

A Generative Game Design Case Study
Relatório Final

Desenvolvido por
André Filipe Monteiro de Oliveira Casaleiro

Orientadores:
Licínio Roque
Rui Craveirinha

Juri:
Luís Pereira
Antonio Silveira Gomes

Mestrado em Design e Multimédia
Faculdade de Ciências e Tecnologia
Universidade de Coimbra
2015 / 2016



RESUMO

Sistemas de Geração Procedimental de Conteúdo são frequentemente usados no Design de Jogos como ferramentas que auxiliam os designers a criar conteúdo. Estes sistemas são utilizados durante a produção de videojogos para apoiar na produtividade e criatividade, como também são usados quando os jogos são lançados no mercado ao utilizar estes sistemas de geração de conteúdo para criar novos materiais como mapas e objetos de dentro do jogo, de forma a que os jogadores tenham sempre algo novo para experimentar e não percam interesse no videojogo.

O objetivo desta dissertação é criar um caso de estudo que demonstre como estes sistemas de PCG (Procedural Content Generation) podem ser mais que ferramentas de apoio ao designer, mas sim um elemento fundamental no design de um videojogo. São raros os exemplos deste tipo de design no mercado e meio académico, o que torna interessante a sua exploração.

O resultado deste caso de estudo vai ser o Design e Protótipo de um jogo, como prova de Design de Jogos Generativo, jogos que são criados a partir de sistemas de PCG. Vai ser estabelecido uma base teórica na investigação de Geração Procedimental de Conteúdo ligada à área de videojogos, nos quais os conhecimentos adquiridos vão ser utilizados para desenvolver o Design do jogo e um protótipo como prova de conceito desse design.

PALAVRAS-CHAVE:

Generative Game Design, Geração Procedimental de Conteúdo,
Design de Videojogos

ABSTRACT

Procedural Content Generation systems are frequently used in Game Design as tools that assist designers to create content. These systems are used during videogame production to support productivity and creativity; they are also used when videogames are released in the market by using content generation systems to create new content as maps and objects inside the game, in a way that players always have something new to play and don't lose interest in the game.

The goal of this dissertation is to create a case study that shows how PCG (Procedural Content Generation) systems can be more than tools that support the designer, but key elements in the design of a videogame. There are few examples of this kind of design in the market and academia, which makes it an interesting theme to explore.

The result of this case study will be the Design and Prototype of a game, as proof of Generative Game Design, games that are built around PCG. A theoretical base will be established in the research of Procedural Content Generation used in videogames, where the knowledge gained will be used to develop the game Design and proof of concept prototype.

KEYWORDS:

Generative Game Design, Procedural Content Generation,
Videogame Design

Obrigado aos meus pais pelo apoio incondicional.

Obrigado aos meus amigos por estarem sempre presentes para me aturar.

Obrigado à minha família de quatro patas por me garantirem todos os dias um sorriso.

Obrigado aos meus orientadores por todo o suporte durante todo este processo.

ÍNDICE

1. INTRODUÇÃO	1
1.1 MOTIVAÇÃO	2
1.2 ENQUADRAMENTO	3
1.3 OBJETIVOS	4
1.4 ESTRUTURA	5
2. ESTADO DA ARTE	7
2.1 VIDEOJOGOS	8
2.2 INTRODUÇÃO À GERAÇÃO PROCEDIMENTAL DE CONTEÚDO	11
2.2.1 PORQUÊ USAR PCG?	11
2.3 O QUE É GERAÇÃO PROCEDIMENTAL DE CONTEÚDO	14
2.3.1 PROPRIEDADES DESEJÁVEIS PARA UMA SOLUÇÃO DE PCG	15
2.3.2 CONTEÚDO GERADO POR PCG	17
2.4 TAXONOMIA DA PCG	19
2.5 PCG APLICADO NA EXPERIÊNCIA DE JOGO	27
2.5.1 DINÂMICAS	28
2.5.2 ESTÉTICAS	33
3. ABORDAGEM METODOLÓGICA	37
3.1 OBJETIVOS	38
3.2 METODOLOGIA	39
3.3 PLANO DE TRABALHO INICIAL	41
3.4 PLANO DE TRABALHO FINAL	42

4. DESENVOLVIMENTO DO DESIGN E PROTÓTIPO DO JOGO	45
4.1 O PRIMEIRO DESIGN	46
4.1.1 DESCOBRIMENTOS	48
4.2 O ESTUDO PARA UMA NOVA SOLUÇÃO	53
4.3 O JOGADOR CONTRA O PLANETA: ADAMASTOR	60
4.4 TECNOLOGIA	68
4.5 PRODUÇÃO DO PRIMEIRO PROTÓTIPO	72
DESIGN & PROTÓTIPO ADAMASTOR	93
5.1 DESIGN FINAL	84
5.2 PROTOTIPAGEM FINAL	95
5.3 ARTE GRÁFICA	101
5.4 DEMO NA DIGRA-FDG 2016	107
5.5 PLAYTESTING E AVALIAÇÃO	107
REFLEXÃO E CONCLUSÃO	113
ANEXOS	115
ANEXO 1	116
ANEXO 2	119
ANEXO 3	122
QUESTIONÁRIO SESSÕES DE PLAYTESTING	122
BIBLIOGRAFIA E REFERÊNCIAS	135

1.

INTRODUÇÃO

Hoje em dia, na nossa sociedade, os jogos digitais tornaram-se numa atividade da rotina diária. Sejam estes jogos feitos para passar tempo em pequenas pausas de cinco minutos, jogos para entreter durante horas ao contar narrativas complexas ou desafiar o lado mais competitivo do ser humano, a realidade é que estes são jogados por um número significativo da população, desde crianças de tenra idade até adultos de idade avançada.

O facto dos *smartphone* e *tablets* terem evoluído até um ponto em que têm uma capacidade de processamento equiparada a computadores pessoais, permite-nos levar uma maior variedade de jogos digitais para todo o lado. Mas este mesmo avanço e crescimento na área dos jogos digitais faz com que a produção dos mesmos se tornem tarefas realmente colossais, com alguns jogos já com equipas de produção na casa das centenas de pessoas. Este facto, entre outros, ajuda a que se queira desenvolver tecnologias que permitam apoiar as equipas que produzem jogos digitais nas suas tarefas, seja como apoio criativo ou até desenvolverem conteúdo de forma automatizada.

É com estas ideias que se tomou a decisão de abordar este tema como dissertação, ao explorar uma ligação entre Design de Jogos, Geração Procedimental de Conteúdo e Jogos Multiplataforma como um caso de estudo.

1.1 MOTIVAÇÃO

A Geração Procedimental de Conteúdo é uma ferramenta que abre um grande leque de possibilidades em Design de Jogos, tanto como apoio aos designers na criação de conteúdo, como também para apoiar a experiência do jogador ao responder à forma como este interage o próprio jogo. Assim é de grande interesse desenvolver um caso de estudo que explore como um jogo digital pode ser construído à volta deste sistema de produção de conteúdo.

Além da Geração de Conteúdo, é importante também referir que o jogo vai ser desenhado como um jogo multiplataforma, possível de ser jogado em vários equipamentos digitais que necessitem de um método de *input* diferente (por exemplo a diferença entre o uso de um rato/*trackpad* num computador *versus* touch num *smartphone* ou tablet *versus* um comando numa consola). Assim esta proposta visa tanto fazer uma exploração entre Game Design e Geração Dinâmica de Conteúdo como também em Interfaces Humano-Computador (UI) e Experiência do utilizador (UX).

1.2 ENQUADRAMENTO

No contexto de Design de Jogos Digitais coloca-se com frequência a necessidade de produzir grandes quantidades de materiais audio-visuais ou regras que influenciem a experiência de jogo. A produção crescente de conteúdos com originalidade e qualidade é necessária para sustentar uma interação continuada e a longevidade da relação do jogador com o jogo. A produção manual destes conteúdos (modelos e suas representações, composições visuais e sonoras, cenários de jogo, regras, etc) é difícil e onerosa, sendo desejável encontrar soluções automatizadas de natureza generativa e avaliativa.

A evolução da tecnologia e diversidade na oferta de equipamentos nos quais é possível jogar jogos digitais criou também uma maior necessidade e oportunidade para desenvolver Jogos Multiplataforma, onde o mesmo jogo pode ser jogado em diversas plataformas com interfaces adaptadas. Assim é importante uma boa aplicação de Design de Interface (UI) e Experiência do Utilizador (UX) para permitir uma experiência de qualidade no jogo em várias plataformas.

1.3 OBJETIVOS

A meta principal desta dissertação é a criação de um protótipo de um jogo multiplataforma original, o qual vai ser desenvolvido em volta de sistemas de Geração Procedimental de Conteúdo.

Pretende-se que ao longo do desenvolvimento deste jogo se consiga atingir os seguintes objetivos:

- Investigação e estudo do estado da arte em Geração Procedimental de Conteúdo e Design de Jogos Digitais;
- Investigação e estudo das tecnologias de desenvolvimento aplicáveis, em particular para as que permitem um desenvolvimento multiplataforma (computador, *mobile*, consola);
- Elaboração de uma proposta para o jogo a desenvolver;
- Design e construção de um protótipo para ensaio do conceito;
- Design da interface e experiência do jogo - Interface do Utilizador e Experiência do Utilizador em videojogos;
- Avaliação da experiência e evolução do protótipo a partir dos resultados deste (*usability & playfulness*).

1.4 ESTRUTURA

Neste subcapítulo apresenta-se a organização e estrutura do documento.

O primeiro capítulo apresenta uma Introdução à dissertação. É feita uma explicação do tema da dissertação, as motivações e interesses que a levaram ao seu desenvolvimento, os objetivos que se pretendem atingir com a mesma e também a estruturação do documento.

O segundo capítulo é dedicado ao Estado da Arte. Aqui é apresentada a investigação feita na área da Geração Procedimental de Conteúdo, o que esta é, os motivos e importância que a levam a ser usada na área dos jogos digitais e ainda taxonomias da sua construção.

O terceiro capítulo explica a Abordagem Metodológica a ser usado no trabalho de suporte a esta dissertação. É introduzido o método que irá ser aplicado para o desenvolvimento desta dissertação, argumentada a sua adequação, e apresenta-se o plano de trabalho correspondente.

O quarto capítulo compreende a Proposta de Jogo. Aqui introduzimos a ideia e o conceito do jogo a ser desenvolvido no contexto desta dissertação.

2.

ESTADO DA ARTE

Nesta secção é apresentada uma introdução a Videojogos e à Geração Procedimental de Conteúdo, o que esta é, como esta técnica pode ser aplicada em jogos e quais os motivos subjacentes ao crescimento da sua utilização e desenvolvimento e algumas aplicações práticas da mesma.

Para facilitar a leitura deste documento, irá ser usada frequentemente a sigla PCG (*Procedural Content Generation*) em vez da sua escrita por extenso Geração Procedimental de Conteúdo.

“Videogames are, before anything else, games.” – Gonzalo Frasca [1]

2.1 VIDEOJOGOS

Não existe uma definição formal sobre videogogos, é um conceito ainda em estudo no meio acadêmico, do qual ainda acontece muita discussão entre as diversas opiniões. Esta dissertação não vai tentar dar uma explicação do que são jogos, é um conceito muito maior do que a palavra aparenta, não são apenas os jogos nas quais crianças participam ou aqueles em que participamos para passar o tempo, mas sim fenômenos que fazem intrinsecamente parte da cultura criada pelo ser humano, desde rituais, desporto até à educação que passamos durante toda a nossa vida [2]. Até um certo nível, animais selvagens participam entre si no que pode ser considerado o conceito de jogo, como as crias de leões brincarem umas com as outras para aprenderem técnicas que vão usar quando chegarem à maioria para sobreviverem sozinhas [2]. Falemos então apenas sobre videogogos.

Numa definição mais simples e aberta, videogogos são aplicações interativas digitais estruturadas por elementos formais, aos quais podem se podem juntar elementos dramáticos que possibilitam criar experiências emocionais e cativantes aos jogadores [3]. Elementos formais são os elementos que tornam os videogogos jogos, são todas as regras, objetivos e procedimentos que estruturam o jogo. Como exemplo, eles elementos podem ser, a partir do popular videogogo de arcadas *Pac-Man* [4]: o pac-man

só se pode se movimentar para cima, baixo esquerda e direita para casas que estejam não são paredes; o pac-man ganha pontos por comer os círculos espalhados dentro do jogo; se o pac-man for apanhado por um fantasma então perde uma vida; o objetivo do pacman é comer todos os círculos do mapa para poder passar ao nível seguinte etc. Se retirarmos estes elementos do Pac-Man, este passa a não ser um jogo é só imaginar: Como seria o Pac-Man se retirasse-mos os objetivos? Como seria o Pac-Man se não o deixássemos movimentar (e daí retirar toda a interação que o jogador tem com o jogo)?

Os elementos dramáticos são elementos que adicionam emoção e criam outro impacto nos videojogos. São a narrativa, as personagens, as conversas, o ambiente criado para criar uma emoção como medo ou felicidade ao jogador [5]. Ter uma forte compreensão de como todos estes elementos interagem entre si são os fundamentos do que podemos chamar de *Game Design* (Design de Jogos) [6].

Uma outra definição menos abstracta é dada por Katie Salen e Eric Zimmerman. “Um jogo é um sistema onde jogadores participam num conflito artificial, definido por regras, que produzem um resultado quantificável” [7]. Um sistema é como várias componentes interagem entre si e os padrões que estas formam, neste caso é como videojogos funcionam [7]. Jogadores são quem participam ativamente num jogo, no qual interagem com o sistema de um jogo de forma a participarem na experiência do jogo [7]. Artificial refere-se a como jogos mantêm uma separação da “vida real” tanto em tempo como espaço, mesmo que este tome lugar no mundo real, o artificial é um dos atributos que definem videojogos [7]. O conflito encarna um competição entre poderes, seja competição ou cooperação entre jogadores, ou um conflito com o sistema de jogos [7]. Regras criam a estrutura de onde vai emergir o jogo, ao especificar o que o jogador pode ou não fazer [7]. O resultado quantificável é a conclusão do jogo, o jogador vence ou perde, ou recebe um resultado numérico no final do jogo [7].

Estes autores dão também outra definição para o que é *Game Design*, “Design de Jogos é o processo pelo qual um Designer de Jogos cria um jogo, o qual vai ser enfrentado por um jogador, de onde vão emergir experiências de jogo.” [7].

Videojogos surgem em computadores a partir da década de 50 do século XX, mesmo que na altura não tivessem ainda ganho esta definição. Um dos primeiros videojogos, desenvolvido em 1950 por Josef Kates, era um computador que conseguia jogar o “jogo do galo”, intitulado de *Bertie the Brain* [8]. Este computador, que media 4 metros de altura, permitia que os jogadores jogassem o jogo do galo contra uma inteligência artificial [8]. Durante os anos seguintes foram desenvolvidas mais experiências como esta, computadores que conseguiam jogar simples formas de videojogos, principalmente devido a limitações tecnológicas dessa época.

Um dos primeiros jogos mais populares desenvolvidos para computador, antes de videojogos terem atingido as massas, foi um jogo intitulado de *Spacewar!* Lançado em 1961, desenvolvido por Steve Russell, Martin Graetz e Wayne Wiitanen para o computador PDP-1 [9]. O jogo consistia num duelo entre duas naves espaciais, cada uma controlada por um jogador a partir de um pequeno joystick. Este serviu durante muito tempo como uma demonstração das capacidades deste computador.

Mas videojogos só atingiram as massas nos anos 70 do mesmo século, quando foi lançado o primeiro jogo que realmente se tornou popular. O jogo foi o



Figura 1. Bertie The Brain(1950)

Pong, lançado em arcadas em 1972 pela Atari, criado por Nolan Bushnell e Ted Dabney [10]. O jogo era uma espécie de simulação de um jogo de ping-pong (de onde deriva o nome), onde dois jogadores competem entre si. Cada jogador controla uma barra em lados opostos do ecrã, que serve como raquete, e tem que tentar marcar o máximo de pontos ao conseguir fazer passar uma bola pelo lado do adversário.

Na década seguinte, nos anos 80, começaram a surgir avanços nas tecnologias dos computadores pessoais (como a Apple-2 e Commodore 64), tal como versões de consolas que se podiam ligar à televisão e permitiram jogar videojogos (como a Atari 5200). Isto fez com que videojogos passassem das arcadas para dentro de casa. Foi também nesta altura que

surgiram as primeiras experiências em Geração Procedimental de Conteúdo para videogames.

2.2 INTRODUÇÃO À GERAÇÃO PROCEDIMENTAL DE CONTEÚDO

Para abordar o tema sobre o que é a de Geração Procedimental de Conteúdo podemos começar com uma breve definição: *Geração Procedimental de Conteúdo é o uso de um algoritmo formal para gerar conteúdo que seria tipicamente produzido por uma sequência manual de ações de um ser humano* [11]. Por outras palavras, são ferramentas e algoritmos que permitem ajudar designers a criar conteúdo durante a produção do jogo ou gerar conteúdo para os jogadores antes ou no momento de jogo.

A definição em Design de Jogos e PCG de Conteúdo corresponde a todas as componentes necessárias para existir um jogo: as regras, níveis, mapas, adereços, objetos, texturas, animações, música, histórias, missões, personagens, armas, veículos, etc [2], isto terá uma explicação mais descritiva no subcapítulo 2.2.2.

2.2.1 PORQUÊ USAR PCG?

Ao conhecer a ideia básica do que é Geração Procedimental de Conteúdo, pode levantar-se a questão: porquê usar esta tecnologia no Design de Jogos? Esta pergunta consegue ter várias respostas. A primeira pode ser apresentada a partir do primeiro uso histórico de geração de conteúdo em jogos digitais: PCG foi uma das soluções encontradas para as limitações de memória dos antigos computadores nos anos 80 [12], [13]. Assim PCG servia para gerar conteúdo quando não havia espaço para guardar informações e outros adereços dos jogos na memória do computador. Neste caso guardavam o conteúdo num estado comprimido até ser necessária a sua utilização. Um dos primeiros exemplos comerciais é o jogo *Elite* lançado

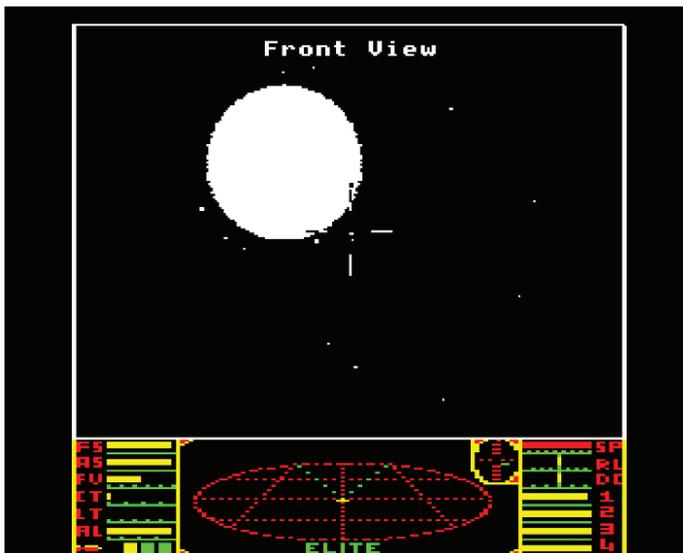


Figura 2. Elite (1984) [14] - Visão sobre uma estrela

Figura 3. Elite (1984) [14] - Visão sobre o mapa de estrelas próximas

em 1984 [14]. Este conseguia manter guardados centenas de sistemas solares (e cada um dos seus constituintes) em apenas alguns kilobytes de memória no hardware, fazia isto ao representar cada astro e objeto espacial com apenas alguns números - esses dados quando descomprimidos possibilitavam o acesso a todos os detalhes de cada objecto no jogo, como por exemplo os nomes, localização, populações, preços, etc [12] de cada planeta.

Outra resposta é o facto que um dos maiores motivos para o uso da Geração Procedimental de Conteúdo, é esta ser um método para substituir o ser humano na produção de conteúdo que muitas vezes pode ser repetitivo ou moroso [12], [15]. Por exemplo: gerar vários níveis de um jogo a partir de um conjunto de regras (ex. *Spelunky* [16]); adaptar conteúdo ao *gameplay* do jogador à medida que este avança no jogo (ex. *Galactic Arms Race* [17]); ou simplesmente na forma de ferramenta para ajudar os designers a produzir conteúdo (ex. *SpeedTree* [18]). A produção de conteúdo para jogos feita por mão de obra humana é um processo lento e caro, e à medida que a tecnologia evolui os jogos tornam-se maiores e o público mais exigente é necessários maiores equipas, tempo e dinheiro para produzir videojogos [15], [19].

Hoje em dia vários estúdios têm equipas com elementos na casa das centenas a desenvolverem videojogos, o que torna o processo de produção de conteúdo num *bottleneck* em termos financeiros e tempo de desenvolvimento para lançar o produto final [15]. Jogos com grandes custos de produção tornam-se menos lucrativos, o que facilmente leva à falência os estúdios que os produziram caso estes não sejam um sucesso financeiro, com

vendas nas de milhões de unidades [15]. O medo deste insucesso financeiro pode também criar problemas no campo criativo: grandes estúdios querem correr poucos riscos, o que faz com que se criem jogos que reciclam sempre as mesmas “receitas” de sucesso anteriores, o que pode ser uma causa para o lançamento de jogos pouco criativos no uso de mecânicas existentes ou o abuso de sequelas. Por outro lado, as equipas são forçadas a lançar jogos antes do tempo, o que resulta no corte ou acabamento prematuro de ideias, mecânicas e conteúdo do jogo que no final acaba por ser um produto com falta de qualidade, e por sua vez tem impacto no sucesso ou no número de vendas.

Estas razões levam a que se abram portas para que a Geração Procedimental de Conteúdo possa substituir a mão humana em tarefas morosas e caras, mas nas quais se consiga manter o grau de qualidade desejado, o que permite poupar tempo e financiamento em tarefas menos críticas ou menos criativas e manter o padrão de qualidade desejado [19].

Além dos grandes estúdios, o uso da PCG pode também ajudar equipas mais pequenas ou até designers de jogos individuais. Ao permitir os criadores de jogos desenvolver grandes quantidades de conteúdo (por exemplo adereços ou configurações de níveis, as quais no modelo de produção manual em grandes estúdios utilizam uma grande quantidade de mão de obra) com o qual iriam gastar muito tempo a produzir individualmente. As técnicas da PCG podem assim conseguir poupar tempo na produção de conteúdo às pequenas equipas, com o qual podem gastar a desenvolver e testar o design do jogo.

A Geração Procedimental de Conteúdo é também um método com o qual se pode adaptar o jogo aos jogadores. Aliar a PCG a *player-modelling*, permite adaptar o jogo ou gerar conteúdo personalizado para o jogador [20]. Por exemplo, ao monitorizar as ações e reações de um jogador aos obstáculos que este encontra num nível, e medir a dificuldade que este tem a superar esses obstáculos, podemos adaptar dificultando ou facilitando os níveis seguintes em resposta a esses resultados.

Outro dos motivos surge nas possibilidades que a PCG abre à criatividade dos designers de jogos. Como o ser humano aprende com a

experiência é natural que se repitam formulas que tenham sucesso em design de jogos, o que leva a pouca inovação. Assim permitir que um sistema gere novas regras, níveis, narrativas, etc. pode dar origem a novas soluções que podem ser exploradas em design de jogos, com as quais os designers podem aproveitar a oportunidade para trabalhar ideias inovadoras [19].

Uma futura exploração da Geração Procedimental de Conteúdo pode criar novos tipos de jogos que só podem ser feitos ao serem construídos em volta das mecânicas de geração de conteúdo. Isto pode possibilitar um futuro onde jogos conseguem gerar conteúdo a uma velocidade suficientemente rápida (tempo real) e com um bom grau de variedade que podem produzir experiências com um valor de *replayability* quase infinito (sem reciclar conteúdo que o jogador já conhece) [21], [19]. Por exemplo jogos online com um mundo persistente podem ganhar imenso com a evolução destes sistemas, já que permite criar um fluxo de novo conteúdo que mantenha um interesse renovado dos jogadores sobre o jogo, um problema que é bastante difícil de resolver a partir de mão humana, já que o tempo para produção de conteúdo com qualidade é muito maior do que o ritmo que os jogadores consomem o conteúdo [12].

2.3 O QUE É GERAÇÃO PROCEDIMENTAL DE CONTEÚDO

Nesta secção irá ser apresentada uma explicação mais descritiva sobre os sistemas de Geração Procedimental de Conteúdo. Vou começar por apresentar quais são as propriedades e características desejadas para as soluções de Geração Procedimental de Conteúdo (PCG), um método para caracterizar o tipo de conteúdo gerado por estes sistemas, e um conjunto de taxonomias pensadas por vários autores e como os PCG afetam a experiência de jogo.

Convém apontar que ainda não existe uma grande quantidade de literatura sobre Geração Procedimental de Conteúdo, mas alguns

investigadores da área já tentaram criar *frameworks* ou taxonomias com os quais descrever um sistema de PCG. Estas taxonomias não são mutualmente exclusivas e complementam-se umas às outras, já que analisam esta área a partir de vários pontos de vista.

2.3.1 PROPRIEDADES DESEJÁVEIS PARA UMA SOLUÇÃO DE PCG

Podemos pensar em implementações de PCG como *soluções para problemas de geração de conteúdo* [19]. Cada problema tem os seus requisitos necessários ou desejáveis para a um sistema de geração de conteúdo. A única constante é o *tradeoff* exigido entre as qualidades permitido pelo estado da tecnologia atual, por exemplo a escolha entre beneficiar velocidade ou detalhe gráfico, ou a escolha entre diversidade ou fiabilidade.

Togelius et al. introduzem alguns dos elementos desejáveis numa solução de PCG [19]:

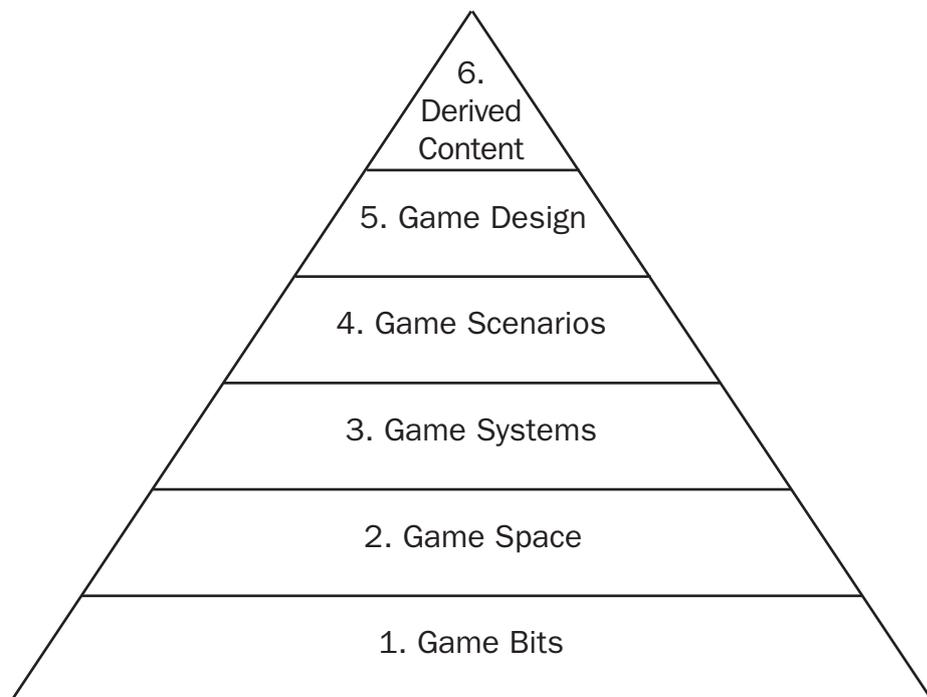


Figura 4. Hendriks et al. [15] - Tipos de conteúdo que podem ser criados por PCG

- **Velocidade:** Requisitos de velocidade dependem do objetivo da solução, ao que para uns é necessários gerar conteúdo em milésimas de segundo durante o decorrer do jogo, como outros têm uma maior liberdade para demorar o tempo que for necessário a gerar o conteúdo com a máxima qualidade possível (logo para estes a velocidade não é uma prioridade);
- **Fiabilidade:** Alguns geradores de conteúdo só têm a preocupação de gerar conteúdo no momento independente da qualidade, enquanto outros têm que garantir que o conteúdo gerado atinge uma determinada qualidade definida. A importância deste requerimento depende do tipo de conteúdo a ser gerado;; o mapa de uma masmorra sem entrada ou saída é considerado uma grande falha no design de um jogo, logo o gerador de mapas tem que ter uma garantida de qualidade mesmo que isso necessite de mais recursos; mas árvores com alguns padrões estranhos de construção podem passar despercebidos no meio de uma floresta com o qual se podem poupar recursos no uso deste gerador;
- **Controlo:** Na maior parte das soluções de PCG é necessário um certo controlo sobre o gerador de conteúdo, seja este controlo feito por um ser humano ou um algoritmo. Neste caso a maior parte do controlo surge de parâmetros ou funções de avaliação sobre o gerador, o qual podem ter várias dimensões de controlo, como por exemplo um algoritmo decidir as cores para um carro ou um jogador decidir o número de continentes a ser gerado para um mapa.
- **Expressividade e Diversidade:** Muitas vezes é necessário que um PCG consiga gerar um leque diverso de conteúdo para evitar que este seja apenas pequenas variações de uma unidade básica. O extremo da não-expressividade é um gerador de níveis que cria sempre o mesmo nível mas com uma pedra sempre no mesmo local mas sempre com uma cor aleatória; o outro extremo é um gerador que gere níveis completamente aleatórios que resultem mapas sem sentido ou até sem possibilidade de serem jogados. Medir expressividade é um tópico pouco trivial,

e criar um gerador de níveis que consiga criar conteúdo diverso sem comprometer a qualidade é ainda menos trivial.

- **Criatividade e Credibilidade:** Não é o objetivo de muitas soluções criar conteúdo que pareça que foi gerado por uma máquina em vez de mão humana.

