

# PsyChO: The Ball

Ricardo Lira da Fonseca

Monografia apresentada  
ao  
Instituto de Matemática e Estatística  
da  
Universidade de São Paulo  
para  
obtenção do título  
de  
Bacharel em Ciência da Computação

*Programa:*  
Bacharelado em Ciência da Computação

*Orientador:*  
Prof. Dr. Flávio Soares Corrêa da Silva

São Paulo, dezembro de 2017



## **PsyChO: The Ball**

Esta é a versão original da monografia elaborada pelo candidato Ricardo Lira da Fonseca, tal como submetida à Comissão Julgadora.



## Agradecimentos



## Resumo

FONSECA, R. L. da **Título do trabalho em português**. 2017. 120 f. Trabalho de Conclusão de Curso - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2017.

Elemento obrigatório, constituído de uma sequência de frases concisas e objetivas, em forma de texto. Deve apresentar os objetivos, métodos empregados, resultados e conclusões. O resumo deve ser redigido em parágrafo único, conter no máximo 500 palavras e ser seguido dos termos representativos do conteúdo do trabalho (palavras-chave).

**Palavras-chave:** palavra-chave1, palavra-chave2, palavra-chave3.



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Objetivos . . . . .	1
<b>2</b>	<b>Conceitos, Técnicas e Ferramentas</b>	<b>3</b>
2.1	Conceitos . . . . .	3
2.2	Técnicas . . . . .	5
2.3	Ferramentas . . . . .	6
<b>3</b>	<b>Proposta</b>	<b>9</b>
3.1	Temática . . . . .	9
3.2	Mecânicas . . . . .	9
3.3	Gameplay . . . . .	10
3.4	Recursos Audiovisuais . . . . .	11
<b>4</b>	<b>Desenvolvimento</b>	<b>13</b>
4.1	STEAMING . . . . .	13
4.2	Gerenciador de Saves . . . . .	14
4.3	Script de Fases . . . . .	14
4.4	Shaders . . . . .	14
4.5	Juiceness . . . . .	15
4.6	Problemas e Desafios . . . . .	15
<b>5</b>	<b>Resultados</b>	<b>17</b>
5.1	Conteúdo . . . . .	17
5.2	Eventos . . . . .	18
5.3	Como Jogar . . . . .	19
<b>6</b>	<b>Conclusões</b>	<b>21</b>
6.1	Matérias Utilizadas . . . . .	21
6.2	Trabalhos Futuros . . . . .	21
<b>Referências Bibliográficas</b>		<b>23</b>



# Capítulo 1

# Introdução

Falar da UGD, seu começo em 2009 e como ela fez a ponte entre interessados em jogos e estudantes de computação.

Falar que entrei em no primeiro ano da faculdade e junto de um colega fiz meu primeiro jogo, um shooter psicodélico chamado psychoball. isso foi o que me inspirou para desenvolver jogos e utilizar o conhecimento aprendido

## 1.1 Motivação

## 1.2 Objetivos



## Capítulo 2

# Conceitos, Técnicas e Ferramentas

Assim como no processo de desenvolver um software, um jogo possui vários conceitos e ferramentas próprias que ajudam na comunicação e organização durante o desenvolvimento. Para familiarizar o leitor com as técnicas e jargões normalmente utilizados, vamos descrever nesse capítulo um pouco da linguagem específica no mundo de *game development*, além das técnicas e ferramentas específicas que foram utilizadas na criação de *PsyChO: The Ball*.

### 2.1 Conceitos

A interatividade de um jogo consiste em que ações um usuário pode realizar, e de que forma isso altera o estado atual do jogo. Essa dinâmica define a jogabilidade de um jogo virtual, também chamada de *gameplay*. Os tipos de interações que o jogador tem acesso são denominadas *mecânicas* do jogo, e estas variam dependendo de qual experiência você deseja transmitir para o usuário.

Por exemplo em jogos de plataforma 2D clássicos como *Mario* você controla um personagem que precisa pular e atravessar obstáculos até terminar o jogo. Para isso o jogador pode correr e pular, estas sendo as mecânicas fornecidas ao jogador para enfrentar os desafios propostos. Essas mecânicas juntas ao *feedback* que o próprio jogo retorna vão gerar uma *experiência* para o jogador, e em geral toda a criação de jogos gira em torno de criar uma certa experiência desejada (como diversão, emoção, medo, aprendizado, etc).

Existe uma área de estudo sobre a criação de experiências em jogos: *Game Design*. Game Designer Jesse Schell define Game Design em seu livro sobre o assunto [Sch08] da seguinte forma:

*Game Design é o ato de decidir como um jogo deve ser.*<sup>1</sup>

É natural esperar que essa área não seja uma ciência exata, de forma que não existe algoritmos ou fórmulas matemáticas que definem exatamente como fazer um jogo "bom". Muito pelo contrário, o estudo na área de Game Design é um conjunto de ideias e linhas de pensamento para auxiliar no desenvolvimento de jogos e mais facilmente atingir a experiência desejada que o desenvolvedor quer transmitir. Também no mesmo livro de Schell [Sch08] ele cria um sistema de "lentes", cada uma representando alguma visão ou crítica para analisar as mecânicas e funcionalidades de um jogo e averiguar a consistência entre experiência desejada e transmitida.

