



Guia de desenvolvimento de jogos para programadores independentes

LUÍS MIGUEL DIAS FERNANDES

Outubro de 2015

Guia de desenvolvimento de jogos para programadores independentes

Luís Miguel Dias Fernandes

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Gráficos e Multimédia**

Orientador: Filipe de Faria Pacheco, PhD

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, Outubro 2015

Resumo

Os vídeo jogos ou jogos de computador têm vindo a crescer na sua relação com o público ganhando terreno e credibilidade nos benefícios que os jogadores obtêm quando usam este tipo de software, indo esses benefícios para além do divertimento associado à palavra jogo no seu sentido etimológico. Nos últimos anos cada vez mais o mercado e algumas das mais reputadas instituições de ensino têm dedicado especial atenção a este tipo de software, englobando nesses estudos diversas áreas desde a engenharia à saúde, incluindo ainda, estudos de cariz psicológico e sociológico reveladores de que estas experiências de entretenimento, cada vez mais disponíveis a todos, têm influência na sua envolvente de integração e relação com outros fenómenos de cariz social.

Apreciado o estado da arte esta tese tem como principal objetivo servir de guia de iniciação a individuais ou pequenas equipas da área de desenvolvimento de software no caminho para o desenvolvimento de jogos de vídeo independentes, apresentando uma análise cuidada capaz de apoiar as equipas desde o momento zero, estando o mesmo estruturado de forma a refletir o entendimento das bases teóricas em que o desenvolvimento deste tipo de software assenta, o estado da arte sobre plataformas, análise de mercado e indústria, metodologias de desenvolvimento e equipas, e ainda analisadas algumas das mais relevantes ferramentas de desenvolvimento e criação de conteúdos.

Como prova de conceito e componente de caráter experimental, este trabalho comprehende ainda o desenvolvimento de um jogo de vídeo guiado pelas orientações e lições apreendidas durante o processo de estudo de forma a ser representativo da aplicação dessa aprendizagem, e ainda capaz de detalhar passo a passo cada fase do processo, com o objetivo de apoiar a preparação de pessoas interessadas em iniciar a aventura de desenvolver os seus próprios jogos e quem sabe dar início aos seus próprios negócios e empresas. O jogo desenvolvido replica um recente êxito, o jogo 2048, que apesar da sua mecânica simples revela-se um excelente desafio do ponto de vista da sua implementação, bem como o processo criativo adjacente no transformar algo tão simples e experimentado numa experiência nova capaz de atrair antigos jogadores e interesse de novos.

O desenvolvimento e publicação da aplicação experimental, e após análise dos dados recolhidos, mostram-se reveladores de que é de fato possível a programadores independentes entrar num mercado de alto potencial, e que adquiridas as bases de conhecimento expostas no documento estarão em condições mais favoráveis para ultrapassar algumas das mais comuns barreiras no atingir desse objetivo.

Palavras-chave: Jogos, plataformas, design gráfico, programação, independente, desenvolvimento software, 2048.

Abstract

Video games and computer games have been growing in their relationship with the public, gaining ground and credibility in the benefits that players get when using this type of software, reaching these benefits way beyond the fun generally associated with the word game in its etymological meaning. In recent years, the market and some of the most reputable educational institutions have devoted special attention to this type of software, encompassing such diverse areas from engineering to health, including further studies of psychological and sociological nature revealing that these entertainment experiences, increasingly available to everyone, have an influence on its surrounding integration and relationship with other phenomena of social nature.

Appreciated the state of the art this thesis mainly intend to serve as a beginner's guide to individual or small software development teams exposing a way to the develop independent video games, with a detailed analysis, the information provided should be enough to serve as support for development teams from the first moment, being structured to reflect the understanding of theoretical basis for development of such software, the state of the art relating to platforms, market analysis and industry, development of methodologies, and also analyzed some of the most relevant tools for development and content creation.

As proof of concept and experimental component, this work also includes the development of a video game guided by the lessons learned during the process of study, being representative of the application of acquired information, still being able to drill down step by step each stage of the process, in order to support the preparation of people interested in starting the adventure of developing their own games and who knows how to start their own businesses and companies. The game chosen to develop had a model of recent success, the game 2048, that even though its simple mechanics, proves to be an interesting challenge from the point of view of its implementation, as well the creative process in turning something as simple and experimented in a whole new experience able to attract old players and gain the interest of new ones.

The development and publication of the experimental game, and after analyzing the collected data, showed up that it is indeed possible for independent developers to get into a high-potential market, and acquired the exposed knowledge presented in this document, they will be on more favorable terms to overcome some of the most common barriers to achieving this goal.

Keywords: Games, distribution platforms, graphic design, independent, programming, software development, 2048.

Agradecimentos

Quero antes de mais deixar uma especial palavra de apreço aos envolvidos neste projeto, em particular ao meu orientador o Professor Filipe de Faria Pacheco por todo o tempo que me dispendeu e pelo seu precioso contributo para que este projeto fosse possível. Sendo esta a etapa final de um percurso académico fundamental para o meu actual posicionamento a nível profissional, aproveito ainda para registar uma palavra de agradecimento a todos os que tornam este curso possível com alto nível de exigência e qualidade.

Agradeço ainda à minha família por acreditar e apoiar as minhas decisões, mesmo quando não parecem ser as mais acertadas, em especial à minha esposa Joana por toda a paciência e tempo apostado num futuro comum melhor, e que irá ser preenchido brevemente com uma nova vida, sendo que é este projeto também dedicado a ele(a).

A todos, um sincero e sentido muito obrigado!

Índice

1.	Introdução	1
1.1	Objetivos e motivação.....	2
1.2	Calendarização	4
1.3	Estrutura do documento	5
2.	Estado da arte	6
2.1	O lado sério dos jogos	7
2.1.1	O que são os jogos?	7
2.1.2	Jogos que alimentam	8
2.1.3	Os jogos e a sociedade	10
2.1.4	Jogos Sérios	12
2.2	Independencia versus a Indústria.....	13
2.2.1	Então queres ser um programador independente	13
2.2.2	A importância dos jogos independentes	14
2.2.3	Plataformas e indústria	15
2.2.4	Quem irá ganhar a corrida da indústria?	19
2.2.5	Então que caminho seguir?	20
2.2.6	Financiamento, distribuição e monetização.....	22
2.3	Desenvolvimento de jogos	24
2.3.1	Metodologias de desenvolvimento e equipas	24
2.3.2	Ferramentas de desenvolvimento	26
2.3.3	Conclusão de análise das ferramentas	38
2.3.4	Media nos jogos e criação de conteúdos	39
2.3.5	Gráficos	40
2.3.6	Som	43
2.3.7	2D versus 3D.....	45
2.3.8	2.5D, um novo estilo de jogo	46
2.4	O desenvolvimento de jogos em Portugal.....	47
3.	Caso experimental	49
3.1	Relação entre estado da arte e caso experimental	49
3.2	<i>Brainstorming</i> , da ideia ao conceito	51
3.3	Design de jogo	52
3.3.1	O jogo: 2048.....	52
3.4	Visão global.....	54
3.4.1	Esquema de navegação.....	54
3.4.2	Requisitos funcionais	55
3.4.3	Requisitos não funcionais	56
3.5	Implementação	56
3.5.1	Gestão do projeto	57
3.5.2	Programação da componente lógica	58

3.5.3	Produção de interfaces gráficos	59
3.5.4	Disponibilização na Google Play Store e na Internet.....	67
4.	Testes e análise de dados.....	69
4.1	Teste Alfa	69
4.2	Teste Beta	72
4.3	Dados de lançamento final	73
5.	Conclusões e apreciações finais	77
5.1	Conclusões sobre o produto criado	77
5.2	Conclusões finais	79
6.	Anexos.....	88

Lista de Figuras

Figura 1 – Estado do mercado de distribuição de jogos (<i>a</i>).....	17
Figura 2 – <i>Share</i> de mercado de sistemas operativos móveis	18
Figura 3 – Decisão sobre desenvolvimento de jogos.....	21
Figura 4 – Fluxo de <i>Minimun Viable Product</i>	Error! Bookmark not defined.
Figura 5 – Visão principal (Unity)	28
Figura 6 – Arquitetura de Cocos 2D-x	30
Figura 7 – Arquitetura de Cocos 2D-JS.....	31
Figura 8 – Exemplo de navegação entre cenas (Cocos2D).....	32
Figura 9 – Sentido de pesquisa do grafo de objetos de cena (Cocos 2D)	33
Figura 10 – Exemplo de <i>sprites</i>	33
Figura 11 – Sequência de acções	34
Figura 12 – Visão geral de Cocos Studio	35
Figura 13 – Game Closure, diagrama de comunicação entre componentes	36
Figura 14 – Gimp	41
Figura 15 – Inkscape, painel principal	42
Figura 16 – Painel de início da aplicação Blender.....	43
Figura 17 – Audacity.....	44
Figura 18 – Visão painel principal da aplicação AudioSauna	45
Figura 19 – Avaliação do jogo 2048 na <i>App Store</i> da Apple	52
Figura 20 – Ecrã do jogo <i>Threes</i> (à esquerda) e 2048(à direita)	53
Figura 21 – Ilustração de regras do jogo 2048	53
Figura 22 – Visão de esquema de navegação global.....	55
Figura 23 – Esboços gráficos de ecrãs de jogo Sticky 2048.....	55
Figura 24 – Fluxo de desenvolvimento e responsáveis por tarefas	57
Figura 25 – Diagrama de classes da componente de lógica do jogo	58
Figura 26 – Esboço de cena da tela inicial anotado	60
Figura 27 – Esboço de personagem (Photoshop)	60
Figura 28 – Componentes estáticos do Goo do ecrã inicial	60
Figura 29 – Animação de personagem em Photoshop	61
Figura 30 – <i>Sprites</i> do Goo do ecrã inicial	61
Figura 31 – Várias imagens da cena principal	61
Figura 32 – Esboços de ecrãs de jogo	62
Figura 33 – Cenas de inicio de jogo e ajuda	63
Figura 34 – Cenas de jogo (dica e desbloqueio de nova personagem).....	63
Figura 35 – <i>Sprites</i> de animações de fusão de personagens	64
Figura 36 – Imagens de <i>feedback</i> durante jogo	64
Figura 37 – Ecrã de vitória.....	64
Figura 38 – Mensagem de fim de jogo.....	65
Figura 39 – Listagem de requisitos Google Play Store(1).....	67
Figura 40 - Listagem de requisitos Google Play Store(2)	68

Figura 41 – Jogo publicado na loja	Error! Bookmark not defined.
Figura 42 – Ecrãs da versão de fase alfa.....	70
Figura 43 – Lista de dispositivos envolvidos no teste alfa.....	71
Figura 44 – Publicação no fórum <i>AndroidDevFrotal</i> para promoção do jogo em fase Beta	73
Figura 45 – Resumo de visitantes da loja Google Play (Agosto).....	74
Figura 46 – <i>Ratings and Reviews</i> loja <i>Google Play Store</i>	74
Figura 47 – Comentário de jogador retirado da <i>Google Play Store</i>	75
Figura 48 – Relatório de problemas da aplicação	75
Figura 49 – Visão de instalações da aplicação ao longo do tempo	76
Figura 50 - Tela principal do jogo Sticky 2048.....	77

Lista de Tabelas

Tabela 1 – Personagens e representação númerica	62
---	----

Lista de excertos de Código

Código 1 – <i>Title.js</i>	59
Código 2 – Construtor da cena <i>Titlescreen.js</i>	65
Código 3 – Exemplo de criação de logotipo na cena principal.....	66
Código 4 – Exemplo de registo de função de gestão de interação com utilizador.....	66
Código 5 – Gestor de som da aplicação	66

Acrónimos e Símbolos

Listas de Acrónimos

HTML5	Hyper Text Markup Language versão 5
ESA	Entertainment Software Association
IAP	In App Purchases
RPM	Revenue per Thousand views
DAU	Daily Active User
MAU	Monthly Active User
ARPDAU	Average Revenue Per Daily Active User
XML	Extensible Markup Language
APROJE	Associação de Produtores de Jogos Electrónicos
HDR	High Definition Resolution

1. Introdução

É objeto desta dissertação a exposição do trabalho desenvolvido no âmbito da tese de mestrado em engenharia informática no ramo de sistemas gráficos e multimédia, projeto este focado no estudo do desenvolvimento de jogos de vídeo. Ao longo deste documento são apresentadas todas as análises teóricas e desenvolvimentos realizados na prossecução da criação de um produto de software passível de ser produzido por pequenas equipas ou individuais que seguem a sua paixão, na já hoje por muitos reconhecida arte de desenvolvimento de jogos, utilizando para isso conhecimentos teóricos de base relacionados com jogos, software, mercado, metodologias e ferramentas de desenvolvimento adequadas ao contexto de programadores individuais ou de pequenas equipas independentes.

Os jogos de vídeo ou computador são hoje, para um conjunto significativo da população, parte do dia-a-dia e representativos de um mercado em franco crescimento (Grubb, 2015) e revelam em si uma complexidade que à partida não lhe é associada e que afeta a sua credibilidade enquanto produto de software de retorno financeiro ou social, tema atualmente motivo de debate um pouco por todo o mundo (ESA, 2015). Este trabalho pretende em concreto abordar de forma realista e objetiva o processo de desenvolvimento deste tipo de software dando especial foco às motivações, métodos e processos de desenvolvimento disponíveis da atualidade, tendo como principal público-alvo programadores experientes que nunca tendo tido contacto profissional no desenvolvimento deste género de software demonstram particular interesse e motivação para no tema.

O documento e caso experimental exposto não têm de forma alguma, qualquer pretensão de que o apresentado seja um caminho exclusivo para a execução deste tipo de projetos, mas sim o objetivo de ser um caminho viável capaz de suster as necessidades mais básicas do desenvolvimento destes produtos de software, orientando potenciais decisões de interessados no tema, de forma a clarificar alguns dos tópicos mais comuns, em particular decisões de cariz mais técnico sobre ferramentas de produção, quer de código quer de conteúdos, e ainda, mesmo que de forma menos aprofundada dado o âmbito e contexto do projeto, orientações sobre mercado, plataformas e aspetos relacionados com gestão de projeto e de negócio. É ainda de referir em particular que as ferramentas de produção

avaliadas e analisadas no documento foram selecionadas, do grande leque de opções disponíveis, tendo em conta questões de popularidade das mesmas, disponibilidade, custo e adequabilidade à anterior experiência profissional dos autores do projeto de forma a viabilizar a produção efetiva de um produto passível de ser publicado em tempo útil neste âmbito.

1.1 Objetivos e motivação

Os jogos são importantes ferramentas de aprendizagem, e mesmo que não implicitamente assumida esta relação, avanços recentes têm vindo a reforçar, através de estudos laboratoriais (Primack, 2012), que os efeitos neurológicos deste tipo de atividades se mostram benéficos para o aumento da qualidade de vida das pessoas, tendo sido apuradas melhorias das capacidades neurológicas e até físicas dos intervenientes. Jogos de mecânica simples como Sudoku, ou até jogos mais complexos como jogos de ação em primeira pessoa têm mostrado que os seus jogadores veem melhorias não desprezáveis das suas capacidades cognitivas mantendo e aumentando a capacidade de resposta a desafios de raciocínio do dia-a-dia (McGonigal, 2011). À luz do referido, parece imprescindível que a formação específica na área do desenvolvimento de jogos de computador, e dada a alta nível disponibilidade dos dispositivos de distribuição destas experiências, seja levada a sério.

Objetivamente o desenvolvimento deste projeto pretende servir como base de lançamento de conceitos do mundo da criação, desenvolvimento e distribuição de vídeo jogos. São especificamente metas a atingir:

- Realização de um estudo aprofundado sobre o que são os jogos e quais as implicações do seu uso a nível social e individual bem como as bases etimológicas do termo;
- Explorar o conceito de programadores independentes de jogos;
- Identificar sucintamente um conjunto de características relevantes para programadores independentes sobre plataformas, indústria, distribuição, financiamento e monetização;
- Selecionar e analisar diferentes ferramentas de desenvolvimento e produção de conteúdos apresentando uma conclusão comparativa e objetiva sobre as mesmas. Não sendo possível apresentar todas as ferramentas disponíveis no mercado a seleção das ferramentas deve seguir uma lógica de diversidade representando de forma abstrata modelos de ferramentas disponíveis;
- Identificar estado do mercado dos vídeo jogos a nível nacional;
- Desenvolver uma solução que demonstre o processo de desenvolvimento de um jogo desde o momento zero;

- Publicar aplicação e recolher dados para análise métrica de forma a apresentar conjunto de conclusões assente em dados reais recolhidos à data.

Para além do seu cariz académico, a escolha do tema desta tese é também pessoal, sendo que o autor enquanto profissional na área de desenvolvimento de software corporativo, mantém uma aberta paixão pelo desenvolvimento de vídeo jogos, primariamente como forma de exposição criativa e exploração de tecnologias onde o desenvolvimento deste tipo de produtos assenta.

A integração de componentes multimédia que compõem as experiências, bem como o processo de gestão dos projetos e algumas noções de mercado e negócio apresentados neste documento são particularmente direcionados para engenheiros de software experientes na área da programação, motivados pelo interesse de trabalhar na área de desenvolvimento de jogos mantendo a pretensão de sensibilizar profissionais que, não descurando o fato indubitável de que se trata de desenvolvimento de software, este tipo de produto comprehende muito mais do que isso, sendo o seu desenvolvimento altamente interdisciplinar.

Finalmente é também fundamental enquanto fator motivacional, o fator financeiro, tendo sido observado que o mercado neste tipo de software representa hoje uma fatia significativa do mercado do entretenimento, atingindo valores de milhares de milhões de euros e o qual prevê um franco crescimento nos próximos anos. Não descurando o aumento de concorrência e luta por cota deste mercado, ganha especial importância a capacidade dos produtores deste tipo de software em otimizar a utilização dos recursos a seu dispor de forma a aumentarem as suas probabilidades de serem bem-sucedidos.

Apresentadas as motivações acima não será irrelevante partilhar que enquanto estudante da área multimédia e sistemas gráficos, e profissional experiente da área de software, a realidade é que não consegui ainda vislumbrar no nosso país um mercado de desenvolvimento de jogos significativo, partilhando que o sonho a nível pessoal é altamente motivador, assumindo ainda alguma presunção pessoal, que a médio/longo prazo o conhecimento exposto neste documento, em conjunto com os conhecimentos adquiridos nos planos de estudos disponibilizados pelas nossas entidades de formação superior, possam ajudar a que esta realidade mude e que possamos utilizar os valiosos recursos ao nosso alcance, na exploração de um mercado, capaz de criar empregos altamente qualificados, à imagem do que já acontece em grande parte dos países mais avançados do mundo onde esse fator foi relevante para a aposta em apoios governamentais e privados.

1.2 Calendarização

Para realização deste projeto foi realizado um plano de execução de forma a manter conformidade e cumprimento de prazos do mesmo não colocando em causa prazos de entrega e metas atingir. Na tabela abaixo são apresentadas as tarefas num planeamento de alto nível.

Data	Descrição das tarefas
Janeiro de 2015 a Abril de 2015	<ul style="list-style-type: none">- Investigação sobre o tema- Levantamento do estado da arte- Experimentação de ferramentas de desenvolvimento de jogos- Experimentação de ferramentas de criação de conteúdos- Elaboração da dissertação
Abril de 2015 a Junho de 2015	<ul style="list-style-type: none">- Análise de metodologias de desenvolvimento e tomada de decisão sobre a metodologia a seguir- Ideia e primeira implementação de protótipo rápido- Avaliação e revisão do protótipo e realização do plano técnico de execução do projeto- Elaboração da dissertação
Junho de 2015 a Agosto de 2015	<ul style="list-style-type: none">- Criação de conteúdos (ficheiros de áudio, imagem e vídeo)- Implementação do jogo de prova de conceito- Realização de teste Alfa- Realização de teste Beta- Finalização do jogo
Agosto de 2015 a Setembro de 2015	<ul style="list-style-type: none">- Publicação do produto- Revisão, acompanhamento de dados de aceitação do jogo- Elaboração da dissertação
Setembro a Outubro de 2015	<ul style="list-style-type: none">- Revisão da dissertação- Conclusão do documento final e preparação da apresentação

1.3 Estrutura do documento

Esta dissertação está dividida em **cinco** capítulos principais sendo o **primeiro capítulo** referente ao enquadramento do tema em análise, apontamento dos principais objetivos da dissertação e apresentado o respetivo planeamento.

No **segundo capítulo** está concentrada a informação relativa à investigação realizada relacionada com a parte mais teórica de análise do que são jogos, análise de mercado sobre plataformas e tecnologias, métodos e desenvolvimento. A estrutura deste capítulo comprehende a apresentação de conteúdos considerados da maior relevância e apresentados numa perspetiva incremental sendo explorados inicialmente conceitos mais teóricos relacionados com vídeo jogos e o seu design. É também abordada, ainda que de forma sucinta dado o âmbito deste projeto, fatores comerciais e de mercado, seguidos de um estudo sobre metodologias de desenvolvimento e ainda analisando e comparando um conjunto de ferramentas de produção capazes de sustentar a realização deste tipo de projetos, concluindo este capítulo com a análise do estado do desenvolvimento de jogos a nível nacional.

O **terceiro capítulo** considerado o guia para execução de um projeto de software de jogos desde a sua génesis até ao produto final. Este capítulo comprehende desde a ideia à definição da arquitetura da solução e ao responder de questões sobre decisões tomadas face a plataformas e público-alvo, justificando essas decisões com parâmetros decididos pelo autor e assentes em escolhas e crenças pessoais.

No **capítulo quarto** são apresentados dados e resultados das fases de teste realizadas no projeto e o resumo analítico de dados das respetivas fases de teste e fase de acompanhamento da aplicação quando em produção.

No **quinto capítulo** são apresentadas as conclusões obtidas com a dissertação bem como um artigo de opinião sobre todo o processo e avaliação do autor sobre o tema.

2. Estado da arte

Os vídeo jogos são já uma parte inevitável das nossas vidas ganhando cada vez mais terreno económica e socialmente ocupando já hoje parte não desprezável do nosso tempo e representando uma indústria que movimenta milhões por todo mundo que não apresenta sinais de abrandar (Grubb, 2015). Como referido por alguns atores importantes desta área, a parte mais complicada do desenvolvimento de jogos é a engenharia do software e respetiva dificuldade na gestão de projetos dada a sua complexidade e aparente perpétua falta de pessoas qualificadas para realizar os trabalhos inerentes à função de programação de jogos (Blow, 2004). O tema em si é difícil de abordar dadas todas as competências envolvidas e potenciais abordagens na análise das diferentes dimensões. Este estudo, dado o seu âmbito, foca os seus esforços num perspetiva mais prática de cariz tecnológica e de gestão de projeto estando orientado para pequenas equipas limitadas em recursos e livres de amarras criativas.

Este capítulo está organizado de forma a que as várias dimensões relevantes quando se parte na aventura de desenvolvimento de um vídeo jogo sejam abordadas de forma incremental e no sentido da criação de jogos independentes. O desenvolvimento de jogos é uma atividade interdisciplinar que requer metodologia e rigor sendo esse o maior desafio na sua produção e nem sempre essas questões são abordadas da melhor forma resultando muitas vezes em projetos que não cumprem o seu propósito (Blow, 2004).

Neste capítulo podemos encontrar tópicos relevantes na prossecução do desenvolvimento de jogos por individuais ou pequenas equipas com objetivo de desenvolvimento de projetos com qualidade que possam ser terminados de forma a ver otimizados os recursos ao seu dispor. A sua organização foi pensada de modo a que exista evolução e ligação de conhecimentos ao avançar na leitura devendo a mesma seguir a ordem apresentada, respetivamente:

- O que são jogos? - qual o sentido etimológico e propósito clássico dos jogos;

- Jogos que alimentam - a criatividade por trás dos jogos, característica dos jogos, teorias do divertimento e efeitos psicológicos nos jogadores;
- Os jogos e a sociedade - os jogos e a sua influência no meio;
- Jogos Sérios - o que são? Principais característica e estado da arte;
- Indie Developers - O que significa ser programador independente de jogos;
- A importância dos programadores de jogos independentes - influências dos independentes na indústria e nos movimentos criativos do mundo dos jogos;
- Plataformas e Indústria - descrição de plataformas e análise de alto nível do estado da indústria;
- Financiamento, distribuição e monetização - negócios, esquemas de monetização, plataformas de distribuição e obtenção de financiamento;
- Metodologias de desenvolvimento - análise de metodologias de desenvolvimento de produto mais relevantes para construção de jogos;
- Ferramentas de desenvolvimento - análise comparativa de ferramentas de desenvolvimento do software do jogo;
- Media e criação de conteúdos - lista de ferramentas de criação de conteúdos de media áudio e imagens geralmente representadas em jogos;
- 2D VS 3D - Diferenças e adequabilidade de modos de representação de jogos para programadores independentes;
- O estado da indústria em Portugal - sucinta análise da história dos vídeo jogos a nível nacional;

2.1 O lado sério dos jogos

2.1.1 O que são os jogos?

Inevitavelmente a primeira questão que se coloca será afinal “O que é um jogo?”

A Teoria de Jogos é uma das áreas científicas que estuda este tema, aprofundando o estudo dos jogos sobre perspetiva de atividades psicológicas mesuráveis e de certa forma passíveis de serem controladas e previsíveis, resultando estes estudos em teorias de como potenciais rivais devem tomar as suas decisões, otimizando as probabilidades de sucesso em cenários políticos e de mercado, que em nada nos ajuda a estruturar uma definição para jogo no propósito procurado de atividade de caráter lúdico e intrinsecamente educativo (Turocy & Stengel , 2001). Quando pesquisado pelo termo jogo rapidamente é nos apresentado um conjunto de tópicos formais e assertivos que descrevem jogo como um atividade competitiva ou desportiva que de acordo com conjunto de regras que promove divertimento.

Com a introdução do computador enquanto ferramenta de transição de media, os jogos de computador vieram reacender o interesse no estudo do termo jogo, e numa era em que os dispositivos que detemos são mais do que apenas transmissores de dígitos e cores frias estes tornaram-se valiosas ferramentas de transmissão da mais poderosa linguagem, a das imagens

e do som dando origem a novas definições da palavra jogo orientando esta definição procurada para jogo de computador (Crawford, 1982), sendo este o termo que nos interessa em contexto.

Ainda que existam várias definições notáveis e quase irrefutáveis do termo jogo, todas compreendem alguns pontos comuns, em que mais interessante do que a definição em si é o potencial de um jogo de computador no manter com os seus intervenientes uma relação neurológica de transmissão de emoções que acabou por originar uma nova área de estudos designada por design de jogos, área esta que estuda e tenta manter atualizado um conjunto de técnicas e fatores a serem consideradas quando envolvidos no processo de criação de um jogo . Em conclusão não parece existir sentido na procura e apontar de uma definição atómica de jogo, sendo sim importante para quem desenvolve jogos manter a consideração de duas propriedades, respetivamente, desafio e divertimento, propriedades estas fundamentais quando se desenha uma experiência de jogo de forma evitar que o potencial jogador entre num estado psicológico de aborrecimento e o afaste da experiência que pretendemos transmitir (Koster, 2013).

2.1.2 Jogos que alimentam

Os jogos são como experiências emocionais que alimentam os nossos cérebros e a criação de jogos prende-se com a criação de padrões que se revelem desafios para os jogadores, assentando no pressuposto de que os nossos cérebros são orientados pela necessidade constante de reconhecimento de padrões e associação desses padrões através de processos cognitivos, evitando situações definidas como ruído face à incapacidade de identificação de padrões e assim criando nos jogadores sentimento de frustração levando mesmo à desistência do desafio. Fundamentalmente o sucesso de um jogo a capacidade dos designers dos jogos, em gerir as emoções dos jogadores de forma a que estes se mantenham num estado de inquisição e curiosidade constante, regida pela evolução do percurso do jogo e resultando isto num sentimento de diversão, aspeto fundamental para a manutenção de interesse dos jogadores (Koster, 2013).

Chris Crawford identificou então diversão como a resposta emocional ao processo cognitivo de aprendizagem, reforçando não só a importância dos jogos no contexto atual, de uma geração que nasceu já na era do digital, habituada a experiências multimédia ricas ao invés dos tradicionais sistemas de educação de escrita e leitura, e que em comparação surgem como experiências pouco apelativas e aborrecidas.