2.3.2 CONTEÚDO GERADO POR PCG

Hendrikx et al. [15] apresentam uma forma de categorizar o tipo de conteúdo criado pelos sistemas de PCG. Estes podem ser apresentados como uma pirâmide que vai dos blocos básicos do jogo (sons e texturas) até conteúdo gerado a partir da experiência de jogo (leaderboards) (figura 4). As categorias são:

1. Game Bits [15]: Estas são as unidades básicas de conteúdo para jogos, estas individualmente não influenciam a experiência do jogador. Cada um destes *Bits* pode ser considerado *concreto* ou *abstrato*: os primeiros podem ser por exemplos árvores com as quais o jogador pode interagir no mundo do jogo, os segundos são por exemplos sons ou texturas que precisam de ser combinados para produzir um bit *concreto* [15]. O autor descreve 6 tipos de *Game Bits*:

- Texturas (abstrato);
- Som (abstrato);
- Vegetação (concreto);
- Edifícios (concreto);
- Comportamento (abstrato);
- Fogo, Água, Pedra e Nuvens (concreto);

2. Game Space [15]: O *Game Space* é o ambiente onde decorre o jogo, e está parcialmente decorado por *Game Bits* por onde o jogador navega. Estes espaços de jogo podem ser definidos como *concretos* ou *abstratos*. Espaços *concretos* são aqueles que estão relacionados com a forma como o ser humano vê o espaço físico ou o mundo real: podem ser florestas, labirintos, planícies, etc. O tabuleiro de xadrez é um exemplo de espaço *abstrato* [15]. O *Game Space* é importante para a criação de um jogo. São

identificados três tipos de espaços:

- Mapas interiores (abstrato ou concreto);
- Mapas exteriores (abstrato ou concreto);
- Corpos de Água (abstrato).

3. Game Systems [15]: Uso da teoria ou modelação de sistemas complexos para gerar ou simular partes de jogos. Estes podem ser *abstratos*, simulações que não mostram o processo de simulação mas sim apenas os resultados (por exemplo sistemas que criam uma ligação entre a vegetação e o tipo de bioma de um local) ou *concretos*, como a simulação de física de cidades (simular os habitantes da cidade) [15]. São identificados quatro tipos destes sistemas:

- Ecossistemas (abstrato ou concreto);
- Redes de estradas (abstrato ou concreto);
- Ambientes Urbanos (abstrato ou concreto);
- Comportamento de Entidades (abstrato);

4. Game Scenarios [15]: Estes descrevem ao jogador a sequência lógica de como se desenrola o jogo. Estes podem ser *abstratos*, os quais descrevem como objetos se inter-relacionam (por exemplo, explicar como uma personagem pode recolher fruta de árvores), ou *concretos*, os quais são apresentados explicitamente no jogo como por exemplo parte de uma narrativa [15]. São identificados quatro tipos de cenários:

- Puzzles (abstrato);
- *Storyboards* (abstrato/concreto);
- História (abstrato/concreto);
- Níveis (abstrato/concreto);

5. Game Design [15]: O design de um jogo é feito de conteúdo como regras e objetivos; componentes estéticas como um arco dramático ou um tema gráficos são também elementos importantes no design [15]. Este tipo de conteúdo pode ser dividido em dois tipos:

- Design do Sistema (abstrato, geração de regras e objetivos do jogo);
- Design do Mundo (concreto, geração do cenário, história e tema do jogo);

6. Derived Content [15]: Este conteúdo é o produto secundário do jogo. Este é importante para o sentido de imersão do jogador com o jogo, este guarda dentro ou fora do jogo as experiências deste dentro do jogo.

- Notícias e *broadcasts*;
- Leaderboards;

2.4 TAXONOMIA DA PCG

Não existe nenhum método “definitivo” para categorizar ou descrever de forma estandardizada um sistema de Geração Procedimental de Conteúdo. Mesmo assim alguns autores criaram possíveis taxonomias para descrever um sistema de PCG.

Togelius et al. [12] apresentaram uma possível taxonomia que categoriza os PCG em 5 critérios ou classes diferentes, cada uma delas representada em dois extremos onde as soluções de PCG podem ser classificadas nalgum ponto entre os dois extremos. As classes são:

Online versus Offline: A primeira distinção baseia-se em que altura é feita o uso de PCG: se a geração de conteúdo é feita *online* durante o decorrer do jogo, ou *offline* durante o desenvolvimento do jogo [12] ou antes de uma sessão de jogo [19] (Vou usar no decorrer deste documento *online* e *offline* como estas definições). Um exemplo de geração de conteúdo *online* pode ser visto no jogo *Left 4 Dead* [22] onde o jogo cria uma experiência



dinâmica ao alterar o estado de jogo a partir do comportamento do jogador a partir de técnicas de PCG (por exemplo enviar hordas de *zombies* contra o jogador caso este fique muito tempo no mesmo local). Um exemplo *offline* é a geração do mapa antes de um jogo de *Civilization IV* [23]. Mas podem existir exemplos intermédios,

Figura 5. Left 4 Dead [22] - Visão do jogador sobre uma horda de zombies

por exemplo um algoritmo a correr num servidor de um jogo de estratégia em tempo real (*real time strategy* ou RTS), pode sugerir aos jogadores novos mapas a partir de registos do estilo de jogo dos jogadores [12].

A geração de conteúdo *online* coloca alguns requisitos no algoritmo do PCG: tem que ser bastante rápido; tem que ter um funcionamento previsível; e dependendo do contexto os resultados têm que ter uma qualidade previsível [12]. Esta previsão é necessária para manter uma qualidade padrão sobre o conteúdo gerado. Pelo contrário, dependendo do tipo de sistema de geração de conteúdo *offline* não tem grandes limitações no tempo de geração [19].



Figura 6. Rogue [24] - Visão da masmorra

Conteúdo Necessário versus Conteúdo Opcional: Outra distinção sobre o conteúdo gerado por um PCG é se este é *necessário* ou *opcional*. Conteúdo *necessário* é essencial para o progresso dos jogadores no jogo ou até mesmo para o jogo em si – por exemplo as masmorras geradas no jogo *Rogue* [24] são necessários para o progresso do jogo. Enquanto que conteúdo opcional pode ser evitado pelo jogador, por exemplo as missões geradas no jogo *The Elder Scrolls V: Skyrim* [25] (estas missões são identificadas como *Radiant Quests*; e são criadas pelo sistema de Inteligência Artificial do jogo *Radiant A.I* [26]) podem ser evitadas pelo jogador e não são necessárias para o progresso da história.

A maior diferença entre ambas é que o conteúdo *necessário* tem que estar sempre correto, não é aceitável gerar conteúdo que impeça o progresso do jogador no jogo, como por exemplo uma masmorra sem saída ou um inimigo invencível [12]. No caso do conteúdo opcional este também precisa de estar num estado que não impeça o progresso do jogador, mas a qualidade não é tão importante [19], voltando ao exemplo anterior do jogo *Skyrim*, algumas missões geradas pelo sistema são apenas colocar um *NPC* a ser assassinado num local aleatório do jogo, o que não impede o progresso do jogo mas não é considerado uma falha uma personagem estar num local que não tenha sentido no seu contexto.

Sementes Aleatórias versus Vetores de Parâmetros: Esta distinção verifica até que ponto pode o algoritmo de geração de conteúdo pode ser parametrizado. Num extremo temos algoritmos que podem usar uma semente gerada aleatoriamente e usá-la como *input* para o gerador; no outro extremo o algoritmo pode usar vetores multidimensionais de parâmetros com valores reais para especificar propriedades do conteúdo a ser gerado [12]. É possível fazer a comparação de ambos estes extremos com a geração de mapas para jogos: No caso do jogo *Minecraft* [27] os mapas são gerados a partir de uma semente, a qual pode ser partilhada entre os jogadores para criar exatamente os mesmos mapas; No jogo *Civilization IV* [23] o jogador pode definir vários parâmetros para os mapas a ser gerado, desde o tamanho do mapa, número de jogadores a competir, o tipo de clima e até se o mapa vai ter apenas um continente ou vai ser um arquipélago de várias ilhas.



Figura 7. Civilization IV [23]

Geração Estocástica versus Geração Determinística: Esta distinção foca-se no nível de aleatoriedade do conteúdo gerado. O nível de variância nos resultados ao correr o mesmo algoritmo com parâmetros idênticos é uma questão de design [12]. Grande parte dos algoritmos de PCG não são determinísticos, assim o conteúdo produzido com os mesmos

parâmetros é na sua maioria diferente. Um exemplo de um algoritmo

determinístico é o usado no jogo *Elite* [14], a qual consegue guardar uma grande quantidade de conteúdo em pouco espaço (e este pode ser igual para todos os jogadores) [12].



Figura 8. Minecraft [27]

Construtivo versus Gerar-e-Testar: Esta distinção é entre algoritmos que podem ser chamados de construtivos e algoritmos descritos como Gerar-e-Testar (*Generate-and-Test*) [12]. Um algoritmo construtivo gera o conteúdo e verifica apenas se está correto durante a geração, não o verifica após gerar o conteúdo. Um algoritmo de Gerar-e-Testar contém um método para gerar conteúdo e um para o avaliar. Assim quando este termina de gerar conteúdo, este é avaliado a partir de critérios definidos no algoritmo (por exemplo se uma masmorra tem entrada e uma saída e estas estão ligadas por um caminho que o jogador consiga percorrer), assim se o teste falhar o gerador gera novamente conteúdo e avalia-o até ter um que esteja correto e possa ser usado no jogo

No primeiro capítulo e introdução do livro “*Procedural Content Generation in Games: A Textbook and an Overview of Current Research*” Togelius et al. [19] descrevem mais duas distinções que podemos anexar às anteriores:

Genérico versus Adaptativo: Conteúdo *genérico* é aquele gerado sem ter em conta o comportamento do jogador durante o jogo, enquanto o conteúdo *adaptativo* é aquele gerado a partir da análise das interações do jogador com o jogo e que a partir daí define o conteúdo a ser gerado [19]. A maioria dos algoritmos usados em jogos comerciais enquadram-se como *genéricos*, mas no meio académico tem dado mais importância a métodos adaptativos [19]. O exemplo dado anteriormente do jogo *Left 4 Dead* [22] é um dos exemplos de algoritmos adaptativos no meio comercial.

Geração Automática versus Autoria Mista: Distinção entre algoritmos que recebem *input* muito limitado dos designers (por vezes o único *input* é aquele dado no desenvolvimento do algoritmo) e algoritmos usados simultaneamente com *input* dos designers ou jogadores para gerar conteúdo [19]. Um exemplo deste paradigma de *Autoria Mista* é a ferramenta *Tanagra* [28], este sistema permite um designer criar partes de um nível 2D e um algoritmo é usado para gerar os blocos que faltam no nível enquanto verificam que este está correto (i.e., que é possível de jogar o nível do fim ao início).

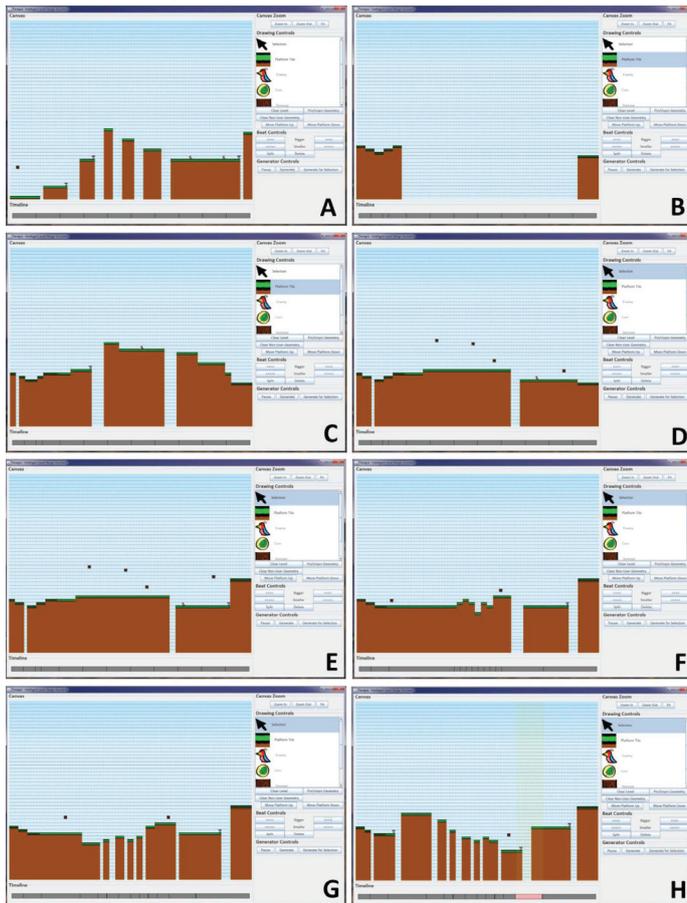


Figura 9. Tanagra [28] - vários exemplos de geração

Hendrikx et al. [15] por sua vez apresentam uma taxonomia mais focada no método de aplicação dos PCG, baseado em algoritmos que já foram usados em aplicações reais. Assim cada sistema de PCG pode pertencer a uma de 6 categorias, mas nada impede que a solução seja mista e use soluções de cada uma destas:

1. Pseudo-Random Number Generators (PRNG): Estes sistemas são usados para imitar a aleatoriedade encontrada na natureza, como por exemplo o algoritmo de ruído de *Perlin* [29].

2. Gramáticas Generativas:

São grupos de regras que ao operar em palavras individuais conseguem gerar frases gramaticamente corretas. Assim conseguem criar conteúdo correto (conteúdo jogável) a partir de elementos codificados como letras/palavras.

3. Filtragem de Imagens: estes têm como objetivo principal melhorar ou modificar uma imagem a partir de uma medida (subjetiva), ou para realçar características de uma imagem para mostrar informação (parcialmente) escondida. Estes métodos por exemplo servem para criar novas texturas a partir de uma anterior.

4. Algoritmos Espaciais: manipulam o espaço para gerar conteúdo. O *output* é gerado a partir de uma estrutura como *input*, por exemplo a partir de uma grelha, fractais ou diagramas de Voronoi.

5. Modelação e Simulação de Sistemas Complexos: é usado quando é impraticável descrever fenómenos naturais a partir de

equações matemáticas, assim modelos e simulações são usados para resolver este problema. Alguns exemplos são os *cellular automata* ou as simulações baseadas em agentes.

6. Inteligência Artificial: tenta imitar inteligência animal ou humana para gerar conteúdo. Alguns exemplos são algoritmos genéticos ou redes neurais.

Gillian Smith em “*Understanding Procedural Content Generation: A Design-Centric Analysis of the Role of PCG in Games*” apresenta um conjunto de abordagens de PCG [21] com uma definição mais abstrata mas que em parte se sobrepõem e expandem a taxonomia de Hendrikx et al. [15]. Estas abordagens são:

1. Otimização: A abordagem de otimização de geração de conteúdo trata o processo de design como uma procura da melhor combinação de elementos que responde a uma lista de critérios. Esses critérios podem ser especificados matematicamente pelo criador do sistema ou curados e avaliados por um humano. Grande parte da investigação feita em Geradores de Conteúdo usa algoritmos evolucionários, que é uma abordagem baseada em otimização. Abordagens de otimização na geração de conteúdo são na maioria dos casos computacionalmente pesados, o que torna difícil o seu uso em jogos que necessitam de um sistema de geração de resposta rápida. Normalmente são usados com um humano como função de avaliação [17], [20] e para criar conteúdo personalizado [28] *offline*.

2. Satisfação de Constrangimentos: Para a abordagem de satisfação de constrangimentos faz-se uma especificação declarativa das propriedades e constrangimentos para o conteúdo a ser criado. Representações declarativas têm a qualidade na qual o designer pode especificar conhecimento sobre como o conteúdo deve aparecer sem ser necessário especificar como o algoritmo de procura deve funcionar. O desafio com este tipo de abordagem advém da determinação de uma representação apropriada para factos sobre o conteúdo gerado, e no *debugging* se um grupo de constrangimentos complexos que se sobrepõem. Esta abordagem é usada extensivamente como ferramenta para os designers [31][28].

3. Gramáticas: Estes sistemas envolvem na especificação de uma gramática na qual o algoritmo a vai expandir sobre o conteúdo criado. Estas não são apenas gramáticas de representação usadas pelos geradores baseados em otimização. As gramáticas podem ser usadas como regras de produção que guiam a geração, sem qualquer preocupação por uma medição externa da qualidade do conteúdo. Nestes casos, a qualidade do conteúdo é implicitamente fundido á gramática baseado em como são definidas as regras de produção. Estes métodos tentam conseguir um balanço entre regras especificadas pelo designer de como as componentes do conteúdo se devem ligar e a exploração feita pelo computador no espaço de design ao expandir a gramática. Gramáticas são usadas na geração de conteúdo *offline* de jogos [32] e em ferramentas para os designers [33].

4. Seleção de Conteúdo: Ainda existe discussões sobre se esta abordagem, a escolha de conteúdo de uma biblioteca e montá-la como peças, é complexa o suficiente para ser qualificada como geração de conteúdo[34]. Segundo o trabalho de Gillian Smith [21], mesmo que simples esta abordagem pode ser considerada uma forma de geração de conteúdo desde que esta sirva para criar ambientes que o jogador possa explorar ou conteúdo que o jogador possa experimentar dentro do contexto de outras mecânicas. O maior problema da seleção de conteúdos é que é simples o suficiente para o jogador encontrar padrões na geração do conteúdo, mas ao mesmo tempo é um dos métodos mais rápidos para geração de conteúdo e é tipicamente usado em jogos onde o gerador tem que funcionar online, durante o tempo de jogo, por exemplo em jogos do estilo *endless runner* [35].

5. Processo Construtivo: Um gerador construtivo constrói conteúdo para a situação e no momento ao ligar vários blocos personalizados (estes podem ser vistos como geradores de seleção de conteúdo que trabalham com peças mais pequenas). Tipicamente tem todo o seu conhecimento do seu design fundido no algoritmo. Este pode fazer uma espécie de procura interna, mas nunca testa os resultados gerados com algum guia heurístico externo que ajude na procura. Geradores construtivos são normalmente específicos ao jogo onde são usados, por exemplo os geradores usados em jogos *rogue-like* [36].

Além destas abordagens, Smith criou também uma tabela de mecânicas com as quais podemos descrever como funciona um sistema de PCG [21] (tabela 1). Esta tabela descreve quatro aspetos principais na mecânica do sistema de Geração de Conteúdos. Estes foram escolhidos numa combinação das preocupações do programador da IA (como funciona o gerador) e as preocupações do designer (como se pode usar o gerador). Cada um destes aspetos pode ser usado para descrever o papel que o Gerador de Conteúdo tem numa ferramenta de design de jogos, tal como nos próprios jogos.

Estas mecânicas resultam em várias dinâmicas que afetam a experiência de jogo para o jogador. Estas vão ser descritas com maior pormenor na secção 2.5

Blocos de Construção	Estado do Jogo	Tipos de Interação	Experiência do Jogador
<i>Bloco Experiencial</i>	<i>Offline</i>	<i>Nenhum</i>	<i>Indireta</i>
Grandes blocos de conteúdo construídos por mão humana.	Conteúdo gerado durante o desenvolvimento ou antes do jogo iniciar.	Sem influência humana.	Sem controlo direto da experiência
<i>Modelos</i>	<i>Online</i>	<i>Parametrizado</i>	<i>Composicional</i>
O computador preenche os espaços em branco de grandes blocos de conteúdo.	Conteúdo gerado durante o jogo.	Indireto, humano coloca valores.	Humano influência componentes disponíveis.
<i>Padrão de Componentes</i>		<i>Preferencial</i>	<i>Experiencial</i>
Pequenos blocos criados por mão humana.		O ser humano é que decide o conteúdo que tem qualidade	O jogador influência a sua própria experiência.
<i>Subcomponente</i>		<i>Manipulação Direta</i>	
Representação interna dos blocos de conteúdo - unidade mais pequena.		Humano manipula o produto diretamente.	

Tabela 1. Gillian Smith [21] - Tabela de Mecânicas de PCG

2.5 PCG APLICADO NA EXPERIÊNCIA DE JOGO

Segundo Smith [21] o resultado das mecânicas apresentadas no final da secção 2.3 são várias *dinâmicas* (as formas como as regras interagem umas com as outras e com o jogador durante o jogo (tabela 2, explicação da tabela na próxima página) e *estéticas* (ou experiências). Ao considerar como a Geração de Conteúdo permite estas dinâmicas, é preferível fazer uma comparação entre jogos com Geração de Conteúdo a jogos com mecânicas similares mas sem Geração de Conteúdo. Isto permito-nos ver que dinâmicas únicas são criadas a partir da Geração de Conteúdo.

Outras Mecânicas	Memorização VS Reação	Estratégia	Procura	Prática	Interação
<i>Centrais</i>	<i>Memorização</i>	O jogador cria estratégias para influenciar o gerador de conteúdo	O jogador procura novo conteúdo num mundo vasto	O jogador pratica mecânicas de jogo em novos cenários	Comunidades de jogadores discutem diferenças das suas experiências no jogo
Necessitam de Geração de Conteúdo.	Testar a memória do jogador				
<i>Parciais</i>	<i>Reação</i>				
Experiência parcial para o jogador	Reagir a circunstâncias imprevistas				
<i>Decorativas</i>					
Não é essencial para a experiência do jogo					

Tabela 2. Gillian Smith [21] - Tabela de Dinâmicas de PCG

2.5.1 DINÂMICAS

a) **Outras Mecânicas:** De forma a perceber onde o sistema de geração de conteúdos se encontra dentro do jogo e as dinâmicas que emergem deste, convém pensar na forma como o jogador vai interagir com o conteúdo gerado e a que ponto é que esse conteúdo vai modificar influenciar a experiência de jogo do jogador.

Centrais: Estes jogos usam a Geração de Conteúdo como uma parte central para a experiência do jogador. Um exemplo é o *Infinite Mario Bros* [37] onde o jogador depende do conteúdo gerado para poder experienciar o jogo.

Parciais: Estes jogos usam o conteúdo para gerar parte da experiência do jogador, mas não todo o jogo. Exemplo é o *Civilizations IV* [23] onde a geração do mapa tem grande influência na exploração e decisões nos primeiros momentos do jogo, mas como o jogo tem tantas outras mecânicas onde o jogador pode criar novas estratégias, ao longo do tempo o mapa gerado torna-se menos importante.



Figura 10. Infinite Mario Bros [37]

Decorativas: Há muitos jogos onde a Geração de Conteúdo tem como objetivo criar apenas conteúdo decorativo. Isto é especialmente usado em gráficos computacionais, onde é uma preocupação em criar texturas, vegetação ou fachadas decorativas para edifícios [33]. Esta distinção entre “decorativa” e as outras duas categorias pode ser similar á declaração de Togelius et al. que algum conteúdo é “desnecessário”, está não é a intenção desta categoria, já que todo o conteúdo criado acaba por influenciar a experiência de jogo no ponto de vista das dinâmicas.

b) Memorização VS Reação: Jogos que são dependentes da Geração Procedimental de Conteúdo para o seu jogo central, e que usam tanto geração *offline* como *online* sem controlo direto dos jogadores são jogos que se tratam de reagir a circunstâncias imprevistas, por exemplo jogos *endless runner* como o *Robot Unicorn Attack* [35], ou jogos onde se tem que explorar um espaço desconhecido, como por exemplo no *Spelunky* [16].



Figura 11. Spelunky [16]

Em contraste, jogos que não tenham geração de conteúdo e usem elementos estáticos pré-criados tendem a criar dinâmicas nas quais os jogadores acabam por memorizar caminhos através dos níveis (por exemplo no jogo de plataformas *Sonic the Hedgehog* [38]), onde o objetivo dos jogadores ao repetir o jogo é conseguir bater o seu anterior melhor recorde.

É importante notar que a dinâmica da reação emerge simplesmente do uso da aleatoriedade. Mesmo assim esta dinâmica pode vir a partir de jogos sem geração de conteúdos que usem um elemento de aleatoriedade, como por exemplo jogos de 3 em linha, no qual se retirássemos a aleatoriedade da cor das peças que aparecem um jogador conseguia memorizar um padrão e praticar jogadas que maximizassem o jogo.

Também é possível criar esta experiência de situações imprevistas através de *crowdsourcing*, por exemplo a partir de uma comunidade que criem níveis para um jogo. Assim o jogador ao experimentar cada nível criado por outros jogadores volta a passar por uma situação de incerteza onde tem que tomar decisões para reagir ao que acontece no jogo, por exemplo a partir de mapas criados pela comunidade para o jogo *Portal 2* [39].

c) Construir Estratégias sobre o Gerador: Geradores de conteúdo *online* e controláveis pelo jogador podem levar a dinâmicas onde o jogador tenta criar uma estratégia à volta do gerador. Isto leva a um ciclo onde o jogador informa o gerador do que deve ser criado a seguir, e o conteúdo gerado em resposta informa o jogador da próxima decisão a ser tomada, por exemplo no *Endless Web* [40].

Não existe um análogo a esta dinâmica em jogos estáticos com conteúdo criado por humanos. Só é possível criar uma estratégia á volta de um gerador quando existe um gerador de conteúdo presente num jogo que consegue responder de forma credível ao *input* humano. Assim, geradores que só tenham controlo indireto sob a experiência do jogador não são adequados a criar esta dinâmica.

d) Procura num Espaço Vasto: Esta dinâmica surge a partir de um gerador que tenha a habilidade de criar vastos espaços com variações que nunca podiam ser feitos razoavelmente por um único autor. Geradores que conseguem produzir esta dinâmica são os que funcionam *offline* ou *online*, ao nível das componentes e subcomponentes (blocos experiências e modelos têm padrões que conseguem facilmente ser detetados por jogadores), que conseguem criar conteúdo central ou parcial à experiência do jogo, e que não necessitam especialmente de controlo do jogador sob o gerador mas que conseguem fazer o design para a experiência tanto diretamente ou através de componentes. Existem vários jogos que usam esta dinâmica, como por exemplo o jogo *Minecraft* [27] ou o futuro jogo *No Man's Sky* [41].

A ideia central aqui é que o jogador consiga produzir conteúdo que o jogador ache surpreendente á medida que exploram o espaço. Por exemplo, no jogo *Borderlands* [42] o gerador de armas produz um grande número de

armas com o qual o jogador pode-se surpreender, mas o jogador pode estar à procura de uma arma em particular à medida que exploram o espaço de jogo.

Para se poder criar esta dinâmica em jogos sem gerador de conteúdo teria que ser necessária uma grande carga autoral. Contudo, há jogos que mostram que isto é plausível, por exemplo o jogo *Animal Crossing: New Leaf* [43]. Este jogo contém um grande elenco de personagens, como uma delas com preferências, personalidade e diálogo únicos. Tem um grande catálogo de objetos que o jogador pode comprar ou procurar à medida que está a jogar. Enquanto que o espaço de jogo é uma pequena cidade, tem tanto conteúdo desenhado que o jogo consegue criar uma dinâmica similar de procura num vasto mundo desenhado.



Figura 12. Borderlands [42] - Exemplo de arma do jogo



Figura 13. Borderlands [42] - Exemplo de arma do jogo

e) Prática em Ambientes Diversos: Existem também jogos que usam a geração de conteúdo para criar espaços onde o jogador pode praticar várias estratégias de jogo. Esta dinâmica surge preferencialmente em geradores *offline* com ou sem controlo do jogador, os quais conseguem criar cenários variados para o jogador testar as suas estratégias, um destes exemplos é o jogo *Civilization IV* [33] onde o jogador tem controlo sobre os parâmetros do gerador que o ajuda a criar os cenários onde o jogador vai jogar e praticar as suas estratégias. Outro exemplo é o jogo *Diablo II* [44] onde o jogador não tem controlo sobre o gerador mas este permite que o jogador pratique diferentes estratégias de combate ou classes de personagens em cenários diferentes.

Em contraste, jogos que tenham apenas um tipo de cenário dificilmente conseguem atingir este nível de dinâmica. Por exemplo, o jogo *Professor Layton and the Curious Village* [45] dá o desafio aos jogadores de resolver uma série de puzzles feitos à mão. Se o jogador ficar preso num puzzle não tem maneira de o praticar, logo para conseguir ajuda para o resolver só pode procurar por pistas ou pedir ajuda a outro jogador. A incorporação de puzzles gerados por um gerador poderia permitir ao jogador praticar vários tipos de estratégia em vários cenários dentro do mesmo contexto. Por outro lado temos jogos como o *Picross* [46] que permite que o jogador pratique as suas estratégias sobre o sistema de puzzles a partir da prática de uma grande lista de puzzles feitos à mão com níveis de dificuldade acrescidas. É plausível mas não muito prático que conteúdo feito pelo ser humano compense pelo criado por um gerador.



Figura 14. Diablo 2 [44]

f) Interação com uma Comunidade de Jogadores: A última dinâmica deste *framework* é como a PCG consegue criar impacto sob a comunidade de jogadores em volta de um jogo.

Investigadores em PCG tipicamente focam-se em criar uma experiência para jogadores individuais. Contudo há provas que a incorporação de Geração de Conteúdo num jogo consegue criar uma comunicação sobre o funcionamento do sistema e o impacto deste sobre a experiência de jogos dentro de uma comunidade de jogadores. Um exemplo, os jogadores do jogo *Civilization IV* entram em longas discussões de como funcionam as opções para gerar os mapas, como estas influenciam as estratégias e porque é que preferem um certo tipo de opções [47].

Compreender a comunidade que cerca um jogo que usa PCG é uma área pronta para futura investigação, tanto pelo ponto de vista da inteligência artificial em PCG (como criar geradores que tenham como alvo uma comunidade de jogadores), como para o ponto de vista do design de jogos ou Interação Humano-Computador (como é que comunidades de jogadores comunicam sobre PCG?). Também existe a oportunidade de estudo de como a Geração de Conteúdo influencia a aprendizagem de jogadores em jogos.