Como temos esse objetivo final de *experiência para o usuário*, é muito comum a utilização de documentos para formalmente especificar e detalhar como será o desenvolvimento do jogo. Um documento famoso desse tipo é o *Game Design Document* ou *GDD*. Seu objetivo é servir como uma "receita de bolo",

---

<sup>1</sup>Tradução livre feita pelo autor

determinando o gênero do jogo, suas mecânicas, detalhamentos de personagens ou roteiro da história, ou até uma breve descrição do que é seu jogo, de forma que todos que trabalhem no projeto possam facilmente consultar e utilizar como guia durante o desenvolvimento. Em seu livro de Fundamentos em Game Design [Ada10], Ernest Adams descreve Game Design Documents da seguinte forma:

*Os documentos gravam decisões feitas e concordadas oralmente [entre os integrantes do grupo];  
(...) mais importante que isso, eles transformam ideias vagas em planos explícitos.<sup>2</sup>*

Desta forma, apesar de parecer um gasto adicional de tempo sem grandes benefícios, a manutenção de documentos de design acaba poupando tempo em resolução de conflitos de ideias e deixa o desenvolvimento de um jogo muito mais fluído. Para o jogo *PsyChO: The Ball* foi criado um documento que brevemente descreve sua jogabilidade e gênero.

No caso, *PsyChO: The Ball* é considerado um jogo *Top Down Shooter*, um gênero muito famoso em jogos onde o jogador controla um personagem que precisa esquivar de inimigos e atirar projéteis para de defender. Além disso a visão do jogo é de cima pra baixo, vendo todos elementos do jogo por cima.

Na parte prática de desenvolvimento vamos precisar de ferramentas para produzir e gerar o jogo. Estas são chamadas de *Motores de Jogos* ou *Game Engines*. Uma *Game Engine* consiste em várias bibliotecas que auxiliam na produção de um jogo, além de alguma estruturação para gerenciar e manipular elementos lógicos. Elas possuem métodos para desenhar objetos na tela, reproduzir arquivos de sons, lidar com input de usuário e em muitos casos outras funcionalidades úteis como métodos para simular interações físicas ou gerenciar janelas. Uma versão mais simples de uma *Game Engine* é uma *Framework* ou *Arcabouço*. Estas também são um conjunto de bibliotecas, mas geralmente mais abertas e sem uma estrutura rígida para desenvolver o jogo.

Mas seja uma *Game Engine* ou um *Arcabouço*, todas ferramentas para criação de jogos utilizam um laço lógico chamado de *Game Loop*. Cada ferramenta tem sua própria versão e implementação, mas em geral todas tem 3 pontos chaves: processar inputs, atualizar objetos no jogo e renderizar elementos gráficos. Podemos visualizar esse laço da seguinte forma, como visto no segundo capítulo do livro de Robert Nystrom, *Game Programming Patterns* [Nys14]:

```
while (true)
{
    processInput();
    update();
    render();
}
```

A função *processInput* é responsável por verificar se o usuário apertou alguma tecla ou de algum jeito fez algo no mundo físico que afete o jogo. A função *update* é responsável pela parte lógica do jogo, como atualizar as posições de inimigos, determinar colisões e qualquer outra atualização lógica necessária. É muito comum que essa função receba um argumento especial que diz quanto tempo se passou desde a última chamada dela (muitas vezes chamado de *dt*). Assim é possível fazer cálculos relativos ao tempo de forma mais realística. Por último a função *render* é responsável por desenhar os objetos na tela, normalmente apagando o que foi desenhado anteriormente.