Platão: "A educação é efetivada quando uma criança pode jogar com coisas agradáveis"

Bruno Bettelheim: "Jogar é a mais útil ferramenta de preparação de uma criança para o futuro e os seus desafios"

Albert Einstein: "Jogar é a mais avançada forma de pesquisa"

Penelope Leach: "Para uma criança não existe divisão entre jogar e aprender"

Esta ideia de que os jogos são poderosas ferramentas de aprendizagem não é uma ideia recente mas é assumido, que com a massificação do acesso a experiências multimédia através de dispositivos modernos como consolas, computadores, smartphones, tablets, smart Tv's e outros, não é possível ser ignorada a potencialidade dessas plataformas enquanto ferramentas de ensino (ESA, 2014). É então clara a responsabilidade de quem faz jogos criar experiências que permitam ao utilizador aproveitar esta capacidade excepcional de ensino e não esquecer que, quando ao desenhar um potencial jogo, esses são puzzles sobre emoções e processos cognitivos de aprendizagem e de análise de padrões, e que sempre que, enquanto não dominados, resultam em diversão ou frustração, o qual termina quando o nosso cérebro entende que não existe mais desafio adjacente ao processo de aprendizagem e prática no domínio de tal atividade, levando a que fique num estado de aborrecimento e que leva o interveniente a abandonar o interesse na experiência e no processo de aprendizagem (Koster, 2013). Com isto podemos então concluir, que mesmo que os diferentes níveis e relevância, todos os jogos são educacionais resultando muitas vezes em processos de prática que nos levam a experimentar de forma mais casual situações reais de forma a preparamo-nos caso essas situações efetivamente surjam.

Se observarmos com atenção esses processos de prática ocorrem não só nos humanos mas em quase todas as espécies enquanto jovens. A título de exemplo não é raro vermos uma luta entre dois felinos pequenos enquanto debaixo do olhar da mãe atenta. De fato podemos imediatamente identificar que se trata de uma 'brincadeira', de um jogo, mas sabemos também que essas brincadeiras são extremamente importantes, representando uma prática de auto conhecimento relativamente avaliação de forças, fraquezas e pontos a melhorar nas suas técnicas de combate e caça que lhes deverão no futuro revelar serem fundamentais para sua subsistência. Nós enquanto seres diferenciados pela nossa maior capacidade cognitiva e de aprendizagem e de memória, não podemos ignorar a importância que as brincadeiras e o divertimento na prática de atividades de simulação têm nas nossas vidas (Saunders, 1996).

Melhoramento de posicionamento espacial, estimulação da vontade de explorar novas temáticas, capacidade reação e ação ao movimento e raciocínio lógico são algumas das melhorias identificadas sobre capacidades de indivíduos envolvidos em estudos neurológicos relativamente às atividades dos seus cérebros durante o processo de jogar vídeo jogos (ESA, 2014).

Se observamos o nosso próprio comportamento quando jogamos conseguimos fazer a rápida leitura que, independentemente do objetivo ou do jogo em si, queremos atingir um objetivo e que esse objetivo está presente, tem interesse, e o nível de desafio é adequado às nossas capacidades, caso contrário a experiência é considerada enfadonha e rapidamente abandonaremos a atividade. Comportada a leitura acima dos jogos enquanto atividades de prática, considero que a atitude de um individuo presente a um desafio num jogo será muito relevante para entender como o mesmo individuo iria comportar-se perante um desafio real similar, refletindo-se na prática da aplicação de pequenos jogos, sendo esta técnica utilizada há vários anos, como ferramenta de avaliação psicológica e comportamental (Saunders, 1996).

Quando o jogador é deparado com um desafio cujo nível se revela ser adequado e com um objetivo claro e relevante para si, o mesmo tende a ficar ‘obcecado’ pelo atingir desse objetivo, procurando muitas vezes não só a solução mas também o método mais otimizado de a atingir, só ficando satisfeito quando o caminho de sucesso for claro e o mais previsível possível. Este comportamento é em si também bastante representativo das nossas atitudes perante a vida, sendo isto normalmente representado como sentimento de segurança e familiaridade, a título de exemplo, o ato de curiosidade na prova de novos alimentos e até novas formas de os cozinar, sendo que mais de metade das vezes comemos o que gostamos da mesma forma vezes sem conta porque ‘gostamos’, porque é seguro, diferindo neste exemplo o fator necessidade, fator este que não existe geralmente num jogo resultando no abandono da atividade a não ser que o design do jogo seja em si algo mutável que evolua ao ritmo do jogador mantendo o interesse e mesmo que não sendo uma necessidade, o faça voltar vezes sem conta, mantendo a comparação com o processo de alimentação podemos comparar os jogos a aquilo que geralmente denominamos por guloseimas. Não necessitamos de as consumir mas o prazer que obtemos a faze-lo, ‘obriga-nos’ a repetir esse ato vezes sem conta (Koster, 2013).

Como atividade não exclusiva dos seres humanos, a prática de jogos é como referido, uma prática importante para o nosso desenvolvimento físico e cognitivo devendo estas serem regidas de forma saudável e equilibrada.

2.1.3 Os jogos e a sociedade

Mesmo que algo natural a todos os jogos são sempre um tema sujeito a múltiplas leituras e opiniões o que comporta um quase eterno debate sobre o que são e qual a sua relevância. Se perguntarmos a um qualquer grupo o que é um jogo muito provavelmente todos terão uma resposta e todas terão pontos coincidentes e diferentes criando quase de imediato motivo para debate e discussão de opiniões. Em relação aos jogos de computador mais concretamente apesar de todas as análises, pesquisas e estudos realizados em algumas das mais importantes instituições de ensino superior mundiais, continuam a ser olhados como algo que levanta as mais diversas contradições, bem como as próprias profissões orientadas para o desenvolvimento deste tipo de produto. Quando a resposta a “Então o que fazes da vida?” é “- desenvolvo jogos” é quase inevitável o olhar de suspeição, não sendo de todo visto sequer como uma profissão para a maioria das pessoas no nosso país, e que não difere de forma alguma de alguns países onde alguns dos mais bem pagos profissionais são programadores de jogos, como por exemplo os Estados Unidos (Warrington, 2015). Esta será em si a primeira barreira com que a maioria dos aspirantes a criadores de jogos se preparam já que o apoio para que um individuo se sinta motivado para seguir uma carreira profissional têm como principal fonte de apoio a família, amigos e invariavelmente da sociedade e mundo económico onde se enquadra. Seria impensável hoje alguém querer seguir a profissão de, por exemplo, ferreiro, ou pelo menos seria um plano extremamente difícil de concretizar visto o cavalo ter já deixado à muito de ser considerado imprescindível meio de deslocação e motor de indústrias. Onde se pretende chegar, sem aprofundar muito em análises dispostas a

discussão, é que não se encontram grandes argumentos negativos contra o fato de hoje em dia alguém ter uma criança a cargo que diga que o seu sonho é ser programador de vídeo jogos, muito pelo contrário, parece ser uma excelente escolha de futuro, ainda que sendo um percurso difícil é desde logo bom que exista predisposição para isso.

No ano de 2014 foi publicado como é já costume uma lista descritiva das funções inerentes ao desenvolvimento de jogos, bem como a média salarial auferida pelos respetivos profissionais sendo que um engenheiro de software/programador de jogos nos EUA no ano de 2014 auferiu um salário base anual de cerca de 93 251\$ USD algo acima da média nacional do país que se situa nos 47 230\$ USD (Gamasutra, 2014).

Outros fatores da maior relevância são previsões económicas, leituras estas geralmente manipuladas para que o ponto de vista da parte interessada prevaleça sobre qualquer outro ponto de vista. É indubitável que a sociedade onde nos inserimos é uma sociedade de consumo, capitalista e orientada a resultados económicos e que esses mesmos resultados moldam as indústrias e a sociedades e selecionam o que podemos ter ou devemos ter e esperar nas décadas seguintes. Geralmente não é verificado que isto seja de forma alguma uma ciência exata ou precisa, mas fato é que os investimentos que podem ou não tornar estes projetos realidade geralmente seguem essas mesmas previsões (ESA, 2014). O mundo económico é de certa forma imprevisível e as previsões falhadas, umas a seguir às outras, desde a revolução industrial, continuam a ter um peso extraordinário nas decisões do que será ou não um bom investimento, e em linha com estas previsões surge uma excelente oportunidade numa indústria que se prevê continuar a crescer nos próximos anos muito pela indexação de novos mercados ao mercado geral dos sistemas de distribuição de bens virtuais onde os vídeo jogos representam uma significativa fatia do bolo (ESA, 2014). Este é provavelmente o maior indicador de estabilidade da indústria e da sua viabilidade. Por outro lado, o mercado apesar de ainda recente, com pouco mais de uma década, é já movimentador de grandes quantidades de capital e de muitos investimentos de sucesso e muitos mais de insucesso, e começa-se já em 2015 a ver uma estabilização deste mercado, surgindo grandes nomes como referências mundiais e que sozinhas captam já de forma quase permanente, percentagens do mercado muito significativas (IGN, 2015). Ainda assim e com estes gigantes na corrida é revisto como um mercado com potencial enorme para criação de empresas e emprego, motivando governos e instituições privadas a investir nesta área e na sua relação com a formação académica e profissional, sendo esse o maior problema da indústria, e já como alguns especialistas veteranos referiram, perpétua uma falta de profissionais, em particular na área de engenharia de software e é então mais do que necessário continuar o investimento na formação específica de profissionais orientados para este tipo de indústria, com particular interesse na área do desenvolvimento de software, não esquecendo nunca, que quando falamos de jogos de vídeo referimo-nos a um tipo específico de software (Blow, 2004). A desconsideração do que realmente é o desenvolvimento de jogos, têm resultado em investimentos falhados que potenciam a monopolização da indústria, e que inflacionam de forma brutal os custos resultando em graves falhas na gestão de projetos e desenvolvimento.

Parece então fundamental mais do que nunca, face a todos os argumentos apresentados até ao momento, que este tipo de software seja olhado com maior atenção e detalhe para que o momento em junção com as suas potencialidades sejam aproveitadas para atingir objetivos que façam acima de tudo a sociedade avançar para um caminho onde daqui a uns anos os jogos sejam a forma ‘clássica’ de se formar não só as próximas gerações, mas também os profissionais mais bem preparados das suas áreas profissionais, capacitando-os de executar as suas funções através de treino em prática simulada (Rouse III, 2004).

2.1.4 Jogos Sérios

Até ao momento foram apresentadas algumas importantes questões sobre vídeo jogos e avaliadas as suas potencialidades num mercado consolidado e que apresenta excelentes oportunidades de penetração e inovação, não apenas e só na perspetiva tecnológica e lúdica mas também enquanto ferramentas de aprendizagem e de formação profissional. Jogos sérios são então consideradas ferramentas inovadoras de aprendizagem colocando sobre o seu desenvolvimento um maior nível de exigência revelando um processo de design mais complexo e orientado a resultados que sejam mesuráveis de forma incremental e adaptada aos seus utilizadores (Tarja, 2007).

Jogos sérios refletem a prática simulada de atividades de forma a preparar os jogadores para atuar quando confrontados com cenários reais. Durante muito tempo os jogos sérios foram exclusivamente utilizados em sectores de grande relevância económica como a defesa militar, aviação, medicina entre outros, onde existia capacidade financeira e uma clara mais-valia na prática simulada levando a prática ao aprimoramento das habilidades dos profissionais reduzindo a taxa de erro humano que muitas vezes resultava na perda de vidas humanas (Luppa, 2011).

Na última década e face à massificação dos computadores pessoais e rápido avanço tecnológico, outros sectores, que já anteriormente olhavam com interesse a utilização de simuladores começaram a vislumbrar a possibilidade de investir em soluções adaptadas aos seus propósitos (Tarja, 2007) sendo atualmente esta uma área do desenvolvimento de jogos também em franco crescimento. O desenvolvimento deste tipo de software procura então criar soluções simuladas cujo design se foque no comportamento humano e nos conteúdos a transmitir aos profissionais envolvidos na simulação criando sinergias multidisciplinares, estudando com maior detalhe o público-alvo e o contexto de aplicação real da prática simulada.

Ao criar um jogo sério é então exigida especial sensibilidade na definição do design de jogo de forma a criar um cenário conceptual relevante, e que seja em si desejável e desafiante (Brathwaite, Brenda, and Schreiber, Ian , 2009). Os jogos sérios ao invés de jogos de carácter exclusivamente lúdico, focam o seu propósito na resolução de problemas simulados onde a aprendizagem dos jogadores é o objetivo máximo da experiência. Ao serem envolvidos nestas experiências os jogadores podem testar novas abordagens a problemas permitindo daí retirar uma componente de aprendizagem (Rowan, 2010). Na sua componente prática de

desenvolvimento, este tipo de jogos pode ser desenvolvido utilizando ferramentas genéricas de desenvolvimento de jogos o que permite um aproveitamento do estado da arte nessa componente sendo na componente de design de jogos onde é notada maior diferenciação deste tipo de jogos (Luppa, Borst, 2011).

2.2 Independência versus a Indústria

2.2.1 Então queres ser um programador independente

Ser *Indie*, ou programador independente de jogos, é um tema bastante discutido e ativo nos dias de hoje, estando a discussão da definição do termo geralmente presa à análise superficial de tipos de jogos, criatividade das histórias, jogabilidade, criatividade... onde alguns mais radicais defendem que isso é apenas aplicado a desenvolvimentos totalmente originais e criativos outros defendem que se trata de liberdade de movimentos de não estar preso a uma editora ou a um mercado específico (Yu, 2013) . Não existe na prática uma definição estática sobre o que é, e deve ser um criador de jogos independente, sendo que existe uma convergência em torno de alguns tópicos, como a não fixação em objetivos puramente financeiros, a exposição de criatividade artística e possibilidade de exploração técnica (Full Sail University, 2014).

Desenvolver jogos é difícil! (Blow, 2004) Esta afirmação é de tal forma repetida e pode ser facilmente encontrada numa qualquer pesquisa sobre criadores de jogos independentes sobre o estado da arte do caminho que iniciaram profissionalmente.

Fazer jogos é fácil! É também uma afirmação também geralmente encontrada numa pesquisa sobre criação de jogos. Com todas as ferramentas disponíveis para criação de conteúdos, criar jogos apresenta-se hoje como uma tarefa fácil.

De fato não existe uma resposta concreta a estas questões que devem ser analisadas de forma contextual. Os cenários mais comuns caracterizam-se essencialmente por desenvolvimentos de jogos enquanto atividade de prática e de aprendizagem, em que a tarefa é de fato acessível, sendo esses desenvolvimentos maioritariamente suportados por ferramentas, geralmente acompanhadas de excelente documentação e outros métodos de ensino, que permitem a que pessoas com escassos conhecimentos técnicos ou até formação criem os seus próprios jogos, simplificando o processo de forma extraordinária e permitindo que conjugadas, vontade e dedicação, qualquer pessoa pode ser um potencial criador de jogos independentes (Full Sail University, 2014). Dimensões como, metas temporais e gestão de recursos humanos e financeiros, são particularmente, e à imagem de qualquer outro negócio, o que determina o contexto mais profissional do desenvolvimento de jogos alterando o paradigma e obrigando a um maior rigor na escolha e gestão de recursos, quer eles sejam técnicos, financeiros ou humanos, posicionando aí a afirmação de que, desenvolver jogos é difícil sendo uma visão otimista considerar que o desenvolvimento de qualquer tipo de

software comercial é simples, acordando que as dificuldades em encontrar e contratar profissionais de qualidade corresponde a processos (GameTrailers, 2011) .

Neste documento o foco está maioritariamente direcionado para criadores de jogos em contexto de aprendizagem com pretensões profissionais, não sendo esperado que indivíduos com objetivo de criar jogos como forma de passatempo veja necessidade de manter conhecimentos de base a pesquisa e investigação apresentada neste documento, sendo que o mesmo pode ser uma ajuda na escolha das tecnologias a experimentar e abordagens a ter numa perspetiva de criação de portefólio curricular que seja fundamento para conseguir iniciar uma carreira nessa indústria.

Profissionalmente o papel de um criador de jogos independente acaba inevitavelmente por chocar no papel de um empreendedor tecnológico onde as necessidades do domínio de várias componentes de negócios se revelam da maior importância e onde conhecimentos técnicos rapidamente se revelam insuficientes na prossecução do objetivo da criação de jogos que possam ser distribuídos de forma massiva e capazes de suster as necessidades de recursos. Apesar de não ser apontado como objetivo maior dos criadores independentes de jogos é um fato indubitável que o sucesso é medido sobre forma do número de vendas e distribuição do software pelo maior número de pessoas, e consequentemente, os lucros financeiros obtidos. Pela normal caracterização do contexto de desenvolvimento de jogos independentes, e relativamente ao baixo nível de investimento geralmente associado aos títulos desenvolvidos por criadores independentes, sucessos recentes têm reforçado o interesse de alguns dos principais distribuidores a apostar na criação de programas de apoio a programadores independentes apoiando os mesmos em todas as áreas de negócio desde o desenvolvimento ao marketing e distribuição (GameTrailers, 2011).

2.2.2 A importância dos jogos independentes

Há alguns anos poucos eram os que poderiam ambicionar ser criadores de jogos independentes a nível profissional, mas nos dias de hoje com a massificação da internet e dos mercados de distribuição de conteúdos digitais, a cena Indie de jogos começa a ganhar força levando a uma maior abertura à criatividade, multiplicidade de géneros de jogos e a revitalização de alguns géneros clássicos com a introdução de novas plataformas. Andy Schatz CEO da Pocketwatch Games afirmou que o movimento Indie é a mais importante transição da indústria dos jogos desde ascensão da Internet, e que criadores de jogos assumindo papel de visionários, sozinhos criaram o novo género de jogos casuais bem como acordaram géneros esquecidos (Gamasutra, 2007). Jonathan Blow um famoso Indie revelou que na sua perspetiva os jogos independentes são os únicos capazes de explorar caminhos considerados desinteressantes para as grandes editoras, como a aposta em jogos que sejam capazes de melhor explorar o poder expressivo do jogos (GameTrailers, 2011) A leitura de opiniões de quem já alcançou e provou irrefutavelmente que este é um caminho possível, demonstram que os criadores independentes podem percorrer caminhos únicos livres de amarras onde o mais promissor caminho financeiro é o único possível aos olhos de investidores, ajudando a criar

experiências enriquecedoras de aprendizagem e estimulando aspectos cognitivos dos jogadores, acabando por servir de bússola tecnológica testando segmentos do mercado que à partida estariam afastados da indústria (Gilbert, 2014).

A título de resumo os programadores independentes são criativos movidos pela paixão de desenvolvimento de jogos e abrindo portas à partida fechadas e caminhos inexplorados de forma livre assumindo responsabilidade sobre todas as adversidades que essa liberdade representa (GameTrailers, 2007). Aproveitando a crescente disponibilidade de ferramentas, cada vez mais profissionais dedicam o seu tempo a este tipo de projetos contando para isso com apoio por parte dos maiores criadores de plataformas de distribuição do mundo provando que agora é o momento de apostar (Wynne, 2014).

2.2.3 Plataformas e indústria

Com o aparecimento da vontade de desenvolvimento de um jogo o primeiro desafio encontrado nos dias de hoje para qualquer indivíduo ou equipa com aspirações a criar uma experiência deste género será a decisão da plataforma ou plataformas onde o jogo irá ser disponibilizado. Apesar da constante convergência tecnológica é necessário considerar um conjunto de requisitos não funcionais, que iram condicionar a escolha a realizar, bem como analisar detalhadamente o segmento de mercado que pretendemos explorar (Chernyak, 2014).

2.2.3.1 Consolas

Atualmente existem disponíveis no mercado dois tipos de consolas de jogos. As consolas de sala, que geralmente são ligadas a um monitor ou TV e as consolas portáteis.

Relativamente às consolas de sala (fixas) encontramo-nos num momento de transição em que as principais plataformas recentemente lançaram os seus mais recentes equipamentos encontram-se em escrutínio sobre o novo hardware e lançamento de jogos exclusivos para cada plataforma.

- Sony Playstation 4
- Microsoft Xbox One
- Nintendo Wii U

Outra subcategoria das consolas é a categoria das consolas portáteis,

- Sony Playstation Vita
- Nintendo 3DS

As consolas são em si sistemas cujo principal objetivo é jogar vídeo jogos, atividade esta que têm sido nas mais recentes versões, também complementada com serviços extra, como acesso à internet, visualizadores de áudio e vídeo, e ainda possibilidade da aquisição de aplicações em sistemas de distribuição internas do sistema bem como a possibilidade de

subscrição de serviços. Estas característica anteriormente apenas acessíveis através de computadores pessoais surgiram de forma a acompanhar os desenvolvimentos tecnológicos noutras plataformas onde as ferramentas de partilhas sociais e aquisição de conteúdos digitais através de micro transações começaram a ganhar cada vez maior relevância (PC Gamer, 2013). De certa forma o que mais facilmente caracteriza estes dispositivos de jogos é estabilidade de hardware e o fato de o mesmo se manter praticamente inalterado durante mais anos do que qualquer outro tipo de plataformas. Cada consola é em si um ecossistema fechado onde todos os desenvolvimentos passam por rigorosos testes de qualidade e de performance de forma a garantir a qualidade do produto onde a maturidade do mercado faz com que seja tipicamente ocupado por títulos designados por AAA ou triplo 'A', títulos estes associados a jogos do mais alto nível de qualidade (Rubino, 2014).

No passado, construtores de consolas lutavam entre si para assinar contratos milionários de exclusividade com as editoras que detinham os direitos dos mais exclusivos e promissores jogos, mas os mais recentes sucessos de jogos desenvolvidos por programadores independentes veio dar especial foco a estúdios de desenvolvimento mais modestos e menos reconhecidos. Apesar de alguns intervenientes no desenvolvimento de títulos independentes não acharem de todo significativo que avanços a nível de performance de hardware sejam em si motivo de alterações ao nível de design de jogo, estando essas motivações mais relacionadas com criatividade e criação de novos conceitos, concordam ser inegável que essas alterações e o acesso a consolas de ultima geração com característica de hardware únicas acabam em si por gerar novas ideias e fomentar a criatividade, existindo a suspeição de que todo este entusiasmo por parte dos construtores de consolas nos títulos desenvolvidos por independentes, e poderá eventualmente ter outra motivação potenciada na captação dos programadores independentes que cada vez estão menos interessados nas consolas e mais focados nos desenvolvimentos para os dispositivos móveis e computadores pessoais (theguardian, 2013).

2.2.3.2 Computadores pessoais

Segundo a companhia *Newzoo* dedicada à investigação da indústria de jogos e análise de desempenho do mercado 2015 será ainda um novo e entusiasmante ano para a indústria sendo espectável um crescimento de 9.4% face ao ano anterior. Ainda de acordo com esta entidade o mercado de jogos é ocupado em 37% pela utilização de computadores pessoais para fins lúdicos mantendo o computador pessoal como a mais utilizada fonte de acesso a jogos de computador e representando uma indústria avaliada em cerca de 33.7 mil milhões de dólares americanos (Sinclair, 2015).

The Global Games Market | 2015^e

Per Screen and Market Segment | Year-on-Year Growth

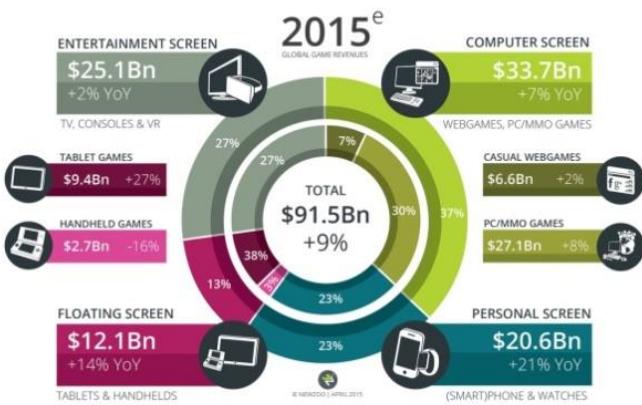
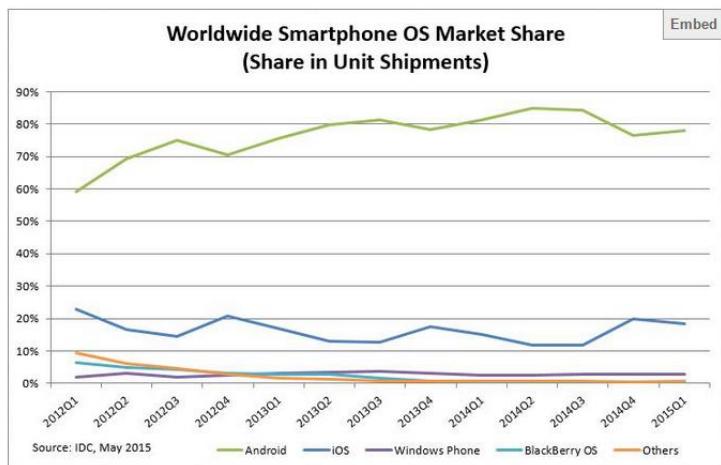


Figura 1 – Estado do mercado de distribuição de jogos (a)

A nível tecnológico os computadores pessoais apresentam uma diversidade de conjugação de componentes única e sem restrições sendo possível comprar computadores prontos para utilização ou comprar componentes individualmente e montar uma máquina à medida das necessidades ou até alterar componentes de acordo com as necessidades, característica exclusiva deste tipo de equipamento e plataforma. Para programadores independentes esta é a plataforma natural, sendo neste tipo de equipamentos onde são desenvolvidos os seus produtos. Nesta plataforma e como podemos ver na imagem acima existe uma subcategoria designada por *webgames* (Thorn, 2008) a qual vem a ganhar cota de mercado associada à recente inovação tecnológica relativa ao estado da arte da Internet que é a massificação dos navegadores com suporte HTML5 que permitem a criação e disponibilização de jogos sem qualquer tipo de instalação. Em última análise esta é a plataforma mais eclética de todas as disponíveis, disponibilizando um sem número de géneros e títulos a custos geralmente bastante mais reduzidos do que nas outras plataformas, quer para aquisição por parte dos utilizadores quer para desenvolvimento por parte de programadores independentes existindo muitas vezes o conceito de comunidades onde os jogadores acompanham e apoiam os criadores de jogos a desenvolver os seus produtos de forma mais orgânica (GameTrailers, 2011).

2.2.3.3 Dispositivos móveis

Os dispositivos móveis têm sido indubitablemente o motivo pelo qual o mercado de jogos têm sido tão agitado na última década. O rápido avanço nas características de hardware e a massificação do acesso à internet através deste tipo de dispositivos aliado ao seu baixo custo de aquisição têm sido catalisadores do crescimento de um dos maiores mercados de distribuição de jogos da atualidade (Sinclair, 2015).



Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q1 2015	78.0%	18.3%	2.7%	0.3%	0.7%
Q1 2014	81.2%	15.2%	2.5%	0.5%	0.7%
Q1 2013	75.5%	16.9%	3.2%	2.9%	1.5%
Q1 2012	59.2%	22.9%	2.0%	6.3%	9.5%

Source: IDC, May 2015

Figura 2 – Share de mercado de sistemas operativos móveis

Empresas como a Apple com o iPhone e o seu sistema operativo iOS, a Google com o seu sistema operativo Android e mais recentemente a Microsoft com o Windows têm em muito criado condições para que tal momento se mantenha, sendo incansáveis na disponibilização de todo tipo de apoio que mudaram radicalmente modelos de negócio, desde produção, distribuição e aquisição de clientes bem como potenciaram a criação de centenas de empresas e milhares de postos de trabalho por todo mundo (Forbes, 2012).

Como podemos ver na imagem, analisando os mais importantes sistemas operativos do mercado de dispositivos móveis podemos verificar que o sistema operativo Android face à sua vertente flexível em relação à elasticidade dos custos e gamas de dispositivos apresenta uns impressionantes 87% de cota de mercado relativo a Maio de 2015 representativos de 334.4 milhões de dispositivos vendidos (IDC, 2015). Outro fator da maior relevância para programadores independentes de jogos é o valor de 93% de utilizadores de dispositivos móveis que dispostos a pagar por conteúdos digitais distribuídos através dos sistemas de distribuição dos sistemas operativos móveis.

Relativamente ao desenvolvimento de jogos de programadores independentes o mercado das plataformas móveis é sem sombra de dúvida o mais promissor e realista considerando não só o potencial numero de utilizadores aos quais é possível chegar, o fato dos seus utilizadores não terem expectativas de encontrar jogos com qualidade de AAA e serem mais tolerantes face ao investimento requerido da sua parte faz com que o desenvolvimento possa ser mais iterativo e dinâmico (Rubino, 2014). Os jogos casuais como o famoso Candy Crush Saga da King (King, 2015) são exemplos de como jogos com mecânicas e gráficos mais simples podem ser efetivos na sua finalidade quando consolidados todos os fatores que fazem um excelente

jogo aliados a uma boa gestão de conteúdos e expectativas dos jogadores. O desenvolvimento de um jogo para as plataformas móveis ainda que teoricamente menos exigente, no que respeita a recursos técnicos e humanos em relação a títulos para as anteriormente referidas plataformas, deve ter particular atenção ao fato de serem produtos com ciclos de vida relativamente mais curtos ou que mantém uma constante relação de proximidade com os seus jogadores através de mecanismos de análise de dados de forma a garantir que o produto se mantém atualizado e interessante para os jogadores, o que em si corresponde a um desafio técnico de extrema exigência (VB, 2014).