2.5.2 ESTÉTICAS

As dinâmicas criadas entre as escolhas do design do sistema de Geração de Conteúdo juntamente com as mecânicas do jogo podem criar várias experiências estéticas. As três estéticas descritas abaixo fazem parte dos oito “tipos de divertimento” identificados por Hunicke et al. no paper do *framework MDA* [48].

a) Descoberta: Geração de Conteúdo suporta o acto de descoberta ao fornecer ao jogador novos ambientes para explorar ou novos sistemas de geração para o jogador descobrir ao longo do tempo. Esta estética é suportada pelas dinâmicas de *Procura num Vasto Mundo* e *Construir Estratégias sobre o Gerador*. No primeiro caso a descoberta vem do simples acto de explorar um grande ambiente desconhecido. No segundo caso, a descoberta vem de uma forma de exploração do espaço gerativo do jogo.

b) Desafio: Geração de Conteúdo actua em suporte do desafio a partir das dinâmicas de *Memorização VS Reação e Prática em Ambientes Diversos*. Cada uma destas dinâmicas leva a diferentes formas de desafio. Jogos que usam Geração de Conteúdo para forçar o jogador a reagir rapidamente fornecem uma forma de desafio de decisões rápidas, onde o jogador tem que tomar decisões de momento a momento a partir do mundo que se desenrola á frente dele. Outros jogos usam a Geração de Conteúdo para forçar o jogador a reagir, mas não necessariamente sensível ao tempo, assim oferece desafio a partir do conteúdo a ser experienciado nunca antes visto pelo jogador. Similarmente, a dinâmica de *Prática em Ambientes Diversos* permite ao jogador experienciar novos desafios (mesmo sendo do mesmo tipo de desafios) ao jogar o mesmo jogo em ambientes diferentes.

A partir da dinâmica de *Construir Estratégias sobre o Gerador* o jogo tem o desafio acrescentado a partir de uma camada de jogo estratégico que advém a partir da aprendizagem de como o gerador funciona, o que ele pode criar e como usar esse conhecimento como vantagem para o jogador.

c) Camaradagem: Geração de Conteúdo funciona em suporte da camaradagem ao criar um sistema emergente que encoraja a comunicação ente os jogadores fora do ambiente de jogo, entre a dinâmica de *Interação entre uma Comunidade de Jogadores*. A geração de Conteúdo encoraja não só a discussão entre jogadores, como também molda como os jogadores falam sobre o jogo, incluída discussão de estratégias para diferentes tipos de configurações de conteúdo.

3.

ABORDAGEM METODOLÓGICA

Esta secção compreende a metodologia e plano de trabalhos que foram tomados para a realização do trabalho empírico subjacente a esta dissertação. Irá por começar a apresentar os objetivos definidos no início da dissertação, passando pela abordagem metodológica tomada para os cumprir e o plano de trabalho aplicado em prática ao longo do desenvolvimento da dissertação.

3.1 OBJETIVOS

Esta dissertação tem como objetivo final a produção de um protótipo de um jogo multiplataforma. Este vai ser desenvolvido como um caso de estudo no qual se pode aplicar técnicas e conceitos em Geração Procedimental de Conteúdo como base no design de um jogo digital. Assim o objectivo principal é mostrar como *PCG* pode tornar possível uma forma de *Game Design Generativo*.

Os objectivos definidos no inicio desta dissertação podem ser dividido entre dois contextos:

A primeira parte destes objetivos encontram-se no contexto do estudo e investigação necessária para a criação do conceito e elaboração do protótipo. Assim é necessário fazer uma investigação e estudo do estado da arte em Geração Procedimental de Conteúdo e Design de Jogos Digitais e como estes trabalham em conjunto e foram aplicados em casos de reais. Para a elaboração do protótipo é necessário também investigar as tecnologias que vão permitir a aplicação do protótipo, em particular para as que permitem um desenvolvimento de jogos multiplataforma (computador, mobile, consola). Além das mecânicas a aplicar no protótipo do jogo, este tem como objetivo também ter algum valor incluído tendo em conta a área académica do curso no qual se insere esta dissertação. Logo o conceito do jogo tem que conter algum valor que possa ser transmitido ao jogador, seja este uma mensagem ou conhecimento.

A segunda parte dos objetivos encontram-se no contexto da prática do desenvolvimento do protótipo. Assim o primeiro objetivo desta parte é a elaboração de uma proposta e conceito de jogo para desenvolver como protótipo. Os objetivos seguintes são o design e construção de um protótipo para ensaio do conceito criado para o videojogo, e também o Design da Interface e de Experiência do utilizador. Ao longo do desenvolvimento do protótipo é também objetivo realizar várias avaliações da experiência de jogo com vários utilizadores e a evolução do protótipo a partir dos destes resultados (os quais se vão focar em *usability & playfulness*).

3.2 METODOLOGIA

A abordagem escolhida para o processo de desenvolvimento do protótipo é uma abordagem com um foco teórico e também experimental.

Esta abordagem pode ser dividida em quatro grupos principais de tarefas:

a) **Investigação e desenvolvimento da Proposta:**

- Estado da Arte;
- *Brainstorm* de ideias;
- Formulação da Proposta;
- Investigação de Tecnologia;

b) **Ciclo de Game Design:**

- Game Design;
- Interaction Design;
- Concept Art;

c) **Ciclo de Prototipagem**

- Desenvolvimento;
- *Playtesting*;

d) **Escrita do Documento**

Descrição dos grupos:

a) Investigação e desenvolvimento da Proposta: Nesta fase o foco centrou-se na investigação do Estado da Arte em Geração Procedimental de Conteúdo e Design de Jogos. O resultado desta investigação foi tida em conta no processo de *Brainstorm* e desenvolvimento do jogo.

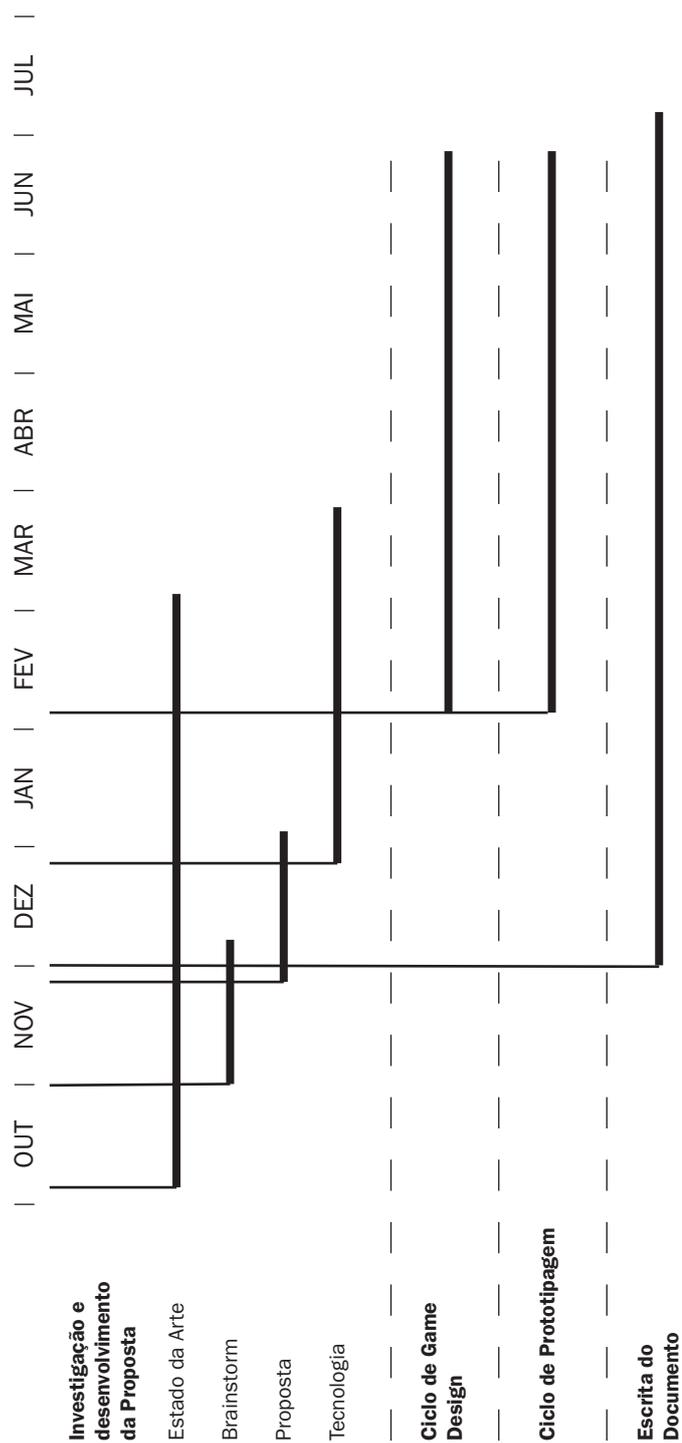
b) Ciclo de Game Design: Este ciclo compreende o desenvolvimento do design do jogo definido na proposta. O primeiro ciclo focou-se no primeiro design do jogo. Existiram mais dois ciclos após este, cada um deles corrigiu os problemas encontrados na versão anterior.

c) Ciclo de Prototipagem: Este ciclo é equiparado ao Ciclo de Game Design, mas em vez de se tratar do design do jogo, este compreendeu o desenvolvimento do protótipo. Este iniciou-se no fim do segundo Ciclo de Game Design, e terminou após o terceiro Ciclo de Game Design.

d) Escrita do Documento: A tarefa de escrita do documento da Dissertação vai acontecer ao longo de todo o processo de desenvolvimento da dissertação.

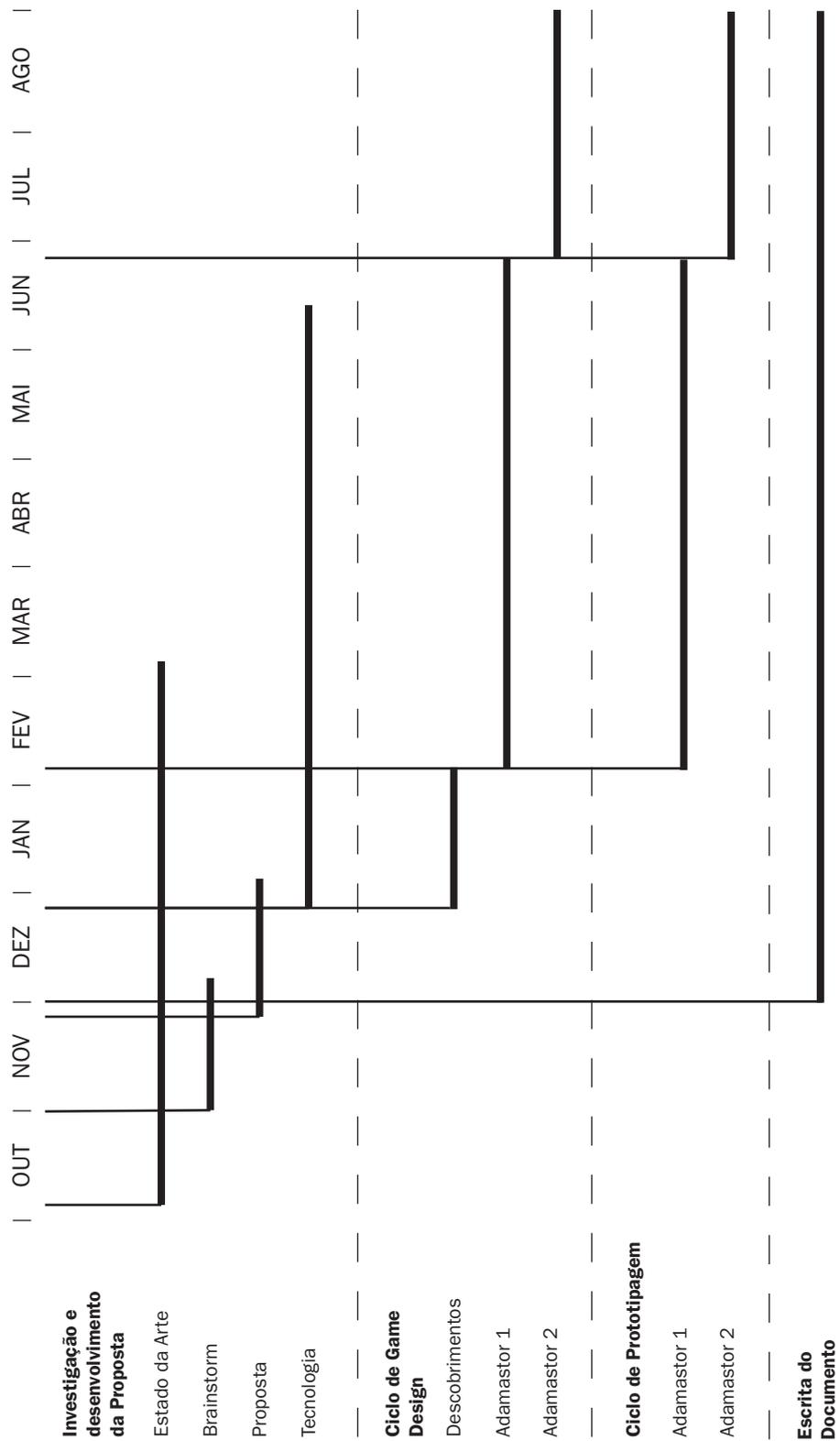
3.3 PLANO DE TRABALHO INICIAL

Diagrama de Gantt:



3.4 PLANO DE TRABALHO FINAL

Diagrama de Gantt:



4. DESENVOLVIMENTO DO DESIGN E PROTÓTIPO DO JOGO

Nesta secção é feito um retrato detalhado sobre o desenvolvimento do design e protótipo do jogo proposto, o seu design e a produção do protótipo.

É feita uma descrição da evolução do jogo a partir de uma sequência temporal, desde a ideia inicial ao design final e o seu protótipo, isto ao explicar os problemas e desafios encontrados e as suas soluções nas diversas áreas que implicam o design e produção de um jogo, e como essas construíram a solução final apresentada nesta dissertação.

Também são exploradas as influências artísticas e de conceito que influenciaram o design do jogo.

4.1 O PRIMEIRO DESIGN

Ao delinear os objetivos desta dissertação, foi decidido que iria criar o design de um jogo, o qual iria dar uma resposta ao que é o *Generative Game Design*, um jogo no qual a sua base assenta a Geração Procedimental de Conteúdo. Foi também decidido um protótipo como proof-of-concept do design do jogo.

Na procura de um conceito inicial para o jogo começou-se por fazer uma sessão de *brainstorming*, na qual se pretendia encontrar a ideia para um jogo no qual o foco do design pudesse ser a exploração do *Gamespace* [15] e a exploração das estéticas de Descoberta e Desafio [48].

Do resultado dessa sessão, fez-se uma exploração mais aprofundada de várias ideias, ao identificar vários pros e contras do seu uso num jogo, tanto nos termos de geração de conteúdo, do *gameplay* e valor atribuído à ideia – foi preferido a ideia ser baseada em algo em concreto, como por exemplo um evento histórico ou problemas e situações atuais, sob ideias mais abstratas. Estas então passaram por outro processo de eliminação, até ter um grupo bastante reduzido que pudessem ser rapidamente explorados na escolha do melhor. Destas sobraram 3 ideias.

Para estudar melhor estes três possíveis futuros conceito, foi feito um exercício no qual fosse possível os ver como um design de um jogo em si e não apenas um conjunto de várias ideias. Este exercício foi feito com recurso a um *Game Design Canvas* baseado no *Participation-Centered Gameplay Experience Design Model* de Luís Pereira et Al. [49]. Este modelo permite auxiliar o processo de design ao colocar a noção de participação como foco no design de um jogo – a forma como o jogador participa na atividade de jogo – de onde surge a experiência e de como esta é interpretada pelo jogador [49]. O modelo propõe seis lentes de como podemos analisar e trabalhar a participação no jogo, cada uma dividida em três pontos operacionais: a Intenção (Qual é o tipo de participação ideal que o jogo está a sugerir?), o Artefacto (Como é que os artefactos suportam as formas idealizadas de participação?) e a Participação (Quais as características da participação do jogador que são consistentes ou demonstram o tipo de participação idealizada?). Estas lentes são:

o

- *Playfulness*: O videojogo como um contexto de participação livre, informal e não estruturada. As Intenções são a exploração, descoberta, recriação e personalização dentro do jogo. Os Artefactos são a natureza da atividade do jogador e a variedade de elementos interativos dentro do jogo. A Participação é representada pelo grau, variedade e tendência de exploração do jogado;
- *Challenge*: O videojogo como um contexto para a participação estruturada, um desafio formal. As Intenções são os desafios, as estratégias, os inimigos ou como dominar habilidades encontradas no jogo. Os Artefactos são a natureza dos desafios propostos, o tipo de castigos ou recompensas, a intensidade e organização dos desafios. A Participação é representada pelo controlo, ritmo, progresso e eficiência do jogador ao realizar tarefas;
- *Sensemaking*: O videojogo como um contexto de participação significativa, de criação de significado. As Intenções são a interpretação de um papel, fantasia e expressão pessoal do jogador. Os Artefactos são o tema e narrativas inerentes, modelos e representações de fenómenos, papéis e motivos, ações com significado. A Participação é representada pelo alinhamento entre ações e papéis, a compressão e análise do fenómeno representado;
- *Embodiment*: O videojogo como um contexto de participação física, tanto virtual e real. As Intenções são o envolvimento e performance física do jogador no jogo. Os Artefactos são a representação física do mundo do jogo, a representação do jogado do mundo do jogo e a interpretação do movimento do jogador. A Participação é representada pelo controlo, ritmo e estética do movimento;
- *Sensoriality*: O videojogo como um contexto de envolvimento multissensorial. As Intenções são a contemplação e admiração. Os Artefactos são o estilo, natureza do estímulo, composições sonoras e visuais, exploração sinestética. A Participação é representada pelo grau de exposição e resposta ao estímulo, interação ou compromisso com as fontes sensoriais;

- *Sociability*: O videojogo como um contexto de participação social, de criar relações entre jogadores. As Intenções são a competição, cooperação, amizade e identificação. Os Artefactos são a diversidade e natureza das interações e relações sociais, modelos de estruturas sociais (equipa, hierarquia, etc). A Participação é representada pela intensidade e tipo de interações entre jogadores.

Estes exercícios encontram-se no capítulo de Anexos como Anexos 1.

Assim concluídos e analisados estes exercícios, foi escolhida a ideia para o design e produção deste jogo. Este conceito vai ser apresentada no capítulo seguinte.

4.1.1 DESCOBRIMENTOS

CONCEITO

A ideia por trás deste conceito é criar um jogo onde é possível demonstrar o método em como foram organizadas e realizadas as expedições que começaram a “Era dos Descobrimentos” – também conhecidas por “Grandes Navegações” – pelos portugueses entre o século XV e XVI. De forma a ilustrar este método ao jogador, torna-se o objetivo deste a descoberta de um caminho marítimo de uma cidade onde o jogo começa, até um local específico dentro de um mundo fictício, isto a partir do comando de uma frota de navios.

Para criar um sentimento de surpresa que permita ao jogador sentir o que é descobrir num novo mundo, mesmo após jogar e terminar o jogo mais que uma vez, este vai ser gerado pelo computador ao usar PCG. O computador irá criar as massas terrestres, os oceanos, correntes marítimas, habitantes, culturas, criaturas e outros eventos.

O JOGO

Este centra-se no ato de exploração e cartografia do mundo e na gestão do grupo de navios da expedição. Isto cria a oportunidade de o jogador explorar, cartografar, participar em eventos, gerir e personalizar a frota, a sua tripulação e o seu navegador.

A exploração é feita através da visão um mapa, onde o jogador tem a possibilidade de cartografar o que explora, desde as massas terrestres, cidades, eventos e até apontar as correntes marítimas. É a partir desta exploração que o jogador vai ser defrontado por vários eventos, na qual a sua solução poderá castigar ou recompensar o jogador pelas suas escolhas. O jogador tem sempre a possibilidade de voltar á sua capital, para guardar o que conseguiu cartografar durante a expedição, como também para reparar os navios e prepara-los para uma expedição que o



Figura 15. Carta náutica - Pedro Reinel (1504). Exemplo de inspiração de como poderia ser visto o mapa do jogo

leve mais longe.

O objetivo da exploração é encontrar uma cidade escondida no mapa, mas o qual é um grande desafio o jogador conseguir encontrar esse caminho com uma só expedição, pelo que é encorajado o jogador seguir o método que foi usado na realidade nos Descobrimientos, criar um processo iterativo de expedições onde cada uma se afasta mais do porto de segurança da capital.

A partir desta noção geral do jogo, podemos dividir o *gameplay* em três componentes distintas: A Exploração, os Eventos e a Gestão.

a) Exploração

A exploração e cartografia são o principal desencadeador de ação, a partir da qual o resto do jogo se vai desenrolar. A exploração é feita através de uma espécie de mapa cartográfico, o qual se encontra branco no seu estado inicial, à exceção da cidade onde começa a expedição e a linha costeira adjacente. Esse local é ponto inicial de onde o jogador vai começar a navegar os oceanos, onde vai ter que cartografar e registar o que encontra – isto feito com apoio do computador, o que jogador pode descobrir numa área em seu redor está como uma sombra sobre o mapa, a função do jogador é fazer uma marca permanente neste a partir dessa sombra, acompanhado dos locais e outros fenómenos que encontra.

Ao desvendar o mundo a partir do mapa, o jogador vai descobrir novas massas terrestres, apontar as correntes marinhas que ajudam ou prejudicam o seu movimento, e também as cidades, criaturas, fenómenos e outros eventos sobre os mares.

Alguns desses eventos que o jogador defronta podem afetar o processo de exploração, como por exemplo, uma tempestade pode abrandar ou mudar a rota da armada.

Para dar como concluída cada expedição, e como forma de poder registar o processo de exploração – uma mimica do que acontecia na realidade, era necessário transportar de volta os conhecimentos adquiridos em cada viagem para o reino. Isto permite também que o jogador possa melhorar a sua frota, a partir de modificações à sua frota que beneficiem o seu desempenho.

No caso de as coisas correrem mal durante uma expedição e a frota for destruída, o jogador perde todo o progresso feito nessa viagem, voltando a tudo o que descobriu no mapa a uma sombra por explorar.

b) Eventos

Os eventos são situações e desafios com os quais o jogador vai ser defrontado. São ao mesmo tempo uma oportunidade para interagir e aprender mais sobre o mundo gerado. Há 3 tipos principais de eventos:

- **Eventos nos Mares:** estes eventos retratam acontecimentos que interagem com o mundo, compostos por batalhas em alto mar contra outros navios ou criaturas marinhas, tempestades, modificações nas correntes marinhas, entre outros eventos decorrentes da exploração. Estes têm um impacto positivo ou negativo para o jogador, muitas vezes numa escala significativa. Por exemplo, um combate pode destruir vários navios o que prejudica o resto da expedição, tal como mudança na corrente marinha pode acelerar a exploração.
- **Eventos da Tripulação:** estes eventos estão ligados a acontecimentos relacionados com a tripulação, reflexo da disposição destes seja na confiança pelo navegador, pela sua saúde ou humor. Exemplos destes eventos são: motins feitos por uma tripulação descontente, um bónus em combate por uma tripulação confiante e sem medo, ou uma maior velocidade na exploração por uma tripulação saudável.
- **Eventos do Navegador:** os eventos do navegador estão ligados a narrativas, como o contacto com novas culturas, negociação de rotas comerciais com povos estrangeiros ou outras narrativas como histórias que podem levar á procura de criaturas marinhas. Estes eventos permitem dar novos traços ao navegador, já que estas experiências refletem se nos novos conhecimentos que o navegador pode adquirir.

c) Gestão da Expedição

O jogador durante as expedições precisa de gerir e controlar a sua frota. É a partir dos elementos que a compõem que se torna possível a interação com o mundo.:

- **Navios:** os navios são o veiculo do ato de exploração. O seu estado físico afeta o processo de exploração, navios bem construídos permitem uma navegação mais rápida e segura. O número de navios que o jogador tem na sua armada limita o número de tripulantes e a quantidade de recursos possíveis transportar. As condições dos navios

têm impacto direto na solução de todos os Eventos nos Mares.

- Tripulação: A tripulação são os elementos que dão “vida” aos navios. O estado da tripulação pode ser visto a partir de três fatores: felicidade, confiança e saúde. Estes fatores afetam a performance da exploração e movimento dos navios. Os recursos têm que ser geridos tendo em conta o tamanho da tripulação e a sua saúde, uma maior tripulação requer mais recursos, tal como uma tripulação doente pode melhorar o seu estado ao ter disponível um grande nível de recursos que podem consumir. Os Eventos da Tripulação refletem as mudanças no estado da tripulação.
- Explorador/Navegador: O navegador é a figura que representa a expedição, ao que pode ser comparado a uma figura herói dos descobrimentos como Vasco da Gama (navegador português que em Maio de 1498 foi o primeiro a chegar à Índia [50]) ou Pedro Álvares Cabral (navegador português que descobriu o Brasil a 22 de Maio de 1500 [51]). Cada navegador tem um conjunto próprio de características, que vão afetar de várias formas a tripulação e o decorrer dos eventos e exploração, por exemplo, um navegador pode ter um maior conhecimento sobre os astros e meteorologia, o que faz dele um navegador que se sabe guiar pelas estrelas ou pressentir que uma tempestade está no seu encalço. Isso é então retratado com opções mais benéficas em eventos relacionados com fenómenos meteorológicos como até uma navegação mais rápida. O navegador tem uma evolução individual, no qual aumenta a sua experiência por cada evento que defrontar e por cada expedição terminada. Estas experiências vão modificar ou dar novas características ao navegador, o qual o jogador pode influenciar dependendo das escolhas que tomar durante o jogo. Caso o navegador morra, é oferecido um novo navegador, com novas características e menos experiência, ao jogador quando este regressar á cidade natal.

GERAÇÃO DE CONTEÚDO

A geração automática de conteúdo é uma componente fundamental deste jogo, já que todo o mundo, organização de elementos e gestão de eventos é feito a partir deste.

É de referir que nenhum mundo gerado pelo computador vai conter todos os elementos, estes vão variando a cada novo mundo gerado para oferecer sempre novo conteúdo ao jogador, mas que seja de uma forma significativa para não serem apenas o mesmo mundo com poucas diferenças além da geometria. Assim algumas culturas e certos grupos de eventos só aparecem em alguns *playthroughs*.

Mundo:

A exploração no jogo é a atividade principal do jogador, e para criar o sentido de explorar algo novo é necessário apresentar ao jogo em todos os jogos um mapa novo e interessante. É este o maior foco do PCG neste design, criar um mapa interessante procedimentalmente, o qual vai criar os oceanos, as massas terrestres, as correntes e outros pormenores para criar uma experiência interessante de exploração.

Como cada mundo é diferente podem existir eventos ou fenómenos que não se repetem entre si, tal como a geografia pode criar um impacto em como o jogador vai proceder à sua exploração e interação, o que cria sempre decisões interessantes e inesperadas.

4.2 O ESTUDO PARA UMA NOVA SOLUÇÃO

Ao terminar a primeira fase de design deste jogo foi realizada uma análise mais profunda ao resultado construído. O objetivo desta análise era determinar se devia avançar realmente com esta ideia, ao analisar os seus pontos fortes e fracos e verificar se seria possível realizar um protótipo, o qual fosse prova de conceito às componentes essências do seu design. Além disso, não estava pessoalmente satisfeito com o conceito, não estava confiante que a ideia fosse realmente algo de novo, mas deixei isso para ser verificado na análise.

Desta análise conclui que os pontos fracos eram demasiado significativos para serem ignorados, os quais devia encontrar uma solução antes de avançar para o desenvolvimento do protótipo e um design mais técnico e detalhado. Os pontos fortes é o que deveria ter em mente ao criar uma nova solução.

Os pontos fracos eram dois: O primeiro era o conceito em si, os Descobrimientos Portugueses nunca foram muito abordados em videojogos, mas o tema onde esse está inserido já tem um nicho no mercado. Esse nicho é ocupado com jogos de exploração e de estratégia marítima. Por exemplo, o jogo *Sid Meier's Pirates!* [52] e algumas das versões mais recentes exploram o mito dos piratas durante o século XVI e XVII nas caraíbas, e a suas interações com os vários impérios que exploravam as regiões, como o Espanhol, Francês, Inglês e Holandês. Nesse jogo o jogador representa um capitão de um navio nas caraíbas, o qual pode dar a sua lealdade a um dos impérios e a nenhum, enquanto explora o arquipélago, faz trocas comerciais, entra em combates em mar aberto e assalta outros navios, encontra tesouros, etc. Outro jogo similar a este conceito é o jogo de estratégia *The Seven Cities of Gold* [53], no qual o jogador fica no papel de um explorador espanhol no final do século XV, e toma rumo ao Mundo Novo desde uma cidade



Figura 16. Sid Meier's Pirates! [52]

na Espanha. Ao chegar á costa do novo continente, o jogador vai poder explorar a costa, construir fortes nela e interagir com os povos nativos, ao mesmo tempo que tenta recolher o máximo de ouro possível para agradar à coroa espanhola. No conceito este jogo é bastante parecido ao que está a ser desenvolvido nesta dissertação, mas tem grandes diferenças. O mapa neste jogo é sempre o mesmo, logo o jogador só pode tentar modificar a sua estratégia se repetir os jogos (para resolver isto os designers do jogo lançaram juntamente com este uma ferramenta que permite ao jogador desenhar novos mundos), a interação com o mundo também é realizado de forma bastante diferente. O jogo de grande estratégia *Europa Universalis IV* [54] permite o jogador governar uma nação, maioritariamente europeias, do século XV ao século XIV, no qual está inserido os Descobrimentos. Neste caso, o jogador pode escolher governar o país de Portugal e tomar as mesmas decisões dos portugueses nessa época: descobrir e colonizar o Brasil, Índia e as colónias Africanas.

O segundo ponto fraco é o grande nível de complexidade no design, principalmente no número de sistemas necessários dentro do jogo. Isto poderia criar grandes problemas no desenvolvimento do protótipo, já que o tempo para desenvolver e testar todas as componentes ia ser insuficiente, e não sustentável com uma prática de design e prototipagem iterativa.

Apesar destes dois pontos negativos, existem pontos a favor do design. A exploração de novos mundos é sempre um tema interessante de ser explorado em videojogos, o qual pode ganhar mais valor ao tentar demonstrar o método de exploração dos navegadores portugueses – exploração com um objetivo, não apenas de navegar o desconhecido sem orientação – e isto é algo que devia ser retido no design do jogo.