---

<sup>2</sup>Tradução livre feita pelo autor

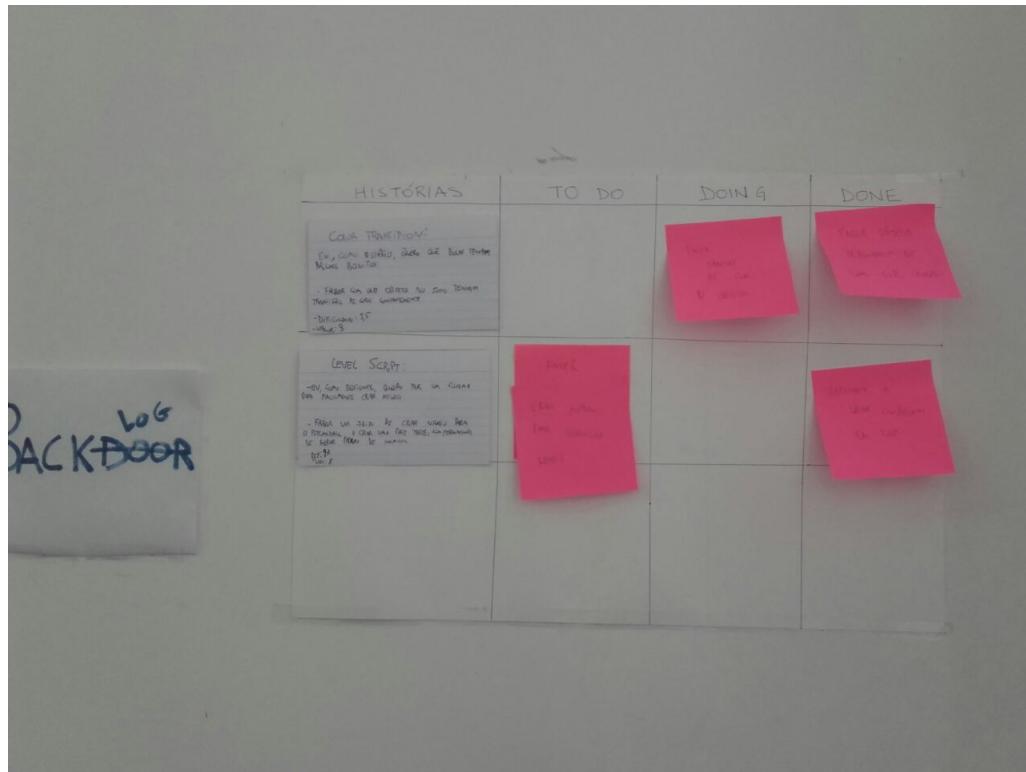
## 2.2 Técnicas

Apesar de grande parte do desenvolvimento de *PsyChO: The Ball* ter sido feita apenas por uma pessoa, foi muito benéfico a utilização de sistemas de versionamento. Estes ajudam tanto na centralização de todo o código do jogo, facilitando na transição entre ambientes de trabalho, quanto na possibilidade de guardar versões antigas e resgatar códigos passados quando necessário. Durante todo o desenvolvimento de *PsyChO: The Ball* foi utilizado um sistema de controle de versão para código, e nele se encontra todas versões de lançamento do jogo.

Uma boa prática na produção de um jogo, seja esse comercial ou um projeto pessoal, é a prática de *releases constantes*. O propósito disto é ter, periodicamente, lançamento de versões estáveis e jogáveis do jogo, de forma que outras pessoas possam jogar e dar feedback frequentemente e assim ajudar no encaminhamento do projeto. Essa técnica é muito comum em metodologias ágeis em desenvolvimento de software, e essa linha de pensamento foi o que guiou todo o desenvolvimento de *PsyChO: The Ball*.

Inicialmente foi utilizado o sistema *Kanban*, uma abordagem moderna de conceitos ágeis muito comum em empresas ou grupos de desenvolvimento de software. Kleber Bernardo, especialista em métodos ágeis, descreve bem o método em um artigo no site Cultura Ágil [Ber14]:

*O Kanban lhe ajuda a assimilar e controlar o progresso de suas tarefas de forma visual. É, normalmente, utilizado um quadro branco com alguns pequenos papéis (Post-it) colados, esses papéis representam as suas tarefas, ao término de cada tarefa o papel é puxado para a etapa seguinte até que a mesma seja finalizada. Ao olhar para um quadro Kanban é fácil enxergar como o trabalho seu e de sua equipe fluem, permitindo não só comunicar o status, mas também dar e receber feedbacks.*



**Figura 2.1:** Kanban utilizado durante o desenvolvimento de *PsyChO: The Ball*

Entretanto após alguns meses utilizando essa técnica, o tempo gasto em sua manutenção acabou se tornando mais trabalhoso do que produtivo para o projeto, e seu uso foi decaindo. Desta forma o *Kanban*

foi substituído por várias pequenas técnicas ágeis, como lançamentos frequentes do jogo, *sprints* (onde era determinado um prazo de uma ou duas semanas para terminar funcionalidades no jogo) e constante re-avaliação do jogo com consulta de feedback.

Por último, uma técnica muito presente no desenvolvimento do jogo foi a utilização de documentação para marcar mudanças e melhorias no projeto. Os dois mais importantes são o *DEVLOG*<sup>3</sup> e *CHANGELOG*<sup>4</sup>. O *DEVLOG* foi um documento necessário para mostrar o meu progresso e tempo gasto no projeto, enquanto este fazia parte da disciplina *MAC0214 Atividade Curricular em Cultura e Extensão*. Nele existem várias entradas detalhando atividades feitas no jogo durante o desenvolvimento. O documento *CHANGELOG* por outro lado serve para marcar lançamentos do jogo, descrevendo features novas adicionadas em cada *sprint* feita. Ambos documentos foram de grande utilidade no inicio do desenvolvimento de *PsyChO: The Ball* pois exibem concretamente o fluxo de progresso na criação do jogo.

## 2.3 Ferramentas

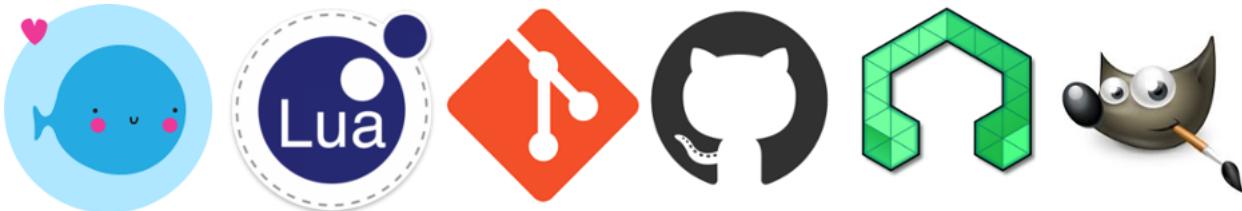
Uma das escolhas para o desenvolvimento de *PsyChO: The Ball* foi a utilização exclusiva de *Software Livre*. Essa escolha se deu tanto pela liberdade que essas ferramentas permitem em sua utilização, quanto pelo objetivo de incentivar mais o uso desse tipo de software, mostrando que são tão eficientes quanto o uso de software proprietários. Assim, como critério de escolha para cada área de desenvolvimento do jogo, foi determinado algum software livre que tenha uma comunidade ativa (para a solução de problemas e disponibilidade de bibliotecas), constante atualização de suas funcionalidades, versões estáveis e uma recomendação de seu uso no desenvolvimento de jogos.

O arcabouço utilizado no desenvolvimento do jogo foi a *Love2d* ou *LÖVE*<sup>5</sup>, uma *framework* grátil que utiliza a linguagem *Lua*<sup>6</sup>.

Como sistema de controle de versão foi utilizado *Git*<sup>7</sup> através da plataforma de uso grátil *Github*<sup>8</sup>.