Um recente caso de sucesso que suscitou curiosidade na comunidade de programadores de jogos independentes para dispositivos móveis foi o jogo Monumental Valey, jogo de mecânica simples de aspecto gráfico limpo e agradável que conseguiu ganhar rapidamente destaque num mercado que anseia por jogos frescos de criatividade e desafio. Apesar de aparentar ser um produto tecnicamente simples envolveu cerca oito elementos ao longo de 55 semanas com um custo de 852 mil dólares e foi até à data o jogo que conseguiu o maior valor de vendas num único dia, atingindo cerca de 145 530.00 dólares e terminando o ano de 2014 com quase 6 milhões de dólares em vendas (Devore, 2015).

São casos como o referido que criam momentos de êxtase em volta de um mercado como o dos dispositivos móveis, mas nunca será de descurar o fato que mesmo os jogos aparentemente mais simples não o são de todo, considerando que o esforço colocado num projeto deste género facilmente passa despercebido aos utilizadores finais e cria muitas vezes a ilusão de que desenvolver um excelente produto do género é simples e é em si sinónimo de vendas e sucesso, desvalorizando todo o esforço necessário no seu desenvolvimento e na árdua tarefa de ganhar visibilidade num mar de aplicações e jogos que é em si um dos maiores desafios encontrados em qualquer plataforma, mas particularmente complicado nos mercados de distribuição digital onde são disponibilizados diariamente centenas de novos jogos (PocketGamer, 2015).

2.2.4 Quem irá ganhar a corrida da indústria?

Nos dias de hoje uma das mais relevantes questões para os envolvidos na indústria dos jogos e que se impõe é a dúvida sobre a possibilidade das plataformas móveis se sobreponem em cota de mercado relação às consolas, encontrando-se aí a mais forte sobreposição de segmentos de mercado relativos às plataformas apresentadas. Na realidade pensar em 2010 sobre a possibilidade de tal ocorrência era apontado como pouco provável e hoje em 2015 tudo aponta para que de fato isso possa vir a ocorrer num espaço curto de tempo (Rubino, 2014).

Algumas das mais reputadas instituições de análise de mercados aponta para que a indústria dos jogos para dispositivos móveis mantenha o seu franco crescimento durante os próximos anos, não se limitando a crescer retirando cota aos segmentos já existentes, mas sim convertendo não jogadores, sendo ainda apontado como o mercado mais apetecível da atualidade para criadores de jogos independentes. Por outro lado é irrefutável que os maiores

construtores de consolas de vídeo jogos têm também vindo a criar infraestruturas para atrair programadores independentes para as suas plataformas e que estudos recentes mostraram que este é também um mercado interessante para programadores independentes apurado que mais de 60% dos utilizadores inquiridos pelo sitio na internet Android Central respondeu que utiliza de fato os dispositivos móveis para jogar jogos sociais e casuais e mantém o interesse nas consolas para jogos mais ‘sérios’ ligando este termo a jogos AAA. Na realidade ainda sobre o mesmo estudo é apurado que mais de 40% dos inquiridos acredita que todas as plataformas se irão manter e crescer nos próximos anos acordando que existe espaço para cruzamento de experiências e que o mercado é suficiente para abranger todas as plataformas. De certo modo existe uma geral convergência de opinião que comporta que os dispositivos móveis ganham a corrida face à sua conveniência, portabilidade e disponibilidade enquanto as consolas e os computadores pessoais mantém a sua liderança em relação ao poder de hardware, o que mantém em aberto a resposta a quem de fato irá dominar o mercado, não sendo de todo possível fazer muito mais que previsões sobre o assunto acordando que é uma indústria em crescimento e que não apresenta caraterística de abrandamento pelo menos até ao fim da década (Rubino, 2014).

2.2.5 Então que caminho seguir?

Esta será em contexto uma das mais repetidas questões por qualquer aspirante a programador de jogos independentes e em boa verdade será uma pergunta sem resposta possível sem que sejam analisadas ao detalhe aspirações, competências e objetivos de cada individual ou equipa. Esses objetivos podem divergir de pessoa para pessoa, e podem estar relacionadas com objetivos financeiros, de criação de próprio emprego ou formação de empresa, como expositor e escape para criatividade, como ferramenta de aprendizagem, para criação de simulações e jogos sérios entre outras motivações.

Programadores independentes dos Estados Unidos, onde o mercado de vídeo jogos assume um papel reconhecido a nível nacional, Jonathan Blow, Chris Hecker e Marcus Persson, criadores de alguns dos mais famosos títulos independentes dos últimos anos, defendem quase em uníssono que a plataforma ideal para independentes são os computadores pessoais onde o chegar ao público é mais acessível e onde é possível criar conteúdos e distribui-los de forma digital utilizando formas mais tradicionais de pagamento como Paypal e servindo-se ainda de páginas de internet próprias para promoção exclusiva dos seus jogos onde lhes é permitido um maior nível de criatividade mantendo afastadas as editoras e fugindo dos modernos sistemas de distribuição designados por ‘marketplaces’ e ‘app stores’ que consideram ser demasiado complexos para programadores independentes. Referido por Chris Hecker seguir o caminho de distribuição através das lojas digitais, apesar de possível é por si considerado um caminho extremamente difícil, revelando não acreditar que seja possível para independentes viver desse tipo de modelo como base de negócio, e assumindo a opinião de que quem o fizer verá o seu trabalho ser considerado um passatempo onde existe a potencial sorte de ser reconhecido e que será na melhor das hipóteses humildemente destacado. Segundo os mencionados acima, existem vários caminhos possíveis, sendo que enquanto

Jonathan Blow segue uma estratégia financeira mais arriscada, em que investe diretamente o seu tempo e dinheiro nos projetos e assume pessoalmente o risco de estes virem a ser sucessos ou fracassos, juntando ainda a isso uma enorme quantidade de sacrifícios pessoais, Marcus Persson defende uma linha mais conservadora em que antes de concluir o seu maior sucesso, o jogo Minecraft vendido à Microsoft por 2.5 mil milhões de dólares, manteve um emprego estável desenvolvendo o seu projeto em tempo parcial (Independent, 2015).

Apesar das diferentes estratégias as opiniões convergem no sentido de que o desenvolvimento de um projeto deste género deve sempre ser guiado pelas pretensões e uma forte crença no projeto, não desistindo à primeira tentativa e adequando o risco de forma a não criar uma situação de insustentabilidade. A primeira e mais importante lição referida é começar pequeno, em projetos que sejam realistas e que estejam ao alcance de serem terminados independentemente da plataforma (Full Sail University, 2014).



Figura 3 – Decisão sobre desenvolvimento de jogos

Nenhum programador independente irá construir como primeiro jogo um sucesso, e caso isso aconteça será tão provável como acertar na lotaria segundo Jonathan Blow . As plataformas móveis são um mercado astronómico e detêm enorme potencial onde programadores independentes podem ter imenso sucesso, mas como refere Chris esse é um caminho bem mais complexo do que à partida sugere. Enquanto no mercado das consolas e computadores pessoais existe o conceito de construir um produto, vende-lo por um preço fixo e obter lucros provenientes desse modelo de distribuição, seja ele digital ou tradicional envolvendo editoras e distribuidores de bens físicos, o mercado das aplicações e jogos móveis é bastante mais complexo e está em constante mudança, e mesmo dentro da plataforma de dispositivos móveis existem modelos distintos e dificuldades inerentes ao processo dos modelos a seguir. Programadores independentes como Jonathan Blow, Chris Hecker e Marcus Persson, defendem a quem pretende seguir esse caminho a realização de uma exaustiva avaliação dos riscos e o assumir de todo o processo ou encontrar parcerias exclusivamente dedicadas a essas questões (GameTrailers, 2011).

Em conclusão, e de acordo com as referidas opiniões e diferentes casos de sucesso descritos é acima de tudo, e independentemente da plataforma ou plataformas de distribuição, importante reconhecer que desenvolver jogos é, sobre tudo para programadores independentes, um trabalho de paixão, e que o manter de uma postura pragmática e focada

no propósito dos projetos que se pretendem realizar é fundamental, concentrando esforços em obter resultados tangíveis não entrando em caminhos que potencialmente levarão a situações insustentáveis. Ainda citando Chris e Jonathan é importante acima de tudo não deixar a tecnologia abranger o projeto e ficar preso a decisões desse género ou ferramentas e estilos, como 2D ou 3D, e acima de tudo começar pequeno.

2.2.6 Financiamento, distribuição e monetização

Como referido um dos desafios em ser independente está invariavelmente ligado ao fato de ser possível manter uma posição de decisão sobre o trabalho a ser desenvolvido e que para que isso seja possível é necessária também uma determinada independência financeira sendo que esta é geralmente a primeira barreira com que qualquer programador independente se irá confrontar. Apesar de não ser da maior relevância para o contexto do presente documento é importante ter em consideração alguns dos modelos de obtenção de financiamento, distribuição e monetização, mais comuns e acessíveis para o desenvolvimento de projetos deste género, pois podem ser uma das referidas características que podem influenciar decisões sobre caminhos a seguir em relação a plataformas e distribuição e até ao produto a desenvolver.

A primeira fase de qualquer projeto é obtenção de financiamento sendo esta relativamente similar a qualquer outro tipo de negócio de desenvolvimento de produto. Modelos de obtenção de financiamento variam de país para país, sendo que o reconhecimento nacional do valor da indústria de entretenimento e vídeo jogos é altamente motivador para o investimento ou falta dele em diferentes contextos. Atualmente existe em particular nos Estados Unidos da América, uma associação designada por ESA que celebrou em 2014 vinte anos de existência e onde a indústria de vídeo jogos representa já uma indústria de mais de vinte mil milhões de dólares por ano contribuindo com mais de seis mil milhões de dólares para a economia do país e (ESA annual report) que resultaram ainda num crescimento de 9% sobre a taxa de empregabilidade na indústria de jogos (ESA, 2014). Este tipo de associações é altamente motivadora de investimentos privados na aposta na criação de novos vídeo jogos e na inovação tecnológica. Infelizmente a nível nacional ainda não vimos nenhum tipo de movimento que venha reforçar o interesse de investidores nesta área e o financiamento mais comum é o auto financiamento e financiamento público, que apesar de ser um excelente contributo ainda não representa fator diferenciador na potenciação da indústria.

Exemplos como o de Alexandre Ribeiro fundador da Game Whizzes são a mostra para o desenvolvimento independente em Portugal e conseguem demonstrar que é possível com algum investimento e uma mão cheia de competências conseguir auto financiar este tipo de projetos tendo conseguido chegar a número um no mercado dos Estados Unidos, uma das mais relevantes lojas de distribuição da Google e da Apple (Ribeiro, 2014).

Quando nos referimos a distribuição é importante conhecer os modelos existentes:

Modelo clássico – comprehende o acordo com uma editora de forma a que a mesma seja responsável por todo processo de marketing e distribuição final do produto. Este modelo está particularmente centrado nas distribuições de jogos para consolas e computadores pessoais sendo a monetização dos produtos acordada entre as partes e resultante do valor de venda ao utilizador final.

Distribuição digital – Apesar de ser mais acessível, sem necessidade de grandes contactos ou formalização, a utilização deste modelo apresenta várias vantagens e desvantagens, sendo este um modelo complexo e altamente mutável. Nesta área existe um sem número de cenários possíveis, desde estúdios independentes que mantém um sitio na internet onde é possível adquirir os jogos utilizando sistemas de pagamento através de mecanismos financeiros, e/ou mercados digitais para plataformas específicas que funcionam como lojas virtuais onde os criadores de jogos acordam inicialmente na cedência parte do valor da transação (Tran, 2014).

Nos últimos anos com a massificação dos dispositivos móveis surgiram conceitos sobre a distribuição digital e métodos de monetização dentro das aplicações, particularmente no mercado dos jogos para dispositivos móveis em que inicialmente o conceito estava limitado a produtos designados por *Premium* onde o utilizador adquire o jogo por valor pré determinado e versões grátis geralmente implementando um sistema de monetização baseado em publicidade, onde o jogador teria como moeda de troca sobre o adquirir de direito de jogar o a apresentação de *banners* publicitários em determinados momentos da experiência de jogo. Recentemente um novo conceito importante na área da monetização é o conceito de *Freemium*, conceito este gerador de bastante polémica (Seufert, 2013).

Na sua essência o *Freemium* consiste na disponibilização de determinado jogo de forma gratuita através de um meio digital, sendo incorporado dentro do próprio jogo todo um sistema de venda de bens virtuais a serem utilizados no próprio jogo, quer seja para permitir ao jogador avançar no jogo quer sejam conteúdos que simplesmente se limitam a embelezar o próprio jogo, personagens ou quaisquer outros elementos presentes na experiência, sendo este modelo conhecido como modelo de compras na aplicação (Seufert, 2013). Alguns dos mais importantes termos a reter sobre o modelo de distribuição digital são:

- IAP (In-App-Purchases)
- RPM (Revenue per 1000 impressions)
- Impressions (Advertisement impressions)
- DAU (Daily active user)
- MAU (Monthly Active users)
- ARPDAU (Average Revenue per daily active user)

Apresentados acima temos então o conjunto de métricas que permitem a realização de cálculos indicadores de performance financeira em relação a transações executadas dentro do ecossistema da experiência de jogo. O valor de retorno é tipicamente calculado sobre a soma

de todas as IAP somadas ao ARPDAU vezes trinta, sendo este valor representativo do número de dias de um mês (Singh, 2014).

$$R[n] = IAP + (ARPDAU * n)$$

Independentemente da plataforma e ainda que apresentados de forma extremamente sucinta, de forma a manter o foco do documento no seu âmbito, o conhecimento do mercado e dos conceitos envolvidos é fundamental para que sejam tomadas as decisões ponderadas na fase mais inicial dos projetos de forma a não os ver inviabilizados a curto prazo.

2.3 Desenvolvimento de jogos

2.3.1 Metodologias de desenvolvimento e equipas

Avaliado o que é um vídeo jogo, quais as motivações e de forma sucinta o mercado, é importante entender como é na realidade desenvolvido um jogo e como deve ser abordado o problema na sua relação com projeto e gestão de recursos técnicos e humanos envolvidos.

O processo de produção de um vídeo jogo à imagem do desenvolvimento de um qualquer outro produto de software compreende um processo complexo que deve ser adaptado à imagem dos recursos existentes mantendo uma visão objetiva e gestão rigorosa de recursos, o que nem sempre ocorre, sendo os métodos de desenvolvimento formal de software desprezados, resultando esta prática muitas das vezes em prazos e orçamentos amplamente alargados sobre o objetivo inicial e/ou versões carregadas de bugs e de problemas técnicos (Novak, 2011), reforçando que planeamento é fundamental para o sucesso deste tipo de produtos de software à imagem de qualquer outro. Por outro lado a aplicação de metodologias como *waterfall* não são geralmente as mais indicadas sendo que metodologias ágeis de desenvolvimento se revelam ser mais apropriadas (Chandler, 2009).

A aplicação de metodologias ágeis como o SCRUM, onde a produção de software é orientada pela definição de *milestones* de prototipagem rápida e onde a cada sprint são avaliados um ou mais requisitos funcionais, é um dos métodos mais efetivos, revendo-se que neste tipo de desenvolvimento nem sempre estarão à partida delineadas todos os requisitos funcionais (Luppa, 2011). No entanto, e apesar de não ser aplicada de forma geral, é possível encontrar na gestão deste tipo de projetos a aplicação do modelo *waterfall* de acordo com processos onde existem fortes dependências entre tarefas, como por exemplo a criação de conteúdos de imagens e animações que dependem exclusivamente da especificação e design de jogo e não estando dependente de nenhuma outra tarefa.

Na sua estrutura clássica o desenvolvimento de vídeo jogos é distribuído por papéis a serem assumidos pelos elementos da equipa, respetivamente:

- Design de jogo
- Programação

- Arte
- Som
- Gestão de projeto
- Qualidade
- Gestão de negócio

No contexto de programadores independentes a limitação de recursos é real e a necessidade de que determinado elemento assuma um ou mais papéis dentro do projeto é caso comum (Reimer, 2005). Outro fator importante refere-se à adoção de metodologias de desenvolvimento onde não existem dados que sustentem a sua utilização no contexto independente, revelando-se o terminar de um jogo um dos maiores desafios dos programadores independentes (YU, 2010). Mesmo que de forma pouco assumida, casos como os criadores de Minecraft ou Braid, recomendam para que exista por parte dos seus criadores uma linha de raciocínio orientada por pequenos protótipos e iterações mesuráveis, ou seja, um plano (Blow, 2004).

“Startup success can be engineered by following the process, which means it can be learned, which means it can be taught.” (Ries, 2012)



Figura 4 – Fluxo de *Minimum Viable Product*

Recentemente o termo MVP (*Minimum Viable Product*) tem vindo a ganhar particular interesse por parte do desenvolvimento de software de jogos. Na prática consiste na aplicação do conceito de prototipagem rápida onde a cada ciclo de desenvolvimento, o produto é reavaliado e apreendidas lições a serem consideradas na iteração seguinte. Apesar de que, para engenheiros de software experientes isto ser entendido como prática comum, quando envolvidos em projetos independentes é relativamente simples alienar-se da utilização destas práticas. Em conclusão, independentemente do tamanho do projeto, será na maioria dos casos boa prática, a criação de um plano e a definição de metas e prazos a cumprir, de forma a obter com a experiência dos erros cometidos e aumentando a viabilidade de planos futuros (Seufert, 2013).

2.3.2 Ferramentas de desenvolvimento

Com o crescimento da indústria dos jogos e a consolidação de mercados segmentados surgiu uma necessidade de criação de motores de jogos, abrindo caminho a um novo mercado de B2B* focado na indústria dos vídeo jogos (Needles, 2011). Seria impensável que cada criador de jogos tivesse de criar de raiz toda a infraestrutura necessária para a implementação de tal produto de software, e face ao crescente número de empresas a entrar na indústria, outra surgiu imediatamente, o negócio dos motores de jogos.

Os motores de jogos compreendem um abstração sólida que permitem o desenvolvimento de vídeo jogos, na prática são como uma qualquer *framework* de software, esta com propósito particular da criação de aplicações multimédia de alto desempenho e com níveis de complexidade e de utilização de recursos computacionais da mais alta exigência (Gregory, 2014).

Apesar das múltiplas opções disponíveis a nível de motores de jogos nos dias de hoje, alguns componentes são em si fundamentais e a base mínima de uma qualquer ferramenta de desenvolvimento de vídeo jogos consiste no agrupamento de módulos mínimos para manipulação de,

- Áudio;
- Animações;
- Desenho gráfico 2D/3D;
- Manipulação de eventos de entrada;
- Gestão de memória.

Como referido, o crescimento da indústria potenciou a aposta na criação destes motores, e com o aparecimento de novas plataformas de distribuição a convergência para a criação de motores de jogos multiplataforma ganhou força resultando em ferramentas de desenvolvimento cada vez mais completas e complexas disponíveis para a generalidade dos criadores de jogos, e hoje em dia, a oferta é tal que escolher a ferramenta adequada para cada equipa ou projeto revela-se em si mesmo um projeto (Gametrailers, 2012).

Existem inúmeras ferramentas disponíveis com característica semelhantes, sendo as característica que mais rapidamente as diferenciam, o seu tipo de motor de rasterização, se 2D ou 3D, a disponibilidade da plataforma de desenvolvimento (Windows, Linux, Mac), se requer ou não conhecimentos de programação, plataformas de distribuição e a tipologia de jogos possíveis de serem construídos (PixelProspector, 2014). Outro ponto relevante é o fato de o motor do jogo ser ou não de código fonte aberto o que por vezes pode indicar que, caso a equipa de desenvolvimento pretenda estender a própria ferramenta terá ou não dificuldades em implementar e adicionar componentes próprios, sendo esta uma característica particularmente importante para projetos complexos e que leva os grandes criadores de vídeo jogos a utilizarem motores de jogos desenvolvidos por si, capacidade esta normalmente não verificada para criadores independentes. É ainda de referir que mesmo que o código da

ferramenta de desenvolvimento não seja de código fonte aberto são complementadas com sistemas de extensibilidade que permitem a criação de *plugins*.

A má escolha da ferramenta a utilizar é geralmente causadora de limitações que podem revelar-se fatais para o desenvolvimento do software colocando em causa todo o projeto sendo da maior importância que esta escolha seja feita, à imagem de qualquer desenvolvimento de software, e que os requisitos funcionais, e particularmente não funcionais, sejam avaliados de forma clara para não comprometerem o produto e a sustentabilidade do projeto a longo prazo (GameTrailers, 2011).

Considerando o contexto e objetivo deste documento foram então após análise extensa de uma lista considerável de soluções, selecionadas três ferramentas disponíveis considerando, a curva de aprendizagem, usabilidade face a conhecimentos gerais de um qualquer profissional de desenvolvimento de software, nível de funcionalidades disponíveis, plataformas de distribuição, tipologia e disponibilidade de versões gratuitas para desenvolvimento, característica estas de bastante relevância para qualquer equipa de programadores e artistas independentes, respetivamente:

- Unity 3D
- Cocos2D
- Game Closure

2.3.2.1 Unity3D

Esta ferramenta é uma das mais utilizadas no mundo dos vídeo jogos sendo muito mais do que uma *framework*, o *Unity* é uma aplicação de desenvolvimento de jogos multi propósito permitindo criar não só o jogo mas também conteúdos de media, sendo designada por IDE de desenvolvimento de jogos. Esta poderosa ferramenta introduz vários conceitos próprios e é suportada por uma comunidade ativa que conta com mais de um milhão de utilizadores estando disponível um sem número de tutoriais e documentação de suporte à aprendizagem da ferramenta.

Em boa verdade esta ferramenta é um conjunto extremamente polido de componentes que permitem a criação de um jogo e a sua disponibilização para múltiplas plataformas dentro de um ecossistema privado. Na sua vertente de programação de jogo consiste essencialmente num sistema integrado de desenvolvimento baseado em *scripts* acompanhado pelo editor *MonoDevelop* disponível para Windows e Mac, ou caso seja pretendido pode ainda optar-se pelo uso do *Visual Studio* em ambiente Windows. A ferramenta comprehende ainda um editor de terreno 3D, um gestor de animações de objetos 3D e um excelente interface gráfico amigo do utilizador. No seu núcleo encontra-se ainda um poderoso e otimizado motor de jogo capaz de processar objetos em 3D, iluminação, física, animações e *scripting*. Apesar de grátils na sua versão básica, consiste ainda numa loja de bens virtuais que podem ser adquiridos através de uma loja própria da aplicação e utilizados de acordo com as regras e leis de proteção de propriedade intelectual respeitante a conteúdos e até código, existindo ainda uma versão designada por profissional que acarreta custos de licenças de utilização adaptáveis a

diferentes cenários e equipas. Assim sendo o Unity3D é uma ferramenta que permite otimizar dentro de um ecossistema os recursos necessários para a construção de uma aplicação multimédia de alto desempenho apresentando, ainda que altamente bem documentada, uma curva de aprendizagem relativamente larga, dado ser uma ferramenta detentora de muitos conceitos e componentes próprios.

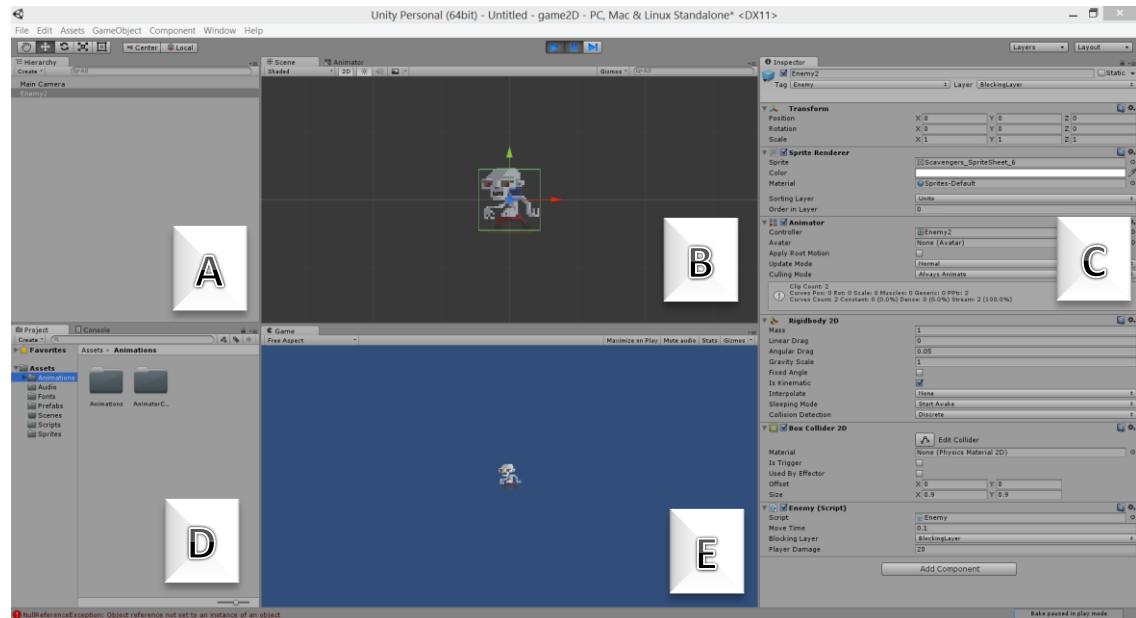


Figura 5 – Visão principal (Unity)

Os conceitos essenciais no desenvolvimento de jogos ou outro tipo de aplicações multimédia em Unity3D são,

2.3.2.1.1 Hierarquia (A)

Como podemos verificar na imagem apresentada acima, a hierarquia consiste num painel da aplicação que lista todos os objetos e subobjectos de cena na forma textual. Desde camaras, luzes, modelos 3D e *prefabs*, todos os elementos constituintes da cena estão aqui representados de forma a que possam ser selecionados e colocados em destaque no painel de cena e abrindo no painel de inspeção (C) as propriedades do respetivo objeto de jogo. Os objetos neste painel apresentam ainda a sua posição em função de dependências de outros objetos em níveis mais altos da hierarquia permitindo a criação de grupos hierárquicos e de forma a permitir aplicação de alterações de transformação a grupos.

2.3.2.1.2 Cena (B)

O painel de cena é a representação visual em forma de imagens dos objetos de jogo descritos no painel hierárquico sendo ainda possível através deste painel a edição em tempo real de objetos 3D. Este painel permite então mover e alterar o nível de detalhe dos objetos apresentados bem como propriedades de iluminação, texturas, *wireframes* ou até combinações de propriedades dos diferentes objetos. As cenas devem ser encaradas como níveis ou fases do jogo contendo apenas os elementos de jogo de cada fase da experiência.

2.3.2.1.3 Inspetor (C)

Quando selecionado um objeto de jogo é possível configurar o mesmo através da atribuição de propriedades e configuração dessas no painel de inspeção de objeto de jogo. Essas alterações são refletidas ainda na sua representação visual apresentada no painel de cena.

2.3.2.1.4 Projeto (D)

Na pasta do projeto são agrupados os *assets* do jogo. Estes estão por defeito organizados por defeito como animações, áudio, fontes de texto, *prefabs*, cenas, *scripts*, e *sprites*. Estes componentes podem ser de diversos tipos desde documentos de vídeo a imagens ou sons.

2.3.2.1.5 Jogo (E)

Para cada cena constituída por um ou mais grupos de objetos de jogo é possível pré visualizar e testar a cena para isso utilizando o painel de jogo que executa a cena particular que pretendemos visualizar. Neste painel é apresentado o que o jogador irá ver durante a cena do jogo que está selecionada permitindo por exemplo validar posicionamentos e fatores de escalamento em diferentes resoluções e teste de ações realizadas pelo jogador.

2.3.2.1.6 Game Objects

Consistem na representação, enquanto objeto programável, de um ou mais assets e ainda camaras, luzes e modelos que compõe a cena de jogo. Geralmente compreendem objetos do jogo visíveis e invisíveis para o jogador e que são transformados pelo comportamento do jogo e pelo input do utilizador em tempo real durante a execução da aplicação. Na prática são todos os objetos que estão representados no jogo e que afetam diretamente a experiência para o jogador.

2.3.2.1.7 Prefabs

Estes objetos consistem em componentes ‘pré-fabricados’ que funcionam como *templates* de objetos compostos por um ou mais *assets* de media. São basicamente classes de objetos de software a serem instanciados em tempo de execução de forma a serem utilizadas em vários contextos. Por exemplo é possível criar um *prefab* de uma animação contínua de um objeto de jogo e sempre que esse objeto for necessário em diferentes fases do jogo a sua estrutura é conhecida o que permite a reutilização interna de código e consequente otimização do programa.

2.3.2.1.8 Scripts

O motor do Unity3D na sua relação face de programação do software utiliza uma estrutura de *scripting* agrupando lógica de uma cena em conjunto com os seus objetos de jogo. Um jogo em *Unity* é então um conjunto de cenas constituídas por objetos de jogo e respetivas propriedades que podem ser alteradas em tempo de execução através de scripts pertencentes à cena. Internamente implementa um compilador designado por MONO que consiste num compilador baseado em C# e que permite aos programadores de jogos em *Unity* programar utilizando o editor específico de MONO ou ainda a utilização de Visual Studio em ambientes Windows. Existe ainda a possibilidade de utilização de uma linguagem de *scripting* designada por BOO caso os programadores tenham interesse e mais recentemente passou a

existir a possibilidade de utilizar ainda *JavaScript*, uma linguagem que cada vez ganha maior popularidade.