A cartografia é outro ponto interessante a ser explorado, não é algo muito usado em jogos, contudo aos exemplos em como pode ser aplicado de forma a criar uma maior ligação entre o jogador e o mundo que está a vivenciar pelo seu avatar dentro do jogo. Um exemplo, o jogo do estilo *Dungeon-Crawler* (jogos que têm como base a exploração de masmorras e labirintos) *Etrian Odyssey* [55] para a Nintendo DS aproveita o ecrã táctil da consola para o jogador apontar os mapas das masmorras que está a explorar, ao permiti-lo desenhar numa grelha 2D os tunicas e salas por onde percorreu

de forma a não se perder e manter o conhecimento do que já descobriu, o jogador pode fazer isso ao colocar peças para marcar os corredores e símbolos a indicar onde estão perigos, tesouros e outras informações uteis.

A tarefa seguinte foi encontrar uma solução para dois pontos fracos do design deste jogo. A complexidade tem uma solução simples, mas que requer criatividade. Essa solução é que o design seguinte fosse “desmontado” até só manter o essencial. Para o tema do jogo foi necessária uma procura de novas ideias, e



Figura 17. Etrian Odyssey 2 [55] - Ferramenta para desenhar mapas

algo que foi também colocado em conta era arranjar um conceito para o jogo o qual permite-se um maior espaço de manobra do que ficar preso à realidade do nosso planeta e história.

Essa procura levou ao encontro de vários autores e obras que inspiraram em várias áreas o que seria o próximo design do jogo a desenvolver para esta dissertação.

O primeiro autor a referir, o qual nesta fase do desenvolvimento estava a ler uma das suas obras, é Jean Henri Gaston Giraud, conhecido como Moebius, um aclamado artista e criador de bandas desenhadas francêss. Os mundos surreais e coloridos criados nas suas bandas desenhadas foram a inspiração retirada dos seus projetos. Uma das obras, principalmente, foi uma das que mais inspirou o conceito seguinte desenvolvido para o jogo a desenvolver. Esta obra foi “Arzarch” [56], publicada a partir 1975, conta uma história só a partir de arte gráfica, a qual começou sem balões de conversação ou legendas – mas que ao lançar novos capítulos desenvolveu uma história com conversações – no qual um guerreiro sentado sobre uma espécie de pterodáctilo explora um estranho mundo desolado mas colorido, com estranhas formações rochosas, florestas de plantas exóticas, enormes criaturas e outras edificações saídas de ficção científica futurista. “L’Incal”



Figura 18. Arzarch [56] de Moebius - uma imagem da banda desenhada

[57], publicado de 1981 a 2014, escrito juntamente com o artista Alejandro Jodorowsky [57], continua a exploração de estranhos mundos futuristas como mostrado em “Arzarch”. Além dos das personagens e criaturas que habitam esse mundo, as peças de tecnologia como veículos, robots e arquitetura também foram uma inspiração para a arte do jogo. A história também tem os seus momentos que inspiraram, já que decorrem na serie fenómenos e manipulações psicológicas que nunca poderiam existir no nosso mundo, mas interessantes a explorar em game design (por exemplo brincar com os sentidos do jogador).

Duas outras obras literárias que ajudaram a aprofundar este interesse da exploração de outros mundos, são os livros da serie “Hyperion Cantos” do escritor Dan Simmons [58] e a obra “Dune” do escritor Frank Herbert [59]. Estas obras narram sobre universos no qual fenómenos surreais geram histórias cativantes e originais, as quais podem ser interessantes ao serem abordadas num meio interativo como videojogos. No livro “Hyperion”, o

primeiro livro dos “Hyperion Cantos”, por exemplo, é relatado sobre um local chamado de Time Tombs (Túmulos do Tempo) que geograficamente se localiza num planeta com o mesmo nome da obra, Hyperion, onde o tempo retrocede de um futuro distante para o passado, enquanto que o resto do mundo continua com uma progressão normal para o futuro. Nesse mesmo local reside uma criatura semi-deus intitulada de Shrike. Durante a narrativa, esta criatura, “brinca” e desafia psicologicamente com várias personagens presentes no livro, um grupo de peregrinos que têm como objetivo encontrar essa mesma figura. Esta abordagem de manipular personagens é interessante, foi visto como uma possível interação do computador com o jogador dentro do jogo, utilizar jogos psicológicos contra o jogador além dos obstáculos e armadilhas físicas que normalmente se encontram em videojogos – assim foi visto como um dos maiores possíveis objetivos do conceito, aliar um mundo desconhecido a uma ameaça maior no qual o jogador não está à espera que se brinquem com os sentidos deste.

No caso de “Dune” são as culturas e história criadas de volta do planeta Arrakis, um planeta na sua maioria coberto por um deserto, onde um povo nativo tribal chamados de Fremen recolhe uma espécie de droga chamada Melange do deserto, este habitado por criaturas gigantes chamadas de Sandworms. Esta obra mostra um grande trabalho na produção de uma cultura e mitologia criado de volta dos habitantes desse planeta, como por exemplo os Fremen verem as criaturas Sandworm como uma representação da vontade dos seus deuses, principalmente como estas ajudam na produção da droga. A droga pode ajudar também a dar algumas ideias de como brincar com as percepções do jogador, já que o seu uso no livro tem vários efeitos, como aumentar a duração de vida em centenas de anos, melhorar a saúde e aumentar a sensibilidade dos sentidos. O uso excessivo da droga causa também alucinações e até permite uma pessoa conseguir aceder à sua memória genética.

Além de obras literárias, as pinturas do artista holandês Hieronymus Bosch criam também imagens do fantástico baseados em temas religiosos. O quadro tríptico “O Jardim das Delicias Terrenas”, uma das suas obras mais conhecidas, e também uma inspiração no tipo de imagem procurado para este projecto, principalmente na forma como são pintadas as várias

estruturas, muitas delas inspiradas em formas geométricas e biológicas.

Desta exploração surgiu assim um novo tema e ideia a adaptar juntamente ao que restou da ideia inicial. Foi trocado um planeta como a Terra para um longe do nosso real, e foi trocado a faixa de tempo consistente com os Descobrimentos, por um futurismo o qual não se pode datar, que permite uma maior lista de possibilidades tanto em temas estéticos, tecnológicos e de interações com o jogador, de forma a permitir um jogo mais simples, mas com um grande número de decisões interessantes. Assim temos uma ideia a explorar sobre a exploração de um planeta desconhecido, o qual vai jogar contra o jogador, tanto ao criar desafios físicos com este, como manipular as suas percepções.

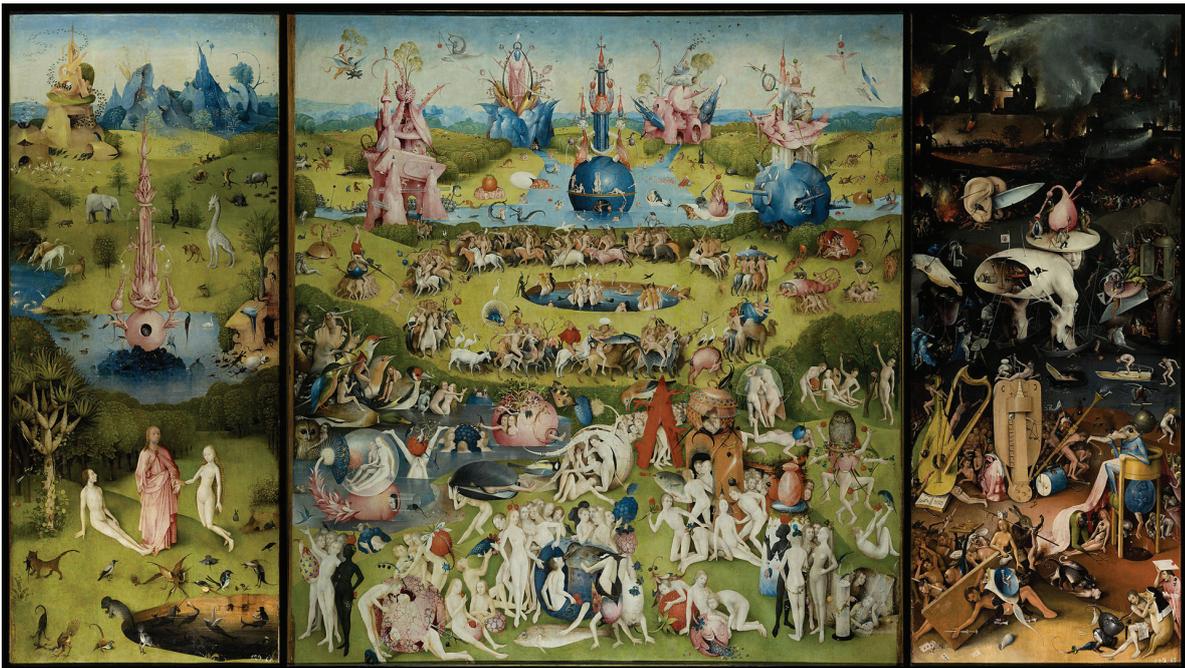


Figura 19. O Jardim das Delícias Terrenas de Hieronymus Bosch (1503-1515)

4.3 O JOGADOR CONTRA O PLANETA: ADAMASTOR

Este novo design do jogo tenta dar resposta como solução aos problemas encontrados no design anterior. Foi decidido num novo conceito, que se pode resumir como Jogador VS Planeta, o qual explora um conceito com um maior leque de oportunidades ao fugir do realismo histórico. Além disso, este novo design consegue invocar um desafio de forma mais simples, ao conter um menor nível de componentes dentro do jogo, ao cortar elementos desnecessários do design anterior, como por exemplo ao retirar a noção de eventos, tripulação e explorador. Ao criar este desafio, é possível também no seu design ter uma ligação mais próxima ao conceito de *Generative Game Design*, ao gerar conteúdo também durante o gameplay e não só para criar o mundo.

CONCEITO

Este jogo decorre num planeta longínquo, no qual o jogador, alienígena a este astro, aterra na sua superfície em busca de um artefacto escondido. O grande desafio, além do mistério inerente ao estar perdido num território inexplorado, é que o planeta contém traços de se encontrar vivo, tem um nível de “sapiência” e mostra isso ao tentar impedir o jogador de obter o artefacto. Este intento do planeta é expresso pela forma como vai manipular a sua própria geografia e as percepções do jogador para que este não consiga atingir o seu objetivo. Neste conceito mantém-se a ideia do método de exploração dos portugueses, no qual é favorável ao jogador regressar à sua nave-mão após uma expedição para recuperar recursos e guardar conhecimentos sobre o planeta. Ter conhecimento armazenado na nave-mão sobre o planeta, impede que ele se modifique nos locais conhecidos

O JOGO

Visão sobre o mundo: O mundo do jogo vai ser feito a partir de uma matriz 2D, onde as dimensões XY representam o terreno do mundo. Este tipo de registo do terreno é muito usado em jogos com visão *Top-Down* (vista de cima para baixo), vista em muitos jogos de estratégia ou roguelike como o jogo *Rogue* [24], ou com uma perspectiva isométrica frequentemente utilizada em RPGs de estratégia desenvolvidos no Japão, como *Final Fantasy Tactics* [60]. Este tipo de visão sobre o mundo consegue também mostrar várias informações sobre o mundo, apesar de não ter o mesmo detalhe ou profundidade de um mundo 3D, a arte do jogo pode compensar esse problema. O tipo de visão decidido para este jogo é *Top-Down*, mas espera-se que durante a experimentação da arte gráfica se decida se os elementos vão representados de cima para baixo ou que tenham um misto de isométrico para criar volume.

Jogador: O objetivo do jogador é encontrar um artefacto perdido no mundo onde se encontra, e vai tentar cumprir esse objetivo a partir do ato de exploração do ambiente que o rodeia, desde o momento em que parte da nave-mãe onde aterrou no início do jogo. O jogador tem vários veículos a seu dispor, logo não perde o jogo por ser destruído apenas uma vez.

A exploração é feita a partir do controlo de um veículo híbrido, que se movimenta em terra e em água, movimentado a partir da energia do vento, o qual o jogador consegue manipular. O mundo gerado vai conter vários tipos de terreno, os quais vão beneficiar ou prejudicar o movimento do veículo, por exemplo florestas podem atrasar a sua locomoção, mas correntes marítimas no oceano podem ser uma grande vantagem se se deslocarem no sentido em que o jogador viaja.

O jogador necessita também de recursos para conseguir sobreviver nesse planeta, esses são recolhidos durante a viagem e armazenados dentro do veículo, até atingirem o número de unidades limite. Esses recursos podem também ser reestabelecidos quando este regressa à base.

O jogador não se encontra totalmente desfavorecido em relação ao planeta e tem uma arma em seu proveito. Este consegue manipular a força e direção do vento, limitado por uma fonte que recupera ao longo do tempo, assim se não tiver energia suficiente guardada na fonte, durante algum tempo

não vai conseguir modificar o vetor do vento. É a partir deste controlo do vento com o qual o jogador se vai movimentar e interagir de várias formas no planeta, como por exemplo fugir de armadilhas que prendam o veículo a partir de uma força excessiva do vento.

Ademais em seu proveito encontra-se a força do conhecimento, ao explorar o planeta o jogador adquire informação sobre o terreno que descobriu, e este ao ser armazenado na nave-mãe, impede que o planeta modifique a sua geografia nesses locais ou use a maior parte dos seus truques mentais para desafiar o jogador. Este fator é importante, se o veículo do jogador for destruído antes de regressar à base, este perde todo o conhecimento adquirido, o qual permite que o planeta, assim que o jogador regressar com um novo veículo, modificar esses locais.

Planeta: Este é o único inimigo do jogador e o gatilho para a ação no jogo. O planeta é também o controlador da Geração Procedimental de Conteúdo, tanto *offline* como *online*. No início de cada jogo um novo planeta vai ser gerado, com um terreno diferente dos outros, mas este não é fixo, o que o torna um desafio é como vai manipular a sua própria geografia durante o jogo para criar obstáculos ao jogador.

A geração do mundo vai ter em conta fatores como altitude e precipitação para decidir onde vai colocar os vários tipos de terreno. Grandes altitudes criam cadeias montanhosas, zonas com muita precipitação criam florestas, a junção de ambas inicia nascentes de rios. A altitude no geral também vai decidir onde vai ser mar, costa ou planícies. O artefacto vai ser colocado aleatoriamente no mapa, e os recursos *pseudo* aleatoriamente, mas de forma a não se agruparem todos no mesmo local.

O desafio criado vai ser apresentado de diversas formas, cada uma delas a gerar um problema diferente ao jogador, desde mudanças simples como alterar a direção das correntes marítimas, criar novas barreiras naturais que impeçam o movimento do jogador, alterar a sua geografia quando o veículo é destruído, até gerar desafios com uma dificuldade elevada, como tempestades que atrapalham o uso do vento ou vulcões que podem eliminar instantaneamente o veículo.

Contudo esta não é a sua única arma, o planeta vai brincar com as perceções do jogador, como limitar-lhe o campo de visão ao tapar a luz

solar, modificar o comportamento do terreno (o jogador está a ver o veiculo a movimentar-se em terra, mas o comportamento e efeitos sonoros comportam-se como água), até teletransportar o jogador para outro canto do mapa sem que este dê conta.

A dificuldade também vai ser manipulada ao longo do jogo, quanto mais próximo o jogador se encontra do artefacto, maior os obstáculos que o planeta vai gerar, isto serve como indicador ao jogador se ele vai na direção correta. Tal como o planeta vai gerar dificuldade, este também acaba por dar algum apoio ao computador, quando este modificar o terreno pode criar recursos que o jogador pode recolher.

CENÁRIOS E COMPONENTES

Veículo do jogador: veículo que o jogador controla;

Base do jogador: base do jogador no mapa;

Elementos da interface do jogador: Indicador do estado do veículo;
Indicador da quantidade de poder de vento que pode usar; Indicador da quantidade de recursos em posse do jogador; Indicador da direção e força do vento que o está a movimentar.

Tiles (peças do terreno) do planeta e os seus efeitos no jogador:

Nome	Descrição	Efeitos
Planície	Planície, tile neutro	n/a
Floresta	Área com vegetação, pode sobrepor-se ao <i>tile</i> de planície e costa	Velocidade mais lenta para o jogador
Costa	Faz ligação entre terra e mar	n/a
Oceano	Área aquática	Tem uma corrente com força e direção variável que afeta o movimento do jogador
Rio	Área aquática de menor dimensão	Tem uma direção fixa com uma força variável que afeta o movimento do jogador
Rochas	Terreno rochoso	Danifica veículo do jogador
Montanha	Zona montanhosa de alta altitude	Barreira; Jogador não consegue passar
Fenda	Fenda na terra	Barreira; Jogador não consegue passar
Lava	Zona de erupção vulcânica, fogo. Pode sobrepor-se a qualquer tile	Danifica e destrói o veículo do jogador
Geiser	Formação geológica, erupção de água.	Danifica o veículo do jogador
Remoinho	Depressão aquática, pode sobrepor-se a <i>tiles</i> de oceano ou rio	Puxa o jogador para o seu centro, o que acaba por destruir o veículo do jogador.
Recursos	Pode sobrepor-se a qualquer <i>tile</i> que não seja uma barreira ao jogador. Representam recursos que o jogador pode recolher. A sua representação varia dependendo da quantidade de recursos.	Fornecer um variável número de recursos ao jogador
Artefacto	Pode sobrepor-se a qualquer <i>tile</i> que não seja uma barreira ao jogador. Representam o artefacto objetivo do jogador	Termina o jogo, vitória do jogador
Base do jogador	Representa a base do jogador e a posição do jogador no início do jogo	Recupera recursos; Guarda o conhecimento sobre o planeta recolhido pelo jogador;

ESPAÇO DE AÇÃO DO JOGADOR

- **Aplicar uma corrente de vento:** Input de slide sobre o ecrã (mesmo movimento como por exemplo arrastar um ícone num ambiente de trabalho) com o indicador da plataforma onde está a jogar, no qual a direção do vetor criado pelo movimento de slide indicam a direção da força do vento, e a distância do ponto onde começa ao ponto onde termina o input a força do vento; esta direção e força do vento movem o veiculo do jogador;
- **Colisão** sobre os obstáculos no mundo;
- **Recolha** de recursos ao colidir com estes no mapa;
- **Recolher** o artefacto para vencer o jogo;
- **Regressar à base** para recuperar recursos e guardar conhecimento;

ESPAÇO DE AÇÃO DO PLANETA

- **Geração** do mapa no inicio do jogo;
- **Avalia** o estado do jogo:
 - ◇ **Verifica** posição do jogador;
 - ◇ **Verifica** em que direção se movimenta o jogador;
 - ◇ **Verifica** tile onde está o jogador;
 - ◇ **Verifica** condição do jogador;
 - ◇ **Verifica** distância do jogador ao artefacto;
 - ◇ **Verifica** desafio (ou dificuldade) estimada do jogo no momento;
 - ◇ **Verifica** quando foi despoletado o ultimo trigger (critérios que iniciam uma ação) de desafio;
- **Aplica** desafios ao jogador dependendo da situação (os desafios estão representados na grelha em baixo);

Desafios (ou eventos) criados pelo planeta:

Nome	Descrição	Objetivo	<i>Trigger</i> (para activar um dos desafios todos os critérios têm que estar satisfeitos)
Modificação geográfica	Modifica totalmente ou parcialmente a geografia das zonas exploradas. Pode ocorrer aleatoriamente mas acontece sempre após a morte do jogador, sobre o território que este explorou nessa expedição mas a qual o conhecimento não conseguiu guardar.	Aumentar desafio ao criar uma incógnita no jogador sobre as áreas que tinha explorado, mas agora podem ter sido modificadas;	Pode ser aplicado em qualquer altura; Morte do jogador;
Criar barreiras	Cria novas cadeias montanhosas ou desfiladeiros em frente e á volta da zona que o jogador está a explorar.	Criar barreiras e obstáculos para impedir ou limitar a exploração do jogador nessa direção.	Não tem critérios, pode ser aplicado em qualquer altura;
Modificar correntes marítimas	Vai modificar as correntes marítimas na área do oceano onde se encontra o jogador.	Cria desafio ao movimento do jogador.	Jogador presente no oceano;
Criar vegetação	Cresce zonas florestais em volta do jogador	Cria desafio ao movimento do jogador	Jogador presente em terra;
Criar vulcões	Cria uma grande atividade sísmica em volta do jogador. Este efeito é retratado pelo aparecimento de desfiladeiros e montanhas em redor do epicentro, numa localização perto do jogador, destruição do bioma em redor (por exemplo queimar zonas florestais ou criar ilhas em volta do oceano), e também de geiseres e erupções vulcânicas que podem destruir o veículo do jogador (estas zonas têm um período de crescimento, logo o jogador tem possibilidade de fugir)	Criar desafio ao jogador, dar uma pista que este se aproxima do objetivo já que é uma ameaça muito maior. Criar um momento de grande desafio já que pode ser fatal para o jogador.	Jogador encontra-se perto do artefacto;

Criar corpos de água	Cria novos corpos de águas em frente do caminho do jogador. Por exemplo pode criar a formação de rios, lagos ou expandir oceanos.	Criar desafio no movimento do jogador.	Jogador presente em terra; Jogador está a média distância do artefacto;
Alteração no	O planeta vai alterar os efeitos e som do terreno do planeta, mas ao manter a aparência. A terra vai-se comportar como oceano e o oceano vai-se comportar como terra,	Criar confusão e desafio ao jogador.	Jogador está a média distância do artefacto;
Limitar visão	A luz solar é diminuída temporariamente, este fica com um campo de visão muito menor que o normal, só consegue ver o que está próximo.	Criar desafio na locomoção do jogador. Fazer com que este explore mais devagar e com mais atenção. Duração de alguns segundos até 2 minutos;	Jogador está a média distância do artefacto;
Teletransporte	O planeta aproveita o efeito de Limitar Visão para teletransportar o jogador para um outro ponto no mapa semelhante ao qual o jogador se encontra (os <i>tiles</i> em redor são similares);	Cria um momento de grande desafio ao jogador, faz com que este se sinta perdido quando retoma outra vez a visão completa.	Limitar Visão; Jogador próximo do artefacto;
Remoinho de água	Cria uma espiral no oceano, esta aplica uma força de atração muito forte sobre o veículo do jogador;	Cria um desafio no movimento do jogador com um potencial mortal.	Jogador está presente no oceano; O jogador está a uma média distância do artefacto;
Tempestade	Cria uma tempestade na área circundante do jogador.	Cria dificuldade no movimento.	Jogador está a uma média distância do artefacto;

O NOME ADAMASTOR

Ao construir este conceito surgiu também um nome para o jogo, o qual fazia falta. O nome dado ao jogo é “Adamastor”, o qual faz uma ponte entre a primeira ideia dos Descobrimentos e esta nova ideia do planeta contra o jogador. Adamastor é uma figura da mitologia greco-romana, adaptada por Camões na obra “Os Lusíadas” [61]. Nesta obra o Adamastor personifica as tempestades que Vasco da Gama teve que enfrentar ao passar o Cabo da Boa Esperança, tempestades estas que tentavam impedir qualquer navegador que quisesse passar do Oceano Atlântico para o Oceano Índico, logo um acontecimento dramático que retrata as forças da natureza contra o homem, ou como nesta ideia do jogo, o planeta contra o homem.

4.4 TECNOLOGIA

O desenvolvimento de um videojogo é sempre uma junção do resultado de artefactos de várias áreas, como imagem, som, lógica, etc. Cada um desses artefactos é o resultado da utilização de vários meios, seja software ou ferramentas analógicas para criar tanto a arte ou som, como uma das várias linguagens possíveis de programação. Assim é sempre um desafio escolher quais as melhores tecnologias a escolher para produzir cada um destes artefactos. Com a grande variedade de tecnologias nos dias de hoje trazem um grande benefício para a área dos videojogos, essa variedade traduz-se numa maior escolha de ferramentas que uma pessoa pode escolher, algumas delas ferramentas gerais que permitem desenvolver várias componentes (por exemplo o motor de jogos Unity 3D [62] o qual permite ao utilizador criar um jogo do início ao fim sem necessitar de componentes externas), como também ferramentas especializadas para problemas de design específicos (por exemplo SpeedTree [18], o qual permite criar e animar vegetação para videojogos e cinema).

No caso da produção do videogame para este projeto, é necessário escolher as ferramentas para produção nas seguintes três áreas: a lógica, a arte e o som.

Lógica: considero a área da lógica o cérebro e corpo do videogame, o elemento que vai juntar todas as componentes necessárias a ter uma aplicação interativa. Ao escolher a ferramenta como solução a este problema, tive em conta o design do videogame. É necessária uma ferramenta que dê um largo espaço de manobra, ou que não imponha muitos limites à experimentação de vários algoritmos de geração de conteúdo e que permita rapidamente os testar num contexto de um videogame. A ideia também era produzir um jogo 2D, assim foi dada prioridade a ferramentas que permitam a criação de jogos 2D.

Isto pode ser resolvido a partir de várias tecnologias, desde a programação do próprio motor de jogo sem auxílio de *Frameworks* ao uso de *Game Engines* (motores de jogo) de terceiros. Mas tendo em conta o curto espaço de tempo existente para a produção de um protótipo para a dissertação, eliminei todas as opções que não permitissem uma rápida aprendizagem ou produção de videogames. O foco da investigação foi em bibliotecas completas no desenvolvimento de videogames e também em motores de videogames completos. Esta dissertação foi iniciada já com alguma experiência prévia em desenvolvimento de aplicações interativas (algumas baseadas em motor de videogames) e videogames, assim software no qual já existe experiência foram beneficiados na escolha.

Devido à grande oferta de ferramentas existentes, para ajudar na pesquisa foi selecionado um grupo restrito de software que correspondesse a determinados critérios:

- Suporte a videogames em 2D;
- Suporte a linguagens de programação que permitam o uso de algoritmos de Geração Procedimental de Conteúdo;
- Suporte multiplataforma (de preferência Windows e Android);
- Bom nível de suporte ao desenvolvimento, como por exemplo documentação técnica, fóruns, wikis, sites de perguntas e respostas (como Stack Overflow [63]);
- Tem que ser open source ou ter um tipo de licença gratuita.

A partir destes critérios foi fácil reduzir a lista a um grupo seletivo de software, que permita uma melhor análise de cada ferramenta.

(Nota: uso GameEngine na lista em vez de Motor de Jogo pela facilidade da escrita)

GameEngine/Framework	Vantagens	Desvantagens
Cocos2D-x (<i>Framework</i>) [64]	Boa documentação; Ferramenta poderosa para jogos 2D; Permite programação em javascript; Fácil de exportar o jogo para as várias plataformas;	Comunidade pouco ativa; Apesar de ser uma ferramenta poderosa algumas funcionalidades necessitam de ferramentas de terceiros (por exemplo para criar partículas ou compactar texturas);
Phaser (<i>Framework</i>) [65]	Programação em javascript; Prototipagem rápida; Boa documentação; Fácil aprendizagem; Comunidade activa;	Não é uma ferramenta aconselhável a PCG pesado;
Unity 5 (<i>GameEngine</i>) [62]	Experiência prévia; Boa documentação; Vasta comunidade ativa; Permite programação em c# e javascript; Bom suporte 2D; Fácil de exportar o jogo para as várias plataformas; Ferramenta poderosa, não impõe grandes limites; Prototipagem rápida;	Apesar de ter um bom suporte 2D, ainda tem alguns problemas ao criar jogos 2D (principalmente a nível gráfico) devido a ser primariamente um motor 3D;

Unreal Engine 4 (<i>GameEngine</i>) [66]	Experiência prévia; Boa documentação; Comunidade ativa; Ferramenta poderosa, não impõe grandes limites; Prototipagem rápida;	Apenas permite programação em C++ e blueprint (programação gráfica); Ferramenta por vezes muito complexa, tem uma grave curva de aprendizagem;
---	--	---

A ferramenta escolhida foi o motor de jogos Unity 5, a escolha foi feita devido à experiência prévia que tinha da ferramenta, o facto de já conhecer as suas limitações e permitir que pudesse transportar para um videojogo quaisquer algoritmos ou ideias que tivesse em mente. O facto de ter uma comunidade ativa e boa documentação acrescentam também confiança na ferramenta, no caso de ter algum problema técnico durante o desenvolvimento acredita-se poder encontrar rapidamente uma solução junto da comunidade.

Arte: a escolha para as ferramentas de para produzir a arte gráfica para o videojogo foi decidida a partir de experiência prévia e o nível de conforto que adquirido com ferramentas que fazem parte do percurso curricular na área do curso do âmbito desta dissertação. Assim, além de grafite, caneta e papel para os esboços do material gráfico necessário, foi escolhido o software Adobe Photoshop CC [67] para produção de arte digital para o videojogo, já que é dos produtos mais poderosos no mercado e permite a criação e edição de imagens, como também tem ferramentas necessárias a criar animações. Se for necessária uma ferramenta mais poderosa para criar animações, o motor de jogo escolhido Unity 5, tem várias ferramentas de animação incorporadas.

Som: o som segue o mesmo caminho na decisão da ferramenta escolhida para o material gráfico. O software foi também escolhido por experiência prévia e nível de conforto adquirido. A escolha foi o software open-source Audacity [68], uma ferramenta simples e poderosa para a edição de clipes sonoros.

4.5 PRODUÇÃO DO PRIMEIRO PROTÓTIPO

PLANEAMENTO

No momento em que tinha o design do jogo e as tecnologias selecionadas, iniciou-se o planeamento da produção da primeira versão do protótipo. O objetivo do primeiro protótipo não é para ser uma versão completa do jogo, mas sim um exemplo mais simples, com o qual rapidamente se pudesse testar a partir de sessões playtesting o design do jogo, e aproveitar também essa simplicidade para aplicar rapidamente as alterações necessárias, encontradas a partir da análise dos resultados das sessões de playtesting. Assim, o necessário a ser criado para este primeiro protótipo é: o gerador do mundo, tiles de terreno o suficientes para criar um mundo coeso (os tiles escolhidos foram o de planície, oceano, costa, montanha e artefacto, que dão um exemplo de quase todas as interações que existem entre o jogador e o terreno, como também permitem terminar o jogo), um método de avaliação do mundo para gerar desafios, o movimento do jogador no mundo, interações básicas como colisões e correntes marinhas e três desafios (Modificação geográfica após a morte, Criação de barreiras e Alteração do comportamento do terreno) com que se pudesse testar a jogabilidade no jogo com algum desafio.