Para a produção de música foi utilizado *LMMS*<sup>9</sup>, um software livre grátil para manipulação e criação de sons.

E por último, para manipulação de imagens foi utilizado a ferramenta *Gimp*<sup>10</sup>.



**Figura 2.2:** Da esquerda pra direita, ícones de *LÖVE*, *Lua*, *Git*, *Github*, *LMMS* e *Gimp*

Dentro dessas ferramentas, vale ressaltar um ponto muito interessante do arcabouço *LÖVE*. Seu *game-loop* tem o diferencial de deslocar todos as verificações de input do usuário como *callbacks assíncronos*. Estes são funções que a própria *framework* vai chamar quando ocorrer algum input do usuário, como ele pressionar um botão do teclado ou mover o mouse. Através de subrotinas, o arcabouço está constantemente aguardando sinais que ativem essas funções, e assim o desenvolvedor não precisa se preocupar em

<sup>3</sup><https://github.com/uspgamedev/Project-Telos/blob/dev/docs/DEVLOG.md>

<sup>4</sup><https://github.com/uspgamedev/Project-Telos/blob/dev/docs/CHANGELOG.md>

<sup>5</sup><https://love2d.org/>

<sup>6</sup><https://www.lua.org/portugues.html>

<sup>7</sup><https://git-scm.com/>

<sup>8</sup><https://github.com/>

<sup>9</sup><https://lmms.io/>

<sup>10</sup><https://www.gimp.org/>

lidar com otimização ou código confuso na hora de esperar e tratar comandos dados pelo usuário. Isso foi uma grande vantagem que determinou a escolha dessa ferramenta na produção de *PsyChO: The Ball*.



# Capítulo 3

## Proposta

Durante o desenvolvimento de um jogo, seja ele digital ou analógico, é de extrema importância que todos os detalhes (como mecânicas, estética ou até mesmo público alvo) estejam bem documentados em algum lugar para que todos que participam da criação do jogo estejam em sintonia. Desta forma é criado o *Game Design Document*, ou *GDD*.

O objetivo do *GDD* é servir como um guia para artistas, desenvolvedores ou músicos em um jogo. Entretanto ele não é um documento estático, pois um jogo nunca está completamente definido desde seu início. Características ou detalhes do jogo vão sendo acrescentadas ou removidas conforme vão se mostrando benéficas ou prejudiciais para a experiência desejada do projeto final. Desta forma fica a cargo do Game Designer de mantê-lo sempre atualizado.

Com isso vamos descrever um breve *GDD* do *PsyChO: The Ball* com todos os conceitos e estéticas que foram planejados até o momento.

### 3.1 Temática

*PsyChO: The Ball* é um *Top-Down Shooter* psicodélico, minimalista e frenético. Ele tem grandes influências de outros grandes jogos do mesmo gênero como *Hotline Miami*<sup>1</sup> ou jogos da série *Touhou Project*<sup>2</sup>. Além disso, foi pensado para ser um jogo difícil, desafiando o jogador constantemente. É esperado que ele fracasse bastante, mas melhore em cada partida nova, conforme vai melhorando seus reflexos.

O estilo psicodélico e minimalista foi inspirado no jogo *Hexagon*<sup>3</sup>. Quase todos elementos do jogo são composições de círculos com cores vibrantes que mudam ao longo do tempo. O jogador desta forma não fica sobrecarregado de informação na tela e pode focar sua atenção em desviar de projeteis e destruir inimigos.

O jogo foi desenvolvido para ser jogado com um teclado e mouse, porém com controles minimalistas para que, no futuro, tenha suporte à joysticks.

### 3.2 Mecânicas

*PsyChO: The Ball* segue a fórmula básica de jogos *Top-Down Shooters*: o jogador controla um círculo colorido, chamado de *PsyChO*, que pode se mover livremente no espaço da tela. Desta forma o jogador consegue esquivar de golpes inimigos ou obstáculos durante o jogo.

---

<sup>1</sup><http://www.devolverdigital.com/games/view/hotline-miami>

<sup>2</sup>[https://en.wikipedia.org/wiki/Touhou\\_Project](https://en.wikipedia.org/wiki/Touhou_Project)

<sup>3</sup><http://terrycavanaghgames.com/hexagon/>

Para interagir com os inimigos, *PsyChO* possui dois golpes: atirar (ataque ofensivo) ou usar um *ULTRABLAST* (ataque defensivo).

Atirar é o jeito normal de atacar e destruir inimigos. Segurando um botão e mirando com o mouse, o jogador escolhe onde vai atirar projéteis que andam em linha reta até colidir com um inimigo ou os cantos da tela de jogo. Os projéteis são bem pequenos e velozes, de forma que uma boa precisão seja necessária para acertar inimigos. Atirar não gasta nenhum recurso e pode ser utilizado a qualquer momento.

*ULTRABLAST* é um golpe extremamente poderoso, porém limitado. Ao pressionar um botão especial, *PsyChO* cria um anel em sua volta que se expande rapidamente. Esse anel consegue destruir um número grande de inimigos antes de desaparecer, sendo muito útil para escapar de situações de extremo perigo. Além disso, ao utilizar o golpe, *PsyChO* fica invulnerável por um curto período de tempo, dando uma folga para o jogador se reposicionar estratégicamente.

Por último, o jogador pode segurar um botão para entrar no modo *Focus*. Nesse modo, o raio de colisão do jogador diminui ligeiramente, e sua velocidade de movimento é drasticamente reduzida. Neste o jogador pode fazer movimentos mais precisos e delicados para desviar de obstáculos ou outros perigos.