2.3.2.2 Cocos2D-x

Outra popular ferramenta de desenvolvimento de jogos é a Cocos2D, originalmente desenvolvida para a criação de jogos para iOS é hoje uma das mais utilizadas ferramentas de desenvolvimento do mundo, utilizada como base de alguns dos mais famosos jogos criados para as plataformas móveis e Web permitindo ainda o desenvolvimento para ambientes de computadores pessoais. Uma das principais características e que a distingue é o fato de ser totalmente gratuita e de código fonte aberto existindo uma comunidade ativa de milhares de utilizadores que de forma livre partilham conhecimento e apoiam a ferramenta na sua evolução contínua. Existem atualmente duas versões que apesar de estarem orientadas para o mesmo objetivo têm particularidades bastante interessantes face à sua implementação de baixo nível.

A versão mais antiga é a versão Cocos2d-x consiste num motor de jogos de licença aberta MIT e pode ser utilizada para desenvolver jogos e aplicações capazes de serem executadas em múltiplas plataformas com a reutilização de um único código base. A linguagem de programação base para o desenvolvimento de jogos utilizando a ferramenta é C++, sendo ainda possível o desenvolvimento utilizando Lua ou *JavaScript*.

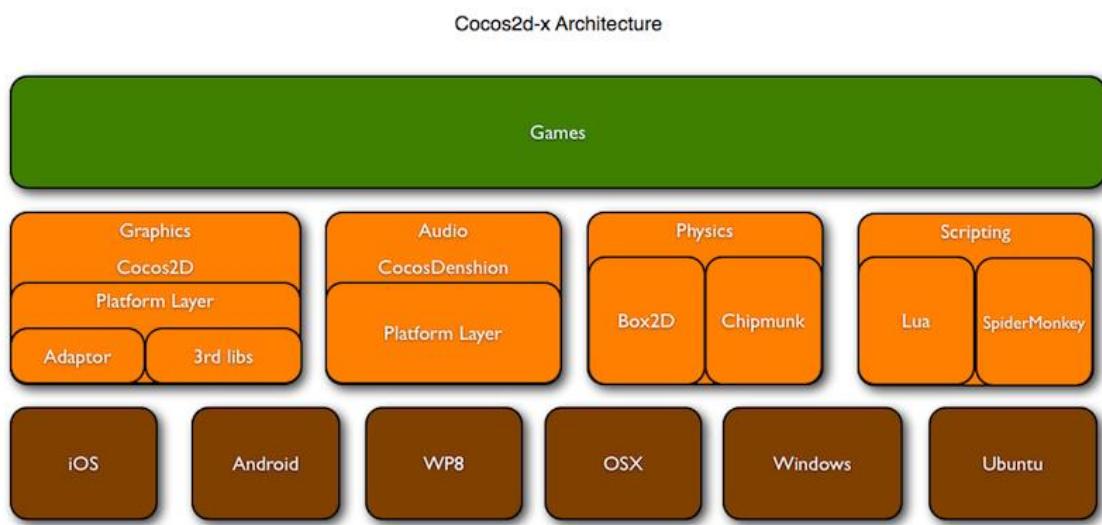


Figura 6 – Arquitetura de Cocos 2D-x

Mais recentemente e face à crescente popularidade e importância que a linguagem *JavaScript* tem vindo a ganhar foi criado um módulo designado por Cocos2d-JS que permite utilizar esta popular linguagem simplista em confronto com C++ de forma a otimizar o fluxo de desenvolvimento de produto com a introdução de uma API mais simples e fácil de aprender.

Na realidade esta versão da *framework* consiste na anterior implementação do motor de jogo em C++ com a particular integração da máquina virtual de *JavaScript* da Mozilla, o famoso *SpiderMonkey* como interface da linguagem de mais alto nível com as librarias nativas da *framework*. Esta implementação patrocinada pela Google em 2011 consiste numa implementação baseada na normalização da API Canvas/WebGL o que a torna 100% compatível com HTML5.

À imagem da implementação de WebGL em navegadores, esta implementa a API de OpenGL ES 2.0 o que permite a rasterização de gráficos pela GPU de alto desempenho em dispositivos móveis e ambiente Web, e ainda a implementação de OpenGL 2.1 para versões dos jogos para desktop, mantendo uma reutilização do código de mais alto nível do jogo.



Figura 7 – Arquitetura de Cocos 2D-JS

Independentemente da versão utilizada, a *framework* apresenta uma estrutura de desenvolvimento de software idêntica e que consiste na implementação de alto nível de uma solução de software global capaz de suster as necessidades de implementação de um qualquer género de jogo, apresentando uma API rica intuitiva e fácil de aprender sendo algumas das bandeiras da ferramenta o alto desempenho demonstrado pelas aplicações criadas em conjunto com o fato de ser totalmente gratuita e de código fonte aberto.

2.3.2.2.1 Inicialização de um projeto

A ferramenta disponibiliza uma consola onde são realizadas as operações de criação de novos projetos e gerar versões para teste e versões finais.

```
$ cocos new MyGame -p com.MyCompany.MyGame -l cpp -d ~/MyCompany
```

Conceptualmente o desenvolvimento de jogos utilizando Cocos2d é bastante similar ao processo de desenvolvimento de software dito comum de desenvolvimento OO sendo que a *framework* disponibiliza objetos de software que pretendem acelerar o processo de desenvolvimento, como objetos de cena, transições, *sprites*, menu, áudio e muitos mais.

Entres os objetos disponíveis para apoiar o desenvolvimento existem ainda alguns conceitos críticos para o processo de desenvolvimento.

O processo de desenvolvimento pode utilizar um qualquer IDE, aconselhando que o mesmo adequado para a linguagem de programação escolhida para o respetivo desenvolvimento.

2.3.2.2.2 Diretor

Cocos2d utiliza o conceito de uso de um diretor como método de gestão de cenas de jogo. O objeto Diretor é uma classe atómica na aplicação, *singleton*, e está disponível em qualquer ponto da aplicação permitindo o controlo entre cenas e transições.

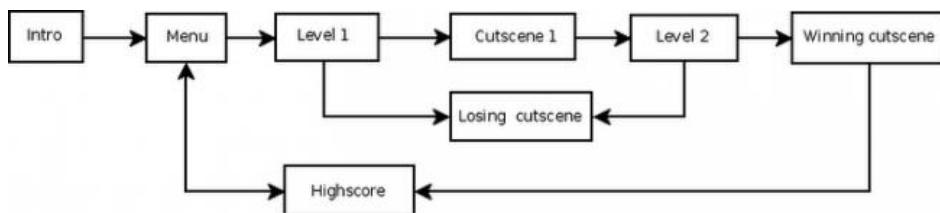


Figura 8 – Exemplo de navegação entre cenas (Cocos2D)

2.3.2.2.3 Cena

Uma cena geralmente representa um ecrã de jogo ou menu e é composto por vários componentes como *sprites*, etiquetas de texto, animações. A cena representa a composição total de um ecrã de menu ou nível de jogo por exemplo.

2.3.2.2.4 Grafo de cena

O grafo da cena representa uma estrutura em árvore em que cada elemento da cena é descrito como um nodo dessa árvore. Esta estrutura é da mais alta importância sendo que para a *framework* é esta a estrutura através da qual o motor de rasterização da ferramenta toma as decisões de desenho dos respetivos objetos que compõe a cena. Internamente o Cocos2d utiliza um algoritmo de pesquisa ordenada da esquerda para a direita. Começa a sua pesquisa no primeiro nodo do grafo (F), caminhando no sentido da esquerda para a direita na sua pesquisa tendo como resultado, os nodos da direita a serem os últimos processados acabando por ficar no topo da pilha (*z-index*). Quando são adicionados novos elementos à cena é importante saber qual o comportamento do motor de rasterização de forma a evitar situações em que os objetos não ficam visíveis evitando perdas de tempo na busca de problemas relacionados com o posicionamento dos objetos na cena.

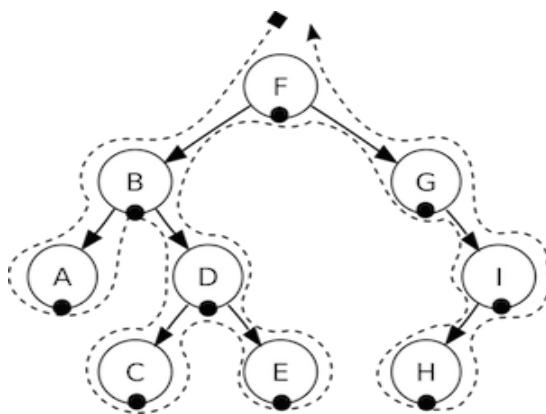


Figura 9 – Sentido de pesquisa do grafo de objetos de cena (Cocos 2D)

2.3.2.2.5 Sprite

Todos os jogos contêm *sprites* independentemente da sua implementação, sendo estes descritos vulgarmente como objetos que se movem no ecrã de jogo e que podem ser dinamicamente manipulados durante a execução da experiência. De fato se pensarmos todos os objetos que são apresentados num ecrã de cena compostos por gráficos podem ser um *sprite* mas tecnicamente caso o objeto não tenha transformações durante a execução, isso faz dele um Nodo dispensando as propriedades que são tipicamente atribuídas a um *sprite* como por exemplo a sua posição, fator de escalamento, rotação, cor entre outras.

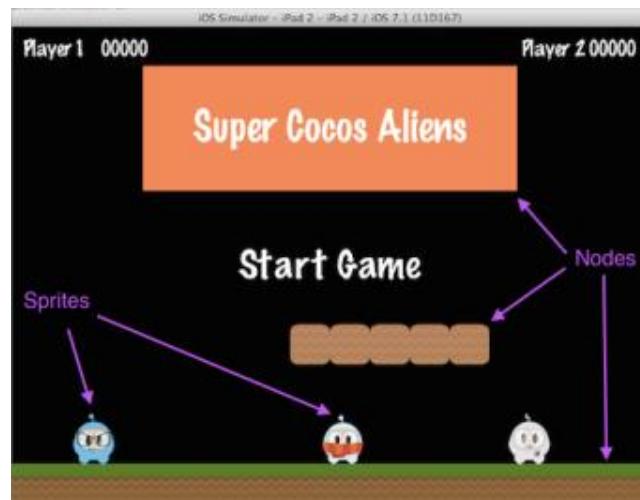


Figura 10 – Exemplo de *sprites*

2.3.2.2.6 Ação

Outra característica importante de uma experiência multimédia e que geralmente é bastante visível são as ações realizadas pelos vários *sprites* dispostos na cena resultando em movimento e animações constantes de forma a cativar e manter o interesse dos intervenientes. Para isso a ferramenta disponibiliza um objeto de ação que permite realizar transformações sobre nodos da cena e movimentação de *sprites* de um ponto para o outro.

Na verdade as ações dentro de uma aplicação de jogo ou outra experiência multimédia são provavelmente o que as distingue de outro tipo de aplicações, comportando um sem número de ações simultâneas sobre um sem número de objetos. Na ferramenta podemos então criar ações sobre nodos da cena e agrupá-las em sequências de forma a poder reutilizar essas sequências ou ações individuais.

2.3.2.2.7 Sequências e *Spawns*

Uma sequência consiste num grupo de ações aplicadas a um nodo da cena.



Figura 11 – Sequência de ações

Um *spawn* consiste na execução de uma sequência de ações com a diferença de que estas são todas iniciadas no mesmo momento no tempo e executadas em simultâneo. A utilização da técnica de *spawn* permite à *framework* executar ações da forma mais eficiente possível recorrendo a todos os recursos disponíveis aplicando paralelismo de forma a que o programador não se tenha de preocupar com essas complexas operações.

2.3.2.2.8 Relacionamentos entre nodos

Cocos2d aplica ainda o conceito de pai e filho entre nodos transpondo determinadas transformações aplicadas ao nodo de cena pai para os filhos. A título de exemplo podemos ter na nossa cena um determinado elemento de jogo composto por uma ou mais *sprites*, como uma personagem e a sua arma, e em determinado momento pretendemos fazer a personagem aumentar de tamanho sendo requerido que essa alteração seja transposta para a arma que a personagem carrega.

2.3.2.2.9 Cocos Studio

Recentemente a Sonar Systems empresa criadora da *framework* Cocos2d deu início a um projeto que continua ainda em desenvolvimento designado por Cocos Studio.

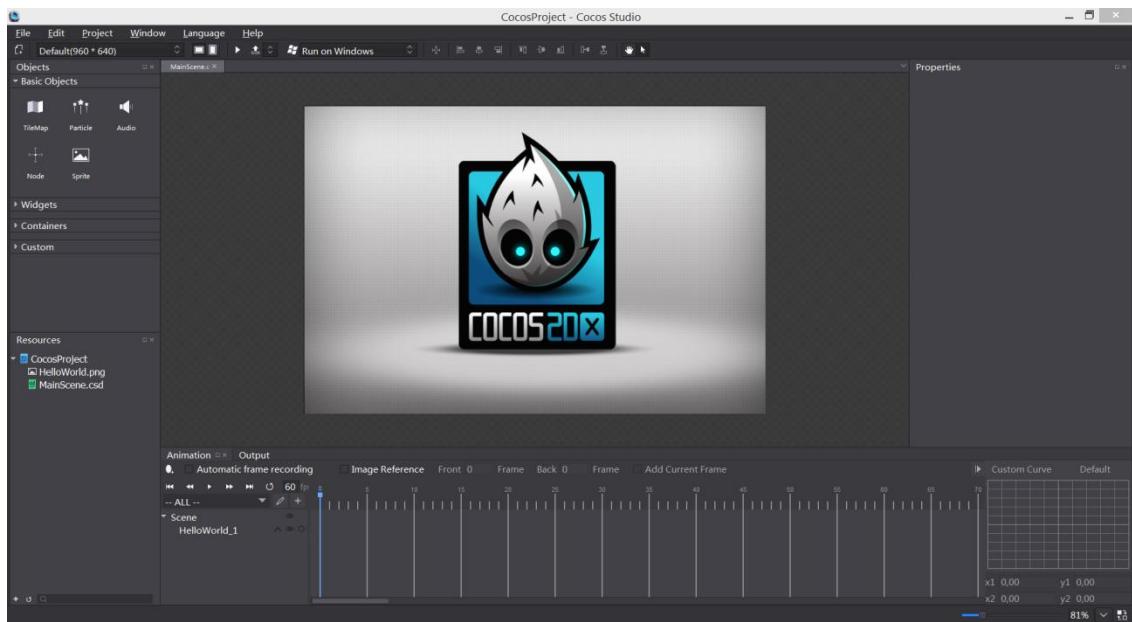


Figura 12 – Visão geral de Cocos Studio

Esta ferramenta propõe ser um editor capaz de apoiar os criadores de jogos a criar as suas cenas de jogo de uma forma mais rápida e simples abstraindo toda a parte de programação dos intervenientes nesta fase do processo. Atualmente a ferramenta disponibiliza quatro componentes principais:

Editor de gráficos – permite a criação e edição de gráficos para artistas abstraindo-os de questões relacionadas com programação e design de jogo.

Editor de animações – o editor de animações apresenta uma solução idêntica à experiência de criação de animações em Adobe Flash disponibilizando ainda ferramentas de animação com baseadas em estruturas de esqueleto.

Compressor de dados para analistas – esta opção permite importar dados de formatos Excel para que sejam convertidos em dados passíveis de serem reconhecidos pela framework e que podem ser utilizados como componentes do editor de cena.

Editor de cenas para designers – O editor de cena permite ligar todos os conteúdos criados pelas funcionalidades anteriores e criar as cenas de jogo.

Depois de criados as cenas é possível publicar esses componentes e importá-los para serem integrados no código de jogo.

2.3.2.3 Game Closure

Esta ferramenta, desconhecida para a maioria, ganhou particular interesse no ano de 2012, quando terá sido alvo de uma tentativa de aquisição por parte de uma das maiores empresas de desenvolvimento de jogos para a internet e para as plataformas móveis, a empresa Zynga

(MacMillan , 2012), ficando reconhecida no meio como a *startup* que declinou uma oferta de aquisição de 100 milhões de dólares em detrimento de obter de forma independente financiamento de cerca de 12 milhões de dólares mantendo o controlo sobre a ferramenta cujo objetivo seria como é atualmente de livre uso e código fonte aberto (Empson, 2012). A nível de engenharia de software a ferramenta assenta numa arquitetura polida baseada numa pilha de componentes que utiliza os mais modernos motores de interpretação de código *JavaScript* convertendo essas instruções em código nativo, permitindo assim a codificação de alto nível baseada em *HTML5* permitindo ainda a disponibilização para navegadores com suporte para *HTML5* (GameClosure, 2013).

Na prática a ferramenta assenta particularmente no desenvolvimento de jogos em 2D permitindo a que programadores experientes possam estender a ferramenta e adapta-la às suas necessidades criando os seus próprios fluxos de desenvolvimento e uma liberdade maior quando comparada com outras soluções disponíveis.

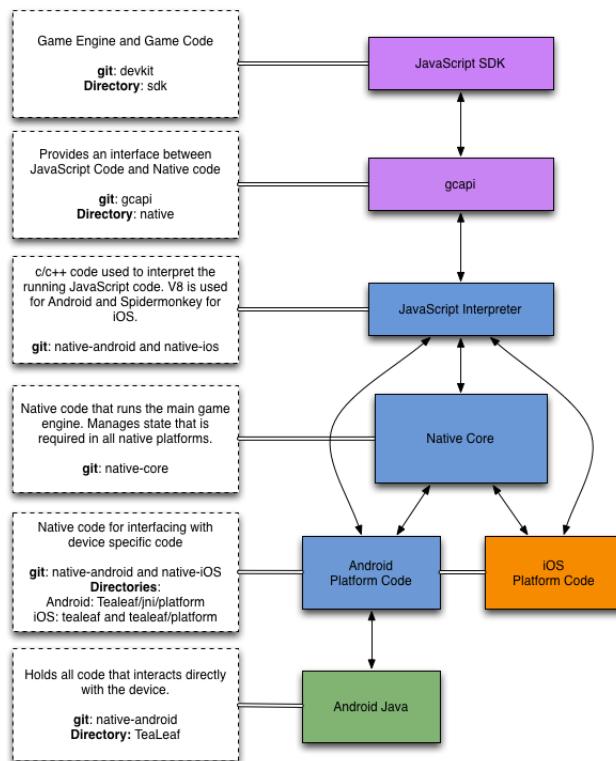


Figura 13 – Game Closure, diagrama de comunicação entre componentes

2.3.2.3.1 Inicialização de projeto

A ferramenta implementa uma estrutura de controlo via consola sendo utilizado para gerir dependências, iniciar projetos e gerir todo o ambiente de desenvolvimento uma ferramenta construída em *nodeJS* (NodeJS, 2015). Após instalada a ferramenta para dar inicio a um projeto é utilizado o comando,

```
$ devkit init [APP NAME]
```

O comando anterior cria uma nova pasta no sistema de ficheiro com o nome da aplicação a criar. Para iniciar o jogo criado é possível correr o comando,

```
$ devkit serve
```

Por defeito este comando inicia um cliente web e abre o navegador permitindo ao programador uma previsualização bastante fiável do jogo que está a desenvolver.

2.3.2.3.2 Estrutura de pastas

Quando inicializado um projeto a estrutura criada pela ferramenta é a seguinte:

```
├── manifest.json
├── modules/
│   └── devkit-core
├── resources/
└── src
    └── Application.js
```

No ficheiro manifest.json podemos encontrar as configurações de base do nosso jogo como o nome, o título e orientações de ecrã disponíveis caso se trate de desenvolvimentos para plataformas móveis.

```
{
  "appID": "adae4be5ggrtldr656t4g4gerg",
  "shortName": "omeujogo",
  "title": "O meu Jogo - ISEP MEI",
  "supportedOrientations": [ "portrait" ]
}
```

2.3.2.3.3 Estilo de desenvolvimento

A ferramenta Game Closure quando comparada com ferramentas como as referidas acima apresenta uma estrutura mais familiar para programadores, sendo que segue um sistema de desenvolvimento baseado em código, deixando o ambiente de desenvolvimento encarregue ao próprio programador, estando disponíveis classes pré definidas para a criação de cenas de jogo, designadas por *Views* com suporte para gestão de eventos de *input*.

2.3.2.3.4 Distribuição

Para criar um ficheiro compilado é executado o seguinte comando:

```
$ devkit release native-[PLATFORM]
```

Corrido este comando é criado na pasta ./build o respetivo ficheiro passível de ser executado na plataforma escolhida.

2.3.3 Conclusão de análise das ferramentas

Das ferramentas analisadas foram consideradas questões tecnológicas que permitissem prestar uma visão técnica avançada dos mais poderosos motores de jogos atualmente disponíveis e que permitem criar conteúdos para as plataformas disponíveis. Em última análise dificilmente será possível indicar uma ferramenta sobre qualquer outra visto todas terem fatores que são mais positivos e outros nem tanto, devendo essa decisão ser tomada de encontro aos requisitos funcionais e possivelmente mais relevante, aos requisitos não funcionais do projeto a desenvolver.

Considerado o âmbito deste projeto é nos possível realizar uma avaliação das ferramentas apoiando a sua adequabilidade ao contexto de programadores independentes com formação na área da engenharia informática e experiência no desenvolvimento de software. Assim sendo podemos começar por analisar a ferramenta Unity3D que demonstra acima de tudo ser uma ferramenta que integra todas as componentes de execução de um jogo num só, permitindo não só a criação do software mas também a edição e criação de ficheiros de jogo, integrados na ferramenta e mantendo desde logo um fluxo formal e próprio focado na produtividade e qualidade. É em si uma ferramenta bastante completa permitindo a decomposição do jogo por ecrãs abstraindo (não que totalmente) os criadores da experiência de questões mais técnicas sobre o desenvolvimento do software. Outro fato relevante é o ser uma ferramenta integrada que acaba por ter uma curva de aprendizagem bastante alargada e que não permite a transposição de conhecimentos de forma muito abrangente quando comparada com outras ferramentas mais nativas face ao desenvolvimento do software. Apesar de simplificada a GUI da ferramenta apresenta um alto nível de maturidade sendo necessária formação e podendo tornando-se frustrante para programadores experientes sendo que o número de horas passadas a arrastar e a preencher caixas com propriedades de objetos cria uma experiência de desenvolvimento pouco flexível e obrigando os programadores a inteirarem-se de inúmeros de conceitos que não são transponíveis para fora da ferramenta, criando uma certa dependência da mesma a médio longo prazo, como por exemplo a utilização de scripts enquanto componentes de cena do jogo que são alocados a objetos de jogo. Se o plano for desenvolver jogos com ambições comerciais a vir a ser um grande título de destaque na categoria AAA e esta a ferramenta a utilizar durante um longo período de anos assumindo as dependências da mesma sobre as decisões da empresa que detêm o software as atualizações e momentos para as realizar, bem como pagamento de todas as licenças e *feeds* acordados é uma excelente ferramenta que pode, como já foi provado, ser o motor de um negócio de sucesso para muitos programadores independentes. Por outro lado falamos de ferramentas que são à partida menos ‘presas’ a uma estrutura ou fluxos de desenvolvimento como a Cocos2D-x. Esta ferramenta não deve de todo ser considerada se o propósito do jogo a construir for um AAA, visto esta ferramenta não disponibilizar opções para compilação de jogos para formatos de consolas onde esse tipo de jogos são dominantes. Confrontada com a existência de uma ferramenta como o Unity3D e todo o seu arsenal de componentes, Cocos2d é uma *framework* de desenvolvimento mais focada no desenvolvimento de software direcionado para pessoal com conhecimentos de desenvolvimento e engenharia de software mais avançados, sendo a base da sua

programação C++, uma linguagem que ainda poderosa afasta até alguns programadores experientes face à sua dimensão e complexidade. Por outro lado, recentemente a equipa de desenvolvimento da ferramenta tem vindo a aproximar a ferramenta de públicos não tão especializados vindo a serem realizados testes sobre a utilização de elementos 3D nos jogos desenvolvidos pela ferramenta, característica já disponível a partir da versão 3 da ferramenta, onde foram introduzidas capacidades de rasterização de modelos 3D. Apesar de ainda ser algo em desenvolvimento considerando a arquitetura baseada na norma OpenGL e OpenGL ES têm sido demonstrado que é algo perfeitamente viável, alargando o espectro de possibilidades da ferramenta. Analisando ainda o *Devkit* da empresa Game Closure, esta apresenta limitações, sendo uma ferramenta mais crua face à sua maturidade de funcionalidades, por outro lado apresenta uma estrutura bem documentada e altamente extensível comportando uma mais fácil transposição de conhecimentos para quem está tecnicamente envolvido em ambientes de desenvolvimento baseados em HTML5, permitindo o rápido desenvolvimento de produtos por programadores mais experientes ou que pretendam apostar numa estrutura que seja a base da sua própria ferramenta de desenvolvimento. Em última análise a ferramenta acaba por ser mais exigente do ponto de vista de conhecimentos de software não sendo aconselhada a indivíduos com poucos conhecimentos na área de programação.

Comparativamente como já referido é difícil à partida indicar a ferramenta ideal visto esse fato estar dependente dos requisitos do projeto e recursos disponíveis mas é inequívoco, e face aos sucessos que ambas as ferramentas destacam, que são passíveis de suster as necessidades do desenvolvimento de projetos multimédia. Na prática todas as ferramentas apresentadas são capazes de produzir produtos de qualidade. Em conclusão podemos ainda colocar em análise os recursos financeiros, que acabam inevitavelmente por caracterizar as equipas de desenvolvimento independentes, aí onde as *frameworks* Cocos2d ou o *Devkit* ganham destaque pelo fato de não acarretarem custos de licenciamento de qualquer natureza.

2.3.4 Media nos jogos e criação de conteúdos

Até aqui analisamos as ferramentas e conceitos base dos vídeo jogos e como referido um jogo é não mais que um tipo de software com característica próprias de apresentação e manipulação de elementos de media (Siapera, 2012). Media é definida como um dos meios ou canais de comunicação gerais de cariz informativo ou de entretenimento na sociedade, como jornais, rádio, televisão ou internet e mais recentemente aberta a discussão de que os jogos de vídeo devem ou não considerados media é algo ativamente discutido entre especialistas defendendo que consideradas todas as característica atualmente existentes nos vídeo jogos, como canais de comunicação de publicidade e de informação, aspetos de integração de redes sociais e até considerado por muitos uma nova forma de arte, realçam nos vídeo jogos similaridades atribuídas aos media tradicionais reforçando esse ponto de vista (Golding, 2012). Não é de todo verdade que todos os jogos contêm essas características daí ser ainda imensamente discutido esse tema. Alguns jogos conseguem ser imensamente desafiantes sem conter grandes conteúdos artísticos e elementos gráficos, como jogos de tabuleiro, que

mesmo assim apresentam-se como jogos altamente populares sem dar grande ênfase a aspectos visuais e sem apresentação de conteúdos de áudio.

Contextualizando é indiscutível que os vídeo jogos criados hoje em dia dificilmente terão qualquer relevância se não forem consideradas questões como o online e aspectos sociais, sendo essa uma das maiores forças da sua distribuição, não colocando de parte que face a uma gigante concorrência e luta por lugar de destaque, coloca sobre os criadores de jogos o dever de apresentar conteúdos visuais e de som de qualidade que permitam aos seus jogadores absorver experiências de jogo de forma mais rica e capaz de transportar os jogadores entre emoções, e nada o faz, como amplamente estudado por especialistas da área da neurociência e psicologia humana, melhor que os mais nativos métodos de comunicação humana como a visão e a audição (Crawford, 1982). O mercado das aplicações de desenvolvimento de conteúdos visuais e de som é um mercado já bastante consolidado, mas em constante mudança e adaptação, sendo hoje possível encontrar inúmeras soluções bastante completas sem qualquer custo para os utilizadores, sendo essa uma preciosa ajuda para que independentes possam criar os seus conteúdos sem custos para além da mão-de-obra humana e criatividade inerente ao processo. Analisadas algumas soluções, apresentamos abaixo uma seleção que mantém a particularidade de serem gratuitas mantendo o alto nível de qualidade para que possam ser criados conteúdos apelativos e representativos da criatividade dos artistas envolvidos nos projetos.

2.3.5 Gráficos

Primariamente analisaremos algumas soluções para criação de conteúdos visuais como imagens e animações 2D e 3D.

2.3.5.1 GIMP

Acrônimo de *GNU Image Manipulation Program*, o GIMP é uma ferramenta de manipulação e criação de conteúdos gráficos 2D grátis e de código fonte aberto que permite a edição, desenho e aplicação de múltiplas transformações a imagens. O seu sistema é baseado em gráficos rasterizados, que significa que o sistema é orientado a transformações sobre os pixéis das imagens (Scratchapixel, 2015). Na prática é um sistema que permite a criação de conteúdos gráficos de qualidade geralmente bastante detalhados, mas que sofrem deformações visuais quando redimensionados, particularmente quando é realizada uma operação de aproximação.