Para planear a ordem com que todas as componentes deste primeiro protótipo iam ser desenvolvidas, estas foram divididas em grupos de tarefas gerais, cada uma ligada a uma determinada secção do jogo. Isto facilitou a ordenação das tarefas a partir da sua importância de desenvolvimento (o que é mais importante produzir primeiro), como também posteriormente durante o desenvolvimento do protótipo, permitiu dividir essas tarefas

em várias sub-tarefas e criar um diagrama de Gantt o qual era usado para organizar as tarefas numa linha temporal (de estimação de tempo de quando se começava e terminava uma tarefa e a data real de quando eram completas) como também o seu estado, se estavam por fazer, incompletas ou incompletas.

A primeira tarefa a ser cumprida tinha que ser o gerador do mapa do mundo, a base do jogo onde se desenrola a ação. Assim encontrar o melhor algoritmo de PCG que respondesse ao design do jogo, aplica-lo para gerar mundos dentro do Unity, como também aplicar os vários tipos de tiles de terreno nesse gerador. A segunda tarefa, com o mapa do mundo já concluído e pronto a testar, é criar as interações do jogador com o jogo: criar o movimento do jogador a partir do vento, aplicar as colisões do jogador com o mundo, e terminar o jogo ao encontrar o artefacto. A terceira tarefa é a interação do jogo com o jogador, a começar como os vários tipos de terreno podem influenciar o jogador (como as correntes marítimas), criar um método que permita o jogo avaliar a ação que está a decorrer e finalmente os três desafios selecionados. A última tarefa seria um rápido estudo sobre a arte do jogo, criar vários elementos gráficos que representem um exemplo do que é esperado para a arte do jogo, e que sirvam como placeholders para um desenvolvimento rápido do jogo. Decidiu-se também deixar função de cartografia, ou registar conhecimento sobre o planeta, para a segunda iteração do protótipo, já que necessitava de testar primeiro o jogo com os desafios para verificar se era necessária alguma modificação a esta componente, como por exemplo a área em redor do jogador que é considerada como descoberta, ou se durante o teste de jogo se nota-se que em determinadas situações ou desafios era uma mais valia o jogo poder criar alguma interação sobre o que foi descoberto.

A lista seguinte descreve os grupos e as suas tarefas, já organizado pela sua ordem de importância:

- **Geração do Mundo de jogo (PCG offline):**
 - a. Seleção do algoritmo para gerar o mundo;
 - b. Aplicação do algoritmo para gerar o mundo;
 - c. Aplicar os tiles definidos ao algoritmo;

- **Interação do jogador com o jogo:**
 - a. Movimento do jogador;
 - b. Colisões com o mundo;
 - c. Terminar o jogo ao encontrar o artefacto;
- **Interação do jogo com o jogador (PCG online):**
 - a. Aplicar modificadores de movimento de cada tipo de terreno ao movimento do jogador;
 - b. Criar um método para avaliar a ação do jogo;
 - c. Criar cada um dos desafios que o planeta pode iniciar;
- **Arte:**
 - a. Design do veículo do jogador;
 - b. Design de cada um dos tiles;
 - c. Design do artefacto;

PRODUÇÃO

Geração do Mundo de jogo (PCG offline):

A primeira tarefa era encontrar uma combinação de algoritmos na qual a sua combinação resulte num sistema de PCG que consiga gerar um mapa completo do mundo do jogo, numa matriz 2D, a qual possa ser representada numa vista *top-down*. Durante a investigação feita para o estado da arte encontrei vários algoritmos que criam estruturas interessantes em matrizes 2D, algumas destacam-se exatamente como solução para este tipo de problema. Neste caso, a ideia para o algoritmo era usar um sistema sobreposto de várias matrizes criadas a partir de algoritmos PRNG (*Pseudo-Random Number Generatos*) [15], neste caso matrizes geradas a partir de Perlin Noise [29].

O sistema de PCG criado para gerar o mundo foi composto por:

- Uma matriz gerada por Perlin Noise[29] que representa um mapa de altitudes. Cada casa da matriz contem um numero que vai de 0 a 1, e que dependendo das variáveis decididas pelo gerador ao criar o mapa criam um limite para o que é oceano, um limite para o que é costa, um limite para o que são planícies e um limite para o que são montanhas (por exemplo, casas que tenham um numero menor que 0.2 representam oceano, o que provavelmente vai gerar um mapa com poucos oceanos);

- Uma segunda matriz de Perlin Noise cria um mapa de precipitação, o qual estabelece um limite de 0 a 1, no qual um valor de um casa seja superior a esse, e se na matriz do mapa de altitude essa casa for uma planície ou costa, então gera nessa casa uma floresta;
- A terceira matriz de Perlin Noise gera o mapa para as correntes marítimas, onde cada valor de 0 a 1 está mapeado para ter o valor de uma rotação de 0 a 360°. Esta matriz acaba por criar rapidamente correntes marítimas semi-realistas.
- Uma matriz onde cada casa consegue guardar a informação do tile em questão, como os valores de cada uma das três matrizes, o tipo de terreno, se for oceano coloca também a força da corrente marítima e se é ou não uma barreira;
- O artefacto vai ser colocado numa casa aleatória que não seja uma barreira, e ao qual o jogador tenha acesso;
- O jogador vai ser colocado no mapa com uma distância mínima do artefacto. Essa distância vai depende do tamanho do mapa.

Para gerar os mapas em Perlin Noise baseei-me no código `c#` criado por Sebastian Lague [69], o qual fornece um método que permite gerar mapas baseados numa matriz de Perlin Noise com várias variáveis que modificam o mapa, o que é uma mais valia para um maior controlo do gerador sobre o conteúdo criado. Essas variáveis são a altura e largura do mapa, as oitavas (camadas individuais de ruído), lacunaridade (controla o aumento de frequência de cada oitava), persistência (controla a diminuição de amplitude de cada oitava). Um exemplo simples sobre estes valores, um mapa baseado apenas na matriz de alturas, se tiver um baixo valor de persistência, e um valor mais elevado de lacunaridade, vai criar um mapa composto principalmente de grandes massas de valores extremos, vai ter portanto grandes cadeias montanhosas, grandes oceanos, e pouco terreno de costa ou planície. Como o código era facilmente manipulável, consegui assim sem perder muito tempo produzir três matrizes diferentes, cada uma com os seus próprios valores nas variáveis para gerar diversos mapas.

Desse mesmo projeto de Lague aproveitei um método para rapidamente criar uma versão teste do mapa numa malha 3D, onde cada par de triângulos da malha representa uma cor (o tipo de terreno). Mais tarde durante o

desenvolvimento do protótipo tive que substituir esse código da malha, já que não permitia um mapear texturas sobre ela.

Apesar disso continuei o uso de uma malha 3D, a qual escrevi um código que permitisse mapear texturas nela, em vez de usar um método alternativo criar um objeto no Unity que represente cada um dos tiles. Apesar de ser mais simples e dar um melhor controlo criar um objeto por cada tile, a partir do momento em que ia ter um mapa que fosse mais de 50 por 50 tiles, isso iria exponencialmente aumentar o numero de recursos necessários para correr o jogo. Ao desenvolver o mapa apontei sempre gerar mapas ainda de um tamanho considerável, de 200 por 200 tiles. A diferença entre criar $200 \times 200 = 40000$ objetos dentro do Unity, que estavam a ocupar uma grande quantidade de memória ram, quase 1 gigabyte de memória, e usar uma malha onde estava mapeado em cada par de triângulos para uma secção de uma só textura (um tileatlas, nome dado a um só ficheiro, por exemplo do tipo .png, que contém várias texturas a ser usadas em modelos num jogo) que num mapa do mesmo tamanho apenas gastava 13 megabytes de memória, é um diferença muito grande em termos de performance e tempo para gerar o mapa.

Ao aplicar as texturas surgiu outro desafio, que foi criar um método que se seleciona a peça correta dependendo da posição. O exemplo disto é termos uma casa de costa, que está rodeada no topo e pela esquerda de água, e na direita e por baixo de outras peças de costa, neste caso a peça a ser escolhida teria que ser um canto superior esquerdo. Antes de ser desenvolvido o algoritmo teve que se criar as peças para cada tipo de terreno. Para cada tipo estas eram:

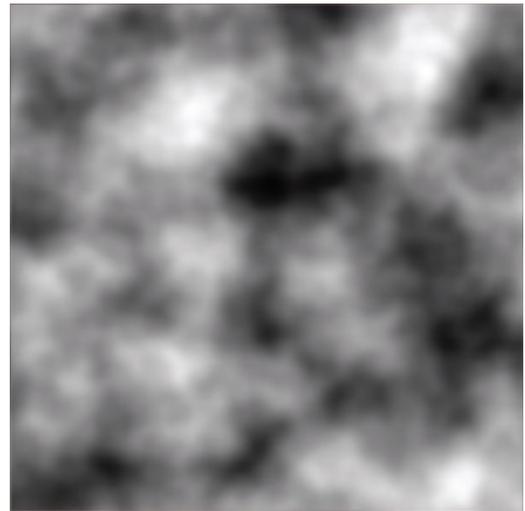


Figura 20. Exemplo de *Perlin Noise* gerado pelo sistema de PCG

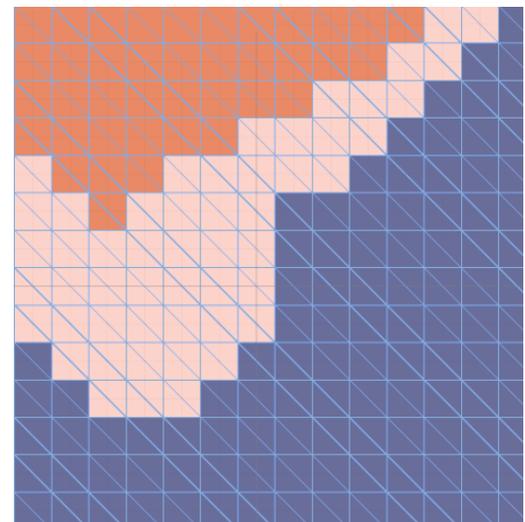


Figura 21. Malha 3D onde é texturado o mapa

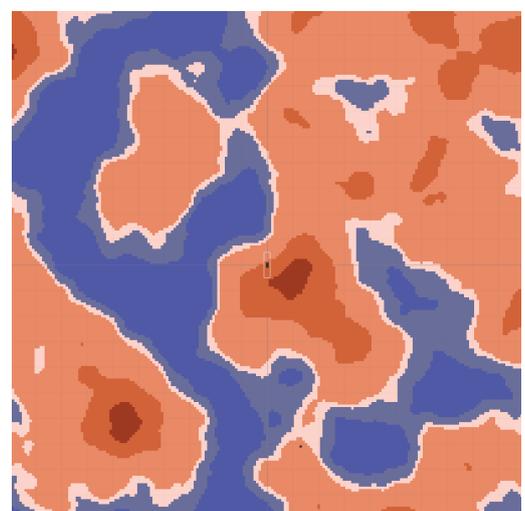


Figura 22. Mapa gerado com vários tipos de terreno

4 peças diferentes para representar o centro (de forma a criar alguma variedade nas peças mais repetidas, que são as que preenchem o interior), canto superior esquerdo, limite superior, canto superior direito, limite direito, canto inferior direito, limite inferior, canto inferior esquerdo, limite esquerdo, casa isoladas (que 3 casas adjacentes sejam de um tipo de terreno diferente) superior, inferior, esquerda e direita.

O algoritmo criado verifica cada casa da matriz 2D do mundo, e verifica as 4 casas adjacentes a esta, a casa que se encontra a norte, sul, este e oeste. Se as 4 casas fossem do mesmo tipo, então o algoritmo seleciona uma peça das várias peças interiores, se não seleciona um dos cantos. Caso uma ou mais das casas adjacentes seja de um tipo diferente, o algoritmo vai comparar a altitude da casa que esta a verificar com essas peças adjacentes. Se tiver uma altura superior, então vai escolher uma das peças que representam um canto ou limite, se a altura for inferior, vai então escolher uma peça interior. Esta escolha de só usar limites e cantos nas peças de altura superior é para ajudar a representar as dimensões do mapa sem ser necessário uma terceira dimensão, já que o jogador dá conta que só os terrenos mais altos é que mostram esta diferenciação para os mais baixos.

Caso a peça seja de uma maior altitude, e esteja só adjacente a uma casa diferente, vai então escolher uma das peças que represente um dos limites (superior, inferior, esquerdo ou direito). No caso de a peça ter duas casas adjacentes diferentes, vai escolher então um dos quatro cantos. No caso de ter três casas diferentes adjacentes, vai então escolher uma das peças que presentem terreno isolado por 3 lados.

(Nota: exemplos das texturas usadas estão no capítulo 5.3)

Este algoritmo foi também desenvolvido de forma a que o seu uso possa ser usada em qualquer situação de jogo, seja no gerador *offline*, como no gerador *online* durante o *gameplay*, sempre que alguma casa mudar o seu tipo de terreno.

Interação do jogador com o jogo

A etapa seguinte foi aplicar o movimento do jogador a partir do controlo do vento, e a interação deste com os elementos presentes no mundo.

O vento é controlado pelo jogador a partir da ferramenta indicadora da plataforma que está a jogar, como o rato no caso de um computador ou a partir de *touch* num *smartphone* ou *tablet*. Ao arrastar o indicador sobre o ecrã é criado um vetor, na qual o angulo de direção representa a direção do vento e a sua magnitude é a força aplicada pelo vento.

No momento em que o veiculo do jogador conseguia-se movimentar pelo mundo, aplicou-se as funções para detetar colisões com os diferentes terrenos, como por exemplo o veiculo chocar com montanhas e não as poder trespassar. Foi também desenvolvida a função que detetar quando o jogador vence o jogo, ao colidir com o artefacto.

Interação do jogo com o jogador (PCG online)

Antes de desenvolver o sistema de PCG em tempo real do jogo, a primeira tarefa realizada deste grupo foi adicionar a interação que cada tipo de terreno tem com o jogador. Começou-se por aplicar as correntes marítimas do mar, ao utilizar os dados registados em cada casa da matriz do mapa com a direção criada pelo gerador do mundo. Para inicializar a força que cada casa do oceano iria aplicar sobre o jogador, decidi aproveitar também o valor da altitude de cada casa, logo, quanto menor fosse a altitude das casas do oceano, maior seria a força aplicada sobre o veiculo do jogador. Esta diferença foi mapeada de forma a ter sempre o mesmo limite mínimo e máximo de força para cada mapa, independentemente do limite de altitude que o mapa tem para definir o que é oceano e terra. Por exemplo, os valores de força podem ser iniciados de 0 a 20 unidades, onde o valor 0 de altitude equivale a 20 de força, e o limite que defini o que é oceano equivale a 0 de força. Usar o valor de altitude cria um efeito engraçado na força das correntes, já que torna sempre as correntes no centro dos oceanos mais fortes do que na costa, valores típicos do uso de Perlin Noise, isto acaba por criar mais dinâmica ao navegar dentro dos oceanos, um retrato das forças das marés em mar alto.

O tile de montanha já teve a sua interação criada quando se desenvolveu a interação do jogador com o mundo, neste caso as colisões. A mesma coisa acontece com o artefacto terminar o mundo. Os tiles de planície e costa não têm efeitos sobre o jogador.

A tarefa seguinte foi criar um sistema que de avaliação do jogo e que decisão de quando criar desafios ao jogador. Para este protótipo era preferível criar um sistema simples, algo aleatório, mas que ainda tivesse alguns critérios básicos para fazer gatilho a dos três desafios que nesta versão podia utilizar contra o jogador.

Neste caso, o sistema conseguia recolher a posição do jogador, em qual direção este se move (norte, sul, este, oeste), a distância a que este estava do artefacto e se estava em terra ou em mar. Além disso, contem um cronómetro que indica quanto tempo se passou desde o ultimo desafio criado.

A ideia, é que de X em X tempo, estes espaços de tempo decidido pelo sistema, o gerador verifica se o estado de situação do jogo com os poucos dados a que tinha acesso, e inicia-se um dos três desafios.

O primeiro desafio desenvolvido foi a Modificação Geográfica, no qual é reaproveitado o algoritmo que gera o mundo para modificar uma seleção da área do mapa. Como o algoritmo para gerar essa área é igual ao algoritmo que gera o mundo, ainda se torna uma função pesada para o jogo, logo as áreas a ser modificadas têm que ser pequenas, no máximo áreas do tamanho 10x10 unidades. Para modificar assim áreas maiores, mas sem retirar performance ao jogo, foi questão de iterar a função ao longo de tempo, para ir modificando uma área de cada vez á medida que o jogo avançava. Estas áreas modificadas são selecionadas por locais por onde o jogador já passou, mas nesta versão do protótipo sem grande atenção, já que nesta fase a cartografia não era um elemento importante terminar rapidamente.

O segundo desafio foi o Criar Barreiras. O método utilizado para este desafio, quando este era chamado pelo sistema de avaliação, era selecionar um ponto no mapa em frente do jogador, dependendo da direção que ia (esta guardada pelo sistema de avaliação). A partir desse ponto e uma pequena área em redor a este, este escolhia um certo numero de casas aleatoriamente (o numero em si também é aleatório entre um certo limite dado pelo sistema), essas casas escolhidas tornavam-se montanhas, com o valor de altura máxima. Assim a qualquer momento o jogador podia embater contra uma destas barreiras.

Mas a partir deste ponto, antes do desenvolvimento do ultimo desafio e estudo da arte do jogo, foram encontrados alguns problemas no design.

AVALIAÇÃO PESSOAL DO PROTÓTIPO

Este subcapítulo é uma leitura pessoal sobre o desenvolvimento do projeto nesta fase e as opções que tomei no decurso do seu desenvolvimento. Ao chegar a este ponto na produção do jogo, pessoalmente não estava satisfeito com o que tinha desenvolvido. Testei o jogo com alguns jogadores, nos quais estes testes não podem ser considerados oficialmente sessões de playtesting, foram uma discussão com colegas sobre as duvidas que tinha sobre o jogo. Estes ao experimentar o que tinha produzido até então, ficaram na maioria com uma opinião semelhante à minha sobre o estado do jogo. Apesar de estarem interessados na ideia do jogo, o estado em que o protótipo se situava, sem conteúdo em termos de jogabilidade ou adereços artísticos, era aborrecido e não levantava interesse para querer continuar a explorar o mundo de jogo, já que não iam encontrar desafios ou situações estimulantes.

O problema de ação que refiro, deve-se ao facto de o jogo não atingir uma ação com movimento dinâmico que eu imaginava ao fazer o design, como também falta o poder de o jogador ter decisões interessantes a tomar durante o jogo. Ao procurar uma solução para este problema, tomei atenção à visão que a câmara do jogo oferece ao jogador sobre o mundo, a forma como o jogador se podia mover e como este podia interagir com o gerador, sem ser apenas reações ao que é gerado.

Ao pensar no movimento tive em conta a visão que estava a dar ao jogador sobre o mundo. Uma camara de visão top-down é um bom tipo de visão para jogos de estratégia, onde o pensamento e decisões tomadas são o elemento principal da jogabilidade, no qual ter uma visão geral do mapa é importante para captar informação sobre o estado de jogo e assim conseguir poder tomar as melhores decisões na estratégia que o jogador está a criar. Apesar de existirem jogos com o mesmo tipo de perspectiva de visão sobre um mapa de jogo em duas dimensões, na qual a ação é dinâmica e necessita de reações instantâneas da parte do jogador (por exemplo o jogo *Micro Machines* [70], um jogo de corrida com vista top-down), no caso

de um jogo exploração, onde o movimento é dinâmico mas qual o mapa não oferece o tipo de desafio esperado, ou momentos onde o jogador pode tomar decisões interessantes e com impacto no jogo, acaba por se tornar numa experiência aborrecida, o qual era o caso do protótipo.

A minha decisão foi mudar o conceito do jogo, de forma a conseguir reaproveitar o máximo possível das ideias do design anterior – o jogador contra o mundo, a exploração, o movimento a partir do vento e a cartografia do espaço – como também todos os artefactos que já tenha desenvolvido para o protótipo. As outras componentes do design anterior seriam modificadas para criar um jogo que melhor aproveita a dinâmica de movimento criado pelo vento, como também as interações que esta pode fazer já que é uma forma de interação que permite experimentação, e assim melhorar a experiência de jogo.

Ao rever o design anterior do jogo, dei conta que este poderia ainda ser mais enriquecido como um jogo exemplo de Generative Game Design. Para a nova ideia tentei dar atenção em como o jogador pode interagir com o mundo, como este pode não só enfrentar o conteúdo gerado, como também usa-lo ou manipula-lo em seu proveito, aumentando assim o numero de decisões que o jogador pode tomar durante o jogo.

5. DESIGN & PROTÓTIPO ADAMASTOR

Neste capítulo vai ser apresentado o design final do jogo desenvolvido nesta dissertação como uma solução ao problema levantado no último parágrafo do capítulo anterior.

Além do design, vai ser apresentado os estudos sobre o estilo gráfico, o desenvolvimento do protótipo, a apresentação de uma demo do jogo na conferência DiGRA-FDG 2016, as sessões de *playtesting* a jogadores e a sua avaliação.

5.1 DESIGN FINAL

CONCEITO

O jogo decorre num estranho planeta alienígena, no qual o jogador assume o papel de uma semente de uma espécie invulgar de plantas, a qual só pode sobreviver num raro tipo de terreno que se encontra num sistema de grutas escondido por baixo da superfície. O objetivo do jogador é encontrar esse bioma onde a semente pode ser semeada e criar uma planta que fique para a posterioridade. Mas a espécie de planta não é tão frágil, e pode gerar noutros tipos de terreno plantas temporárias, as quais produzem novas sementes que vão continuar a explorar o mundo, e a cada geração que passa estas sementes vão evoluindo ao adaptarem-se aos desafios que encontram. Para se deslocar no mundo a planta consegue manipular o comportamento do vento, o qual o seu arrasto a faz flutuar na direção para as correntes se deslocam. Mas o planeta que o jogador está a explorar sofre uma constante mudança, demonstrando uma animosidade perante as sementes da planta ao modificar a sua estrutura e a utilizar outros truques sobre os sentidos do jogador, ao tentar impedir que esta encontre uma zona do mundo onde pode crescer em segurança.

O JOGO

Visão sobre o mundo

O mundo do jogo é feito a partir de uma matriz 2D, com uma vista lateral para o mundo de jogo, onde as dimensões XY representam o terreno do mundo (X dimensão horizontal, Y dimensão vertical). Este tipo de perspetiva sobre o mundo é usado frequentemente em jogos de plataformas como o jogo *Spelunky* [16]. A vista lateral permite criar um mundo com movimento dinâmica, onde forças físicas como a gravidade podem ser representadas no movimento de objetos, as quais são usadas neste jogo. A visão permite representar um planeta constituído por conjuntos de várias grutas e ligações sob a superfície, onde o jogador vai passar a grande parte do seu tempo a explorar.

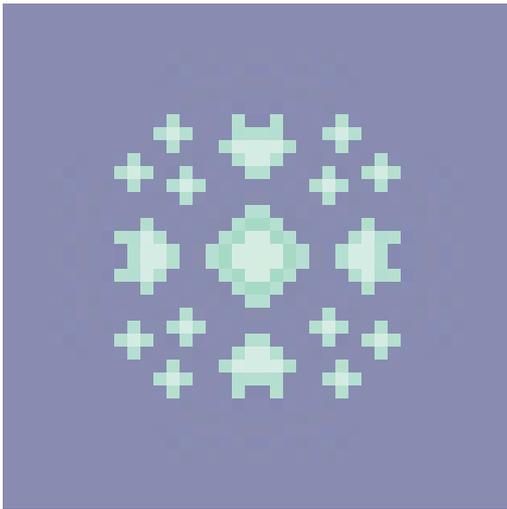


Figura 24. Exemplo da construção de uma semente

Jogador

O jogador representa o papel de uma semente, que tem como objetivo semear e criar uma planta num tipo de terreno fértil na qual pode permanecer viva e em segurança.

Essa semente é constituída por várias partes: um núcleo central, e dois grupos de dois tipos de fragmentos diferentes em volta do núcleo. Estas vão dar informações ao jogador sobre o estado da

semente, como a sua fragmentação (que representam os pontos de vida, representado como os fragmentos em volta do núcleo serem eliminados) ou a energia representada no seu núcleo (que representa a energia de vento acumulada). Os dois grupos de peças representam a composição e poderes da semente (e estas vão ter apresentações gráficas diferentes dependendo da constituição da semente). Cada semente vai ter quatro atributos principais: um nível limite de pontos de vida, limite de energia de vento, velocidade máxima e resistência contra colisões. No início do jogo o computador vai escolher aleatoriamente qual vai ser o seu atributo principal, no qual o jogador recebe um bônus. O jogador durante o jogo também vai ganhar poderes como permitir que flutue em cima de água, conseguir ver armadilhas ou blocos escondidos, ou renascer numa semente mais fraca após a destruição da semente que o jogador controla. Os efeitos da semente não vão ser explicados ao jogador, o objetivo é o jogador experimentar criar plantas em cada tipo de bloco e descobrir por si como estes afetam a semente. Ao descobrir como estas funcionam, o jogador pode criar estratégias que apoiem o seu tipo de jogo.

Esta composição vai ser manipulada á medida que o jogador sênea as sementes e cria uma nova geração de plantas. Cada vez que o jogador sênea a semente recupera uma percentagem de vida, dependendo do tipo de terreno. Este tipo de terreno também influência como a nova geração de sementes vai evoluir na sua constituição, por exemplo criar uma semente num terreno rochoso vai recuperar menos vida à semente, mas vai aumentar o seu nível de resistência. Ao atingir certos critérios que o planeta define no início do jogo, o jogador pode receber poderes, como poder flutuar na água, ver a maiores distâncias ou não ter uma semente auxiliar caso a sua

seja destruída.

Além da modificação feita na nova geração de sementes quando se semeia, a planta que o jogador cria também modifica o terreno em volta, criando um tipo de terra que vai ser seguro para o jogador, o qual o planeta não vai poder modificar.

O jogador tem o poder de manipular o vento, a partir de uma quantidade limitada de energia que recupera ao longo do tempo. Essas correntes de vento são criadas a partir de vetores lineares que o jogador desenha no ecrã, onde a direção do vetor representa a direção da corrente de vento, e a magnitude é a força do vento. Assim, um vetor com uma grande magnitude vai gastar mais da energia de vento armazenada, e caso o jogador não tenha energia suficiente para alimentar o vetor, nada vai acontecer.

Manipular o vento é o que vai criar a deslocação da semente no mundo. Esta é um objeto leve, e qualquer força de vento, seja uma brisa ou uma corrente forte, vai levar a semente consigo na direção que estão a percorrer. Este sistema vai permitir as interações do jogador com o terreno à sua volta.

Ao chocar contra as peças do terreno do mapa, o jogador vai sempre receber dano. A quantidade de dano infligido sobre o jogador nessas colisões é definida pela velocidade que o jogador se desloca e o tipo de terreno com que chocou, logo uma colisão a maior velocidade vai infligir mais dano sobre o jogador, e chocar contra pedras inflige mais dano do que colidir contra areia. Mas o jogador pode também querer sacrificar parte dos seus pontos de vida de propósito, já que a semente consegue destruir blocos de terra ao atingi-los a uma certa velocidade. Quanto maior a resistência da semente, menos velocidade perde ao destruir blocos (o que pode até permitir destruir vários blocos em serie) e até perder menos vida nas colisões.

O jogador vence o jogo quando consegue semear a semente e criar uma nova planta no terreno que procura. Perde o jogo caso a sua semente seja destruída e não tenha uma outra na planta criada anteriormente – ter ganho durante o jogo a habilidade de ter mais que uma semente.

Planeta:

Este é o único inimigo do jogador e o gatilho para a ação no jogo. O planeta é também o controlador da Geração Procedimental de Conteúdo, tanto offline como online. No início de cada jogo um novo planeta vai ser

gerado, cada um com uma estrutura de cavernas, corredores e superfície único. A estrutura não é estática, pelo que vai ser modificada durante o jogo pelo planeta.

A geração do mundo é feita a partir de vários pontos espalhados pelo mapa do mundo, cada um com o seu próprio tipo de bioma e de terreno. A profundidade e posição horizontal de onde se encontra cada fonte vai modificar as probabilidades dessa fonte ser de um tipo de terreno, por exemplo, terreno rochoso é mais provável que exista a maior profundidade. O terreno seguro que o jogador procura é colocado num canto aleatório do mapa, não tem nenhum sistema para o jogador não saber onde o procurar caso tenha experiência no jogo.

O planeta vai criar desafio a partir de dois métodos: manipulação da sua geografia e manipulação da percepção do jogador. A manipulação da sua geografia pode ser rápida e acompanha o movimento do jogador, ao gerar novos blocos, ou retira-los do caminho do jogador, o que podem criar situações com o qual o jogador se depara com uma grande barreira atrás de si (caso esse terreno antes não tinha sido modificado) ou novos corredores entre cavernas. Além dessa geração rápida, quando o jogador cria uma nova planta e geração de semente, o planeta pode também decidir se manipular a si. Ao ler dados durante o jogo sobre a situação do jogo, se o planeta vir que o jogador se está a adaptar a um tipo de terreno (verifica nos dados que o jogador passou mais tempo num tipo de terreno no qual recebeu pouco dano), pode fazer com que os pontos que decidem um tipo de terreno mudem para outro, onde o jogador recebeu mais dano.

O gerador tem também duas variáveis que controlam o seu comportamento, dificuldade e simpatia. Apesar de estar contra a semente, o planeta pode tomar decisões irracionais em relação ao seu objetivo, o que podem ajudar o jogador. Um maior nível de simpatia faz com que o terreno crie mais blocos de um tipo com o qual o jogador recebeu menos dano, e que a probabilidade de retirar blocos do terreno seja maior. Uma maior dificuldade faz o contrário, cria terrenos onde o jogador recebeu mais dano e cria barreiras maiores. Estas duas variáveis podem estar com valores altos e valores baixos, não estão ligadas entre si, o conteúdo gerado torna-se é mais extremo com os níveis elevados, por exemplo gerar barreiras difíceis de ultrapassar ou uma maior área modificada para um tipo de terreno.