O número reduzido de mecânicas (em comparação à outros jogos do mesmo gênero) foi uma escolha proposital no design. Com um arsenal limitado de ações para fazer, se torna um desafio tanto para o jogador (que vai ter que masterizar cada detalhe e nuância das ações disponíveis para enfrentar os obstáculos do jogo), quanto para o game designer, que vai ter de pensar em níveis e interações inovadoras que aprofundem o uso de cada mecânica para manter o interesse do usuário.

### 3.3 Gameplay

*PsyChO: The Ball* segue o modelo clássico e encontrado em video-games: níveis. O jogo será composto de 5 níveis sequenciais, cada um apresentando novos desafios e uma curva crescente de dificuldade. No fim de cada nível, o jogador enfrenta um *chefão*, sendo este um inimigo bem mais forte e que resiste muito mais aos golpes de *PsyChO*. Se consegue derrotá-lo, avança para o nível seguinte. Se sobreviver ao chefão final do último nível, o jogo termina em sucesso.

O jogador começa cada partida do jogo no primeiro nível com 10 vidas, e precisa enfrentar todos desafios que cada nível fornece. Estes desafios podem ser inimigos que atiram projéteis em direção ao jogador, inimigos que se multiplicam ou até mesmo inimigos invencíveis, onde o jogador precisará evitá-los até desaparecerem. Porém *PsyChO* é extremamente frágil, e qualquer objeto inimigo que o atinja é o suficiente para destruí-lo e lhe tirar uma vida. Cabe ao jogador aprender a utilizar seus golpes eficientemente e desvendar quando é mais prudente atacar ou fugir dos inimigos. Se a qualquer momento o jogador perde todas suas vidas, o jogo termina em fracasso e o jogador precisa recomeçar desde o início do nível em que perdeu.

*PsyChO* irá normalmente atirar para destruir inimigos, porém quando estiver em apuros, se o jogador tiver reflexos rápidos poderá utilizar seu golpe especial *ULTRABLAST* para matar inimigos próximos e ficar invulnerável temporariamente. *PsyChO* começa cada vida com 2 *ULTRABLASTs* e precisa destruir inimigos para ganhar mais.

Ao destruir inimigos, o jogador ganha pontos. Além do uso comum em jogos de usar pontos para recompensar o jogador e servir de métrica para seu progresso no jogo, ao acumular pontos o jogador ganha vidas e *ULTRABLASTs* para gastar. Desta forma o jogo influencia o jogador a jogar agressivamente para chegar mais longe.

Por último, cada nível é dividido em seções. Isso ajuda no level design do jogo de forma a organizar o fluxo do gameplay e melhor apresentar elementos novos. De maneira geral, as primeiras seções de um nível servem para apresentar inimigos ou conceitos novos que estão relacionados com o tema do nível. A

penúltima seção junta todos elementos novos apresentados, que deveriam se encaixar organicamente já que seguem um tema em comum. Por último, a seção final é reservada para o *chefão* do nível, que vai testar todo conhecimento aprendido pelo jogador até então.

### 3.4 Recursos Audiovisuais

Os recursos audiovisuais do jogo foram pensados para complementar a jogabilidade minimalista e psicodélica presente. Todos objetos no jogo são composições geométricas abstratas, com o intuito de facilitar a associação do jogador com o que é benéfico ou prejudicial. Cada tipo diferente de inimigo segue um padrão próprio de cores. *PsyChO* permuta entre muitas cores vibrantes. Além disso o fundo do jogo fica transitando entre cores com baixa saturação para não distrair o jogador.

Para complementar a experiência do jogo, a trilha sonora original é baseada em gêneros eletrônicos e psicodélicos, que combina muito bem com a jogabilidade frenética e cores pulsantes. Os efeitos sonoros possuem vários efeitos de ressonância e eco, típicos em jogos com a mesma temática.

Para deixar o jogo mais emocionante, *PsyChO: The Ball* é repleto de efeitos de partículas e explosões para criar a ilusão de que todas ações do jogador tem um impacto enorme no jogo.



# Capítulo 4

## Desenvolvimento

*PsyChO: The Ball* teve seu início em 2013 como *PsyChObALL*. Durante os primeiros meses o projeto servia como um meio prático de se aprender técnicas de programação e desenvolvimento de jogos, e aos poucos foi tomando proporções maiores até ter seu primeiro lançamento em agosto de 2013. O jogo ficou em desenvolvimento até meio de 2014, mas lentamente começou a receber menos atualizações conforme seus criadores foram focando em outros projetos.

Somente no segundo semestre de 2015 que ele ressurgiu como *PsyChO: The Ball* na disciplina **MAC0214 - Atividade Curricular em Cultura e Extensão**. O objetivo do projeto era revitalizar o *PsyChObALL* original, fazendo tudo do zero e utilizando todo conhecimento acumulado durante a graduação. O fim de 2016 teve como resultado um protótipo bem elaborado do jogo com todas mecânicas originais do *PsyChObALL* implementadas e melhoradas.

O desenvolvimento do jogo passou por várias fases. Segue neste capítulo alguma das infraestruturas mais importantes desenvolvidas durante e para o projeto.

### 4.1 STEAMING

Antes de começar o desenvolvimento direto do jogo, os primeiros meses foram direcionados em criar um template de jogos para o arcabouço *Love2D*. Esse template se tornou o *STEAMING* (*Simple TEmplAtE for MakINg Games*), e todo o código do jogo foi construído em cima dele.