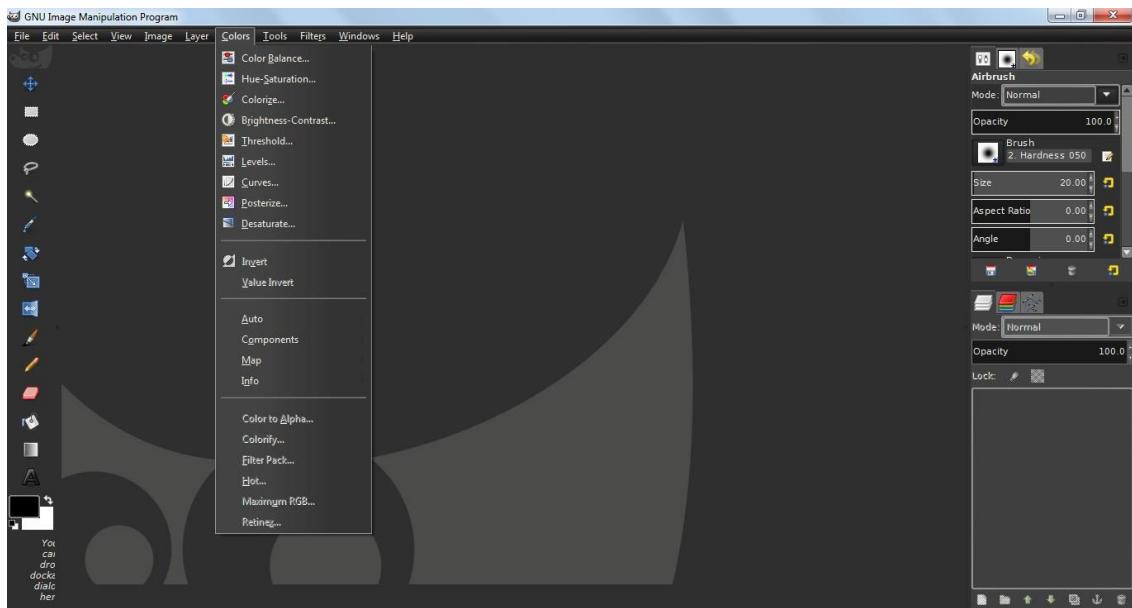


Figura 14 – Gimp

A ferramenta foi originalmente criada em 1995 tendo a sua primeira versão sido disponibilizada ao público em 1996 (Gimp, 2015). Hoje e após longos anos de contínua evolução é uma ferramenta capaz de se debater com os seus mais diretos rivais comerciais, concretamente com o Photoshop, a popular e poderosa ferramenta da Adobe (Adobe, 2015). Algumas das principais características do GIMP são o seu sistema de desenho e pintura, disponibilizando paletas de escolha de cores e pincéis, utilização de sistemas de camadas e canais de cores. Disponibiliza ainda um sistema integrado de criação de animações baseadas em *frames* permitindo a utilização de camadas e *Onion Skins* nesse processo e ainda a manipulação e gestão de ficheiros nos mais utilizados formatos existentes incluindo importação e exportação de imagens para formato SVG (Scalable Vector Graphics) (SVG, 2010).

2.3.5.2 Inkscape

Inkscape é também uma poderosa ferramenta de uso gratuito que permite a criação e manipulação de gráficos, este no formato *Scalable Vector Graphics*, significando isto que este internamente implementa a normalização W3C baseada na normalização de XML como formato nativo para representação gráfica computacional (Inkscape). Comparativamente à ferramenta anteriormente mencionada a grande diferença é a de ser possível trabalhar na criação de conteúdos os quais não sofrem perdas de qualidade quando redimensionadas pelo fato de serem criadas de forma dinâmica que utiliza um sistema de formas primitivas para desenhar no ecrã.

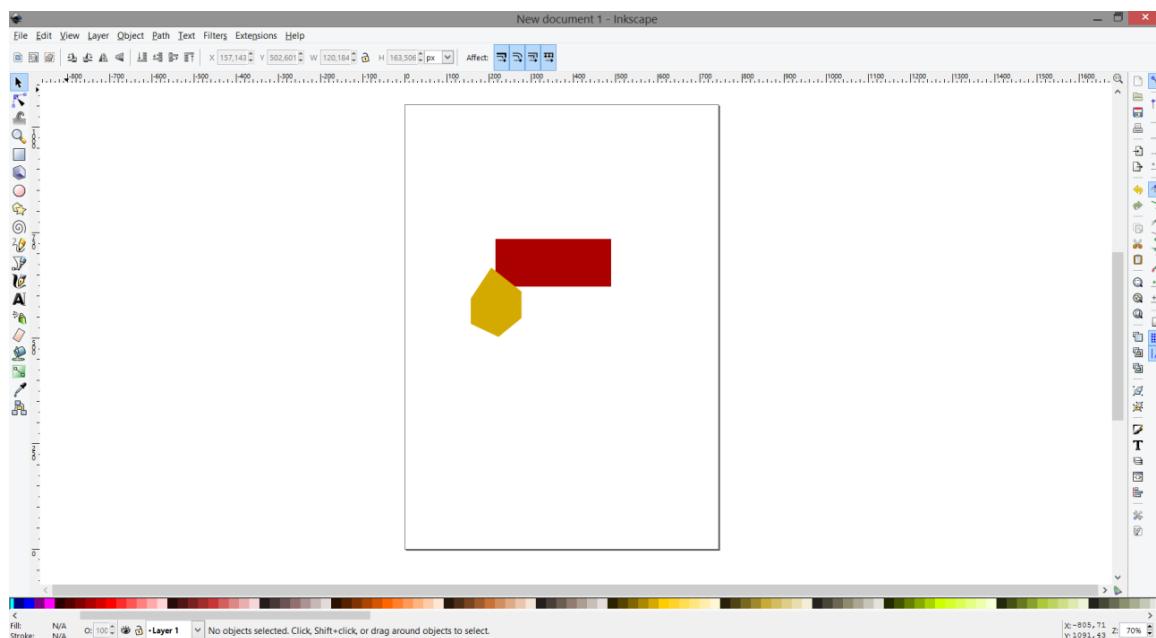


Figura 15 – Inkscape, painel principal

No cenário de criação de jogos este tipo de ferramenta permite a criação de imagens de forma a serem depois exportadas para formatos rasterizados em múltiplas resoluções mantendo um nível de qualidade exigido.

2.3.5.3 Blender

Criada pela fundação Blender esta ferramenta multi propósito é também de uso inteiramente gratuito e de código fonte aberto permitindo a criação de conteúdos 3D.

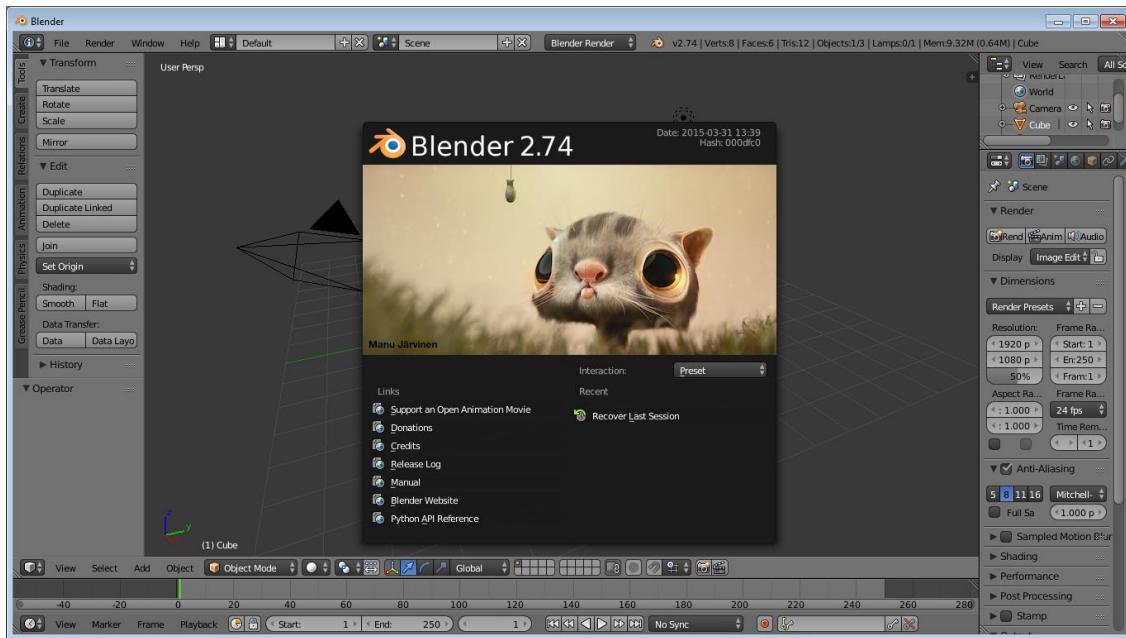


Figura 16 – Painel de ínicio da aplicação Blender

Este poderoso editor 3D disponibiliza um vasto leque de funcionalidades incluindo um rasterizador 3D de alto desempenho capaz de utilizar o hardware disponível de forma excepcional criando cenas foto realistas com suporte de HDR e pré visualização em tempo real.

Inclui ainda um conjunto de ferramentas que apoiam os seus utilizadores na modelação de objetos 3D de forma rápida e intuitiva, permite a aplicação de materiais realísticos sobre modelos 3D, um avançado sistema de *rigging* que permite aplicar estruturas esqueléticas sobre objetos de forma a agilizar processos de animação com suporte de um conjunto de ferramentas para o efeito.

Outras características incluem ainda uma interface com utilizador bastante avançada e intuitiva, a exportação de imagens, vídeos e modelos 3D nos mais variados formatos, edição de vídeo e integração de elementos 3D em cenas, criação de jogos e de extensões para a aplicação e integração da API da ferramenta com outras aplicações, uma avançado sistema de aplicação de texturas e pintura direta sobre objetos 3D modelados e muito mais (Blender, 2015).

2.3.6 Som

Outra componente fundamental numa experiência de jogo de computador é o som de forma a que o jogador possa sentir uma ligação natural à experiência. A criação e captação de sons é em si uma tarefa complicada e ligar todos os elementos de som pode ser uma tarefa morosa e de difícil execução. Considerando isto existem algumas ferramentas que permitem não só a criação de sons como a mistura de sons captados e criados.

2.3.6.1 Audacity

Audacity é uma ferramenta ainda que simples na sua apresentação, bastante capaz na edição, mistura e transformação de elementos de som previamente captados. Disponibiliza um sem número de funcionalidades capaz de permitir transformações a elementos de som existentes como a redução de ruído e eco por exemplo. A aplicação permite ainda exportar as composições para os mais comuns tipos de ficheiros de áudio passíveis de serem importados para projetos multimédia (Audacity).

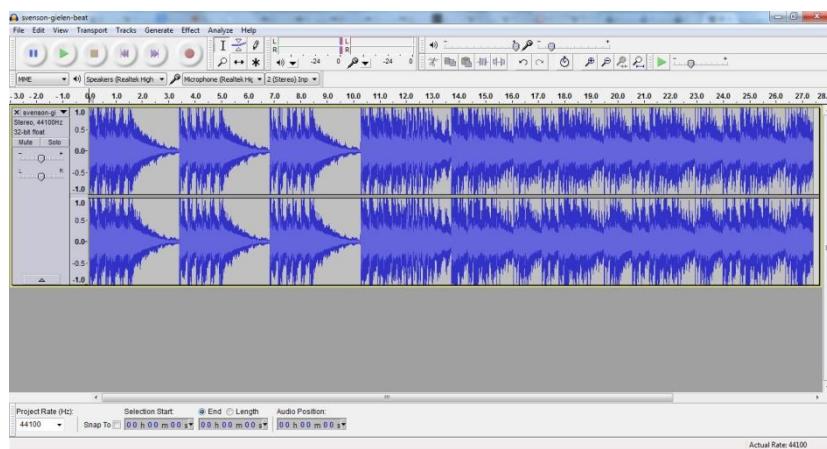


Figura 17 – Audacity

2.3.6.2 AudioSauna

Esta fantástica ferramenta de última geração que permite a criação de áudio diretamente através de um navegador. Permite captar áudio, editar e misturar efeitos de áudio e disponibiliza ainda um sintetizador virtual analógico e um modulador de frequências. É uma ferramenta 100% grátis e funciona *online* permitindo ainda a exportação de áudio para o formato de ficheiro WAV.

As ferramentas acima para além de permitir a criação de gráficos e componentes de áudio permitem a criação de animações fundamentais para que os vídeo jogos sejam altamente apelativos. Criadores de jogos independentes têm hoje ainda disponíveis, acesso a milhares de conteúdos através da Internet, alguns deles restritos para utilização não comercial e outros completamente grátis que apoiam os criadores na busca de artistas que sejam capazes de representar visualmente os conceitos que pretendem transmitir, o que em si é um elemento altamente facilitador e motivador para o estímulo de criatividade e relação cinegética entre as áreas profissionais necessárias para a realização de um projeto de jogo de vídeo (*Audiosauna*).



Figura 18 – Visão painel principal da aplicação AudioSauna

2.3.7 2D versus 3D

Apesar das ferramentas disponíveis não é nunca demais salientar, que se a perspetiva é criar jogos e fazer disso um caminho profissional, é da mais alta importância que desenvolver jogos é difícil (Blow, 2004)! Por essa razão e analisando o percurso de alguns dos mais proeminentes criadores de jogos dos dias de hoje como Jonathan Blow, criador de Braid (Braid, 2006) ou a equipa ‘Play’ que criou o famoso jogo Limbo (Limbo, 2010), vêm reforçar que o desenvolvimento de jogos 2D é o formato mais adequado para equipas pequenas, independentemente do nível de experiência dos intervenientes, sendo importante reter que o desenvolvimento de um vídeo jogo com expetativas de penetração de mercado , por muito pequeno que seja, implica um enorme esforço e conhecimentos de inúmeras áreas, e que por muitos recursos que sejam atribuídos a áreas de criação de conteúdos, design do jogo, áudio etc. A maior e mais difícil parte de desenvolvimento de um jogo é a sua engenharia de software que aumenta a sua complexidade de forma exponencial (Naser, 2015). Considerando estes fatores e seguindo exemplos de sucesso, a escolha pelo desenvolvimento de jogos 2D ou em alguns casos 2.5D é apresentada como a solução mais realista e viável de ser implementada por pequenas equipas. Para desenvolver jogos 3D são necessários conhecimentos mais avançados da parte de programação e criação de componentes gráficos o que cria imediatamente um problema quer para encontrar tais recursos quer relativamente aos custos que esses comportam ainda aumentando em muito o tempo de desenvolvimento do projeto (GameTrailers, 2011). Outro fator em contexto de equipas de desenvolvimento independentes, refere-se à como referida acima, concorrência no mercado *mainstream* de jogos 3D que é incomportável para qualquer equipa de desenvolvimento independente que não tenha à sua disposição recursos financeiros consideráveis. Com a migração na década de 90 para as plataformas 3D a maioria das editoras abandonaram o 2D deixando hoje essa fatia de mercado quase exclusivamente dedicada a criadores independentes, que têm nos últimos anos, e com o aparecimento e massificação dos smartphones e tablets, ocupado essa fatia de mercado, demonstrando que existe bastante espaço para criadores independentes utilizarem

a sua criatividade e capacidade técnica para sustarem uma contínua necessidade de mercado que simplesmente não é atraída para os grandes títulos hoje disponíveis nas mais avançadas consolas de jogos. Outro fator importante é a sua distribuição nas mais variáveis plataformas sendo tecnicamente muito mais simples o desenvolvimento de jogos 2D para multiplataformas de distribuição estando o suporte para *render* de conteúdos 2D altamente suportado e apoiando o desenvolvimento na depuração e testes de integração relacionados com especificações técnicas de *hardware* dos dispositivos capazes de reproduzir aplicação (Rampant, 2013).

Em conclusão parece-nos ser muito mais relevante para uma equipa independente o desenvolvimento de uma experiência de jogo focada no jogador nas regras do jogo e no aspeto mais artístico e social, e menos focada na criação de experiências tecnicamente muito avançadas, com a utilização de texturas de grande detalhe, plataformas de jogo ‘Open World’, sistemas de jogo em tempo real e outras situações que consomem recursos insustentáveis para a maioria das equipas, e que rapidamente podem criar situações de desvio do plano original colocando em causa todo o projeto. A aplicação destas recomendações, ainda que a ser consideradas dependendo dos recursos disponíveis, tendem a garantir um cenário mais otimizado para pequenas equipas e *startups* em que o desenvolvimento deve ser apoiado por equipas pequenas multidisciplinares onde a gestão do projeto, a responsabilidade de cada elemento e o foco num produto pequeno e polido são receita para resultados sustentáveis (Naser, 2015).

2.3.8 2.5D, um novo estilo de jogo

Referido acima este é um novo estilo de jogo que se encontra entre o 2D e o 3D. Consiste essencialmente na utilização de elementos 3D na composição das cenas de jogo mas limitando toda ação de jogabilidade ao plano de dois eixos (x, y). Apesar de não ser um género na realidade tão novo como falado (alguns jogos mais antigos já utilizavam a técnica de jogos de perspetiva isométrica) têm em particular ganho notoriedade em desenvolvimentos de jogos de vídeo direcionados para plataformas móveis. Existem já bastantes títulos que aproveitam o estilo de forma excepcional, de tal forma que deu origem a um motor de jogo criado pela famosa editora Ubisoft designado por UbiArt Framework inteiramente focado neste género de perspetiva e que infelizmente é de carácter proprietário e apenas disponível para a editora e estúdios com a qual a mesma trabalha (Ubisoft, 2010). Alguns dos títulos lançados mostram a potencialidade da ferramenta e do género que rapidamente e graças às características artísticas apresentadas por jogos polidos e altamente apelativos a um público-alvo que ganha cada vez mais importância na indústria. Jogos como Rayman Legends (HardwareHeaven, 2013), chegaram rapidamente a vendas de milhões de cópias e no caso de Trine (Trine, 2014) que começou como um título de um equipa independente e rapidamente atingiu os mercados mais exigentes das consolas garantindo um lugar na história dos vídeo jogos.

2.4 O desenvolvimento de jogos em Portugal

Com início nos anos 80 deu-se a entrada de Portugal na indústria dos vídeo jogos, cerca de 20 anos depois das primeiras experiências internacionais, e hoje tal como na altura, apesar da diferença de recursos disponíveis, mantém-se como uma indústria residual no nosso país, que apesar de algumas tentativas de criação de jogos executadas com modesto sucesso, dessas, poucas foram distinguidas de forma efetiva no mercado internacional destacando-se o *Under Siege*, criada pela extinta Seed Studios, como o maior título alguma vez desenvolvido no nosso país que atingiu o custo de produção de 1.4M€ chegando a ser destacado como um dos melhores jogos de programadores independentes do ano de 2011 pela *Sony Entertainment* para a sua consola, a *Playstation3* (Moriarty, 2011).

Na realidade a última década e meia foi uma autêntica montanha russa para a indústria dos jogos em Portugal e enquanto alguns festejam uma indústria em franco crescimento no nosso país, entidades, como o maior sítio da web de notícias de negócios de jogos, Gamasutra.com, apontam como residual e infantil o estado nacional face à competição do mercado internacional. Segundo Mike Rose, editor do sítio, identifica que apesar do crescimento da indústria e aparecimento de cada vez mais criadores de jogos no país, a cena de desenvolvimento de jogos em Portugal continua tão invisível como sempre foi. Paulo Gomes CEO da Bigmoon Interactive Studios, viu a indústria crescer e ganhar força de forma a ser reconhecida, mas pouco tempo depois e contra todas as expetativas acabou por desfalecer (Rose, 2013). Em 2004 a cena dos jogos não existia em Portugal mas o interesse na indústria começou a surgir, e foram criados os primeiros estúdios dedicados inteiramente ao sector tendo em 2005 surgido o APROJE onde foram criados os primeiros cursos de formação específica para desenvolvimento de jogos no país e organizando ainda as primeiras conferencias dedicadas ao tema alguma vez realizadas em território nacional. Um ano depois fora criada a GameInvest, fundação dedicada ao financiamento de startups focadas no desenvolvimento de estúdios dedicados ao desenvolvimento de títulos próprios e por momentos a sensação de que a cena do desenvolvimento de jogos tinha finalmente arrancado em Portugal invadia a cena do desenvolvimento de jogos, resultando desse movimento alguns dos ainda falados grandes nomes da indústria de jogos em Portugal como a Seed Studios, hoje Camel101, Bigmoon Studios, MadPuppet entre outros que com a crise financeira conseguiram acabaram por ser encerradas, como aconteceu com a Seed Studios e a própria GameInvest (EA, 2007). Felizmente alguns dos estúdios sobreviveram, voltando a entrar em cena com projetos mais pequenos identificando-se como criadores independentes mais focados nas plataformas móveis, como a Nerd Monkeys e a Battle Sheep Studio, conseguindo o país ainda atrair alguns dos estúdios mais importantes da indústria a nível internacional como a Mini-Clip que abriu um estúdio em Lisboa mantendo viva a cena da criação de jogos no país (Miniclip, 2015). Segundo Filipe Pina, CEO da Nerd Monkeys, a indústria em Portugal não tem atualmente qualquer chance de vingar em Portugal dado que o país não tem condições para reter talentos, acabando estes, por deixar o país em busca de oportunidades onde sintam valorizadas as suas capacidades, apontando como falta de liderança e de projetos interessantes como principal causa (Bonifácio, 2015).

Recentemente com a ascensão dos mercados digitais a cena de desenvolvimento mudou, e Portugal, dados fatores como os baixos custos de vida, espaços de trabalho acessíveis e uma carteira de recursos humanos altamente valorados, parece ser mais do que nunca o local perfeito para se ter sucesso nesta indústria (Guerra, 2014). Nos últimos anos o interesse na aposta na educação de profissionais para sustentar entre portas essa indústria e explorar alguns mercados onde Portugal tem uma vantagem competitiva, como o mercado do Brasil, apontado como um dos mercados onde a previsão de crescimento da indústria é das maiores esperadas dos próximos anos, devido ao número de dispositivos a serem vendidos e o fato do acesso à internet estar cada vez mais disponível e adaptado a nível de preços à realidade do país com mais de duzentos milhões de pessoas e atualmente com mais de 50 milhões de utilizadores de ligações de dados (Carmona, 2015).

Apesar de tudo vários programadores de jogos independentes continuam acreditar que a indústria tem futuro em Portugal e que é possível colocar o desenvolvimento nacional em destaque internacional, sendo que o trabalho árduo e a perseverança são apontados como as mais fortes armas que definem os profissionais do nosso país (Suckley, 2015).

3. Caso experimental

Neste capítulo é apresentado todo processo de desenvolvimento de um vídeo jogo em contexto de experimentação e processo de aprendizagem com objetivo de promover a partilha de experiências e apresentação de dados recolhidos, ambicionando que esta partilha seja capaz de elucidar algumas questões mais práticas do desenvolvimento e publicação de um jogo de vídeo. Indiscutivelmente o produto final, em última análise, e dado o contexto de produção de um jogo independente, acaba sempre por espelhar a criatividade e visão dos executantes da ideia, particularmente em relação ao aspeto gráfico do mesmo e decisões de uso das ferramentas de desenvolvimento e plataformas de distribuição alvo, representando o produto criado uma visão do seu executante em prova de conceito do analisado neste documento.

3.1 Relação entre estado da arte e caso experimental

Todo o estudo apresentado no capítulo anterior foi fundamental para a execução do caso experimental descrito no presente capítulo, e apesar de não ser visível de forma clara, existe intrinsecamente uma forte ligação entre todo o trabalho de pesquisa executado e o produto criado, inclusive sobre a ordem de como os tópicos estão organizados, de forma a permitir ao leitor um avanço incremental de conhecimentos sobre os temas.

O estudo dos mais básicos princípios do que é um jogo e a análise aprofundada sobre aspectos mais formais desse tipo de produtos de software revela-se fundamental para que se possa avançar de forma clara e séria para o desenvolvimento deste tipo de projeto, de modo a que não seja, como geralmente o é, entendido como algo menor quando comparado com outras soluções de software ditas mais sérias e maioritariamente focadas em lógicas de negócio. Assim sendo, foi considerado da maior relevância e dentro do contexto formal do âmbito deste projeto, que este estudo fosse realizado, no sentido de estimular sensibilidades sobre a importância dos jogos e efeitos que este tipo de experiências tem nas pessoas. Os jogos

devem ser, e à luz de todos os dados e informações apresentadas, levados muito a sério, e essa informação e sensibilidade deve, mesmo que não espelhada de forma transparente no projeto desenvolvido, estar bem presente na mente de todos os envolvidos na produção deste tipo de software.

Sensibilizados para a relevância dos jogos na sua relação com as pessoas foi ainda da mais alta importância clarificar o posicionamento de programadores independentes na sua relação com a indústria, mantendo a relação de que a motivação para desenvolver jogos, apesar de puder ser puramente lúdica, não sendo de todo o foco deste documento, é também financeira, considerado que este fato é determinante para que profissionais e potenciais investidores dediquem os seus recursos financeiros e humanos a este género de projetos. Ainda à luz desta análise de mercado e posicionamento dos programadores independentes, foi ainda relevante considerar as plataformas disponíveis e adequabilidade das mesmas para projetos realizados por independentes, sendo que como descrito na análise apresentada, é fundamental que a tecnologia não seja fator de limitação e que a escolha de plataformas, mercados e sistemas de distribuição sejam considerados nessas tomadas de decisão, de forma a que tentativas de entrada no mercado não sejam destruídas por incapacidades técnicas ou financeiras de modo a evitar que a oportunidade de criação de produtos e ganho de visibilidade seja perdida.

Todo o conteúdo analisado e fundamentado acaba por revelar-se determinante nas escolhas tecnológicas analisadas, sendo que consolidados os pontos referidos acima, a análise de ferramentas para a produção do software, foi ajustada de acordo com os recursos humanos disponíveis de acordo com a visão e capacidade dos intervenientes na criação de um produto a ser desenvolvido e disponibilizado para as plataformas alvo mais relevantes no contexto atual de programadores independentes, estabelecendo assim as bases necessárias para tomadas de decisão sobre o caso experimental face a plataformas, tecnologias, ferramentas de produção, metodologias e objetivos, decisões estas ainda resultantes da visão e motivação pessoal do autor. Assim sendo enumeramos os tópicos que fundamentaram a tomada de decisão dentro do âmbito proposto e à imagem do analisado, respetivamente:

- Utilização de uma ferramenta multiplataforma com foco nas plataformas móveis e internet;
- Transposição de conhecimentos de programação baseados em tecnologias da internet;
- Exploração de mercados de distribuição da internet e dispositivos móveis;
- Publicação e posterior análise de dados recolhidos.

Para que o descrito fosse atingido e de acordo com as ferramentas analisadas no capítulo anterior foi decidido utilizar uma ferramenta que permitisse o desenvolvimento de um produto de software passível de ser distribuído nas plataformas móveis e internet, de forma a permitir obter experiência na publicação de produtos nas plataformas de distribuição móveis, particularmente a plataforma *Android* que de acordo com os dados mais atualizados recolhidos demonstram que esta plataforma representa a maior fatia de mercado a nível mundial sobre as plataformas móveis disponíveis (IDC, 2015).

Particularmente adequada ao proposto apresenta-se a ferramenta descrita no capítulo anterior, a ferramenta *Game Closure*, que dado ser uma ferramenta baseada puramente em técnicas de programação e conciliadas as competências transversais ao desenvolvimento de software resultam numa mais curta curva de aprendizagem e maior transposição de conhecimentos, sustendo a tecnologia todas as necessidades gerais para a construção de jogos para as plataformas alvo.

Para a produção de elementos de áudio foi utilizada a ferramenta *Audacity*, também referida no capítulo anterior, utilizada ainda para gravação de áudio através de hardware próprio disponível no dispositivo utilizados para desenvolvimento (Surface, 2015), considerando também a utilização de recursos disponibilizados em fontes de distribuição de amostras de áudio gratuitos e livres de licenças em portais como *FreeSFX* (FreeSfx, 2015).

Acordado o objetivo de produção de um jogo com característica criativas executadas pelo autor, foi decidido utilizar a ferramenta *Gimp* também descrita no capítulo anterior e ainda a ferramenta *Photoshop*. Ambas apresentam característica idênticas sendo que aproveitando conhecimentos em *Photoshop*, obtidos durante o curso na cadeira de Imagem e Vídeo, revelam um processo mais produtivo e resultados finais de maior qualidade não deixando de referir que foi utilizada uma versão gratuita em modo experimental, requerendo esta para uso profissional a aquisição de uma licença (Photoshop, 2015).