Contudo esta não é a sua única arma, o planeta vai brincar com as percepções do jogador, como limitar-lhe o campo de visão, modificar o comportamento do terreno (blocos de terreno na realidade não existem, uma alusão a passagens secretas de jogos como *Sonic the Hedgehog* [38]), até teletransportar o jogador para outro canto do mapa sem que este dê conta.

A superfície do planeta é um desafio ao jogador, já que esta é devastada por uma enorme tempestade permanente com fortes rajadas de vento. O jogador pode explorar este espaço para encontrar caminhos que consigam aceder a cavernas mais rapidamente, mas é sempre um risco que tem que ser calculado, porque pode ser facilmente destruído caso seja lançado contra um bloco a grande velocidade.

CENÁRIOS E COMPONENTES

Elementos da semente: Núcleo; Fragmentos da semente;

Planta: Planta criada quando o jogador semeia a semente;

Elementos da interface do jogador: Indicador da direção e força da corrente de vento criado pelo jogador; Indicador de vida da semente (o número de fragmentos em volta do núcleo); Indicador da capacidade de vento (uma barra que preenche o interior do núcleo da semente);

Elementos e modificadores da semente:

Nome	Descrição
Núcleo básico	O núcleo que a semente contem no inicio de cada jogo;
Fragmento básico 1	Um dos dois tipos de fragmentos que a semente contem no inicio de cada jogo;
Fragmento básico 2	Um dos dois tipos de fragmentos que a semente contem no inicio de cada jogo;
Capacidade de Vento	Quanto maior for a capacidade de energia do vento disponível para o jogador, maior vai ser o tamanho do núcleo da semente;
Limite de Pontos de Vida	Quanto maior for o limite de pontos de vida da semente, mais fragmentos esta vai conter em volta (começa com 8 fragmentos, de dois tipos diferentes);
Nível de resistência	Quanto maior for a resistência do jogador, maior vai ser a parede em volta do núcleo;
Nível de velocidade	Este vai ser representado na animação da semente, os fragmentos da semente vão ser sempre mais lentos que a velocidade do núcleo, logo a diferença de distâncias consegue mostrar a diferença de velocidade.
Fragmento de Água	Este tipo de fragmento mostra ao jogador que ele pode flutuar em cima de água, logo não é destruído sobre esta, mas tem que gastar uma quantidade de energia de vento para se conseguir libertar;
Fragmento de Visão;	Este tipo de fragmento mostra ao jogador que consegue detetar as armadilhas mais básicas do planeta, como blocos falsos;
Fragmento de Regeneração;	Este tipo de fragmento mostra ao jogador que no caso da sua semente ser destruída, ele vai poder regenerar uma nova semente mais fraca no local de onde brotou a ultima planta;

Tiles (peças do terreno) do planeta:

A matriz 2D que represente o planeta é constituída a partir de casas que podem-se encontrar em duas situações diferentes, mas que partilham biomas e tipos de terreno entre si: blocos físicos (os quais com que o jogador consegue interagir e colidir) e os blocos de *background* (que podem causar efeitos sobre o jogador, mas este não os consegue atingir, representam o fundo do jogo). Qualquer espaço que não é ocupado por um bloco físico, é ocupado por um bloco de *background*.

Deslocação: Atravessar / estar em frente dos blocos do *background*;

Colisão: Colisão com blocos físicos.

Blocos / tiles do planeta

Bioma	Tipo	Efeitos na Deslocação	Efeitos na Colisão (Força para destruir bloco; Dano infligido na semente)	Evolução da semente (HP – pontos de vida)
Deserto	Areia	Existe uma probabilidade de ocorrer rajadas de vento que afetam o movimento da semente;	Força: Pouca; Dano: Pouco;	HP: recupera 50% do limite máximo de HP; Melhora o atributo de velocidade;
Deserto	Terra	Existe uma probabilidade de ocorrer rajadas de vento que afetam o movimento da semente;	Força: Média; Dano: Pouco;	HP: recupera 75% do limite máximo de HP; Melhora o atributo de Pontos de Vida;
Deserto	Rocha	Existe uma probabilidade de ocorrer rajadas de vento que afetam o movimento da semente;	Força: Indestrutível; Dano: Muito;	HP: recupera 25% do limite máximo de HP; Melhora o atributo de resistência;
Selva	Terra	N/A	Força: Média; Dano: Pouco;	HP: recupera 50% do limite máximo de HP; Melhora o atributo de Pontos de Vida;
Selva	Planta -Trepadeiras	Existe uma probabilidade de a semente ficar presa numa trepadeira;	Força: Pouca; Dano: Pouco; A semente pode ficar presa no bloco;	HP: recupera 75% do limite máximo de HP; Melhora o atributo da capacidade de vento;
Selva	Planta - Venenosa	N/A	Força: Pouca; Dano: Pouco; A semente pode ficar envenenada;	HP: recupera 25% do limite máximo de HP; Melhora o atributo de resistência;

Glaciar	Neve	Existe uma probabilidade de diminuir a velocidade da semente;	Força: Pouca; Dano: Médio;	HP: recupera 25% do limite máximo de HP; Melhora o atributo de velocidade;
Glaciar	Gelo Frágil	Existe uma probabilidade de diminuir a velocidade da semente;	Força: Pouca; Dano: Médio; A semente pode ficar colada ao bloco;	HP: recupera 25% do limite máximo de HP; Melhora o atributo da capacidade de vento;
Glaciar	Gelo Sólido	Existe uma probabilidade de diminuir a velocidade da semente;	Força: Forte; Dano: Médio; A semente pode ficar colada ao bloco;	HP: recupera 25% do limite máximo de HP; Melhora o atributo de resistência;
Terreno Seguro	Terra	N/A	Força: Média; Dano: Médio; Não pode ser modificado pelo planeta;	Planta semeada num local seguro, o jogador ganha o jogo;
Terreno Criado pela planta	Terra	N/A	Força: Média; Dano: Média; Não pode ser modificado pelo planeta;	N/A
Água	Água	N/A	Força: Indestrutível; Dano: Destroi a semente, a não ser que ela tenha a habilidade para poder flutuar;	N/A

Espaço de Ação do Jogador

- **Aplicar uma corrente de vento:** Input de slide sobre o ecrã (mesmo movimento como por exemplo arrastar um ícone num ambiente de trabalho) com o indicador da plataforma onde está a jogar, no qual a direção do vetor criado pelo movimento de slide indicam a direção da força do vento, e a distância do ponto onde começa ao ponto onde termina o input a força do vento; esta direção e força do vento movem o veiculo do jogador;
- **Colidir** com os obstáculos no mundo;
- **Destruição** de blocos ao colidir com eles a grande velocidade;
- **Semear** a semente no que resulta no crescimento de uma planta onde esta foi semeada;
- **Semear** a semente num bloco de terreno seguro para vencer o jogo;
- **Transformação** do terreno à volta das plantas que brotam das sementes;
- **Modificar** os atributos de novas gerações de sementes ao criar uma planta;
- **Recuperar** parcialmente os pontos de vida dependendo do bloco onde semeou a semente;
- **Recuperar** toda a energia de vento ao criar semear a semente;
- **Usar** fortes correntes de vento para conseguir fugir de armadilhas que prendam a corrente, como trepadeiras ou blocos de gelo;
- **Aprender** a manipular a evolução da semente ao perceber os efeitos que cada tipo de bloco tem sobre o nascimento de uma nova semente;
- **Aprender** a manipular o planeta ao perceber como este cria os blocos a partir das ações do jogador;

Espaço de Ação do Planeta

- Geração do mapa no início do jogo;
- Avalia o estado do jogo:
 - ◇ Verifica posição do jogador;
 - ◇ Verifica em que direção se movimenta o jogador;
 - ◇ Verifica o bioma e tipo de terreno no local onde está o jogador;
 - ◇ Verifica a condição do jogador;

- ◇ Verifica a composição da semente do jogador;
- ◇ Verifica distância do jogador ao terreno seguro;
- ◇ Verifica as variáveis de dificuldade e simpatia;
- ◇ Verifica quando foi despoletada a ultima armadilha para o jogador;
- ◇ Verifica uma área do mapa em redor do jogador;
- ◇ Verifica os blocos que pode modificar nessa área;
- Adaptar os níveis de desafio e simpatia quando o jogador planta a semente;
- Modifica os blocos que definem áreas de cada bioma quando o jogador planta a semente;
- Modifica os tipos dos blocos numa área em redor do jogador;
- Cria novos blocos físicos numa área em redor do jogador;
- Destrói blocos físicos numa área em redor do jogador;
- Envenena o jogador quando este colide com alguns tipos de bloco;
- Prende o jogador quando este colide com alguns tipos de bloco;
- Adapta as correntes de água às modificações do terreno em sua volta;
- Aplica armadilhas ao jogador dependendo da situação (os desafios estão representados na grelha em baixo);

Armadilhas e desafios criados pelo planeta:

Nome	Descrição	Objetivo	Trigger
Criar correntes de vento	O planeta consegue gerar correntes de vento que vão afetar a trajetória de movimento da semente.	Criar dificuldade no movimento do jogador;	Pode ativar em qualquer momento do jogo;
Limitar visão	A visão da semente é coberta, com o qual só pode ver o que está próximo dela.	Limitar a visão do jogador, criar um desafio onde o jogador tem que se movimentar com mais calma e atenção; perdura por alguns segundos a dois minutos;	Nível médio de desafio;
Criar blocos físicos falsos	O planeta pode transformar ou criar novos blocos que são falsos.	Tenta enganar o jogador ao esconder caminhos e outros acessos ao jogador, como também pode impedir o jogador de semear a semente, pelo que esta passa pelo bloco invisível.	Nível baixo de desafio;
Teletransportar jogador	O planeta aproveita o efeito de Limitar Visão para teletransportar o jogador para um outro ponto no mapa semelhante ao qual o jogador se encontra (os blocos em redor são similares);	Cria um momento de grande desafio ao jogador, faz com que este se sinta perdido quando retoma outra vez a visão completa.	Nível alto de desafio;
Criar Nascentes de Água	O planeta pode criar uma nova nascente de água, a partir desse ponto a água vai fluir e criar um novo rio dentro do mundo.	Cria novos obstáculos ao jogador;	Nível médio de desafio;

5.2 PROTOTIPAGEM FINAL

PLANEAMENTO

O protótipo final foi planeado da mesma forma que o primeiro protótipo, ao delinear o objetivo desta versão do protótipo, os componentes a desenvolver e as tarefas gerais necessárias a realizar para construir cada uma das componentes. O objetivo, como no primeiro protótipo, é criar uma versão simples do jogo, com elementos mais importantes para o gameplay concluídos para se poder testar o novo conceito do jogo e a partir dos testes iterar numa nova versão do design.

Os elementos a criar para este protótipo é o gerador de conteúdo offline, que constrói o mundo de jogo, blocos (ou tiles) para que permitam uma variedade interessante de interações durante o gameplay (neste caso todos os tiles do bioma do deserto, área segura e área criada em volta da planta), movimento da semente, as interação do jogador com o mundo (colisões e destruição de blocos), o sistema de plantar e criar novas sementes com modificações dos atributos, um sistema para avaliar o jogo e o controlador do planeta que irá modificar o mundo em tempo real.

Para este protótipo é importante aproveitar ao máximo o que restou da versão anterior, assim a primeira tarefa a ser realizada para este protótipo é analisar o que foi desenvolvido anteriormente, para retirar o que não é necessário, e adaptar o que pode ser usado ao novo protótipo.

O novo design do jogo tem um tipo de estrutura diferente para representar o planeta, este é constituído por um conjunto de cavernas, ligações e a superfície desse mundo, diferente da visão sobre a superfície do planeta da versão anterior, pelo que é preciso encontrar um novo algoritmo para construir este terreno.

O resto das tarefas seguintes seguem o mesmo modelo que o protótipo anterior. Após ter o construtor do planeta desenvolvido é necessário criar a interação do jogador com o mundo, como o movimento, permitir colidir com o terreno, destruir blocos e plantar a semente. O passo seguinte é criar um sistema que permita as modificações da semente a partir do tipo de terreno onde esta é semeada e de onde nasce uma nova geração de semente. Neste caso só vai ser desenvolvido a modificação dos quatro atributos

básicos da semente, pelo que as habilidades especiais são desnecessárias para esta versão. O próximo passo é a construção de um novo algoritmo que avalie o estado de jogo, pelo que segue e gera conteúdo rapidamente em redor do jogador. Como a modificação do terreno em volta são importantes e necessitam de mais testes, as armadilhas do gerador ficam para uma versão futura do protótipo, à de Criar Correntes de Bento, já que no protótipo anterior tinha criado um modelo que gerava esse movimento do vento controlado pelo computador. Por fim, criar a arte necessária para dar vida ao jogo, que represente todos os elementos necessários ao jogo.

A lista de tarefas a realizar, ordenada por ordem de importância, é a seguinte:

- **Protótipo Anterior:**
 - a. Analisar todos os elementos que podem ser aproveitados;
 - b. Retirar e adaptar esses elementos ao novo protótipo;
- **Geração do Mundo de jogo (PCG offline):**
 - a. Seleção do algoritmo para gerar o mundo;
 - b. Aplicação do algoritmo para gerar o mundo;
 - c. Aplicar os tiles definidos ao algoritmo;
- **Interação do jogador com o jogo:**
 - a. Movimento do jogador;
 - b. Colisões com o mundo;
 - c. Destruição de blocos;
 - d. Plantar a semente;
 - e. Vencer o jogo ao plantar a semente no terreno seguro;
- **Sistema da Semente:**
 - a. Criar uma nova semente quando a anterior é plantada;
 - b. Modificar os atributos da semente;
- **Interação do jogo com o jogador (PCG online):**
 - a. Criar um método para avaliar a ação do jogo;
 - b. Modificar os tipos de terreno em volta do jogador;
 - c. Criar e destruir blocos do terreno;
 - d. Adaptar dificuldade e simpatia a partir do que está a acontecer no jogo;
- **Arte:**
 - a. Design dos elementos da semente (núcleo e fragmentos);
 - b. Design de cada um dos blocos;

PRODUÇÃO

Protótipo Anterior

Para decidir o que manter do protótipo anterior, comecei por comparar o design de ambos os jogos e manter o que tinham em comum. Selecionei sistema de movimento a partir do vento (que é igual ao design anterior) e o sistema para selecionar e organizar os tiles (o jogo continua a ser 2D e usar o mesmo formato de tiles) sem necessitarem, nesse momento, de nenhuma modificação.

Apesar do algoritmo do Gerador do planeta anterior não ser necessário neste design, achei melhor manter o sistema que gera as matrizes a partir de Perlin Noise para este protótipo. Este pode ser usado no controlador do planeta no novo design, para criar as correntes de vento contra o jogador durante o jogo.

Geração do Mundo de jogo (PCG *offline*)

Como já tinha feito pesquisas para algoritmos durante o Estado da Arte e para o gerador do design anterior, ao fazer este design soube desde início o algoritmo que ia usar para gerar as estruturas do mapa.

Cellular Automata [15] (ou autómata celular) foi o algoritmo selecionado para criar o sistema de cavernas do mapa. Este algoritmo consiste numa grelha de duas dimensões, na qual cada casa pode ter um valor, 0 ou 1 (on/off). O algoritmo começa por definir um estado de tempo ($t = 0$) para cada célula, e a partir dessa célula é criada uma vizinhança à volta desta. A cada estado de tempo ($t = 1, t = 2, \dots$) que passa, é aplicada uma regra matemática sobre cada célula da grelha, que muda o seu valor dependendo dos valores da sua vizinhança e do seu próprio valor. No caso deste algoritmo aplicada à geração de mapas, a regra passada alguns estados acaba por gerar grandes grupos, de espaço negativo ou positivo, que se assemelham a estruturas naturais de cavernas [71].

O Sistema de PCG para gerar o mundo é então composto por:

- Uma grelha que é preenchida pelo algoritmo do Cellular Automata, que cria uma estrutura básica de cavernas (valor negativo é o espaço por onde o jogador se move, o espaço positivo são os blocos do terreno);

- Um sistema que avalia a estrutura das cavernas de valor negativo verifica se todas elas estão ligadas entre si. Ao verificar que duas cavernas não estão ligadas, ele cria um caminho entre elas. Quando a função verificar que o espaço negativo é uma única estrutura, é uma garantia que o jogador consegue percorrer todo o mapa e este tem um fim;
- Um sistema que gera um numero de pontos que criam áreas para cada bioma e tipo de terreno. Estes pontos têm regras que gerem as probabilidades de cada um ser de um tipo de bioma ou terreno. As células da grelha do mapa herdam o bioma e tipo de terreno da área do ponto onde se encontram. Cada área dá também um valor de energia a cada casa, que diminui o quanto mais longe a célula tiver do centro do ponto, isto serve para escolher de qual área é que uma célula escolhe essas informações no caso de duas áreas se sobreporem. Neste protótipo o gerador só cria os terrenos do bioma do deserto;
- No final do mapa estar completo, um dos pontos aleatoriamente é escolhido como a área segura. O jogador começa sempre numa área afastada desta.

Interação do jogador com o mundo

O movimento foi adaptado diretamente do protótipo anterior, tal como as colisões.

O sistema de destruição de blocos foi ligado diretamente ao sistema que controla o dano que o jogador leva a colidir com cada tipo de terreno. Estes têm valores de resistência e dano diferentes.

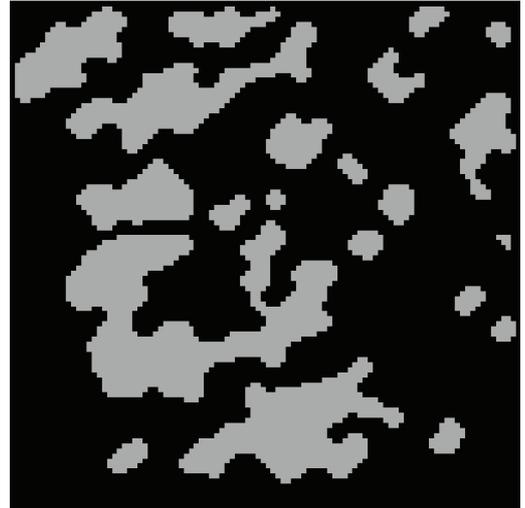


Figura 23. Sistema de cavernas básico gerado (celular automata)



Figura 24. Sistema de cavernas gerado com ligações



Figura 25. Sistema de cavernas com texturas

A velocidade do jogador interage com estas, para calcular o dano o jogo calcula a velocidade do computador juntamente com quanto dano deve esse terreno dar ao jogador. Segundo a 3ª lei de Newton, “A toda ação há sempre uma reação oposta e de igual intensidade”, o dano que o terreno causa ao jogador na colisão é assim também aplicado a si mesmo. Se o valor for superior à resistência, então o bloco é destruído – contudo existem blocos indestrutíveis, por mais dano que se lhes aplique. Tem que se ter em conta que o jogador também tem um nível de resistência, pelo que é sempre retirada essa resistência ao dano que lhe é causado.

Ao fazer um clique sobre o jogo, no caso do computador na tecla “espaço”, e na tablet um simples tap, a semente do jogador é lançada a grandes velocidades para baixo, e ao colidir com um terreno vai neste criar uma planta. Os blocos próximos em redor da planta são transformados noutra tipo de terreno, que não dá bónus ao jogador, e os torna destrutíveis. Existem estratégias criadas de volta desta transformação, ao tornar, por exemplo, blocos indestrutíveis como rochas em destrutíveis, que é a terra criada pela planta. Por outro lado, esse terreno não pode depois ser modificado pelo planeta, o que torna um local seguro para o jogador regressar, caso esteja em sarilhos.

Ao plantar a semente no tipo de bloco seguro, o jogador vence o jogo.

Sistema da Semente

Quando o jogador semeia a semente, à exceção se for plantada no terreno seguro no qual vence o jogo, o jogo vai criar uma nova semente para o jogador. Esta nova semente vai ter certos atributos melhorados, dependendo do tipo de terreno em que o jogador plantou a semente (a modificação de cada atributo por terreno pode ser revista na secção de Design).

Nesta versão do protótipo, os atributos eram mostrados ao jogador a partir de números no ecrã e não a partir da própria semente.

Interação do jogo com o jogador (PCG *online*)

Durante o jogo o computador vai estar a desafiar o jogador ao modificar o terreno próximo deste para lhe criar barreiras e desafios.

Para este sistema funcionar, o controlador do planeta está

constantemente a avaliar o estado do jogo. Este sistema de avaliação teve que ser feito para ser eficiente, ao só analisar os dados mais críticos para não criar atrasos durante o jogo. Para o protótipo o computador verifica a posição do jogador, em que direção se movimenta, quanto tempo este passou e levo de dano em cada tipo de terreno e as variáveis de simpatia e desafio.

Após avaliar o sistema, o planeta aleatoriamente decide quando deve fazer uma ação. Ao realizar uma ação, o computador cria uma lista de probabilidades do que pode fazer nesse turno. Por exemplo, se o nível de dificuldade estiver elevado o computador vai ter mais probabilidade de criar blocos novos, num maior numero, caso contrário ele reduz os blocos que cria. Um alto nível de dificuldade faz com o que computador tente mais vezes criar novos blocos na direção para onde o jogador se movimenta.

Além de criar e destruir blocos dependendo dos valores de simpatia e desafio, o computador pode também mudar o tipo de terreno dos blocos em volta, como também decide os terrenos dos novos blocos criados. Ele decide estes a partir dos valores de quanto tempo passou e dano o jogador levou em cada tipo de terreno – neste protótipo estes não fazem muito diferença além do dano que podem dar ao jogador e se os blocos são ou não destrutíveis, mas dentro dos outros biomas existem mais efeitos que podem criar situações interessantes a partir da modificação do terreno, ao tornar o jogo realmente difícil ou não. Caso um jogador passe muito tempo num tipo de terreno, o computador vai diminuir a probabilidade de escolher um bloco desse tipo, caso contrário, este aumenta a probabilidade de esse tipo ser escolhido. A mesma coisa acontece com o dano, caso o jogador receba muito dano por um tipo de terreno, o computador vai ter tendência a escolher esse tipo de blocos para desafiar o jogador. Mas estas probabilidades são também afetadas pelas variáveis de desafio e simpatia.

No caso de a variável de simpatia ser mais forte que de desafio, o computador vai tentar escolher o novo bloco a partir do tempo que este passou em cada terreno. No caso do desafio ser maior, este vai tentar escolher pelo tipo de dano, o que aumenta a dificuldade. Isto pode causar diferenças interessantes dentro do jogo. Por exemplo, o nível de desafio pode ser maior, logo o computador vai escolher blocos com o qual o jogador recebeu mais dano durante o jogo, mas esse tipo de terreno pode ser o qual o jogador

passou mais tempo dentro do jogo, pelo qual ficou habituado a desviar-se desses blocos (caso tenha recebido mais dano no início do jogo, mas depois esses danos acabaram de diminuir ao longo do tempo), o que acaba por ser benéfico ao jogador, mesmo que o perigo seja maior.

As variáveis de desafio e simpatia são adaptadas dentro do jogo, de formas diferentes dependendo das condições do jogo. No caso do jogador ainda se encontrar longe do sitio seguro, então o computador vai aumentar a simpatia e diminuir o desafio caso o jogador esteja a receber muito dano num curto espaço de tempo, como se vir que o jogador está muito tempo sem levar dano ele aumenta o desafio e diminui a simpatia. Mas estes valores são diferentes, o quanto mais próximo o jogador está do terreno seguro, o desafio vai aumentar em incrementos cada vez maiores e a simpatia em menores. O desafio também aumenta se os atributos da semente estiverem muito fortes ou o jogador tenha ganho habilidades.

5.3 ARTE GRÁFICA

Ao longo do desenvolvimento do protótipo do jogo desde o primeiro design do Adamastor, o objetivo para a arte do protótipo foi sempre feito a pensar num modelo 2D, simples, que permitisse criar um mundo coeso e esteticamente apelativo rapidamente, mas que tivesse também potencial de ser evoluído num futuro desenvolvimento após a dissertação. Por outro lado, tinha em mente criar um estilo estético que, ao nunca negar a possibilidade, se pudesse gerar também por computador. Tive isto em conta ao desenvolver o sistema que cria a semente, no qual numa versão futura ia ser aplicado parcialmente por geração.

A escolha do uso de Pixel Art deve-se ao facto de responder ao que era desejável para a arte do jogo. É um estilo simples, rápido para protótipar, tem um grande potencial para evoluir quando é mais trabalhado e ainda é um dos métodos mais simples para fazer arte generativa, já que se baseia-se em matrizes de cores.