O template possui duas características peculiares que foram vantajosas para o desenvolvimento do jogo: sua infraestrutura orientada objetos para manipular elementos num jogo e sua estrutura de desenhar objetos na tela.

Todos objetos no jogo herdam de uma classe básica chamada *Element*. Um *Element* pode possuir três atributos fundamentais: um tipo (chamado de *type*), um subtipo (chamado de *subtype*) e uma identificação única (chamada de *id*). Com esses três atributos é possível agrupar, identificar, selecionar ou até mesmo destruir elementos no jogo com muita facilidade, facilitando imensamente o desenvolvimento de jogos.

*STEAMING* reduz todo pipeline gráfico do jogo em camadas. Elementos do jogo desenháveis são atribuídos a alguma camada, e a própria infraestrutura do template vai percorrer todas elas em ordem e desenhar os objetos dentro de cada uma, chamando o método correspondente da classe. Com essa organização é possível diagramar o design gráfico do jogo bem mais facilmente, e não se preocupar com objetos sendo desenhados na ordem certa. Entretanto objetos na mesma camada são desenhados em uma ordem imprevisível, já que são armazenados dentro de uma tabela *hash* sem ordem específica.

## 4.2 Gerenciador de Saves

Para armazenar dados entre partidas, foi desenvolvido um gerenciador de saves. Este é responsável por armazenar as informações relevantes do jogo, como maiores pontuações, configurações de som escolhidas pelo usuário, se é a primeira vez que o jogador abriu o jogo, etc.

Todas informações são guardadas em tabelas de lua, que depois são transformadas no formato JSON e salvas no disco rígido. Para recuperar essas informações, o gerenciador de saves faz a transformação inversa toda vez que o jogo é inicializado e atribui os valores para suas respectivas variáveis.

## 4.3 Script de Fases

Para organizar o fluxo de eventos em cada nível do jogo, foi implementado um gerenciador de níveis. Assim foi possível abstrair a lógica de cada fase em scripts.

Cada script descreve em que sequência inimigos ou ações se desenrolam em cada nível do jogo. Isso facilitou imensamente no design dos níveis, especialmente quando foi necessário balancear o jogo, pois o script permite editar e visualizar facilmente cada nível em poucas linhas. Vários métodos foram criados para gerar padrões específicos de inimigos, dar vidas-extras para o Psycho ou até mesmo criar chefões.

Porém, para manter um ritmo durante o nível, e não fazer o gerenciador rodar todas as linhas de uma vez, foi necessário usar corotinas. Elas permitem paralelizar o código e gerar múltiplos pontos de retorno para que o jogo aguarde certas condições antes de continuar o script de um nível. Essas condições pode ser de tempo (como esperar 10 segundos antes de enviar a próxima sequência de inimigos) ou de aguardar até que não tenha um inimigo vivo na tela (utilizada por exemplo para esperar que o jogador destrua todos inimigos antes de avançar para o nível seguinte).

Para lidar com corotinas paradas ou criar novas corotinas, vários métodos auxiliares foram implementadas, sempre com o objetivo de abstrair o máximo possível para que mesmo um não-programador consiga criar seus próprios níveis no jogo.

## 4.4 Shaders

Uma boa descrição de *PsyChO: The Ball* seria: contém muitos círculos. Logo, como este é o elemento mais característico do jogo, é desejável que os círculos sejam visualmente agradáveis.

Inicialmente os círculos eram desenhados com os métodos imbutidos no arcabouço *LÖVE*. Entretanto estes não utilizam nenhum algoritmo de suavimento, então as bordas dos círculos ficam serrilhadas e não combinam com a estética do jogo.

Desta forma escrevemos um *shader* que desenha círculos com bordas suavizadas (ou como é conhecido esse efeito, *anti-aliasing*). Utilizando a linguagem de *shaders* da *LÖVE* (muito semelhante à linguagem de *shaders* do *OpenGL*, *GLSL*), um script genérico consegue desenhar qualquer círculo com bordas suavizadas dado o tamanho do seu diâmetro.

Além disso o jogo precisa de outro efeito realizado por *shaders*: borrar a tela, ou *blur*. Esse efeito acontece quando o jogador pausa o jogo, de forma que somente a interface da tela de pause fique nítida, enquanto todos outros elementos do jogo fiquem desfocados como se estivessem borrados.

Para atingir tal efeito criamos dois scripts de *shader*, um que aplica um efeito de borrão horizontal, e outro vertical. Assim ao utilizar os dois em sequência é possível chegar no resultado esperado de uma tela desfocada, criando um efeito muito legal enquanto o jogo estiver pausado.

## 4.5 Juiceness

Além desses efeitos com shaders, um grande foco no desenvolvimento do *PsyChO: The Ball* foi gasto em deixar o jogo mais *juicy*. Isso é realizado com vários efeitos pequenos mas que tentam melhorar a experiência do jogador.

Quase todas transições do jogo são feitas com alguma interpolação em vez de instantâneas. Um bom exemplo disso é a movimentação do personagem principal, *Psycho*. Em vez de ele ter uma velocidade constante quando se move, o *Psycho* tem uma leve aceleração quando começa seu movimento, e um leve desaceleração quando o jogador solta o comando de andar. Isso cria uma ilusão de movimento muito mais realística e agradável pro jogador em contraposto a uma velocidade que se inicia e acaba instantaneamente.

Outro efeito implementado no jogo para aumentar a *juiceness* é a explosão de partículas toda vez que um inimigo morre. Vários pequenos círculos de tamanho e velocidade variada explodem assim que um inimigo é destruído, dando mais emoção ao jogo e satisfação ao jogador.