3.2 *Brainstorming*, da ideia ao conceito

Determinadas as questões relacionadas com mercado, aquisição de utilizadores, visibilidade e outras relacionadas com meios de produção, a decisão seguinte recai sobre o design do produto a desenvolver, tendo sido realizada uma pré análise que permitiu uma tomada de decisão assente nos recursos humanos e técnicos disponíveis bem como as metas temporais e objetivos a atingir. Assim sendo e considerando as variantes descritas para tomada de decisão sobre o produto a desenvolver e de acordo com o estudo teórico apresentado foi considerada a criação de uma versão de um jogo que estivesse, à data, a captar o interesse do público de forma a que não existisse necessidade um esforço significativo para criar e educar potenciais jogadores, reduzindo risco de não conseguir obter um número mínimo de jogadores que permitissem obter uma quantidade de dados significativos para análise. Sucintamente o modelo de jogo a desenvolver deve seguir os seguintes conceitos:

- Modelo de jogo provado de forma a reduzir o nível de educação e promoção necessária para obter utilizadores;
- Jogo que apresente características de evolução e cenários de aprendizagem por repetição de tarefas e ajuste de ações por parte do jogador;
- Modelo de jogo capaz de ser plataforma de prova de potencial criativo sobre criação de conteúdos multimédia como imagem, áudio e animações;

Avaliadas as condições descritas acima fora decidido criar uma versão do popular jogo 2048, de forma a criar uma versão visualmente mais rica e interativa do que a versão original, disponível para as plataformas móveis e ambientes de navegadores de internet.



Figura 19 – Avaliação do jogo 2048 na App Store da Apple

À data de Junho de 2015 o jogo apresentava uma avaliação de 4.5/5 com mais de um milhão de avaliações apenas na App Store da Apple (iTunes, 2015) representando uma comunidade de jogadores significativa. Assim com o jogo proposto é objetivo atingir jogadores que tenham já, ou não, experimentado esta experiência e que possam ser atraídos pelo novo aspecto visual e criativo do jogo desenvolvido.

3.3 Design de jogo

A fase de design de jogo é a primeira fase do desenvolvimento, sendo que nesta fase é explorado o conceito de jogo a desenvolver, as suas implicações em relação a regras, jogabilidade, relação com os jogadores, gestão de expectativas e definição de objetivos (Thorn, 2014).

3.3.1 O jogo: 2048

3.3.1.1 Original

O jogo 2048 consiste numa experiência de mecânica simples que explora os mais simples princípios dos jogos, particularmente aspecto cognitivos de pesquisa e procura de diferentes soluções para problemas similares, numa perspetiva de otimização de resultados (Kosher, 2013). Popularizado em Março de 2014 por Gabrielli Ciruci, o jogo é em si uma versão do popular jogo ‘Threes’ criado pela equipa de desenvolvimento independente constituída pelo designer de jogo Asher Vollmer, Greg Wohlwend enquanto ilustrador e pelo compositor Jimmy Hinson. O jogo ‘Threes’ foi originalmente disponibilizado ao público em Fevereiro de 2014 para a plataforma iOS e mais tarde exportado para Android, XboxOne e Windows Phone tendo sido clonado por Gabrielli Ciruci e lançado em Março de 2014 para a plataforma Android e iOS. Os objetos de jogo consistem muito simplesmente em blocos numerados e dispostos numa matriz 4x4 que podem ser movimentados pelo jogador no sentido vertical ou horizontal.

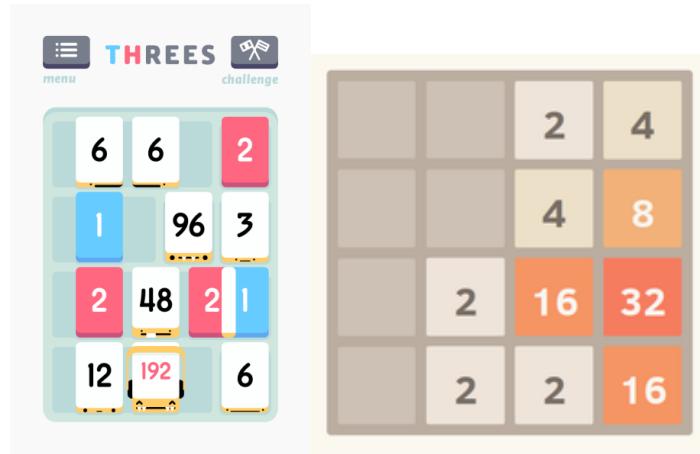


Figura 20 – Ecrã do jogo *Threes* (à esquerda) e *2048* (à direita)

O jogo tem início com a geração de forma aleatória dois blocos com os valores de dois ou quatro e o jogador deve executar a sua vez escolhendo a direção para a qual os blocos devem ser movidos e sucessivamente somados. Quando movidos os blocos deslocam-se até às posições limite da matriz ou até encontrarem outro bloco de valor diferente. Caso durante o movimento o bloco encontre outro bloco de igual valor os mesmos juntam-se adquirindo o valor da soma dos dois não sendo possível esta operação acontecer mais do que uma vez por jogada e por bloco movimentado. O jogo apresenta ainda um sistema de atribuição de pontos que se refletem na soma do todos os valores de blocos somados durante um jogo, sendo uma meta do jogo o atingir da soma de blocos que perfeçam o valor de 2048, podendo ainda o jogador optar por recomeçar ou continuar a jogar tentando atingir resultados finais superiores.

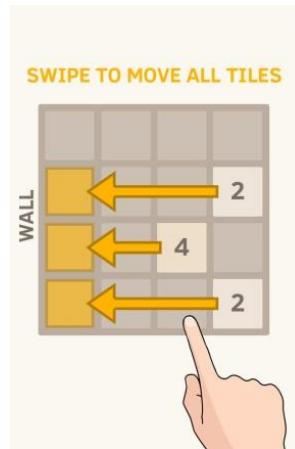


Figura 21 – Ilustração de regras do jogo *2048*

Objetivamente o jogo explora a vontade dos jogadores de obterem resultados cada vez melhores permitindo ainda através da existência de quadros de honra a comparação do seu resultado com o de outros jogadores, criando no jogador a sensação de que é possível fazer sempre melhor, promovendo competitividade como característica de socialização.

A mecânica simples e fácil de aprender revelou ser receita para enorme sucesso obtendo em particular o original jogo *Threes* aclamação mundial sendo galardoado com as mais altas pontuações atribuídas a jogos do género nas plataformas móveis (Eurogamer, 2014).

3.3.1.2 Sticky 2048

A versão criada como prova de conceito consiste numa versão de mecânica similar ao referido 2048 e *Threes* com objetivo de explorar uma maior faceta de criação de conteúdos multimédia e desenvolvimento de contexto sobre a mecânica base de jogo.

De forma similar o objetivo do jogo consiste na obtenção de um resultado o mais alto possível contextualizando um conjunto de personagens que objetivamente criam uma relação com o jogador de forma a promover o conceito de que o jogador não está apenas a jogar na prossecução de obter um resultado elevado mas também na ajuda às personagens, de forma promover o encontrar o seu rei que neste caso se apresenta como resultado 2048 e consequente conquista da vitória.

De forma a manter o desafio do jogo incremental para além do esquema de resultados o jogo apresenta ainda a integração de objetivos intermédios e avançados:

- Encontrar ‘Fernando Pessoa’ ;
- Encontrar um *Viking*;
- Salvar o Rei;
- Encontrar três reis de forma consecutiva;
- Jogar cem jogos.

Mantendo a integridade do design de jogo original o produto criado mantém uma relação com jogador privilegiando o desafio, objetividade, raciocínio estratégico e diversão.

3.4 Visão global

3.4.1 Esquema de navegação

O esquema de navegação segue uma lógica intuitiva de modo a que o jogador nunca se sinta perdido na experiência, mantendo sempre uma lógica de promoção do jogo e de retenção do jogador. Considerado o enunciado o esquema de navegação dentro da aplicação é apresentada abaixo.

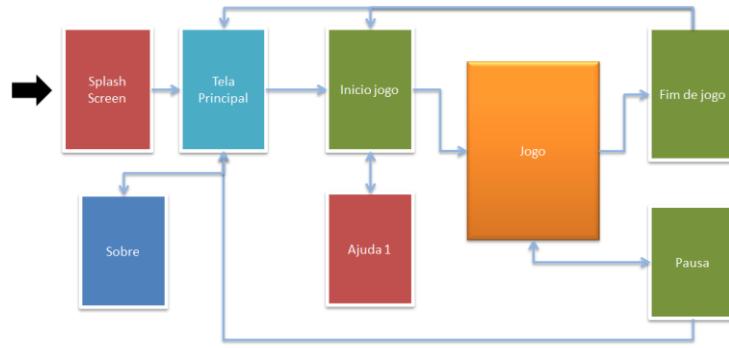


Figura 22 – Visão de esquema de navegação global

A experiência tem como ponto de entrada o *SplashScreen* representando este o ecrã de entrada da aplicação e de carregamento de todos os componentes necessários e constituintes do jogo passando depois para a tela principal do jogo onde o jogador têm a oportunidade de alterar opções relativas a efeitos de som e música, ver informações sobre o jogo e ainda dar inicio a um novo jogo. Depois de iniciado um novo jogo é apresentada uma tela que permite ao jogador escolher a opção de início de jogo ou aceder a uma nova tela sobre as regras do jogo permitindo o mesmo assim voltar à tela de início de jogo e dar efetivamente início a um novo jogo. O jogador pode a qualquer momento optar por pausar a experiência e abandonar o jogo a decorrer ou continuar assim que tiver interesse. A experiência termina assim que o jogar ficar sem movimentos possíveis ou atingir o objetivo final.

3.4.2 Requisitos funcionais

Considerada a necessidade de levantamento dos requisitos funcionais foram desenhados os primeiros rascunhos de como a aplicação deveria ser apresentada aos jogadores de forma ainda a permitir uma maior legibilidade de requisitos funcionais. Assim sendo foram criados os esboços concordantes com o esquema de navegação apresentado acima.



Figura 23 – Esboços gráficos de ecrãs de jogo Sticky 2048

Analizados os esboços criados de acordo com os objetivos da experiência a desenvolver foram revelados os seguintes requisitos funcionais:

- Activar / Desactivar efeitos de áudio e música;
- Ver informações do programador;
- Ligar aos serviços externos de tabelas de resultados e objetivos;
- Iniciar jogo;
- Ver tutorial;
- Pausar o jogo;
- Submeter objetivo;
- Submeter resultado;
- Voltar à tela principal.

3.4.3 Requisitos não funcionais

Tratando-se de desenvolvimento de software é indispensável a avaliação de requisitos não funcionais do produto a desenvolver, sendo que estes acabam por ter um peso muito significativo correspondente a escolhas tecnológicas, definição de interfaces de utilização e gestão de projeto. Assim sendo e considerando o âmbito de desenvolvimento e o projeto analisado como objetivos foram identificados como requisitos não funcionais os seguintes:

- **Usabilidade** – a experiência deverá ser extremamente interativa e fácil de utilizar, devendo a implementação das interfaces com os utilizadores deverá ter em consideração excelente *feedback*;
- **Performance** – o jogo não deve em momento algum, e independentemente da plataforma onde é disponibilizado, apresentar falhas e efeitos gráficos indesejados como ecrãs congelados ou baixas frequências de apresentação de imagens por segundo;
- **Mobilidade** - o jogo deve ser construído de forma a que possa ser executado no maior número de dispositivos possíveis mantendo um nível ubíquo de experiência;
- **Segurança** – os dados dos utilizadores devem ser protegidos de forma a garantir um sentimento de confiança para com os intervenientes na experiência;
- **Disponibilidade** – o jogo deve estar sempre disponível garantindo que mesmo que não exista ligação aos serviços externos possa ser executado.

3.5 Implementação

Após realizada análise de requisitos e considerado o objetivo de criar um jogo que fosse cem por cento funcional no espaço de tempo disponível dadas as competências técnicas da equipa envolvida no seu desenvolvimento, foi selecionada como ferramenta de desenvolvimento, a *framework Game Closure* analisada no capítulo anterior, visto ser uma ferramenta que sistem

as necessidades necessárias para o projeto a desenvolver, promovendo esta, a reutilização de conhecimentos transversais ao desenvolvimento de software.

3.5.1 Gestão do projeto

A implementação do projeto foi baseada em metodologias ágeis de desenvolvimento de forma a que todo o processo fosse baseado em pequenos *sprints* com entregas capazes de serem testadas e reavaliadas de forma rápida e eficaz.

Tratando-se de um desenvolvimento independente os recursos humanos não estão disponíveis como seria ideal, assim sendo os recursos disponíveis acabam por desempenhar múltiplos papéis no desenvolvimento do projeto não sendo irrelevante numa perspetiva orgânica de entrada e saída de recursos num projeto com estas característica e importante para o seu planeamento que sejam definidos os papéis dos recursos necessários.

- **Product owner:** Responsável por conhecer e orientar todo o processo de decisão e produção do jogo. Deve ainda ser capaz de orientar e esclarecer sempre que existam dúvidas sobre o produto;
- **Designer de jogo** – responsável por definir conceptualmente o jogo e em parceria com o *product owner* definir os objetivos e parte relevantes da experiência;
- **Designer gráfico** – Responsável por produzir componentes gráficos da experiência;
- **Programador** – Profissional capaz de desenvolver de forma programática o jogo de forma a que este seja reflexo do idealizado pelo design de jogo;
- **QA** – realização de testes e reportes de resultados para reavaliação do estado do produto antes de se avançar no processo de desenvolvimento e mais importante, antes de se passar à fase de entrega para o público;

Definidos os papéis foi criada uma estrutura de base de trabalho de forma a promover um processo de trabalho fluído capaz de ser o mais otimizado e focado na qualidade e produtividade.



Figura 24 – Fluxo de desenvolvimento e responsáveis por tarefas

Para a produção de design de cenas e programação do código de jogo não existem dependências diretas sendo que podem ser executadas de forma simultânea representando componentes distintas do software privilegiando o desenvolvimento de uma arquitetura de software que promove a separação de responsabilidades (MSDN, 2015).

3.5.2 Programação da componente lógica

Este processo do desenvolvimento consiste na implementação da parte de software referente à lógica do jogo. Compreendendo o jogo 2048 descrito acima, o jogo é programado com a utilização de três classes sendo uma delas relativas à grelha de jogo, outra aos blocos de jogo e ainda uma classe que concentra toda a lógica de gestão de jogo e de blocos dentro de uma grelha de jogo.

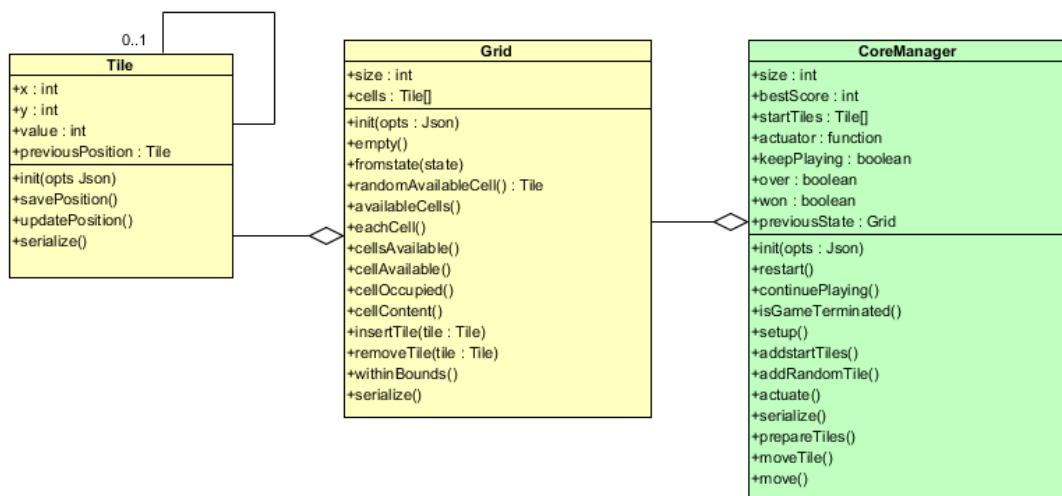


Figura 25 – Diagrama de classes da componente de lógica do jogo

Sendo o jogo desenvolvido utilizando a *framework* Game Closure em que a linguagem de programação é *JavaScript* sobre o paradigma de orientação a objetos.

```

Exports = Class(function(supr) {
    this.init = function(position, value) {
        this.x = position.x;
        this.y = position.y;
        this.value = value || 2;

        this.previousPosition = null;
        this.mergedFrom = null; // Tracks tiles that merged together
    };
    this.savePosition = function() {
        this.previousPosition = {x: this.x, y: this.y};
    };
    this.updatePosition = function(position) {
        this.x = position.x;
    };
})

```

```

        this.y = position.y;
    };
    this.serialize = function() {
        return {
            position: {
                x: this.x,
                y: this.y
            },
            value: this.value
        };
    };
});

```

Código 1 – *Title.js*

Acima temos a representação do código da classe *Title* representativa de um bloco de jogo. Na ferramenta utilizada, o sistema de classes é construído sobre *JavaScript* sendo o seu conceito baseado em protótipos em que os seus métodos são criados através da atribuição de funções a variáveis, revelando esta como uma linguagem funcional. Outra característica da ferramenta é a criação de ficheiros com nome do objeto utilizando para isso a propriedade global *exports*. No exemplo podemos verificar que à propriedade *exports* do ficheiro *Title.js* é atribuída uma instância de um objeto *Class* onde é definido uma função de *callback* com a estrutura de um bloco de jogo de acordo com o diagrama de classes apresentado. Particularmente a *framework* utiliza a função definida *init* como construtor da classe.

3.5.3 Produção de interfaces gráficos

O processo de produção de componentes gráficos teve inicio numa versão de rascunho em papel (ver anexos) tendo sido posteriormente redesenhasadas de forma digital utilizando hardware específico, no caso concreto a utilização de ecrã digitalizador com auxílio de caneta com capacidades de deteção de diferenças de pressão de forma a obter um resultado idêntico ao desenho em papel ou outro meio manual. Todo o processo teve em conta um *pipeline* definido como ideal de acordo com os meios disponíveis, sendo o processo iniciado pelo desenho em esboço da cena a criar, seguido da segmentação da cena em objetos de cena, permitindo depois de forma paralela entregar tarefas de refinamento de gráficos aos artistas e informações relevantes para os programadores da cena de forma a criarem as interfaces e programarem os comportamentos esperados para cada objeto de cena. Esta abordagem permitiu ainda a possibilidade de cada tela ser considerada como uma componente parcial independente permitindo o teste de funcionalidades por tela.

3.5.3.1 Tela principal e opções

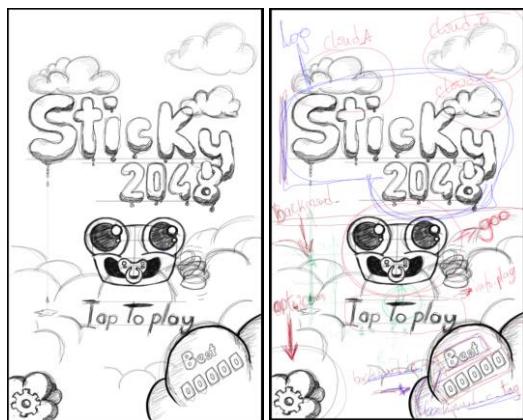


Figura 26 – Esboço de cena da tela inicial anotado

3.5.3.1.1 Descrição de objetos de cena

• Goo

A personagem Goo é a personagem central do jogo sendo estes apresentados de várias formas ao longo do jogo e sendo objetivo do jogador encontrar o seu Rei.

Na cena de entrada do jogo o primeiro Goo apresentado é o Goo Baby sendo este uma composição de vários componentes gráficos criados utilizando a ferramenta Photoshop. A ferramenta Photoshop apresenta a possibilidade de trabalho em camadas sendo possível executar na mesma ferramenta todas as tarefas necessárias desde a fase de rascunho até à fase de criação de animações baseadas em imagens individuais. Utilizando a ferramenta pincel foi desenhado o esboço da personagem,



Figura 27 – Esboço de personagem (Photoshop)

Seguidamente foi realizada uma segmentação da personagem de forma a obter objetos independentes e utilizando a mesma ferramenta foram ilustrados com cores.



Figura 28 – Componentes estáticos do Goo do ecrã inicial

Depois de criadas a componente estática do Goo foi utilizada uma camada de vídeo que permite no Photoshop o desenho e ilustração de imagens individuais em sequência permitindo depois a exportação dos elementos selecionados. Estes grupos de imagens são designados por *sprites* e quando conjugados com os elementos estáticos da nossa personagem permitem criar animações.

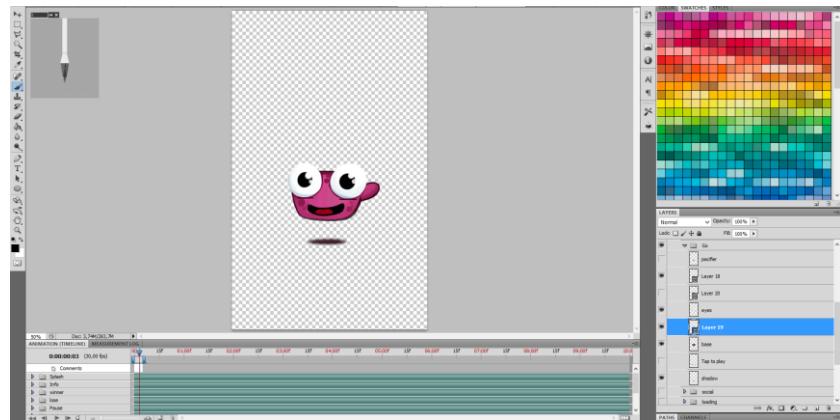


Figura 29 – Animação de personagem em Photoshop

O resultado final da personagem compreende então os três componentes estáticos de base apresentados e três conjuntos de *sprites*.



Figura 30 – Sprites do Goo do ecrã inicial

- **Componentes estáticos da tela inicial**



Figura 31 – Várias imagens da cena principal

Todos os componentes gráficos apresentados foram produzidos utilizando a ferramenta Photoshop e as técnicas acima descritas sendo todos os conteúdos originais e criados especificamente para o jogo Sticky 2048.

3.5.3.2 Tela de início de jogo, jogo e opções

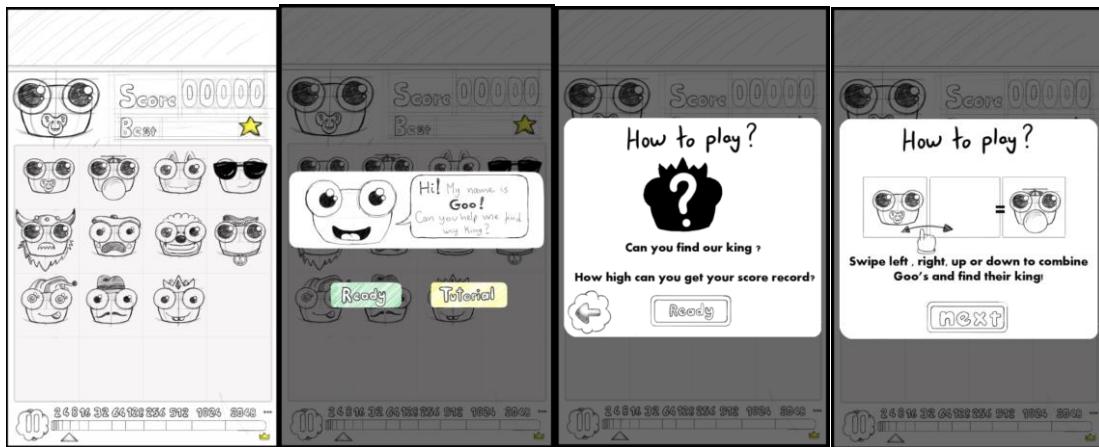


Figura 32 – Esboços de ecrãs de jogo

3.5.3.2.1 Personagens

As personagens do jogo vão sendo reveladas à medida que o jogador vai avançando no jogo e representam individualmente um valor numérico múltiplo de dois. A cada jogada em que um Goo é unido com outro de igual valor, a personagem muda para a seguinte, incrementando para o dobro e sendo esse resultado somado ao valor do resultado total da sessão.

	Valor		Valor		Valor
	2		64		2048
	4		128		4096
	8		256		8192
	16		512	-	-
	32		1024	-	-

Tabela 1 – Personagens e representação numérica

Quando o jogo é iniciado é apresentado um ecrã informativo da posição atual do jogador, permitindo ainda ao jogador carregar num botão onde são apresentadas de forma sucinta as regras do jogo e objetivos.



Figura 33 – Cenas de inicio de jogo e ajuda

De forma a que a experiência de jogo fosse diferenciadora são apresentados regularmente sugestões de movimentos e animações que permitem ao jogador um maior nível de envolvimento sentindo que ojogo responde aos seus movimentos e captando a sua atenção.



Figura 34 – Cenas de jogo (dica e desbloqueio de nova personagem)

Quando um Goo é libertado pela primeira vez é ainda apresentado um ecrã de desbloqueio servindo este mecanismo como forma de prémio e sensibilização para o progresso do jogador no jogo. A utilização de efeitos visuais e de áudio durante o jogo foram criados de forma a garantir uma experiência que não se revelasse enfadonha e estática sendo que quando é executado uma jogada é sempre executada uma ação de carácter visual ou auditivo sendo apresentada uma animação e um som relativo ao resultado da ação do jogador.



Figura 35 – *Sprites* de animações de fusão de personagens

Sendo que o jogador avança no jogo através da soma de valores de blocos de igual valor representativos de personagens iguais, sempre que são somados blocos, são apresentados efeitos designados por partículas e ainda, através de um *sprite* criado para o efeito, é apresentada uma animação de efeito explosão.

Durante o jogo e de forma a manter como referido o interesse e um nível satisfatório de interatividade o jogador é frequentemente presenteado com mensagens de incentivo.



Figura 36 – Imagens de *feedback* durante jogo

Durante a experiência de jogo existem ainda objetivos a serem cumpridos sendo que o objetivo máximo para além do melhor resultado possível é o atingir a personagem do Rei 2048 sendo este o objetivo base do jogo. Esse objetivo é atingido quando o jogador consegue criar um cenário em que a soma de dois blocos de valor 1024 são somados, sendo depois o jogador presenteado com o ecrã de vitória e questionado se pretende continuar a jogar ou reiniciar o jogo.



Figura 37 – Ecrã de vitória

À medida que o jogo vai avançando o espaço de movimentos, face ao fato de a cada jogada ser gerado um novo bloco e adicionado à grelha de jogo, o jogador vai sendo desafiado a pensar numa estratégia de jogo capaz de manter espaços em aberto para que o bloco a surgir não gere uma situação de jogo onde não será possível realizar mais movimentos sendo o jogo terminado nesse cenário.



Figura 38 – Mensagem de fim de jogo

3.5.3.3 Codificação

A framework Game Closure permite de forma programática a representação de elementos gráficos dentro de uma cena designada por *View* ou no caso apresentado uma *ImageView*, representando esta classe um objeto que é apresentado como imagem. A criação da cena consiste então na instância de uma classe do tipo *ImageView* que no caso concreto é instanciada com a imagem de fundo apresentada acima no construtor da classe.

```
Exports = Class(ImageView, function(supr) {
    this.init = function(opts) {
        loop = true;
        this.sound = soundcontroller.getSound();
        opts = merge(opts, {
            x: 0,
            y: 0,
            image: "resources/images/new/titlescreen/background.png"
        });
        supr(this, 'init', [opts]);
    };
    ...
});
```

Código 2 – Construtor da cena Titlescreen.js

Depois de instanciada uma cena que representa uma *View* é possível adicionar objetos filhos da cena numa estrutura de hierarquia sendo que cada objeto gráfico apresentado acima representa uma nova instância com referência à *View* pai.

```
Exports = Class(ImageView, function(supr) {
    ...
    var logo = new Image({url: "/logo.png"})
    this.logoView = new ImageView({
        superview: this,
        image: logo,
        width: logo.getWidth() * scale,
        height: logo.img.getHeight() * scale,
```

```

        x: width * 0.009,
        y: smallScreen? (width * 0.05): (width * 0.25)
    });
    ...
});

```

Código 3 – Exemplo de criação de logotipo na cena principal

Outra característica e fundamental é a gestão de eventos despoletados pelo jogador sendo que a *framework* disponibiliza um sistema capaz de detetar eventos de entrada.

```

Import event.input.InputEvent as InputEvent;
...
this.playView.on('InputSelect', function (evt, pt) {
    // Iniciar jogo ...
});

```

Código 4 – Exemplo de registo de função de gestão de interação com utilizador

3.5.3.4 Efeitos de som e música

Para serem reproduzidos sons como música e efeitos de som a ferramenta disponibiliza uma abstração de um gestor designado por *AudioManager* que permite tocar sons devidamente pré selecionados. Na implementação do jogo foi criada uma estrutura que pudesse ser reutilizada ao longo da aplicação de forma a que a mesma instância fosse reutilizada sempre que necessário.

```

import AudioManager;
exports.sound = null;
exports.getSound = function () {
    if (!exports.sound) {
        exports.sound = new AudioManager({
            path: 'resources/sounds',
            files: {
                swipe: {
                    path: 'effect',
                    background: false
                }
            ...
        });
    }
    return exports.sound;
};

```

Código 5 – Gestor de som da aplicação

Sempre que necessário manipula um determinado ficheiro de áudio é utilizada a função acima descrita para obter acesso à instância e depois chamada a função respetiva disponibilizando algumas funcionalidades básicas de manipulação de áudio com *play*, *pause* e *mute*.