BLOCOS

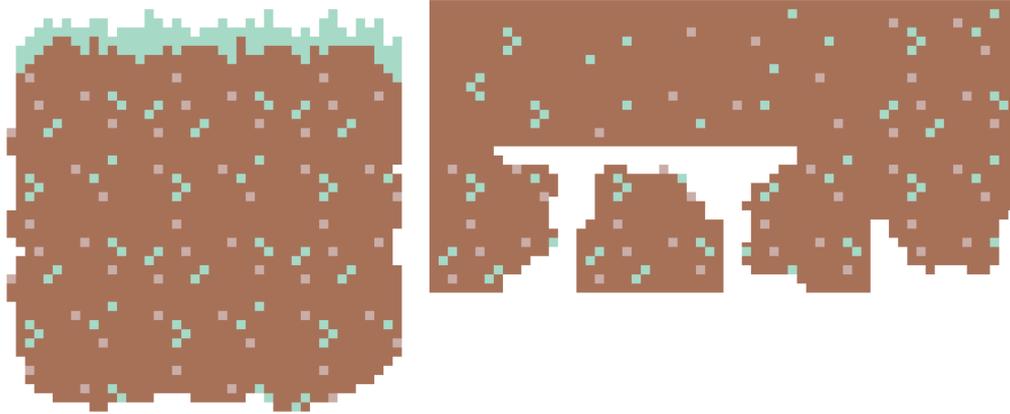


Figura 26. Blocos do terreno criado pela semente

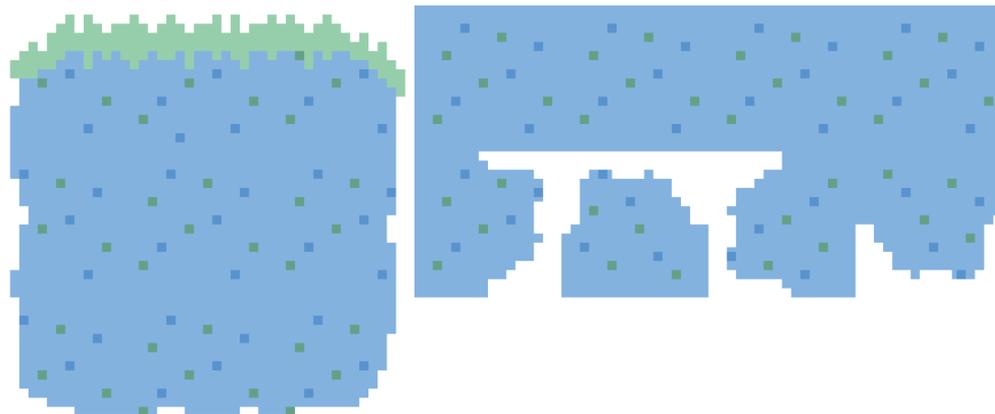


Figura 27. Blocos da área segura

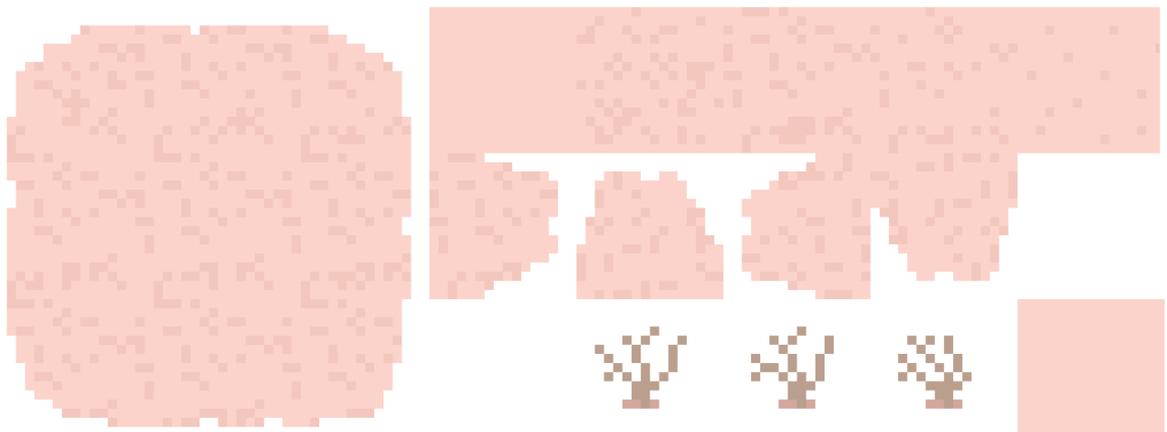


Figura 28. Blocos da areia do bioma deserto

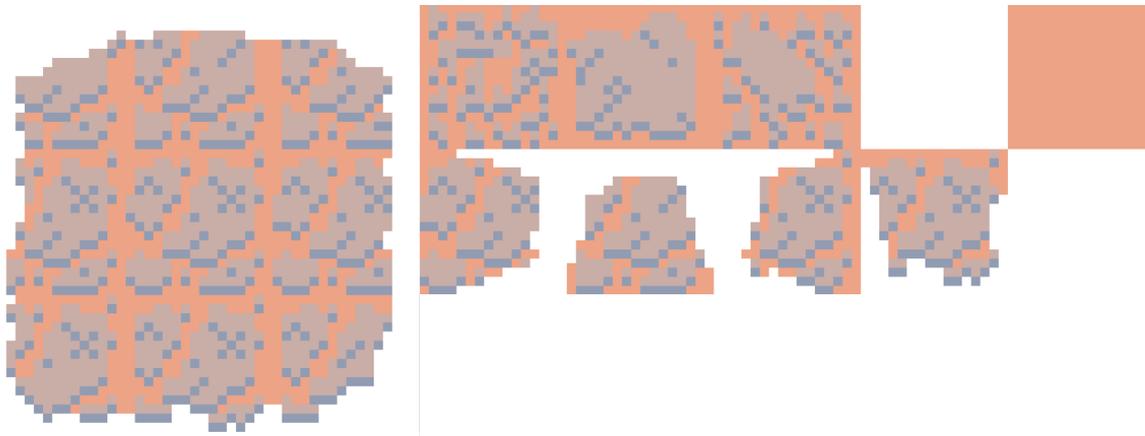


Figura 29. Blocos de rocha do bioma deserto

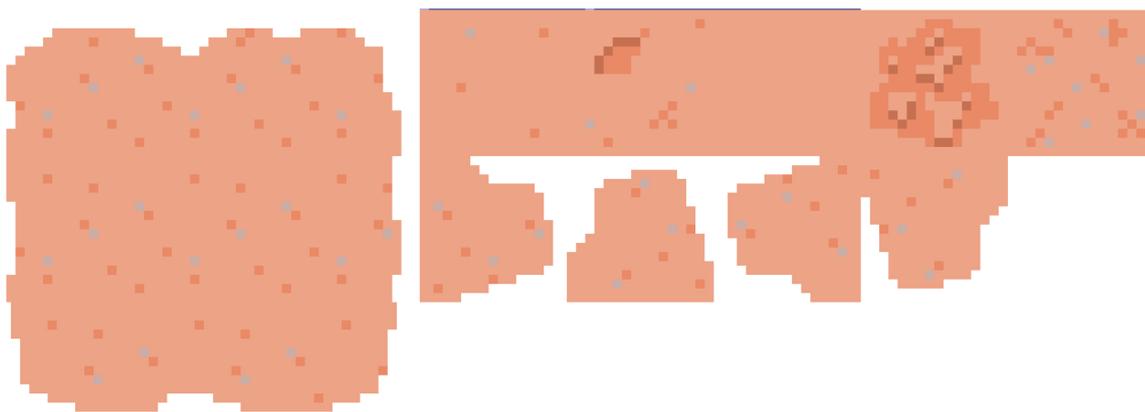


Figura 30. Blocos de terra do bioma deserto

PERCURSOS DE ÁGUA

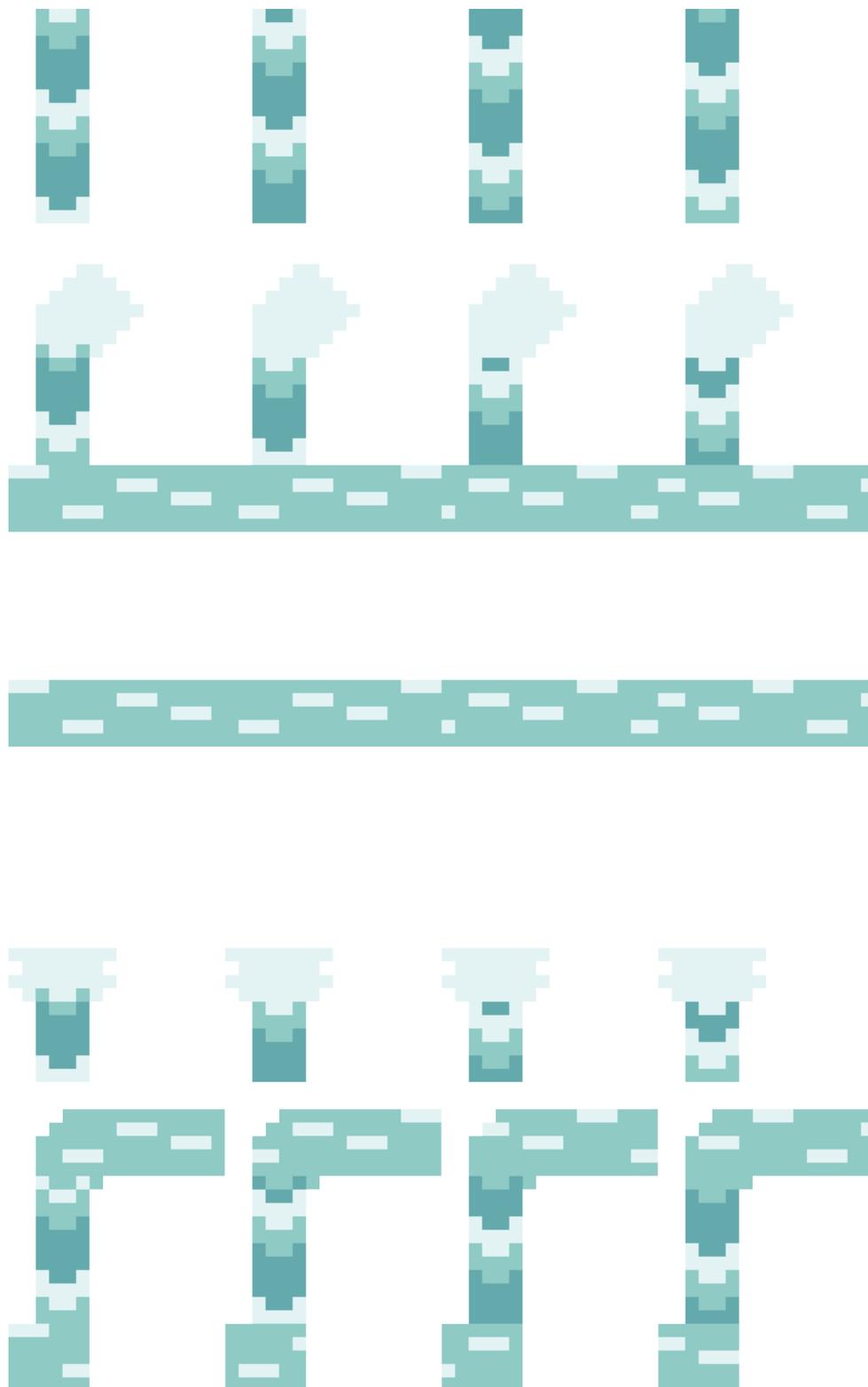


Figura 31. Animação para a aplicação de futuros lençóis de água e cascatas

SEMENTE E PLANTA

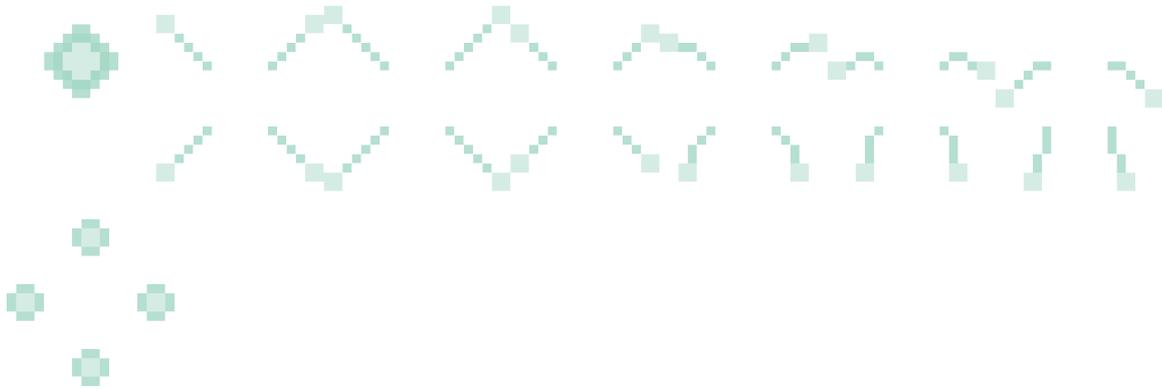


Figura 32. Fragmentos, núcleo e animação criara para a semente utilizada no protótipo

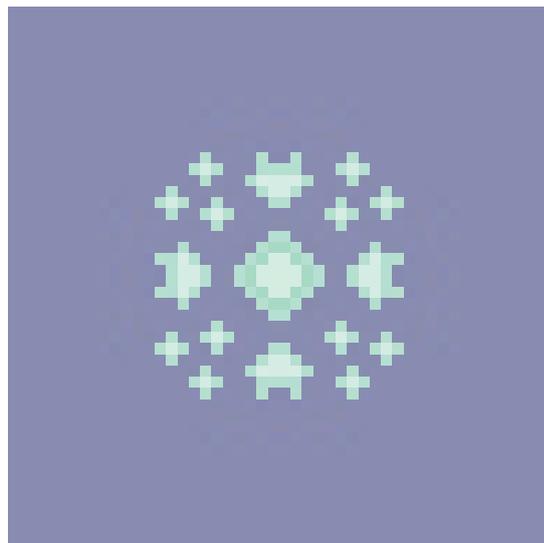
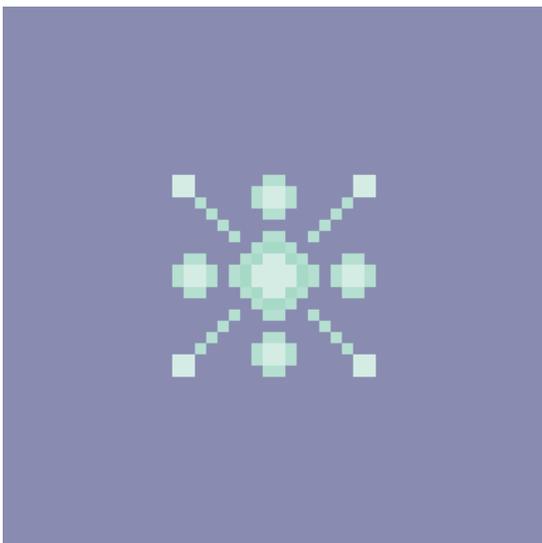


Figura 33 e 34. Exemplos de sementes construídos

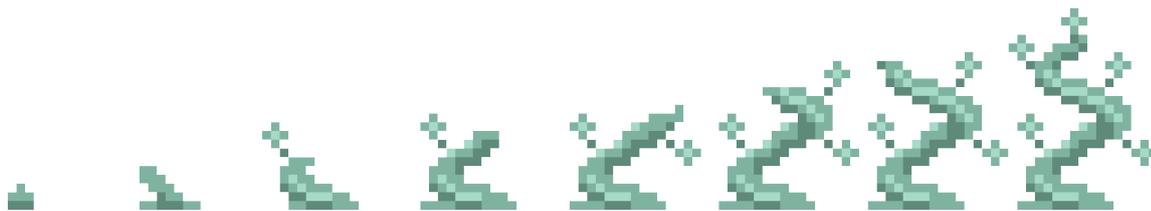


Figura 35. Animação do crescimento da planta

PROTÓTIPO EM AÇÃO



Figura 36. *Screenshot* do protótipo em ação

5.4 DEMO NA DIGRA-FDG 2016

Na semana de 1 a 5 de Agosto de 2016 tive a oportunidade de ser aceite para fazer uma para apresentar a demo do jogo produzido na primeira conferência DiGRA-FDG, em Dundee na Escócia, hospedada pela Abertay University.

Além de ter sido uma experiência incrível de aprendizagem, onde consegui, após ter desenvolvido esta dissertação e estar dentro da área da Geração Procedimental de Conteúdo e Design de Videojogos, conseguir aprofundar o meu conhecimento desta matéria na área académica.

Esta experiência foi também importante na reflexão sobre esta dissertação. A discutir sobre esta área com os vários participantes presentes na conferência, aprendi que o tema desta dissertação é uma área com interesse e procura ativa para solução. Não existem muitos jogos ou investigação académica na qual a PCG é a base, mesmo que seja muitas vezes usada para ajudar no design de um jogo ao produzir conteúdo. Assim, como um caso de estudo, esta dissertação passa a tomar um lugar relevante nesta área de estudo.

Por outro lado, consegui que participantes conhecedores desta área pudessem experimentar o meu jogo. O protótipo teve problemas, como vai ser explicado no capítulo seguinte sobre as sessões de playtesting, mas a maioria achou que a ideia por detrás do jogo é interessante e da opinião que esta devia ser explorada futuramente.

A participação é assim importante para a conclusão desta dissertação, já que permitiu adquirir conhecimentos e opiniões dentro da área académica antes de dar esta por terminado.

(Nota: O abstract aceite para a apresentação da demo no DiGRA-FDG está na secção de Anexos, como Anexo 2.)

5.5 PLAYTESTING E AVALIAÇÃO

Foram realizadas 7 sessões de *playtesting* do protótipo produzido, o qual o resultado foram respostas a um questionário.

Os jogadores das sessões de *playtesting* foram escolhidos como

Confidants [72], pessoas de confiança. A escolha de selecionar pessoas de confiança foi feita pelo facto de as conhecer, e conseguir juntar um grupo de pessoas de várias áreas diferentes, grande parte jogadora ávida de tipos diferentes de videojogos, para darem uma nova visão sobre o jogo, como também um *feedback* mais qualitativo do que quantitativo [72]. Usou-se um protótipo com dois mapas de tamanhos diferentes para fazer as sessões de *playtesting*, um mapa pequeno de 100 por 100 blocos, e um mapa grande de 200 por 200 blocos. A única informação dada aos jogadores antes da sessão de *playtesting*, foi uma breve apresentação do conceito do jogo e a indicação dos controlos básicos, como o uso do rato para movimento, botões para reiniciar o jogo ou fazer *respawn* da semente caso fica-se presa num bloco (um bug comum no protótipo).

Os resultados destas sessões estão na secção de Anexos como Anexo 3.

QUESTÕES

As primeiras 2 perguntas do questionário servem para mostrar onde foi testado o protótipo, e dar uma noção de quem é a pessoa que respondeu. Os questionários foram anónimos e realizados após as sessões de *playtesting*.

Da pergunta 3 à pergunta 6 foi para saber quais os tipos de mapas que os jogadores experimentaram, quanto tempo duraram no jogo, quantas vezes morreram e se conseguiram sequer completar o jogo. O resto das perguntas é para encontrar os pontos fortes e os pontos fracos do jogo, encontrar problemas a resolver e receber opinião de jogadores sobre o que gostam e gostavam de ver na versão final do jogo.

1- Sistema operativo: Windows, Linux, Mac.

2- Sobre ti: Que tipo de jogos mais gostas (alguns exemplos) e o porque?

3- Jogaste quais versões: small map, large map

4- Concluíste quais versões: small map, large map

5- Quantas vezes perdeste: small map, large map

6- Quanto tempo demoraste a terminar o jogo? Small map, large map

7- Consideras o jogo: Fácil ... difícil

8- Por pensamentos gerais, qual é a tua opinião sobre o jogo?

- 9- Foste capaz de aprender rapidamente os controlos?
- 10- O que pensas sobre o movimento da semente?
- 11- Qual foi o maior desafio no jogo?
- 12- Que emoções mais sentiste durante o jogo? (por exemplo: curiosidade, atenção, adrenalina, aborrecimento, frustração etc)
- 13- Após jogares, que informação seria útil pela primeira vez que jogaste?
- 14- Houve algum momento de confusão? Se sim, qual?
- 15- Reparaste que é possível destruir blocos se colidires com uma parede a uma maior velocidade?
- 16- Reparaste que ao criar uma nova planta existem vários efeitos na semente?
- 17- Se sim, o quê?
- 18- Reparaste que dependendo do tipo de tile há um efeito diferente na semente?
- 19- Se sim, o quê?
- 20- A tua opinião sobre o estilo estético?
- 21- O que gostaste no jogo? Porquê?
- 22- O que não gostaste no jogo? Porquê?
- 23- Quais achas que são as maiores falhas no jogo?
- 24- O que farias para melhorar o jogo?
- 25- Comentários

AVALIAÇÃO

A partir dos resultados consegui adquirir um panorama geral do estado do protótipo e encontrar alguns problemas a resolver no design do videojogo. Vou levar a avaliação do playtesting como uma discussão sobre o resultado do protótipo e design do jogo, vou começar por fazer uma breve apresentação dos resultados obtidos, seguido dessa discussão do razão de ser desses resultados e o que posso fazer para resolver os problemas encontrados.

A partir do resultado das primeiras perguntas, reparou-se que foi o mapa pequeno no qual mais jogadores completaram o jogo, com 6 de 7 jogadores a termina-lo. No caso do mapa grande, só 3 dos 7 jogadores o terminaram.

A opinião sobre a dificuldade do jogo esteve espalhada, sendo que houve um numero quase equilibrado de jogadores que consideraram o jogo fácil, de média dificuldade e difícil.

Os controlos foram considerados fáceis de aprender e a maior parte dos jogadores achou o movimento da semente interessante.

Grande parte dos jogadores teve dificuldade em descobrir qual era o objetivo do jogo.

A emoção mais comum sentida pelos jogadores foi frustração.

Todos os jogadores sentiram momentos de confusão no jogo, tanto por não perceberem bem os controlos ou não saber qual é o objetivo.

Grande parte dos jogadores não descobriu as interações por trás do jogo, as quais não lhes foram indicadas, como a destruição de blocos (onde só 2 de 7 jogadores se aperceberam que era possível), como também foram os poucos que se aperceberam que ao plantar a semente esta recuperava vida e os seus atributos modificavam.

A maior parte dos jogadores gostou do uso de pixel art no jogo.

Para os jogadores, os elementos mais interessantes do protótipo foram o plantar da semente e a exploração dos mapas gerados.

Os jogadores não gostaram do facto de não descobrirem o que é o objetivo.

Destes resultados considero mais importantes apontar foi como grande parte dos jogadores considerou o protótipo incompleto, frustrante e pouco interessante. Para por estes resultados em perspetiva, é necessário ter em mente que este é um protótipo inicial do que poderia ser um futuro videojogo completo. O protótipo no momento acaba por ser apenas um proof of concept do Gerador do Planeta, Gerador de conteúdo em tempo real e do sistema de plantar a semente. Bugs são um problema comum deste protótipo, e por vezes podem arruinar a experiência de jogo ou criar situações desagradáveis. Por outro lado, o facto do mundo estar vazio, apenas com três tipos de blocos diferentes que acabam por reagir da mesma maneira, acaba por não criar um grande interesse no jogo. Outro elemento que penso que faz falta no jogo era toda a interface criada pela semente, por onde que o efeito dos vários atributos, e a evolução destes ao longo do jogo enquanto o jogador planta a semente, acabam por ser facilmente visíveis

na semente (como o numero e tipo de fragmentos em volta da semente mudarem).

Defendo que o maior problema nestas sessões de *playtesting* foi que o protótipo experimentado estava demasiado incompleto. Era necessário mais conteúdo para se conseguir ter uma melhor experiência de jogo. Apesar disso, o gerador do planeta teve um bom resultado, já que grande parte dos jogadores gostou dos mundos criados, apesar de conterem poucos elementos.

Um ponto que tenho que corrigir é o facto do jogador poder criar quantas plantas quiser. É necessário criar um sistema que limite o numero de vezes que a semente pode ser plantada. A ideia de momento, é ter uma espécie de energia que o jogador carrega a partir ou de tempo, movimento ou de algum novo elemento que se pode colocar no mundo. Ao chegar a um certo limite dessa energia, o jogador pode criar uma nova planta.

Outros pontos em ter consideração é aumentar o campo de visão do jogo, permitir que o jogador tenha mais informação do que acontece à volta para não se sentir tão perdido. Assim o jogador tem mais facilidade em conseguir ver por onde já passou, o terreno já modificado e por onde pode mais explorar. Alguns jogadores recomendam também que o gerador gere espaços mais abertos, o que é fácil de fazer ao alterar só uma variável no gerador já produzido.

Para concluir, penso que os resultados desta sessão de *playtesting* são legítimos para o artefacto que foi concluído, mas como esta é uma versão bastante incompleta do jogo final e com alguns *bugs* que estragam a experiência de jogo, não conseguem dar uma verdadeira opinião sobre o design final do jogo. O próximo paço é desenvolver um protótipo mais completo e ai sim permitir uma maior sessão de *playtesting* de forma a que se consiga avaliar verdadeiramente o design do jogo. Apesar disso, sei que a maior parte dos jogadores achou a ideia interessante quando discuti com eles pessoalmente, o que me deixa confiante que este seja um projeto que deva ser continuado no futuro

6.

REFLEXÃO E CONCLUSÃO

O Design de Jogos Generativos foi o tema desta dissertação, e o seu objetivo demonstrar o que é Design de Jogos Generativo no campo de Design de videojogos, de forma a mostrar como é possível o uso de Geração Procedimental de Conteúdo para criar a base de um videojogo, e não apenas uma ferramenta que assiste o design de jogos.

Para encontrar a resposta a este tema foi decidido criar o design de um jogo e um protótipo que servisse como prova de conceito desse design. Daqui surgiu o jogo Adamastor como um exemplo de Generative Game Design.

O processo de procura para chegar a este resultado começou por uma pesquisa sobre o que é o Design de Jogos, a Geração Procedimental de conteúdo e como a simbiose de ambos conseguem criar diversas experiências únicas e interessantes. Desta pesquisa resultou o conhecimento, de como já foi aplicado o PCG em Design de Jogos, as várias ferramentas e algoritmos utilizados para chegar a esse fim. Foi a partir deste conhecimento que tomei as decisões que tomei durante o processo de design.

Até chegar ao design final do jogo Adamastor foi preciso várias ideias diferentes, encontrar os problemas e prontos fracos nestas e os corrigir. É por isso que o resultado final desta dissertação só surgiu após a produção e análise de dois designs de jogos distintos. Mas na minha opinião esse é o processo de Game Design, um jogo não é criado à primeira vez, mas sim

após um sistemático processo de análise e refinação do que foi construído. E por vezes, certas ideias em certos contextos simplesmente não resultam.

Durante o processo de desenvolvimento e construção deste jogo, eu como estudante e futuro profissional também cresci e evolui. Grande parte desta dissertação foi eu a aventurar-me por áreas que nunca tinha estudado e ferramentas na qual não tinha muita prática, mas esse é o processo de aprendizagem, uma pessoa só aprende ao enfrentar esses desafios. Construir e tentar perceber porque uma ideia ou design de jogo resultam ou não resultam por si também é um processo de aprendizagem, saber e conseguir encontrar os defeitos nas próprias coisas que construímos, e assim os conseguir arranjar e melhorar é uma habilidade essencial para qualquer Designer, seja de qual área for, a nossa profissão resume-se em resolução de problemas.

Apesar do protótipo construído ter ficado aquém das expectativas do protótipo que iria ser criado no início do projeto, penso que não foi tudo em vão. O design do videogame tem potencial, e como desafio merece ser explorado. Isto é uma prova que realmente Generative Game Design é um bom método de pensamento para fazer o design de jogos. Este permite encontrar novas soluções para videogames, é possível que futura exploração deste tema origine jogos completamente originais, diferentes do que a história de videogames mostrou até agora.

7.

ANEXOS

Ideia - Jogo de sobrevivência

Playfulness	Challenge	Sensemaking	Embodiment	Sensoriality	Sociability
<p>1. Que espaços de liberdade o vídeojogo suporta? Total de liberdade na exploração do mundo. Interação simples com o mundo (ex. comer plantas ou animais, beber água, reprodução) Modificar o mundo a partir da consequência dessas ações. Criar espécie/personagem?</p>	<p>4. Quais os objetivos que o vídeojogo propõe? Sobreviver até ao final de um determinado espaço de tempo (o jogo vai sempre terminar em derrota sem conhecimento prévio do jogador).</p>	<p>7. Que fenômeno é representado no jogo ou interpretado pelos jogadores? O meio ambiente e os habitantes do mundo. Representação realista/icônica das ações do e sob o ambiente. A mutação do mundo ao longo do tempo (destruição).</p>	<p>10. Como se manifesta a presença dos jogadores no mundo de jogo? O jogador é representado por um animal no mundo. O avatar do jogador transmite a maioria das informações sobre o seu estado pela sua representação (se tem fome, sede, de boa ou má saúde, etc).</p>	<p>13. Qual a natureza e intenção dos estímulos sensoriais? Os estímulos são feitos através de imagem, animação e som. Estes vão representar o ambiente natural onde reside o jogador. O jogo começa com uma representação natural do mundo (verde, sons da natureza, etc), mas ao passar do jogo este vai-se tornando</p>	<p>16. Que relações o jogo propõe ou promove? Discussão? N/A</p>
<p>2. Que elementos suportam a atividade livre dos jogadores? Todo o meio ambiente pode ter interação com a criatura do jogador (tanto criaturas, plantas e objetos no mundo).</p>	<p>5. Qual a natureza do desafio encontrada no vídeojogo? Estratégia e reflexos do jogador.</p>	<p>8. Que eventos significativos são representados/elicitados/interpretados no jogo? Comer, beber, caçar, etc. Mutação do mundo.</p>	<p>11. Como se caracteriza o mundo de jogo? O mundo de jogo é finito e de média/grande dimensão (demora algum tempo a ser explorado). Este representa o habitat do animal o qual vai sofrer mutações ao longo do tempo.</p>	<p>14. Quais as oportunidades de contemplação que o jogo oferece? A destruição visual e sonora da natureza pela ação do Homem. Sons de resposta às ações do jogador (morte de outro animal, do jogador, sons a comer e a beber).</p>	<p>17. Qual é estrutura ou topologia que o jogo promove? Jogador vs jogo. Jogo sem estrutura e experimental, o jogo tem liberdade de interagir sobre o mundo, não há objetivos fixos.</p>
<p>3. Como se caracteriza o espaço de possibilidades explorável pelos jogadores? Mundo de grandes dimensões. O jogador tem um leque de interações simples mas as quais podem ser usadas em quase todo o ambiente. O jogador pode criar a sua personagem?</p>	<p>6. Qual o feedback de desempenho dos jogadores? Conseguir reproduzir animais da sua espécie. Perda no final do jogo.</p>	<p>9. Quais são os papéis assumidos pelos jogadores? Um animal que tenta sobreviver no seu meio natural.</p>	<p>12. Como os jogadores interagem com o espaço de jogo? Teclado e rato no computador. Touch.</p>	<p>15. De que oportunidades de exploração estética o jogador dispõe? Mutações do habitat natural do animal, para um ambiente sujo e devastado pelo ser humano.</p>	<p>18. Qual o tipo de mediação entre os jogadores? N/A</p>

Ideia - Jogo criar herói

Playfulness	Challenge	Sensemaking	Embodiment	Sensoriality	Sociability
<p>1. Que espaços de liberdade o vídeojogo suporta? Exploração do mundo. Exploração dos poderes da personagem.</p>	<p>4. Quais os objetivos que o vídeojogo propõe? Conseguir chegar ao final do nível sem morrer. O jogador pode ter várias possibilidades de como o fazer.</p>	<p>7. Que fenómeno é representado no jogo ou interpretado pelos jogadores? Pixel art. Mundo, personagem e npcs. O mundo é alterado dinamicamente ao reagir ao gameplay do jogador.</p>	<p>10. Como se manifesta a presença dos jogadores no mundo de jogo? O jogador é representado por uma personagem em jogo. Esta interage direta e indiretamente com o mundo de jogo. (indiretamente é o mundo que lê o que a personagem faz). 2D.</p>	<p>13. Qual a natureza e intenção dos estímulos sensoriais? Estímulo visual e sonoro. O estilo visual e sonoro do jogo mudam com as decisões do jogador, dinâmico.</p>	<p>16. Que relações o jogo propõe ou promove? N/A</p>
<p>2. Que elementos suportam atividade livre dos jogadores? O movimento e poderes da personagem.</p>	<p>5. Qual a natureza do desafio encontrada no vídeojogo? O desafio é o próprio jogo em si, que se adapta ao estilo de jogo do jogador. Essa adaptação serve para manter um constante ambiente de ameaça e desafio ao jogador.</p>	<p>8. Que eventos significativos são representados/ elicitados/interpretados no jogo? Movimento, saltar, uso dos variados poderes, matar inimigos, terminar o nível.</p>	<p>11. Como se caracteriza o mundo de jogo? Mundo finito 2D em estilo pixel art. Cada nível e mundo é gerado pelo computador e o qual se adapta e ganha mutações a partir das ações do jogador.</p>	<p>14. Quais as oportunidades de contemplação que o jogo oferece? Contemplação do ambiente e da adaptação que o jogo faz ao jogador, o facto de lhe estar sempre a criar desafio. Leitura do mundo juntamente com a dinâmica criada com a conversa subtextuada entre o jogador e jogo.</p>	<p>17. Qual é a estrutura ou topologia que o jogo promove? Jogador VS jogo. O jogador tenta vencer o mundo criado pelo jogo e este tenta derrotar e criar desafio ao jogador.</p>
<p>3. Como se caracteriza o espaço de possibilidades explorável pelos jogadores? Finito e dinâmico. O mundo vai sofrer modificações à medida que o jogador explora e interage com o mundo. A partir dessa interação os níveis seguintes vão ser afetados pelas decisões e ações do jogador. O possibilidade de movimento dada à personagem e ainda os vários poderes que o afetam a si e o</p>	<p>6. Qual o feedback de desempenho dos jogadores? Estética, regras e reação do mundo e personagens em relação às ações e escolhas do jogador. Completar o nível. Pontos?</p>	<p>9. Quais são os papéis assumidos pelos jogadores? O herói do jogo.</p>	<p>12. Como os jogadores interagem com o espaço de jogo? Teclado. Comando. Ecrã touch?</p>	<p>15. De que oportunidades de exploração estética o jogador dispõe? A ação do jogador é o gerador de experiências audiovisuais. O jogo muda esteticamente e sonoramente para se adaptar ao jogo do utilizador.</p>	<p>18. Qual o tipo de mediação entre os jogadores? N/A</p>

Ideia - Descobrimientos

Playfulness	Challenge	Sensemaking	Embodiment	Sensoriality	Sociability
<p>1. Que espaços de liberdade o vídeojogo suporta? Exploração do mundo do jogo. Controlar uma frota de navios dentro desse mundo. Interação com personagens dentro do jogo. Combate.</p>	<p>4. Quais os objetivos que o vídeojogo propõe? Descobrir um dos locais e rotas definidos no início do jogo no mapa do mundo. Conseguir regressar à capital para guardar as informações recolhidas durante uma viagem.</p>	<p>7. Que fenómeno é representado no jogo ou interpretado pelos jogadores? O mapa, a frota do jogador e outras personagens e eventos são representados áudio e visualmente.</p>	<p>10. Como se manifesta a presença dos jogadores no mundo de jogo? A partir de uma frota de navios. God view.</p>	<p>13. Qual a natureza e intensão dos estímulos sensoriais? Som ambiente da vida no oceano. Sons que antevem eventos (por exemplo tempestades) e sons que representam eventos.</p>	<p>16. Que relações o jogo propõe ou promove? N/A</p>
<p>2. Que elementos suportam a atividade livre dos jogadores? A frota que o jogador controla para explorar o mundo. Personalizar a frota com tripulação, equipamento e navegadores (heróis).</p>	<p>5. Qual a natureza do desafio encontrada no vídeojogo? Estratégia. Arriscar ou não arriscar.</p>	<p>8. Que eventos significativos são representados/ elicitados/interpretados no jogo? Descobrir, contactar, lutar, trocar, comunicar.</p>	<p>11. Como se caracteriza o mundo de jogo? Um grande mundo de jogo finito, na sua maioria um oceano. Uma simplificação do que poderá ser um mundo real.</p>	<p>14. Quais as oportunidades de contemplação que o jogo oferece? Como foi pensado e organizado na realidade os descobrimentos. Não foi apenas enviar barcos com a esperança de descobrir algo novo e estava aí feito, mas sim um método científico de viajarem e voltarem com provas e novos conhecimentos para alimentar as expedições seguintes.</p>	<p>17. Qual é a estrutura ou topologia que o jogo promove? Jogador VS jogo</p>
<p>3. Como se caracteriza o espaço de possibilidades explorável pelos jogadores? Um mapa de grandes dimensões. A frota à disposição do jogador. Os navegadores, tripulação e equipamento utilizável pelo jogador. Outras personagens e locais dentro do mundo de jogo.</p>	<p>6. Qual o feedback de desempenho dos jogadores? Recompensas quando o jogador consegue regressar à capital após fazer novos descobrimentos, que permitem melhorar a frota e os seus constituintes. Caso a frota seja destruída o jogador tem que retomar a exploração com uma frota inicial e sem os descobrimentos da última viagem.</p>	<p>9. Quais são os papéis assumidos pelos jogadores? Um líder omnipotente que comanda a viagem de uma frota nos descobrimentos.</p>	<p>12. Como os jogadores interagem com o espaço de jogo? Rato, teclado. Touch.</p>	<p>15. De que oportunidades de exploração estética o jogador dispõe? Personalização da frota. Exploração do mapa do jogo.</p>	<p>18. Qual o tipo de mediação entre os jogadores? N/A</p>

ANEXO 2

ABSTRACT

Adamastor is an exploration game with rogue-like elements that uses a Generative Game Design approach. The game takes place on a living planet where the player must search for a sanctuary hidden in the game world. The planet is sentient, and using procedural generation shifts its own land masses and oceans, while adapting its terrain types and geology to create obstacles, affect the player's perceptions to trying and stop the sanctuary to be found. The idea is to use PCG (Procedural Content Generation) to mediate emotions such as the feeling of challenge of going against the elements and uncertainty of exploring the unknown.