Como último exemplo de efeito *juicy* no jogo, *PsyChO: The Ball* utiliza um clássico em jogos hoje em dia: tremer a tela quando o jogador morre. Outro efeito bem simples de se implementar, já que só é preciso transladar todos elementos na tela alguns pixels para a frente e para trás, porém que gera grande impacto e imersão ao jogador.

## 4.6 Problemas e Desafios

Durante o desenvolvimento de *PsyChO: The Ball* surgiram vários problemas e desafios. Para consertar bugs ou otimizar o código foi necessário aprender a fundo a linguagem lua, explorar bibliotecas disponíveis e até mesmo relembrar algoritmos aprendidos na graduação.

Um dos desafios mais interessantes que surgiu durante o desenvolvimento do jogo foi a transição de cor dos objetos. Para manter o tema psicodélico, quase todos objetos ficam transitando entre uma tabela de cores própria, porém essa transição nunca parecia natural e orgânica já que eu apenas interpolava os valores *RGB* de um objeto para os valores alvo. Para resolver isso aprendi a forma de representar cores *HSL*, que representa uma cor através de sua matiz, saturação e luminosidade. Com essa mudança, o jogo ficou com transições entre cores muito mais naturais e visualmente belas.



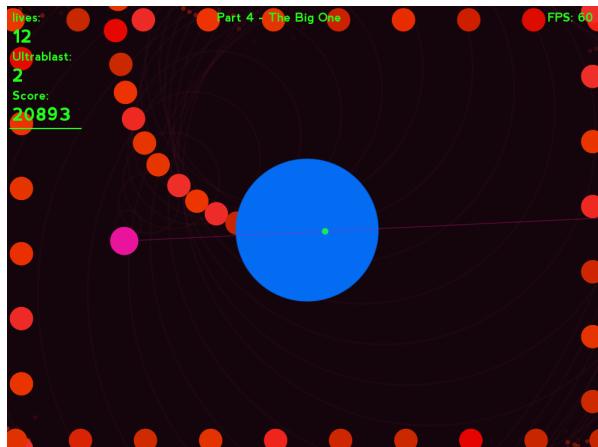
# Capítulo 5

## Resultados

Em dezembro de 2016 o protótipo de *PsyChO: The Ball* foi finalizado, e teve seu primeiro lançamento em um evento expositivo da *UspGameDev*. Nele vários alunos da *Universidade de São Paulo* puderam comparecer e jogá-lo gratuitamente, dentre outros jogos de membros do grupo de extensão. Foi uma imensa satisfação observar outras pessoas jogando e se divertindo com o jogo, mesmo ele sendo um protótipo.

Uma das maiores importâncias nesses eventos é poder receber *feedback* de outras pessoas. Estes variam dentre opiniões sobre mecânicas, descobrir se os conceitos de design escolhidos cumpriram seu propósito, ou até mesmo descobrir novos *bugs* durante o evento. Isso facilita imensamente o processo após uma exposição de polir e consertar onde for necessário.

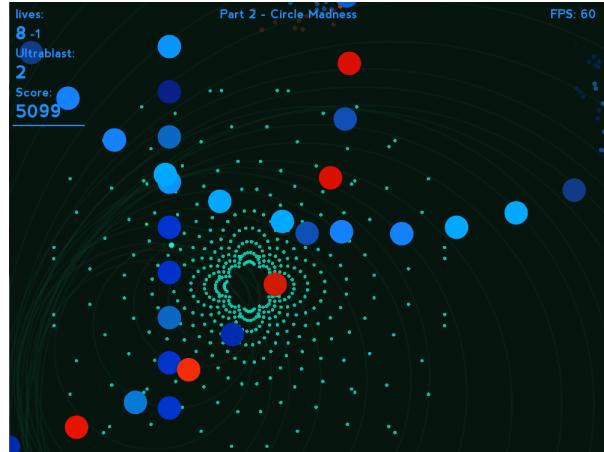
### 5.1 Conteúdo



**Figura 5.1:** Primeiro chefão do jogo atacando o jogador

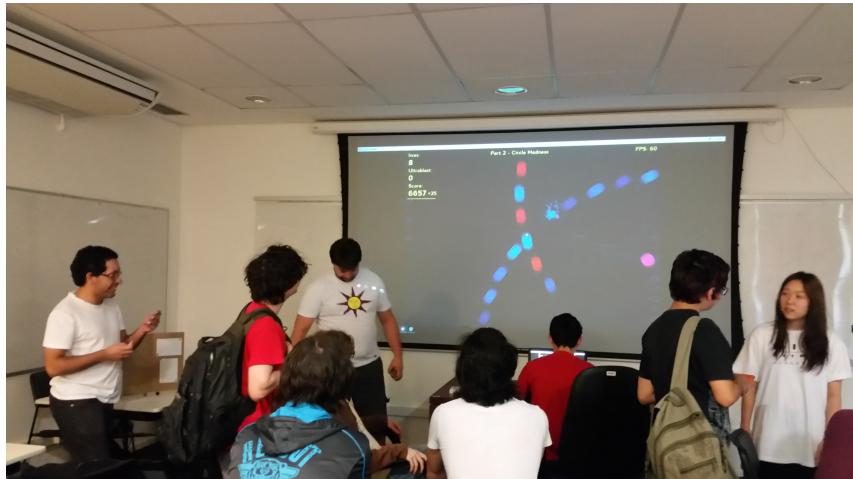
O jogo em seu estado atual possui 2 níveis dos 5 planejados. Cada nível tem sua própria trilha sonora, e é dividido em quatro partes únicas, sendo a última composta por um *chefão*. O jogo possui vários efeitos especiais visuais e sonoros, e abrange 5 inimigos diferentes para desafiar o jogador.