3.5.4 Disponibilização na Google Play Store e na Internet

Considerado o contexto do projeto e objetivo de chegar ao maior número de utilizadores e explorar o mercado das plataformas móveis para além da disponibilização do jogo na internet foi ainda criada uma versão disponível de forma nativa para a plataforma Android, permitindo ainda obter experiência prática do processo. A disponibilização na plataforma Android prevê uma lista de requisitos mínimos de forma a que o jogo possa ser exposto na sua plataforma de distribuição a Google Play Store (REF). Para além do ficheiro de extensão APK é ainda necessário o preenchimento de alguns dados sobre o jogo respetivamente o título, uma breve descrição e uma descrição completa.

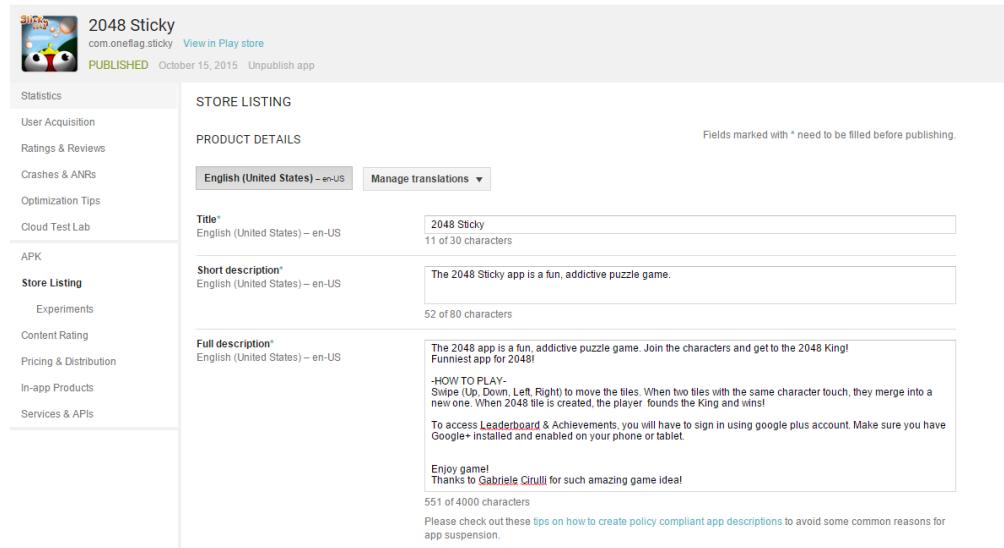


Figura 39 – Listagem de requisitos Google Play Store(1)

Para além da descrição técnica é ainda requisito mínimo submeter algumas imagens do jogo e da experiência de jogo de forma a que possa ser visualmente mais fácil para os jogadores encontrarem e ganharem interesse em fazer download da aplicação depois de visitar a aplicação na loja.

Hi-res icon *
Default – English (United States) – en-US
512 x 512
32-bit PNG (with alpha)

Feature Graphic *
Default – English (United States) – en-US
1024 w x 500 h
JPG or 24-bit PNG (no alpha)

Promo Graphic
Default – English (United States) – en-US
180 w x 120 h
JPG or 24-bit PNG (no alpha)

TV Banner
Default – English (United States) – en-US
320 w x 180 h
JPG or 24-bit PNG (no alpha)

Promo Video
Default – English (United States) – en-US
YouTube video
Please enter a URL.
<https://www.youtube.com/watch?v=4T7rfqkZ7fc>

CATEGORIZATION

Application type *: Games

Category *: Casual

Content rating *: Everyone

New content rating *:
APPLIED RATING:
 IARC Certificate ID: 188c1959-b638-4bce-b36b-596e136785ed
 Submitted: Apr 20, 11:44 AM
[View details](#) [Learn more](#)

Figura 40 - Listagem de requisitos Google Play Store(2)

Não sendo mandatório foi ainda criado um pequeno vídeo introdutório de forma a criar um maior interesse por parte dos visitantes em fazer o download e experimentar o jogo. Para a produção do vídeo foi utilizada a ferramenta Sony Vegas que permite a edição e manipulação de vídeo de forma rápida e simples.

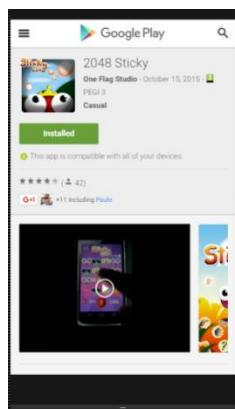


Figura 41 - Jogo publicado na loja

4. Testes e análise de dados

4.1 Teste Alfa

A primeira versão do jogo foi disponibilizada ainda em modo de produção com o intuito de validar compatibilidade com dispositivos, jogabilidade e aceitação por parte do público. Visualmente apresentava um aspeto pouco polido sem animações mas já com contexto de personagens sem ainda implementação de objetivos e integrações com serviços externos.

Não tendo sido publicada a aplicação foi disponibilizada na internet em portal próprio e ainda partilhada uma versão de testes para a plataforma Android.



Figura 42 – Ecrãs da versão de fase alfa

Depois de partilhada a versão com cerca de 20 pessoas e dessas sete responderam a um questionário sobre a versão disponibilizada. O questionário referia claramente tratar-se de um teste alfa, referindo que o projeto estaria ainda em fase de desenvolvimento inicial. Apesar da fraca resposta ao questionário conseguiram obter-se algumas informações relevantes bem como alguns *bugs* que teriam de ser corrigidos de imediato. Abaixo estão apresentados os resultados desta fase.

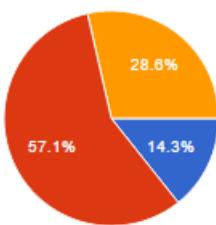
Se este jogo estivesse disponível no Google Play Market qual seria o valor de 1-5 que lhe atribuiria?



Costuma jogar este género de jogos?



Com que regularidade usa o seu telemóvel para jogar?



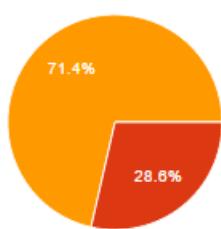
Once a day	1	14.3%
More than once a day	4	57.1%
Less than once a day	2	28.6%

Se uma versão final deste jogo fosse disponibilizada publicamente faria download do jogo?



Yes	7	100%
No	0	0%

Como classifica a experiência de jogo?



Bad	0	0%
Satisfatory	2	28.6%
Good	5	71.4%
Very good	0	0%

Analizados os resultados do questionário, ainda que tratando-se de um número reduzido de dados, é possível entender que o protótipo estaria ainda muito longe de ser um produto final de qualidade razoável, considerando que uma amostra com esta proximidade tende a inflacionar os resultados.

Samsung Galaxy 4
samsung galaxy s3
Samsung GT7582
Ig g3
Samsung S7390
Samsung i5801
Motorola moto g2

Figura 43 – Lista de dispositivos envolvidos no teste alfa

No questionário foi ainda pedido aos utilizadores se possível que indicassem o modelo do dispositivo onde experienciaram o jogo, de forma a entender se existiriam problemas com modelos específicos ou com rácios e resoluções de ecrã pelo que foi possível compreender que não tendo existido comentários sobre esse tema assumindo que não haveria problemas dessa natureza. Foi ainda pedido no questionário caso existisse interesse para partilharem

algumas opiniões e sugestões de melhoria, sendo que todos intervenientes nas respostas ao questionário fizeram o favor de dar o seu contributo.

<p><i>"Apart from the already identified missing features, I would say it looks to be on the right track. My device is rather slow, but the game was not, which is very nice. I also like the game colors, however I think overall they're a little bit too dark, but that's just my personal opinion. :) Good luck!"</i></p>
<p><i>"The gameplay is a little slow and no animations"</i></p>
<p><i>"Back button not working and the characters should by identified by their values."</i></p>
<p><i>"Quando faço movimentos para cima ou para baixo, se descair um pouco para os lados o jogo assume como se tivesse feito o movimento para os lados, mas não devia ser."</i></p>
<p><i>A alteração do boneco principal devia ter mais enfase; Botão do dispositivo de voltar não funciona."</i></p>
<p><i>"Falta algo a explicar o objetivo do jogo. Algo que apareça na primeira vez que se lança a app e depois se aceda num menu qualquer. A cor "rosa" esta muito próxima do vermelho."</i></p>
<p><i>"- invés de teres a seta verde a dizer que ganhou pontos podias por uma animação no sapo grande. Ex: se juntou 2 blocos baixo valor apenas sorri, se juntou 2 blocos grande valor dá um salto ou cambalhota. - as animações do "Play" e da nuvem são muito lentas e parecem irritar dá uma vista de olhos - falta botão sair (nem com o botão de voltar do android dá para sair)"</i></p>

Tabela 1 - Comentários ao teste alfa

Em última análise foi concluído após o primeiro teste alfa que existia interesse por parte das pessoas acima de tudo em contribuir para o projeto com opiniões e sugestões de melhoria, afirmindo isso que o jogo teria despertado interesse.

4.2 Teste Beta

Depois do teste Alfa e consideradas todas as opiniões e problemas referidos pelos envolvidos no teste foram revisitadas algumas considerações iniciais ao produto e revistas particularmente questões relativas à experiência de jogo e correção de problemas relacionadas com escalamentos de objetos e ações esperadas pelos utilizadores de dispositivos móveis em particular para a plataforma móvel *Android*. Para o teste Beta foi utilizada a plataforma de distribuição do Google. A publicação para testes permite aos criadores de jogos definir um grupo de utilizadores enviando-lhe um convite com uma ligação segura onde os utilizadores podem obter uma cópia do jogo de forma segura. A utilização deste mecanismo permitiu ainda obter conhecimentos sobre a plataforma de distribuição da Google o que foi utilizada mais tarde para distribuição a nível global.

Foram então selecionados alguns grupos públicos como grupos de teste e realizado em cada um deles uma publicação para informar os seus participantes que caso estivessem interessados poderiam aceder à versão disponibilizada e ser um *Beta Tester*.

Os grupos selecionados foram:

- androidbrasil-dev@googlegroups.com ([Google Group](#))

- androiddevfortal@googlegroups.com ([Google Group](#))
- androidbrasil@googlegroups.com ([Google Group](#))

AndroidDevFortal ›
 NEW [JOGO] Beta na Store - Sticky(Pegajoso) 2048, seja um testador de meus jogos e me ajude a criar conteúdos
 7 posts by 5 authors (G+)

 me (Luis Fernandes change)
 ★ [Translate message to English](#)

Olá pessoal,
 Se juntem ao meu test Beta. Esse é meu primeiro jogo de sempre é grátis não custa ajudar e se divertir um pouco. Obrigado!
[Click aqui para ser um testador](#): <https://play.google.com/apps/testing/com.oneflag.sticky> e obter o jogo na Google store!
 Melhores agradecimentos,
 Luis

[Click here to Reply](#)

★ Victor Hugo Parabéns pela iniciativa cara!

Figura 44 – Publicação no fórum *AndroidDevFortal* para promoção do jogo em fase Beta

Com o lançamento da versão Beta no início de Julho e com alguma promoção através de publicações em fóruns como exemplo acima o número de *downloads* atingiu o número de 35 downloads o que se revelou decepcionante. Por outro lado apesar do número reduzido de downloads a aceitação da versão Beta revelou-se extremamente satisfatória com uma média de 5 estrelas na Google Play Store.

4.3 Dados de lançamento final

O jogo foi oficialmente publicado como versão final no dia 31 de Julho, tendo o mesmo graças à sua versão *beta*, sido alvo de interesse de uma empresa que promove publicidade nos dispositivos móveis. Esse contacto permitiu consolidar uma parceria de utilização de exposição de publicidade no jogo em troca de publicidade gratuita a partir do dia 10 de Agosto e durante o período de 20 dias consequentes.

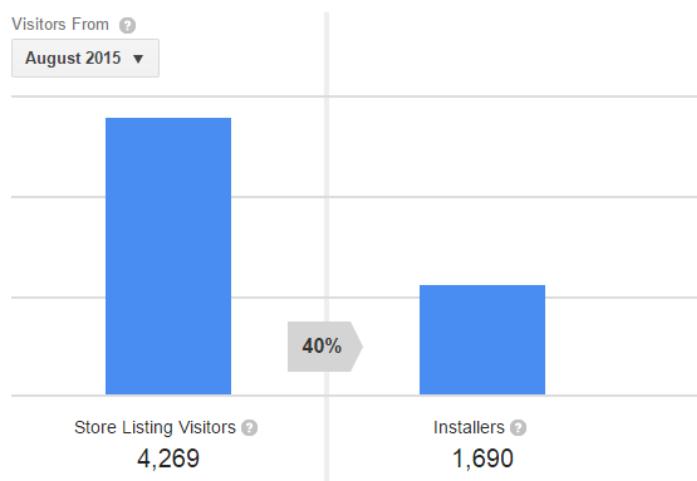


Figura 45 – Resumo de visitantes da loja Google Play (Agosto)

Durante esse período o número de *downloads* atingiu uma taxa de interesse de 40% com 1690 downloads com 4269 visualizações na loja do Google. Este acordo permitiu de forma extraordinária a exposição do jogo que no dia 10 de Agosto explodiu em número de *downloads* e jogadores ativos com uma aceitação do jogo excepcional que o coloca com uma pontuação média de 4.29 (04 de Outubro de 2015).

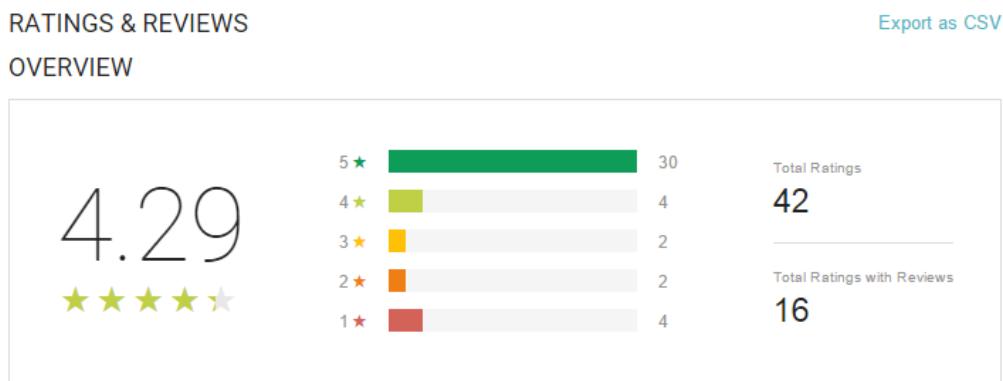


Figura 46 – Ratings and Reviews loja Google Play Store

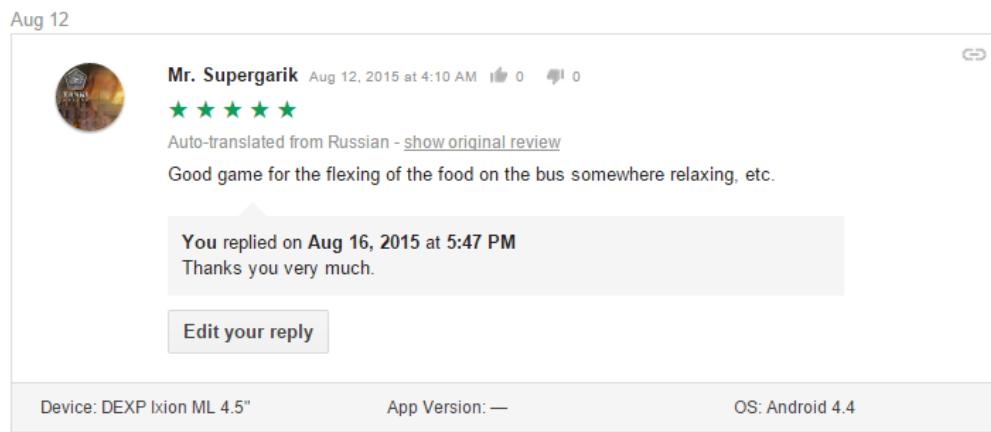


Figura 47 – Comentário de jogador retirado da *Google Play Store*

Ainda de reforçar que a versão disponibilizada não reportou nenhum *bug* refente a congelamento da aplicação o que em si revelou ainda que o desenvolvimento e os testes funcionais foram realizados com sucesso revelando que a ferramenta de desenvolvimento apresenta uma estrutura extremamente estável.

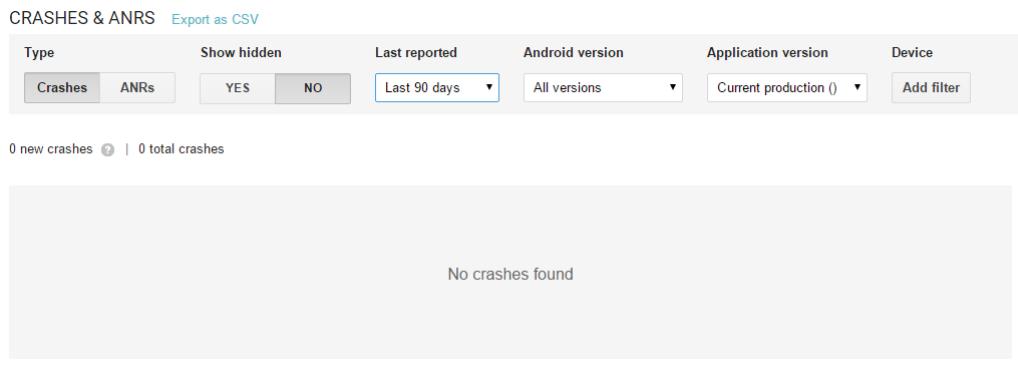


Figura 48 – Relatório de problemas da aplicação

Em conclusão o processo de disponibilização do jogo, com particular foco nos dispositivos *Android* revelou ter sido um sucesso, dado o fato de todo o processo de publicação ter sido realizado pela primeira vez. Não é de esquecer o fator ainda de que para que tal pudesse ser possível foram realizados contactos de forma a encontrar um parceiro que de forma gratuita consegue-se proceder a uma campanha de publicidade à imagem do que sucedeu, sendo este um cenário típico de programadores independentes sem orçamentos disponíveis para marketing e aquisição de jogadores. No momento de pico o jogo atingiu um máximo de 305 instalações representativas de pelo menos uma partida foi jogada no dia a que se refere.

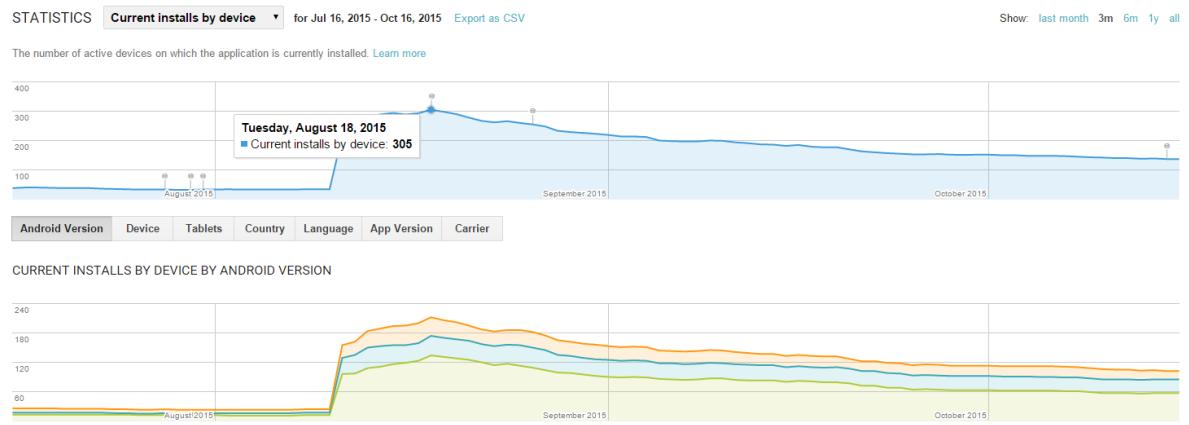


Figura 49 – Visão de instalações da aplicação ao longo do tempo

5. Conclusões e apreciações finais

5.1 Conclusões sobre o produto criado

Apresentado este documento e nele contida informação de base necessária para quem ambiciona desenvolver este tipo de software parece-nos relevante referir que para primeiro produto desenvolvido este apresenta excelente resposta por parte dos jogadores, tendo registado cerca de 2000 downloads nos dispositivos *Android* e jogadas mais de cinquenta mil sessões, mantendo ativos mais de 100 jogadores por dia até à data. Considerados os resultados do produto desenvolvido e o nível de aceitação descritos acreditamos que o propósito de sucesso foi atingido de forma clara.

Assente numa lógica de negócio direcionada para as plataformas digitais de distribuição e para a internet o desenvolvido serviu acima de tudo de prova de conceito de que a aplicação de conhecimentos transversais a profissionais de desenvolvimento de software corporativo são suficientes para suster as necessidades de construção deste tipo de produto sendo que comparativamente com soluções mais direcionadas para as empresas, como soluções de gestão de negócio integradas, estes apresentam desafios diferentes e informações escassas em relação a normas de construção de interfaces, sendo estas centradas na interação com os utilizadores e efeitos de resposta rápida baseadas em lógicas de negócio mais simples.

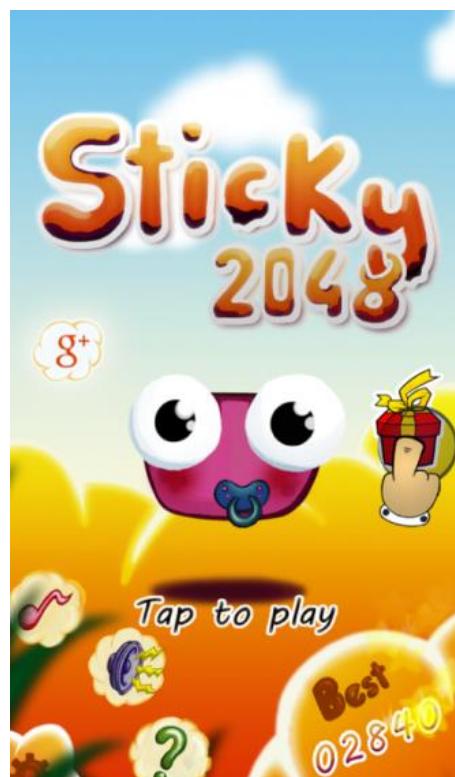


Figura 50 - Tela principal do jogo Sticky 2048

O jogo em si foi então desenvolvido de forma iterativa sobre um planeamento definido à medida para a equipa e de acordo com os recursos humanos e técnicos disponíveis, privilegiando sempre a utilização de ferramentas de software de uso gratuito à excepção do uso da ferramenta *Photoshop* que surgiu como forma de suster atrasos na criação de conteúdos, demonstrando-se mais adequada para o cenário de rápido desenvolvimento e reutilização de conhecimentos prévios. Depois de analisado o mercado à data da realização do projeto, foi decidido avançar para desenho conceptual de um jogo que permitisse focar a atenção no processo de desenvolvimento, e não na complexidade do jogo em si de acordo com o âmbito do documento. A parte gráfica partiu acima de tudo de um processo original e criativo sendo que todas as personagens e contexto apresentados no jogo são da total responsabilidade do seu autor não existindo ligação criativa a nenhum outro produto. Os principais ecrãs e personagens do jogo foram esboçadas inicialmente em papel e respetivamente transportadas para meio digital, sendo desenhadas de forma manual utilizando para isso a ferramenta *Photoshop* e hardware de digitalização com suporte a caneta com sensibilidade à pressão. Esta fase de desenvolvimento foi provavelmente onde foram encontrados os maiores desafios por parte dos executantes dada a falta de experiência resultando em desvios temporais significativos no planeamento, concluindo-se que não será nunca de desprezar todo o trabalho necessário por parte dos artistas gráficos geralmente envolvidos no desenvolvimento deste tipo de produtos e particularmente quando esse é em si o maior fator de diferenciação face à concorrência direta.

O processo de desenvolvimento e programação da experiência teve particular uso de um *pipeline* capaz de manter a integração e testes funcionais a cada fase do desenvolvimento evitando o escalamento de problemas e bugs. Com a utilização da *framework* Game Closure foi ainda possível a fácil transposição de conceitos de desenvolvimento de software como boas práticas de separação de responsabilidades de classes e criação de componentes de software testáveis de forma a garantir a qualidade do software desenvolvido.

Após concluído o desenvolvimento e integração das componentes gráficas finais com o programa desenvolvido o processo passou para a fase de distribuição tendo o jogo sido disponibilizado para todas as plataformas através de um portal de internet e ainda disponibilizado especificamente para a plataforma *Android* onde o processo de integração requer a satisfação de alguns requisitos. Nesse sentido foram ainda criados componentes gráficos como um vídeo de apresentação e ainda alguns ícones para integração de serviços prestados pelo Google.

Em última análise o resultado do produto criado foi extremamente satisfatório tendo sido conseguido o atingir de todos os objetivos propostos com a criação deste produto, nomeadamente, a criação de uma experiência interativa, visualmente apelativa, com contextualização, capaz de chegar às massas e acima de tudo, ser produzida em tempo útil por pequenas equipas independentes.

5.2 Conclusões finais

Todo a análise, pesquisa de conhecimento e trabalho realizado permitiram aprofundar significativamente processos de desenvolvimento de jogos assentes num processo de prática e experimentação de forma a obter os resultados mais viáveis e reais possíveis. O conhecimento do que são jogos desde a sua definição etimológica ao entendimento de como este tipo de produtos afeta as pessoas a nível social e psicológico são bases fundamentais para quem entende o desenvolvimento de produtos com propósitos maiores, sejam eles no sentido lúdico sejam eles direcionados no sentido de avançados sistemas de educação.

O desenvolvimento do trabalho apresentado, apostado no conhecimento geral do estado da arte sobre o conhecimento da indústria e movimentos dos maiores do meio, particularmente no meio de desenvolvimento independente, permitiu a apaixonados pelo desenvolvimento de software com vontade de explorar o potencial que este tipo específico de experiências apresentam na sua relação com as artes e com elementos de media, apreender alguns dos mais basilares conceitos do tema, criando sensibilidade para as dificuldades do meio, sejam elas relativas à criação de conteúdos, ou à criação de expectativas e processos de criatividade que resultem no desenvolvimento de produtos que sejam eficazes no seu propósito. Os vídeo jogos, sejam eles lúdicos ou educativos, procuram através do conhecimento profundo do funcionamento da mente humana, e através das suas características únicas de interatividade, promover a prática e a experimentação em forma de elementos de média como motor de transmissão de informação, promovendo ainda a competição por melhores resultados.

Não menos relevante, ainda que dado o âmbito do projeto menos explorado, são fatores relacionados com sustentabilidade financeira, modelos de negócio e distribuição, sendo estes muitas vezes motivadores ou desmotivadores de aposta nesta área. A verdade é que o mercado está em constante mudança e comparativamente temos hoje um público menos disposto a investir em jogos de forma direta e a apostar mais num modelo similar ao das máquinas de jogos dos anos 80 e 90 onde o jogador pagava para jogar, com as diferenças que hoje em dia o pagamento é feito de forma digital e as máquinas de jogos não estão estáticas e pertencentes a terceiros, mas sim são móveis, são nossas e acompanham-nos para todo o lado. Dito isto conclui-se que face ao mercado é necessário explorar e analisar não só o estado atual, mas sim as tendências, visto que nesta indústria a mudança não é estranha e apenas quem a conseguir acompanhar terá uma hipótese de se manter na corrida.

O desafio para programadores de jogos independentes, abstraindo os seus propósitos, está hoje centrado numa concorrência gigante num mercado de trajetória ascendente onde as barreiras à entrada são muito maiores do que fatores de criatividade e vontade de fazer. São necessárias análises profundas aos mercados e às ferramentas de produção, de forma a criar elementos diferenciadores e que permitam criar produtos capazes de fazer a diferença. Como referido no inicio deste documento, o apresentado não é um caminho único para a prossecução do objetivo de desenvolvimento de jogos por programadores independentes mas sim servir como pilar para o embarcar nesta, por muitos descrita, aventura.

Foi ainda possível concluir que independentemente da ferramenta utilizada ou do jogo a desenvolver é acima de tudo importante manter uma postura metódica e objetiva, é necessário definir o que se pretende construir e distinguir quais são os fatores de diferenciação dos produtos e promover esses mesmos fatores. A escolha de uma ferramenta de desenvolvimento e dos elementos da equipa são ainda fundamentais, sendo que estas terão maior peso no futuro em caso de necessidade de alteração. O foco, a organização e a capacidade de produção sustentável serão eventualmente algumas das mais relevantes características de um programador independente que aspire a criar carreira profissional no meio, e como referido ainda neste documento, desenvolver jogos não é fácil, não é barato e requer dos envolvidos vontade, espírito de sacrifício e acima de tudo, paixão pelo que fazem.