The focus of this design is in exploring the Discovery and Challenge aesthetics from the MDA framework (Hunicke, LeBlanc, and Zubek 2004), using the dynamics described in (Smith 2014) as Searching in a Vast World, Building Generator Strategies and Practicing in Different Environments. It takes also inspiration from various games, such as the map generator of Civilization IV (Firaxis Games 2005), exploration of large open spaces like in Elite (Acornsoft 1984) and the use of diverse biomes in Terraria (Re-Logic 2011). The design uses a mix of Procedural Content Generators that will work on different moments. The first one works offline to create the world before the beginning of the game, based on Pseudo-Random Number Generators to create various layers of data for the world, such as height maps and ocean currents. The second one works online, during gameplay, and will generate content based on the analysis of player behaviour. This includes changing the terrain unexplored by the player to adapt to the playstyle, creating obstacles such as natural barriers, dynamic challenges such as moving obstacles, or manipulating player's perceptions of the environment.

Conceptually, the game design was inspired by the method of exploration the XV and XVI century Portuguese explorers undertook during the "Descobrimentos" (The Portuguese Discoveries), as well as the feelings they experienced during their voyage, recorded in works such as *Os Lusíadas* (Camões 1572). The objective of those discoveries was finding

the maritime route to India, and to accomplish this they adopted a careful method of exploration, sending the explorers into the sea, always keeping in mind that they should at all costs return home with what they found, even if nothing. The knowledge they found on each expedition allowed for them to perfect the methods and technologies they used to explore the seas, adapting and surpassing the challenges they encountered. This is the challenge that the design of this game will create for the players.

In the game the player won't be a human explorer but a seed of a plant, which is trying to find a place in the world where it can grow and prosper far from danger. The seed is made up of various fragments that are connected by a core, randomly chosen at the start of the game and each with a special skill. As the game progresses the player will be able to trade and enhance the fragments in order to adapt to the challenges created by world. To control the seed's movement the player is given the power of the wind, as to create a flow with a direction and velocity which the seed will follow while floating around the world.

While exploring the world, there are things the player has to manage in order to be successful: The wind is a resource that declines whenever the player uses it, but it will constantly recharge until it hits a maximum limit given by the seed; Since the seed is floating, the player has to control its altitude, if the seed hits the ground on a wrong kind of terrain it can be destroyed or drowned; Also, the player must be careful with the seed as the fragments can break up, meaning the end of the game. The seed resistance is determined by the number of fragments it's composed, and the abuse of the wind power like creating high speeds or hitting obstacles will make those fragments break up.

But falling in ground is also the method which the player can adapt to the world, by sowing the seed and growing a new temporary plant that can create a new seed. So, depending where the seed falls on the ground, it can create a new plant or be destroyed. And if the plant grows, the terrain type will affect how the player will be able to modify the new seed.

Adamastor is an experiment in designing a game around a PCG and not use it only as a tool to generate game content, but also to create the experience.

KEYWORDS

Generative Game Design, Game Content Generation, Digital Game

BIBLIOGRAPHY

Acornsoft. (1984). *Elite*. [BBC Micro, Acorn Electron]. Acornsoft.

Camões, Luís Vaz de. 1572. *Os Lusíadas*.

Firaxis Games. (2005). *Civilization IV*. [PC Computer]. 2K Games.

Hunicke, Robin, Marc LeBlanc, and Robert Zubek. 2004. "MDA: A Formal Approach to Game Design and Game Research." *Workshop on Challenges in Game AI*, 1–4. doi:10.1.1.79.4561.

Re-Logic. (2011). *Terraria*. [PC Computer]. Re-Logic.

Smith, Gillian. 2014. "Understanding Procedural Content Generation: A Design-Centric Analysis of the Role of PCG in Games." *Proceedings of the 2014 SIGCHI Conference on Human Factors in Computing Systems (CHI 2014)*, 917–26. doi:10.1145/2556288.2557341.

ANEXO 3

QUESTIONÁRIO SESSÕES DE PLAYTESTING

Sistema operativo onde foi testado o jogo

- Windows

Que tipo de jogos gostas? Dá exemplos e explica porque é que gostas

- RPG, Acção, RTS, Simuladores, qualquer coisa que me mantenha interessado, seja através de mecânicas, história ou conceito.
- simulação por ser desafiante
- Gosto de jogos de contra relógio. Gosto sempre de testar se consigo fazer melhor e em pouco tempo, tipo o fruit ninja.
- Gosto de jogos de aventura, fantasia, sci-fi, com um combate third person/ first person. Por exemplo: Dark Souls, Witcher.
- Diferentes genres, mas especialmente jogos tenham algo que nos façam ficar agarrados, sejam por ser desafiantes ou por despertarem sensações de realização ou recompensa ainda que subconscientemente.
- Jogos de acção ou que nos façam pensar. Deus-Ex, Assassins creed

Quais versões do jogo testaste?

- Small Map, Large Map
- Small Map, Large Map
- Small Map, Large Map
- Small Map

- Small Map
- Large Map
- Small Map, Large Map

Terminaste o jogo em quais versões?

- Small Map, Large Map
- Small Map
- Small Map
- Small Map
- Small Map
- Large Map
- Small Map, Large Map

Quanto tempo demoraste a terminar o jogo em cada mapa?

- 15-45 segundos no pequeno, 2-3 mins no grande.
- 3 minutos no Small Map e minutos no Large Map
- Small Map 1 minuto. Large Map não terminei.
- estive a jogar durante 2 min e não percebi se terminei ou não.
- 2 minutos e 35 segundos
- 5 minutos
- Small - 1min06seg ; Large - 1min34seg

Quantas vezes perdeste no jogo?

- 1
- 3
- 0
- nao sei, nao percebi como funciona o jogo
- 1 vez
- 4 em 5 tentativas
- Small - 1 ; Large - 0

Consideras o jogo: (fácil 1 a difícil 5)

- 1
- 3
- 3

- 5
- 1
- 4
- 2

Em poucas palavras qual é a tua opinião sobre o jogo?

- Muito espartano. Não tem que chegue para interessar a curto nem longo prazo, a geração de terreno não é rápida nem caótica o suficiente para se tornar um obstáculo e o mapa gerado não tem zonas abertas o suficiente para tirar partido dos aumentos de velocidade e wind supply. O facto de ser possível plantar sementes sem pausa trivializa imenso o jogo.
- Considero-o desafiante, bem construído e simples de compreender
- Incompleto.
- está confuso, nem percebi qual o objetivo.
- É um conceito interessante, no entanto achei um pouco estranho a maneira como se movimentava a planta. Outra coisa que achei que precisa de melhoramento é a indicação de que a zona “azul” é o lugar que ao ser alcançado indica o fim do jogo, na altura pensava que fosse outro “Biome” que tivesse um comportamento diferente, mas mal entrei apercebi-me que era o fim.
- O jogo está bastante fraco a nível de conteúdo e usabilidade mas o conceito está interessante e provado que pode funcionar e ser divertido se existirem algumas modificações.
- O jogo é difícil de entender ao início, mas depois é simples e intuitivo

Qual o grau de dificuldade para aprender os controlos? (1 fácil a 5 difícil)

- 1
- 4
- 1
- 1
- 2
- 1

- 1

Qual é a tua opinião sobre o movimento da semente?

- É frustrante. A semente não responde sempre aos movimentos, e mover com precisão é difícil quando não é possível prever se o input vai ser tido em conta, ignorado, ou mal lido.
- Achei interessante o movimento com o rato, dando mais fluidez ao jogo e criando um contraste com a imagem retilínea do mapa e das figuras.
- Esta orgânico.
- As sementes algumas semeavam-se sozinhas, outras eu tentava secar e não acontecia nada.
- Achava que seria interessante, em vez de ela flutuar, ter umas espécie de slingshot effect, em que se atirava como se fosse uma bala no Worms (com efeito de gravidade), que assim que se conectasse uma parede carregava-se no espaço e plantava-se a semente.
- Bem conseguido, nada a criticar.
- Parece uma nave do space invaders, mas está engraçada

Qual foi o maior desafio no jogo?

- Nada em particular.
- Tentar perceber quais os melhores espaços de plantação da planta
- Não bater com a semente nos tiles que fazem ricochet.
- Perceber qual o objetivo e como se joga efetivamente
- Descobrir como se atingia o fim
- Saber para onde ir.
- O maior desafio do jogo foi entender como se comportava o mundo e como ganhar vida quando se semeia plantas

Que emoções sentiste durante o jogo? (por exemplo: Diversão, curiosidade, desafio, aborrecimento, frustração, etc)

- Aborrecimento e frustração.
- Confusão, curiosidade, desafio, satisfação
- Frustração.
- frustração

- Senti algum desafio, pois a cada momento que passava a minha vida diminuía. Para tal tinha que plantar sementes para sobreviver enquanto procurava o meu destino.
- Desafio, frustração, curiosidade, diversão e pânico (no bom sentido, por temer que o mundo me fosse apanhar)
- Curiosidade e frustração (ao início)

Após o jogo, que informação seria útil ter ao jogar pela primeira vez?

- Não há nada de particular interesse que falte explicar depois de ter lido a introdução dada no questionário.
- Apresentar, de modo subtil, as indicações de fertilidade (saúde, vento, velocidade)
- Como se completa o jogo.
- Perceber qual o objetivo e saber mais do que somente que controlos usar
- A indicação do objectivo do jogo no início: “Atinge a área azul para concluir o jogo.”
- Saber se renovar ou não a planta cria terra firme.
- O que significam algumas acções

Houve momentos em que te sentiste confuso? Se sim, quais?

- Apenas ao início, quando demorei a reparar que a vida vai desaparecendo.
- Sim, até compreender como usar os controlos e compreender que o WindSupply era essencial ao jogo. A maior confusão surgiu quando descobri a terra fértil, pois era um espaço novo, desconhecido e diferente de todo o resto do mapa.
- Sim, o objetivo.
- Todos os momentos
- Na primeira vez que experimentei, não sabia muito bem para que lado me virar, mas pronto, isso faz parte da componente de exploração deste jogo.
- Sim grande parte do tempo por não saber por onde ir
- Não

Deste conta que é possível destruir blocos ao colidires com eles a uma maior velocidade?

- Sim
- Não
- Não
- Não
- Não
- Não
- Sim

Reparaste que ao semear a semente existem vários efeitos na nova semente? Se sim, quais?

- Sim. Respondo na seguinte.
- Não
- Não
- nao reparei
- Não
- Recuperar vida foi a única coisa que reparei
- Não

Reparaste que ao semear a semente em cada tipo de bloco os efeitos na semente são diferentes? Se sim, quais?

- Sim. Em solo castanho claro aumenta a velocidade máxima, em castanho escuro aumenta o Wind Supply. Suponho que não hajam mais solos, já que o único que faltou referir foi o azul, e sempre que plantei nesse o jogo terminou.
- Sim. Ao semear nos blocos mais “húmidos” a área cultivada era maior, ainda que apenas junto à margem. Os bolcos mais “secos” só eram cultivados se estivessem ao lado de lugares cultivados. Os blocos férteis concluíam o jogo.
- Não
- Eu vi que eles iam mudando, mas não percebi o porquê de mudarem
- Não, teria que jogar mais vezes para sentir esse detalhe. Como estava preocupado em aprender como funcionava, não reparei
- Não reparei

- Não

Qual é a tua opinião sobre o estilo artístico?

- Desinteressante. Dá mais uma sensação de ser um estilo minimalista para evitar ter de detalhar do que por limitações ou homenagem.
- Simples e eficiente, de fácil compreensão de figuras.
- Esta bem feito.
- Gosto deste estilo..
- Faz-me lembrar os estágios iniciais do desenvolvimento dos jogos há duas décadas atrás.
- Retro fica sempre bem e os gráficos estão simples mas engraçados. O fundo preto é que poderia ser mudado para algo com conteúdo.
- 8-bits parece-me bem

O que gostaste no jogo? Porquê?

- Nada de particular.
- O modo de “cultivo” da semente, ou seja, todo o desafio até chegar à compreensão do ato de cultivar e a satisfação resultante da primeira experiência, com o “nascer” de uma planta.
- Do movimento permanente da semente, está orgânico.
- Gostei do design.
- Gostei da arbitrariedade dos espaços criados e do desafio que é explorá-los, porque torna cada experiência um pouco diferente
- A sensação de desafio e inquietação ao passar por passagens estreitas pois despertou um pânico um pouco primal em mim. Não querer ficar subterrado é um pensamento instintivo.
- A aleatoriedade do mapa

O que não gostaste no jogo? Porquê?

- A geração do terreno é desinteressante, e os ambientes com que me deparei eram fechados demais para poder andar depressa com a semente e não permitiam grande criatividade no movimento.
- Apenas o início, pois o jogo, ao abrir, não indica o que fazer, pelo que os únicos recursos serão testar os controlos e observar o que acontece, explorar.

- Não dá para perceber muito bem quais são os obstáculos, pois as vezes em sítios de pedra a semente passa e outras vezes não, por vezes colide com terreno normal e só passado alguns segundos é que ele muda para rochoso.
- Achei muito confuso no geral.
- Não gostei muito da parte gráfica, acho que o jogo poderia ter umas texturas mais futurísticas, mesmo sendo um conceito simples, talvez lhe desse mais potencial
- Sensação de não estar a caminhar em direção a um objectivo concreto ou que as minhas ações não estavam a levar-me a lado nenhum, descobrir o final é apenas graças a sorte e aleatoriedade.
- Nada

O que corrigias no jogo?

- “Aumentava o campo de visão e enviezava o gerador para criar mais áreas abertas. Neste jogo o factor mais relevante é o controlo do jogador sobre a semente, e com um campo de visão e áreas abertas tão pequenos há pouca liberdade para experimentar mexer a semente, atingir velocidades altas, etc. Factores externos que afectem o movimento (ajudando ou impedindo o jogador) ou que impossibilitem plantar a semente em certas zonas poderiam aumentar o interesse.”
- A falta de um ponto de orientação, pois todas as voltas possíveis, além de o facto do jogo adicionar novos blocos, podem “desnortear” o jogador no mapa e fazê-lo perder a percepção de onde poderá estar localizada a terra fértil.
- Tornaria o jogo mais intuitivo de alguma forma, faltam pistas para entender os conceitos do terreno e do objectivo do jogo.
- Talvez se percebesse ao certo como funciona poderia dar uma sugestão
- O problema de se sentir preso debaixo de terra e se ficar lá preso algumas vezes.
- É mais fácil ver nos comentários que aponte durante o jogo.
- a função que a tecla R faz pois é possível saltar de um buraco para fora dele.

O que farias para melhorar o jogo?

- Opção 1
- Opção 1
- Torna-lo de alguma forma mais interessante e divertido.
- Talvez se percebesse ao certo como funciona poderia dar uma sugestão
- Mudava as suas texturas, tentava acrescentar alguma lore ao jogo, dando algum significado na história que é o salvamento desta planta
- Ver comentários.
- talvez mais acções na semente, como disparar algo com tempo limite

Comentários

- Além das sugestões já mencionadas, acho que o as indicações no início do jogo serão as mais importantes. Durante o jogo há, por vezes, momentos em que não se consegue ler muito bem as indicações pois ficam por cima de áreas mais claras. Acho que uma pequena edição de sons no jogo seria interessante e iria apelar ainda mais o jogador - por exemplo, efeitos quando a planta é cultivada, sons distintos no movimento da semente (dentro dos blocos e fora dos blocos) e o fim (perder ou concluir o jogo)
- “A ideia que se tem é que o jogador esta a ver a semente por cima mas depois é estranho o facto de só se poder plantar nas bordas do mapa. Também não faz igualmente muito sentido poder se andar dentro da parte castanha e assumir que estamos dentro de uma gruta. Essas duas coisas principalmente são um obstáculo para perceber o mapa.
“
- Porque não tornar isto num mobile game?
- “Estes apontamentos foram tirados durante o gameplay, à medida que fui jogando fui apontando coisas, é mais fácil perceber a minha mentalidade a jogar o jogo a partir daqui. Falta de loading screen é frustrante Objectivo do jogo é pouco claro. Fartei-me de pôr plantas durante o meu primeiro playthrough e perdi antes de conseguir perceber o objectivo. Crashes constantes, não consegui reiniciar o jogo. O mundo a mover-se para interromper o caminho é

bastante interessante mas podia ser mais agressivo. Não fez grande diferença no caminho que eu tentava seguir.

Ter de fechar o jogo para reiniciar um novo é frustrante também.

A segunda vez que joguei consegui completar o jogo facilmente. Final muito abrupto, não me senti recompensado por chegar ao fim, devido a ter sido demasiado fácil.

À terceira senti mais dificuldade em certas passagens com menos espaço. Eventualmente fui mesmo preso pelo mundo, não conseguindo terminar o que me despertou mais vontade de jogar pois senti-me derrotado especialmente sabendo que a culpa foi minha por me meter num caminho mais estreito e sem saída. Sensação interessante que pode ser explorada.

Quarta tentativa, voltei a perder, provando que afinal o meu pensamento de que o jogo é fácil, foi apenas uma primeira impressão.

Depende muito da geração do mapa inicial. Se for muito espaçoso, perde a piada, pois é preciso estar bastante próximo da terra para esta reagir à nossa presença.

Outro pormenor que não cheguei a perceber, o renovar da semente para além de renovar o HP também fixa a terra? Fiquei com essa impressão mas não tenho a certeza.

Deveria existir um limite de renovações para tornar as coisas mais interessantes. O nível de vento disponível também deveria ser menor e recuperar mais lentamente, para obrigar o jogador a ter que jogar com o “planar”. Ter de aproveitar o momento para conseguir chegar onde se quer é muito mais difícil do que simplesmente estar sempre a usar o vento (o que é possível pois renova muito rápido), torna o jogo muito mais interessante.

O jogo está bastante fraco a nível de conteúdo e usabilidade mas o conceito está interessante e provado que pode funcionar e ser divertido se existirem algumas modificações. A sensação de desconforto por não saber quando e onde o mundo se iria transformar para interromper a jornada é interessante e apelativa, no entanto poderia ser ainda mais agressivo o mundo. A dificuldade de jogo para mim foi não saber para onde me dirigir principalmente, o que me levava a caminhos onde o mundo se aproveitava para me barrar. Durante a jornada não senti qualquer sensação de progresso o que não é uma boa sensação e faz com que encontrar o final do jogo seja um pouco aleatório e não nos faça sentir recompensado pois não sentimos que trabalhamos para isso. Uma solução poderia ser ir mudando aos poucos a cor do solo quanto mais perto do final nos encontrássemos mais forte seria a componente azul do solo.

“

7.

BIBLIOGRAFIA E REFERÊNCIAS

- [1] Noah Wardrip-Fruin, P. Harrigan, and G. Frasca, “Videogames of the Opressed: Critical Thinking, Education, Tolerance, and Other Trivial Issues,” in *First Person: New Media as Story, Performance, and Game*, The MIT Press, 2004, p. 85.
- [2] J. Huizinga, *Homo Ludens: A study of the play-element in culture*. Routledge & Kegan Paul Ltd, 1944.
- [3] T. Fullerton, “Defining Games,” in *Game Design Workshop: A Playcentric Approach To Creating Innovative Games*, Morgan Kaufmann, 2008, pp. 42–43.
- [4] Namco, “Pac-Man.” Namco, 1980.
- [5] T. Fullerton, “Working with Dramatic Elements,” in *Game Design Workshop: A Playcentric Approach To Creating Innovative Games*, Morgan Kaufmann, 2008, pp. 86–106.
- [6] T. Fullerton, “Working with Formal Elements,” in *Game Design Workshop: A Playcentric Approach To Creating Innovative Games*, Morgan Kaufmann, 2008, pp. 49–85.
- [7] K. Salen and E. Zimmerman, “Unit 1: Core Concepts | Defining Games,” in *Rules of Play: Game Design Fundamentals*, The MIT Press, 2004, p. 80.
- [8] A. Smith, “THE PRIESTHOOD AT PLAY: COMPUTER GAMES IN THE 1950S,” 2014. [Online]. Available: <https://videogamehistorian.wordpress.com/2014/01/22/the-priesthood-at-play-computer-games-in-the-1950s/>.
- [9] N. Landsteiner, “Spacewar!,” 2012. [Online]. Available: <http://www.masswerk.at/spacewar/>.
- [10] D. Winter, “Pong Story,” 1996. [Online]. Available: <http://www.pong-story.com/>.
- [11] G. Smith and H. Ave, “An Analog History of Procedural Content Generation.”

- [12] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *IEEE Trans. Comput. Intell. AI Games*, vol. 3, no. 3, pp. 172–186, 2011.
- [13] R. Khaled, M. J. Nelson, and P. Barr, "Design metaphors for procedural content generation in games," *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. - CHI '13*, p. 1509, 2013.
- [14] Acornsoft, "Elite." 1984.
- [15] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, "Procedural Content Generation for Games: A Survey," *ACM Trans. Multimed. Comput. Commun. Appl.*, no. February, pp. 1–24, 2011.
- [16] Mossmouth, "Spelunky," 2008.
- [17] E. J. Hastings, R. K. Guha, and K. O. Stanley, "Automatic Content Generation in the Galactic Arms Race Video Game," *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 4, pp. 245–263, 2009.
- [18] I. D. Visualization, "SpeedTree." .
- [19] J. Togelius, N. Shaker, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2015.
- [20] G. N. Yannakakis and J. Togelius, "Experience-Driven Procedural Content Generation," vol. 2, no. 3, pp. 147–161, 2011.
- [21] G. Smith, "Understanding Procedural Content Generation: a Design-Centric Analysis of the Role of PCG in Games," *Proc. 2014 SIGCHI Conf. Hum. Factors Comput. Syst. (CHI 2014)*, pp. 917–926, 2014.
- [22] Valve, "Left 4 Dead." 2008.
- [23] Firaxis Games, "Civilization IV." 2005.
- [24] G. W. Michael Toy, "Rogue." 1980.
- [25] B. Softworks, "The Elder Scrolls V: Skyrim." Bethesda Softworks, 2011.
- [26] T. E. S. Wiki, "Radiant A.I." [Online]. Available: http://elderscrolls.wikia.com/wiki/Radiant_A.I.
- [27] M. Persson, "Minecraft." 2011.
- [28] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive planning and constraint solving for mixed-initiative level design," *IEEE Trans. Comput. Intell. AI Games*, vol. 3, no. 3, pp. 201–215, 2011.
- [29] K. Perlin, "An image synthesizer," *SIGGRAPH Comput. Graph.*, vol. 19, pp. 287–296, 1985.
- [30] J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, J. T. Folsom-

Kovarik, and K. O. Stanley, “Picbreeder: a case study in collaborative evolutionary exploration of design space,” *Evol. Comput.*, vol. 19, no. 3, pp. 373–403, 2011.

[31] R. Smelik, K. Galka, K. J. de Kraker, F. Kuijper, and R. Bidarra, “Semantic constraints for procedural generation of virtual worlds,” *Proc. 2nd Int. Work. Proced. Content Gener. Games - PCGames ’11*, pp. 1–4, 2011.

[32] J. Dormans, “Adventures in level design: generating missions and spaces for action adventure games,” ... *Work. Proced. Content Gener. Games*, pp. 1–8, 2010.

[33] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool, “Procedural modeling of buildings,” *ACM Trans. Graph.*, vol. 25, no. 3, p. 614, 2006.

[34] J. Togelius, E. Kastbjerg, D. Schedl, and G. N. Yannakakis, “What is Procedural Content Generation? Mario on the borderline,” *Proc. 2nd Int. Work. Proced. Content Gener. Games*, p. 6, 2011.

[35] A. Swim, “Robot Unicorn Attack.”

[36] “Rogue Basin.” [Online]. Available: http://www.roguebasin.com/index.php?title=Main_Page.

[37] M. Persson, “Infinite Mario Bros!” 2008.

[38] T. Sonic, “Sonic the Hedgehog.” 1991.

[39] Valve, “Portal 2.” 2011.

[40] G. Smith, A. Othenin-Girard, J. Whitehead, and N. Wardrip-Fruin, “PCG-based Game Design: Creating Endless Web,” *Proc. Int. Conf. Found. Digit. Games*, no. c, pp. 188–195, 2012.

[41] “No Man’s Sky.”

[42] 2KGames, “Borderlands.” 2009.

[43] Nintendo, “Animal Crossing: New Lead.” 2013.

[44] Blizzard, “Diablo II.” 2000.

[45] Level-5, “Professor Layton and the Curious Village.” 2008.

[46] Jupiter, “Picross.” 2007.

[47] Mortac, “The Complete Guide to Map Generation.” [Online]. Available: <http://forums.civfanatics.com/showthread.php?t=246788&s=b667c50191bb40c93402f38a7560b7ac>.

[48] R. Hunicke, M. LeBlanc, and R. Zubeck, “MDA: A Formal Approach to Game Design and Game Research,” *Work. Challenges Game AI*, pp. 1–4, 2004.

[49] L. Pereira and L. Roque, “Towards a game experience design model centered on participation,” in *CHI ’12 Extended Abstracts on Human Factors in Computing Systems*, 2012, pp. 2327–2332.

[50] A. Leitão, “Vasco da Gama,” 2006. [Online]. Available: <http://cvc.instituto->

camoes.pt/navegaport/d23.html.

- [51] D. N. G. J. P. da Rocha, “Pedro Álvares Cabral,” 2002. [Online]. Available: <http://cvc.instituto-camoes.pt/navegaport/d01.html>.
- [52] MicroProse, “Sid Meier’s Pirates!” MicroProse, Hunt Valley, 1987.
- [53] O. Softscape, “The Seven Cities of Gold.” Electronic Arts, 1984.
- [54] P. Interactive, “Europa Universalis IV.” Paradox Interactive, 2013.
- [55] Atlus, “Etrian Odyssey.” Atlus, Tokyo, 2007.
- [56] J. “Moebius” Giraud, Arzarch. Les Humanoïdes Associés, 1975.
- [57] J. “Moebius” Giraud and A. Jodorowsky, *L’Incal*. Les Humanoïdes Associés, 1981.
- [58] D. Simmons, Hyperion. Doubleday, 1989.
- [59] F. Herbert, *Dune*. Chilton Books, 1965.
- [60] SquareSoft, “Final Fantasy Tactics.” 1997.
- [61] L. V. de Camões, *Os Lusíadas*. 1572.
- [62] U. Technologies, “Unity 3D.” 2005.
- [63] S. E. Inc, “Stack Overflow,” 2008. [Online]. Available: <http://stackoverflow.com/>.
- [64] I. C. Technologies and R. Quesada, “Cocos2d-x.” 2010.
- [65] P. S. Ltd., “Phaser.” 2013.
- [66] E. Games, “Unreal Engine.” 1998.
- [67] A. Systems, “Adobe Photoshop CC.” 2015.
- [68] Many Authors (Open Source), “Audacity.” 2000.
- [69] S. Lague, “Procedural Landmass Generation,” 2016. [Online]. Available: <https://github.com/SebLague/Procedural-Landmass-Generation>.
- [70] Codemasters, “Micro Machines.” 1991.
- [71] L. Johnson, G. N. Yannakakis, and J. Togelius, “Cellular automata for real-time generation of infinite cave levels,” *Proc. 2010 Work. Proced. Content Gener. Games - PCGames ’10*, pp. 1–4, 2010.
- [72] T. Fullerton, “Playtesting,” in *Game Design Workshop: A Playcentric Approach To Creating Innovative Games*, Morgan Kaufmann, 2008, pp. 248 – 276.