Além disso o jogo salva as maiores pontuações entre partidas, então jogadores podem sempre tentar superar a pontuação de colegas, ou tentar aumentar seu próprio recorde. O sistema para rodar níveis através de scripts permite facilmente que usuários criem e joguem seus próprios níveis sem precisar de muito conhecimento computacional.



**Figura 5.2:** Jogador morrendo para uma onda de inimigos lhe atacando

## 5.2 Eventos

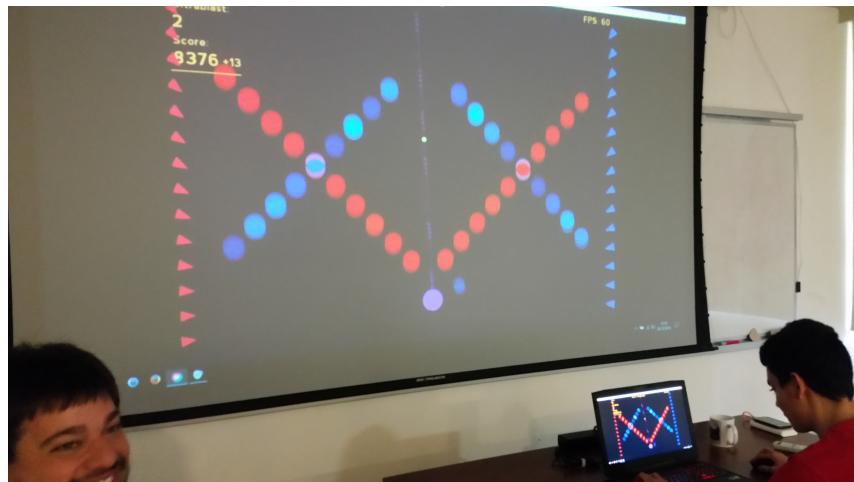


**Figura 5.3:** Evento **Let's (test) Play!**. No fundo *PsyChO: The Ball* - 02/12/2016

A primeira exposição pública de *PsyChO: The Ball* foi em dezembro de 2016, no evento aberto organizado pelo grupo *UspGameDev: Lets (test) Play!*. Nele todos integrantes do grupo tiveram uma chance de mostrar todo o progresso de desenvolvimento em seus projetos para a comunidade uspiana. Desta forma alunos, professores e funcionários puderam jogar, darem opiniões e acima de tudo se divertirem durante uma tarde nos jogos.

Foi um momento marcante no desenvolvimento de *PsyChO: The Ball*, pois receber o feedback de outras pessoas traz tanto uma utilidade para o aperfeiçoamento do jogo, quanto uma satisfação pessoal ao ver pessoas se envolverem num projeto que demorou meses para ser desenvolvido. Uma das maiores mudanças que surgiu nesse primeiro evento foi a implementação do sistema para armazenar pontuações máximas, assim jogadores podiam comparar resultados no jogo entre si.

O segundo grande evento ocorreu em julho de 2017, **II Let's (test) Play**. Nesta sequência do primeiro evento foi apresentado mais projetos de membros da *UspGameDev*. Além disso teve uma quantidade maior de pessoas comparecendo, até mesmo de pessoas não atreladas à *Universidade de São Paulo*. Foi possível mostrar todo o progresso no desenvolvimento do jogo nos 6 meses que se passaram, e novamente se fortalecendo com todo o feedback recebido pelos alunos e professores que quiseram jogar *PsyChO: The Ball*.



**Figura 5.4:** Aluno jogando *PsyChO: The Ball* no Let's (test) Play! - 02/12/2016



**Figura 5.5:** Foto do evento II Let's (test) Play - 05/07/2017

### 5.3 Como Jogar

*PsyChO: The Ball* possuí todos seus lançamentos disponíveis online em seu próprio repositório no domínio *Github*:

<https://github.com/uspgamedev/Project-Telos/releases>

Neste link é possível encontrar instruções de como instalar e jogar *PsyChO: The Ball* nos sistemas operacionais *Linux* (suporte direto para distros baseadas em *Ubuntu* ou *Debian*), *Windows* ou *Macintosh*.

Além disso é possível acessar diretamente a versão mais atualizada do código fonte através da página principal do repositório do projeto:

<https://github.com/uspgamedev/Project-Telos/>



# Capítulo 6

## Conclusões

Falar de coisas importantes que vou levar comigo para o resto da vida

### 6.1 Matérias Utilizadas

Falar de quais matérias da graduação eu usei e de que forma.

Também falar de atividades extracurriculares que me ajudaram, especialmente a UGD

### 6.2 Trabalhos Futuros

Falar em trabalhar mais no jogo para lançar em lugar X e meu interesse em continuar estudos na área de game design e desenvolvimentos de jogos.



# Referências Bibliográficas

- [Ada10] Ernest Adams. *Fundamentals of Game Design*. Segunda edição, 2010. [4](#)
- [Ber14] Kleber Bernardo. Kanban: Do início ao fim! <https://www.culturaagil.com.br/kanban-do-inicio-ao-fim/>, 2014. [5](#)
- [Nys14] Robert Nystrom. *Game Programming Patterns*. Primeira edição, 2014. [4](#)
- [Sch08] Jesse Schell. *The Art of Game Design: A Book of Lenses*. Primeira edição, 2008. [3](#)