O desenvolvimento deste projeto ainda que muito assente em aspectos de curiosidade e ligação pessoal sobre o mercado e desenvolvimento de jogos, serve como base de início de um negócio próprio e aposta no criar de uma empresa que venha a ser sustentável e geradora de empregos na área. A sensibilização para as dificuldades e experiência prática obtida na execução do projeto desenvolvido bem como a recolha e análise de dados do jogo criado permitem hoje dizer de forma clara que este é o caminho a seguir e que o mesmo teve já inicio, ainda que de forma lenta e humilde. A nível pessoal acredito que este projeto representa a etapa de viragem do culminar de todo investimento realizado na aposta educacional e que a curto prazo as bases apresentadas neste documento serão os pilares onde o sucesso estará estabelecido.

Futuramente a criação de novos produtos, a melhoria dos já disponíveis e contínua aposta na investigação e aprimoramento de conhecimentos sobre o tema dos jogos, design, desenvolvimento e distribuição é um claro objetivo resultando isso na proceção de um sonho em que Portugal poderá vir a ser reconhecido na área do desenvolvimento de jogos, sejam eles sérios ou lúdicos, tal como já o é em várias outras áreas.

Referências Bibliográficas

[Crawford, 1982]	Crawford C., The Art of Computer Game Design, http://www-rohan.sdsu.edu/~stewart/cs583/ACGD_ArtComputerGameDesign_ChrisCrawford_1982.pdf [último acesso: Outubro 2015]
[Koster, 2013]	Koster R., 2013. The Theory of Fun for Game Design, Paperback 2nd edition, O'Reilly Media Inc. 2013
[Salen and Zimmerman, 2004]	Salen K., Zimmerman E., 2004, Rules of Play: game design, Massachusetts Institute of Technology, ISBN-10: 0262240459
[McGonigal, 2011]	McGonigal J., 2011, Reality Is Broken: Why Games Make Us Better and How They Can Change the World, Penguin Books, ASIN: B004G8Q1Q4
[Thorn, 2008]	Thorn A., Cross Platform Game Development, ISBN 10: 1-59822-056-X, p.266
[Novak, 2011]	Novak J., 2011. Game Development Essentials: An Introduction 3th edition, Cengage Learning, ISBN-10: 1111307652
[Luppa and Borst, 2011]	Luppa N., Borst T., 2012, End-to-End Game Development: Creating Independent Serious Games and Simulations from Start to Finish, Focal Press Elsevier Direct, ISBN: 978-0-240-81179-6
[Rabin, 2009]	Rabin S., 2009, Introduction to Game Development, Second Edition, Course Technology a part of Cengage Learning, ISBN-10: 1584506792
[Turocy et al., 2001]	Turocy T, Stengel B., Game Theory*, London School of Economics, http://www.cdam.lse.ac.uk/Reports/Files/cdam-2001-09.pdf [último acesso: Outubro 2015]
[Rouse III, 2004]	Rouse III R., Game Design: Theory and Practice 2nd edition, Wordware Publishing, ISBN-10: 1556229127
[Saunders et al, 1996]	Saunders D., Games and Simulations to Enhance Quality Learning, The Society for Interactive Learning, ISBN-10: 0749418664
[Tarja,2007]	Tarja S, Serious Games – An Overview, http://www.scangames.eu/downloads/HS-IKI-TR-07-001_PER.pdf [último acesso: Outubro 2015]
[Brathwaite and Schreiber, 2009]	Brathwaite B., Schreiber I., Challenges for Game Designers (Paperback), Cengage Learning, ISBN-10: 158450580X
[Chandler, 2009]	Chandler M., The Game Production Handbook 2nd edition, Jones & Bartlett Learning, ISBN-10: 1934015407
[Gregory, 2014]	Gregory J., Game Engine Architecture, 1st edition, AK Peters/CRC Press, ISBN-10: 1568814135
[Needles, 2011]	Needles A., Balancing the Demand Equation: The Elements of a Successful, Modern B2B Demand Generation Model, New Year Publishing, ISBN-10: 1935547364
[Seufert, 2013]	Seufert E., Freemium Economics: Leveraging Analytics and User Segmentation to Drive Revenue (The Savvy Manager's Guides) 1st edition, Morgan Kaufmann, ISBN-10: 0124166903
[Siapera, 2012]	Siapera E., Understanding New Media, SAGE Publications Ltd, ISBN-10: 1848607792
[Thorn, 2013]	Thorn A., Game Development Principles 1st edition, Cengage

Referências Webgráficas

[Grubb, 2015]	Grubb J., Mobile gaming will be a \$45B market by 2018, leaving console gaming way behind, http://venturebeat.com/2015/05/04/mobile-gaming-will-leave-console-software-behind-with-45b-in-spending-by-2018/ , [último acesso: Outubro 2015]
[Primack,, 2012]	Primack B., Carroll M., McNamara M., Klem M., King B., Rich M., Chan C., Nayak S., Role of Video Games in Improving Health-Related Outcomes, http://www.ajpmonline.org/article/S0749-3797(12)00172-9/pdf , [último acesso: Outubro 2015]
[Rowan, 2010]	Rowan, D 2010, Serious Games - Can games change your business and change the world?, http://www.trusim.com/files/SeriousGames_Nov2010.pdf , [último acesso: Outubro 2015]
[King, 2015]	King, Candy Crush Saga, http://candycrushsaga.com/ , [último acesso: Outubro 2015]
[ESA, 2014]	ESA, Industry Facts, http://www.theesa.com/about-esas/industry-facts/ , [último acesso: Outubro 2015]
[Blow, 2004]	Blow J., Game Development: Harder Than You Think, http://dl.acm.org/citation.cfm?id=971590 , [último acesso: Outubro 2015]
[Warrington, 2015]	Warrington G., Dear Teacher, A Video Game Developer Is A Real Job And Should Be Celebrated, https://www.linkedin.com/pulse/dear-teacher-video-game-developer-real-job-should-glenn-warrington , [último acesso: Outubro 2015]
[Gamasutra, 2014]	Gamasutra, Salary Survey 2014, http://www.gamasutra.com/salarysurvey2014.pdf , [último acesso: Outubro 2015]
[IGN, 2015]	IGN, Top 50 Video Game Makers, http://www.ign.com/top/video-game-makers , [último acesso: Outubro 2015]
[Full Sail University, 2014]	Aguilo Ricardo A., Life as an Indie Game Developer , https://www.youtube.com/watch?v=iMkynKGliBQ , [último acesso: Outubro 2015]
[Jonathan et al, 2011]	Blow J., Hecker C., Persson M., Indie Game Revolution! , http://www.gametrailers.com/full-episodes/5mqiab/bonus-round-indie-game-revolution- , [último acesso: Outubro 2015]
[Yu, 2013]	YU Derek, 2013, Making it in Indie Games: Starter Guide, http://makegames.tumblr.com/post/44181247500/making-it-in-indie-games-starter-guide , [último acesso: Outubro 2015]

[Yu, 2010]	YU Derek, 2010, Finishing a Game, http://makegames.tumblr.com/post/1136623767/finishing-a-game , [último acesso: Outubro 2015]
[Gilbert, 2014]	Gilbert S., Why indie videogaming is so important, http://www.denofgeek.com/games/indie-games/30768/why-indie-videogaming-is-so-important , [último acesso: Outubro 2015]
[Gamasutra, 2007]	Gamasutra, Independent Game Creators On Importance Of Indie Movement, http://gamasutra.com/view/news/106695/QA_Independent_Game_Creators_On_Importance_Of_Indie_Movement.php , [último acesso: Outubro 2015]
[Chernyak, 2014]	Chernyak U., Video Game Market Overview: Console vs. PC vs. Mobile, http://www.gamasutra.com/blogs/UlyanaChernyak/20140527/218626/Video_Game_Market_Overview_Console_vs_PC_vs_Mobile.php , [último acesso: Outubro 2015]
[Wynne, 2014]	Wynne J., Here's everything announced at Twitter's Flight developer conference VentureBeat Dev by Ken Yeung, http://venturebeat.com/community/2014/07/19/indie-games-are-more-important-than-ever/ , [último acesso: Outubro 2015]
[Braid, 2006]	Braid, http://braid-game.com/ , [último acesso: Outubro 2015]
[Limbo, 2010]	Limbo, http://playdead.com/limbo/ , [último acesso: Outubro 2015]
[Naser, 2015]	Naser A., Why to Develop 2D Games for Mobile Instead of 3D Games?, https://www.linkedin.com/pulse/why-develop-2d-games-mobile-instead-3d-ahmad-hammad?goback=%2Egna_4433494 , [último acesso: Outubro 2015]
[Rampant, 2013]	Rampant C., Why Are Most Indie Games 2D Instead of 3D? , http://rampantgames.com/blog/?p=5934 , [último acesso: Outubro 2015]
[Ubisoft, 2010]	Ubi Art Framework, http://ubi-art.uk.ubi.com/ , [último acesso: Outubro 2015]
[HardwareHeaven, 2013]	HardwareHeaven, How Rayman Legends Is Made! , https://www.youtube.com/watch?v=y-chi097uV4 , [último acesso: Outubro 2015]
[Surface, 2015]	Surace Pro, http://www.microsoftstore.com/store/msusa/en_US/cat/All-Surface/categoryID.69403400 , [último acesso: Outubro 2015]
[FreeSfx, 2015]	FreeSfx, http://freesfx.co.uk/ , [último acesso: Outubro 2015]
[Trine, 2014]	Trine, http://www.frozenbyte.com/games/trine-enchanted-edition , [último acesso: Outubro 2015]
[PC Gamer, 2013]	PC Gamer, Microtransactions: the good, the bad and the ugly, http://www.pcgamer.com/microtransactions-the-good-the-bad-and-the-ugly/ , [último acesso: Outubro 2015]
[Rubino, 2014]	Rubino D., Michaluk K., Nickinson P., Ritchie R., Can Mobile Gaming Kill, http://www.androidcentral.com/talk-mobile/can-mobile-gaming-kill-consoles , [último acesso: Outubro 2015]
[Sinclair, 2015]	Sinclair B., Gaming will hit \$91.5 billion this year

	, http://www.gamesindustry.biz/articles/2015-04-22-gaming-will-hit-usd91-5-billion-this-year-newzoo , [último acesso: Outubro 2015]
[theguardian, 2013]	theguardian, PS4 v Xbox One: indie game developers have their say, http://www.theguardian.com/technology/2013/nov/20/ps4-xbox-one-indie-developers-think-game-console-war , [último acesso: Outubro 2015]
[IDC, 2015]	IDC, Smartphone OS Market Share, 2015 Q2, http://www.idc.com/prodserv/smartphone-os-market-share.jsp , [último acesso: Outubro 2015]
[VB, 2014]	VB, Takahashi D., EEDAR deconstructs how to succeed in mobile and tablet games, http://venturebeat.com/2014/11/07/eedar-deconstructs-how-to-succeed-in-mobile-and-tablet-games/ , [último acesso: Outubro 2015]
[Forbes, 2012]	Forbes, 10 Jobs That Didn't Exist 10 Years Ago, http://www.forbes.com/sites/meghancasserly/2012/05/11/10-jobs-that-didnt-exist-10-years-ago/ , [último acesso: Outubro 2015]
[Devore, 2015]	Devore J., Games aren't cheap, http://www.destructoid.com/games-aren-t-cheap-it-cost-1-4-million-to-make-monument-valley-and-its-dlc-286263.phtml , [último acesso: Outubro 2015]
[PocketGamer, 2015]	http://www.pocketgamer.biz/metrics/app-store/ , [último acesso: Outubro 2015]
[Independent, 2015]	http://www.independent.co.uk/news/business/news/minecraft-creator-sold-his-25b-stake-to-microsoft-after-one-tweet-10088955.html , [último acesso: Outubro 2015]
[Ribeiro, 2014]	http://multimediacomtodos.pt/entrevista-alexandre-ribeiro-criador-do-fangz , [último acesso: Outubro 2015]
[Tran, 2014]	Jimmy Tran, http://venturebeat.com/community/2014/05/06/how-digital-distribution-is-changing-the-scope-of-gaming/ , [último acesso: Outubro 2015]
[Singh, 2014]	Gurinder Singh, http://www.gamasutra.com/blogs/GurinderSingh/20140123/209055/5_Metrics_every_mobile_game_developer_must_track.php?print=1 , [último acesso: Outubro 2015]
	http://www.gamasutra.com/blogs/AlistairDoulin/20100107/86323/Building_A_Strong_Indie_Game_Development_Team.php
[Reimer, 2005]	REIMER Jeremy , http://arstechnica.com/features/2005/11/crossplatform/1/ , [último acesso: Outubro 2015]
[Seufert, 2013]	Seufert B. , http://mobiledevmemo.com/minimum-viable-metrics/ , [último acesso: Outubro 2015]
[Ries, 2012]	Eric Ries, http://theleanstartup.com/principles
[PixelProspector, 2014]	http://www.pixelprospector.com/the-big-list-of-game-making-tools/ , [último acesso: Outubro 2015]
	http://gamedevelopment.tutsplus.com/articles/9-tips-for-indie-

	game-developers-i-learned-at-gdc-2013--gamedev-6591, [último acesso: Outubro 2015]
[MacMillan , 2012]	Douglas MacMillan , http://www.bloomberg.com/news/articles/2012-04-17/zynga-flashes-1-8-billion-searching-for-the-new-farmville-tech , [último acesso: Outubro 2015]
[Empson, 2012]	Rip Empson, http://techcrunch.com/2012/02/24/game-closure-turns-down-facebook-zynga-to-raise-12m-for-html5-mobile-gaming/ , [último acesso: Outubro 2015]
[Golding, 2012]	DANIEL GOLDING, MAR 06, 2012, http://blogs.crikey.com.au/game-on/2012/03/06/videogames-are-not-a-young-media-form-so-stop-saying-they-are/ , [último acesso: Outubro 2015]
[Photoshop, 2015]	Photoshop, http://www.adobe.com/products/photoshop.html , [último acesso: Outubro 2015]
[Gimp, 2015]	http://www.gimp.org/about/ , [último acesso: Outubro 2015]
[Adobe, 2015]	Adobe, www.adobe.com/ , [último acesso: Outubro 2015]
[SVG, 2010]	SVG, http://www.w3.org/Graphics/SVG/ , [último acesso: Outubro 2015]
[Scratchapixel, 2015]	Rasterization: a Practical Implementation, http://www.scratchapixel.com/lessons/3d-basic-rendering/rasterization-practical-implementation , [último acesso: Outubro 2015]
[Inkscape]	Inkscape, https://inkscape.org/en/about/overview/ , [último acesso: Outubro 2015]
[Blender, 2015]	Blender, http://www.blender.org/features/ , [último acesso: Outubro 2015]
[Audiosauna]	Audiosauna, http://www.audiosauna.com/features.html , [último acesso: Outubro 2015]
[Audacity]	Audacity, http://audacityteam.org/about/ , [último acesso: Outubro 2015]
(Game Closure, 2013)	GameClosure, Native Stack Overview, http://docs.gameclosure.com/native/native-stack.html#native-stack-overview , [último acesso: Outubro 2015]
[Moriarty, 2011]	Moriarty C., Under Siege Review - Holly **** this game is hard, Copyright 2003-2015 IGN Entertainment, Inc. , http://www.ign.com/articles/2011/06/15/under-siege-review , [último acesso: Outubro 2015]
[Rose, 2013]	Rose M., Portugal: The rise and fall of a video game industry, http://www.gamasutra.com/view/news/200343/Portugal_The_rise_and_fall_of_a_video_game_industry.php , [último acesso: Outubro 2015]
[EA, 2007]	Exame Informática, Direcção da APROJE demite-se, Exame Informática, http://exameinformatica.sapo.pt/noticias/mercados/2007-01-01-direccao-da-aproje-demite-se , [último acesso: Outubro 2015]
[Bonifácio, 2015]	Bonifácio J., 12 Abril 2015, Eles fazem videojogos e são muito bons nisso, http://www.publico.pt/tecnologia/noticia/eles-fazem-videojogos-e-sao-muito-bons-nisso-1691866 , [último acesso:

	Outubro 2015]
[Miniclip, 2015]	Miniclip, SWISS POWERHOUSE MINICLIP CONTINUES TO GROW, WITH THE HELP OF COCOS2D http://blog.cocos2d-x.org/2015/06/swiss-powerhouse-miniclip-continues-to-grow-with-the-help-of-cocos2d/ , 2015 Cocos2d-x Developer Blog, [último acesso: Outubro 2015]
[Carmona, 2015]	Carmona S., Tango, Samba and Mariachis: The Latin American Mobile Market, https://www.youtube.com/watch?v=yUt-SXjuVQ8 , [último acesso: Outubro 2015]
[Suckley, 2015]	Suckley M., Battlesheep's Nélio Codices on how to live off game development in Portugal , http://www.pocketgamer.biz/news/61968/battlesheep-living-off-game-dev-in-portugal/ , [último acesso: Outubro 2015]
[iTunes, 2015]	2048, Ketchapp, https://itunes.apple.com/en/app/2048/id840919914?mt=8 , [último acesso: Outubro 2015]
[NodeJS, 2015]	NodeJS, https://nodejs.org/en/ , [último acesso: Outubro 2015]
[Eurogamer, 2014]	Threes Review, http://www.eurogamer.net/articles/2014-03-05-threes-review , [último acesso: Outubro 2015]
[ESA, 2015]	http://www.theesa.com/wp-content/uploads/2015/04/ESA-Essential-Facts-2015.pdf , [último acesso: Outubro 2015]
[MSDN, 2015]	Key Principles of Software Architecture , https://msdn.microsoft.com/en-us/library/ee658124.aspx , [último acesso: Outubro 2015]
[Guerra, 2014]	Guerra A., Videojogos: a indústria milionária que está a nascer em Portugal, http://www.dinheirovivo.pt/Imprimir.aspx?content_id=4244092 , [último acesso: Outubro 2015]

6. Anexos

Sticky 2048 - One Flag Studio Alpha Test

This survey should only be answered by anyone that has played the alpha version of Sticky 2048 created by One Flag Studio.

First thank you for your time playing the preview version of the game and hope you've enjoyed the experience.

Please answer this survey with honesty providing helpful information for the product developers that will take in account your time and opinion in creating a most suitable and fun experience in the next releases. Thank you!

If this game was featured in the Google Play market how much stars you would give it?

- 1 Stars
- 2 Stars
- 3 Stars
- 4 Stars
- 5 Stars

Do you like this type of games?

- Yes
- No

How often do you use your cell phone to play games?

- Once a day
- More than once a day
- Less than once a day

If a final version of this game is released in the google play market will you download it?

- Yes
- No

How do you rate the game play

- Bad
- Satisfactory
- Good
- Very good

Device used in test

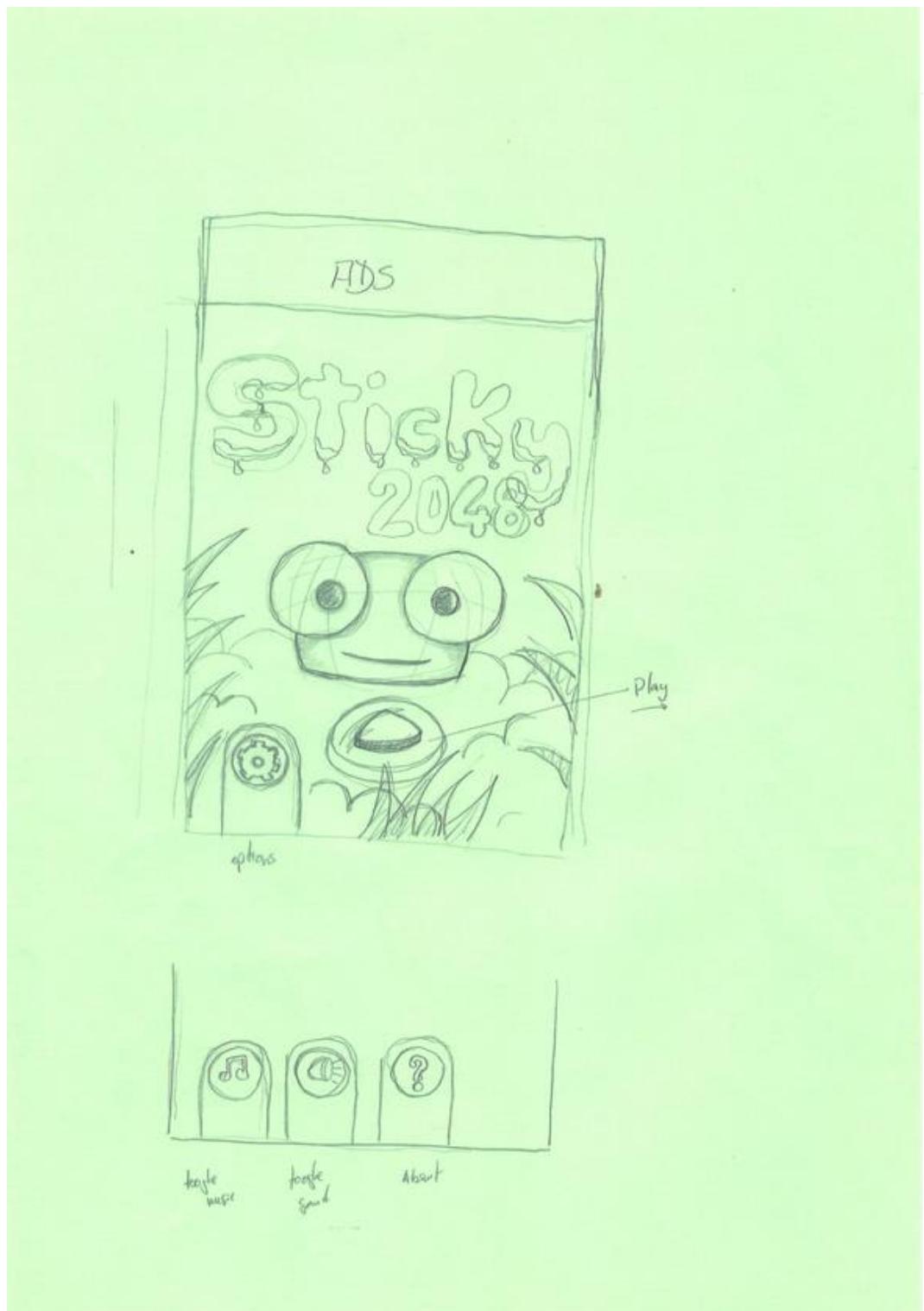
For example: Samsung i5801

Time played

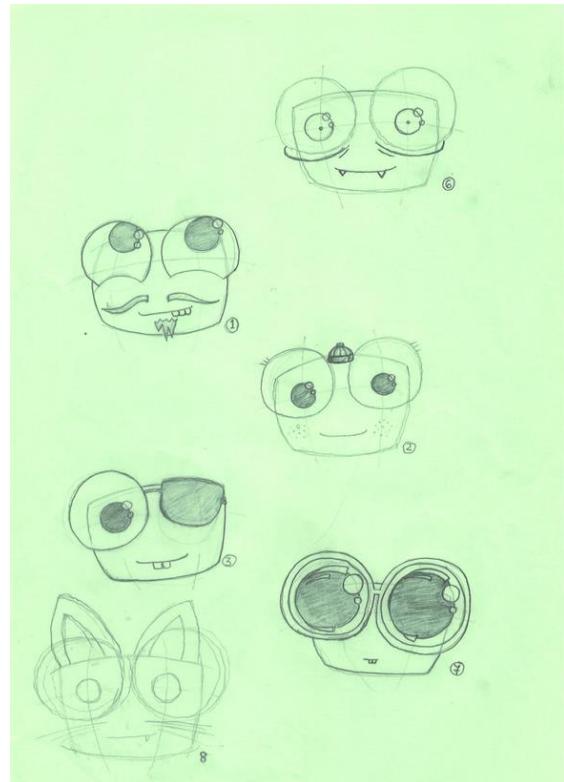
How much time did you played the game? Ex. 3 hours

Can you provide some tips and things you would like to see in this game?

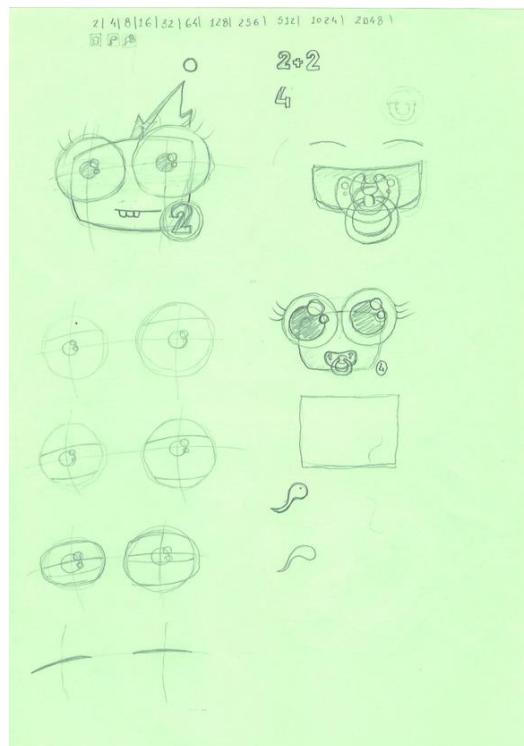
Anexo 1 – Questionário teste alfa



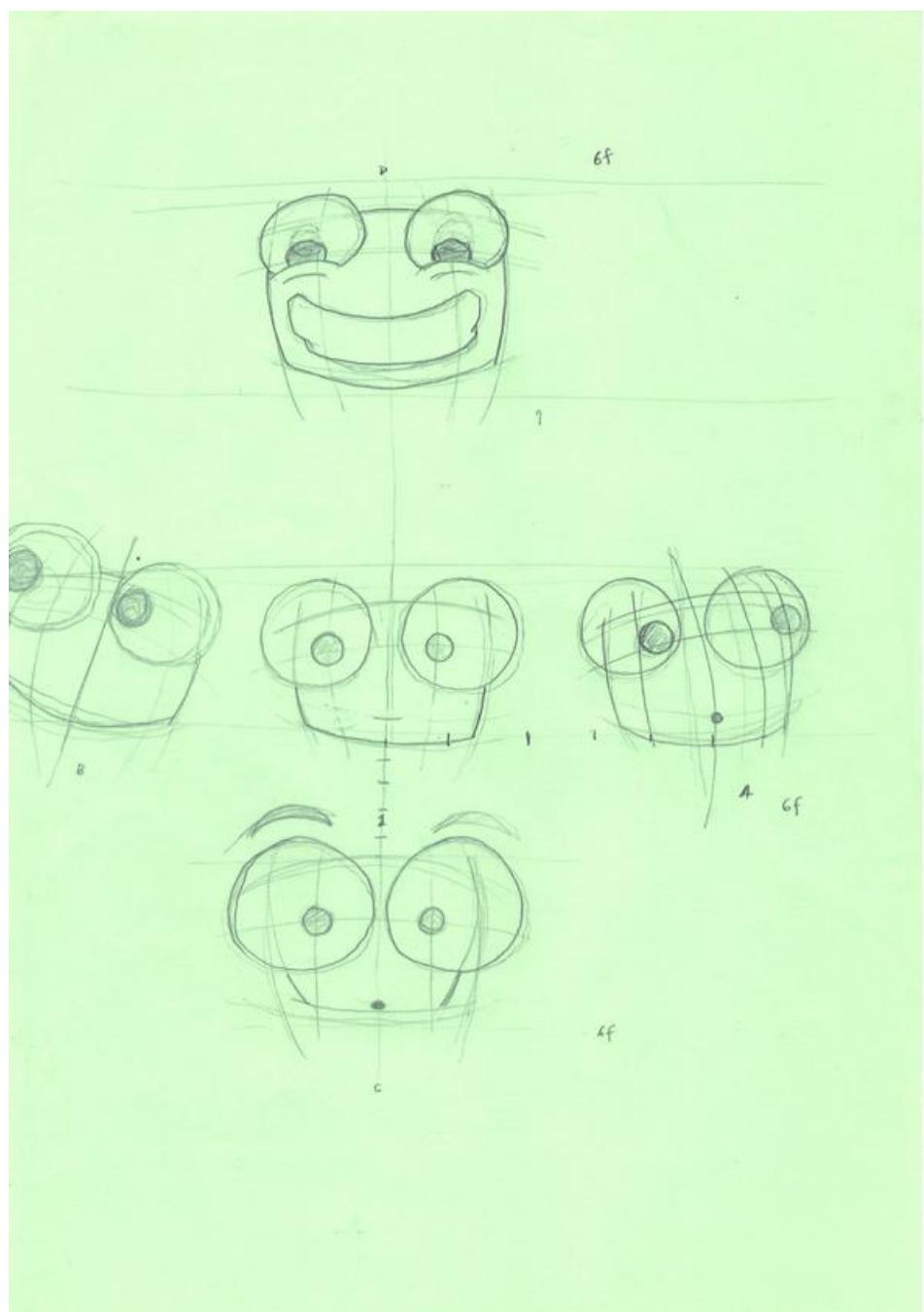
Anexo 2 – Esboço em papel (1)



Anexo 3 - Esboço em papel (2)



Anexo 4 - Esboço em papel (3)



Anexo 5 - Esboço em papel (4)