

# 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.



## Agradecimientos

[illegible]



# Resumo

SOBRENOME, A. B. C. **Título do trabalho em português**. 2010. 120 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2010.

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).

[illegible][illegible]

**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 . . . . .	3
2.3	Ferramentas . . . . .	3
<b>3</b>	<b>Proposta</b>	<b>5</b>
3.1	Temática . . . . .	5
3.2	Mecânicas . . . . .	5
3.3	Gameplay . . . . .	6
3.4	Recursos Audiovisuais . . . . .	7
<b>4</b>	<b>Desenvolvimento</b>	<b>9</b>
4.1	STEAMING . . . . .	9
4.2	Gerenciador de Saves . . . . .	10
4.3	Script de Fases . . . . .	10
4.4	Shaders . . . . .	10
4.5	Juiciness . . . . .	11
4.6	Problemas e Desafios . . . . .	11
<b>5</b>	<b>Resultados</b>	<b>13</b>
5.1	Conteúdo . . . . .	13
5.2	Eventos . . . . .	13
5.3	Como Jogar . . . . .	13
<b>6</b>	<b>Conclusões</b>	<b>15</b>
6.1	Matérias Utilizadas . . . . .	15
6.2	Trabalhos Futuros . . . . .	15
	<b>Referências Bibliográficas</b>	<b>17</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

[illegible]

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto  
 texto texto texto texto texto texto texto.

## 1.2 Objetivos

[illegible]

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto  
 texto texto texto texto texto texto texto.



## Capítulo 2

# Conceitos, Técnicas e Ferramentas

Textotextotextotexto

## 2.1 Conceitos

Texto texto texto

## 2.2 Técnicas

[illegible]

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto  
texto texto texto texto texto texto texto.

## 2.3 Ferramentas

[illegible]

Texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto  
 texto texto texto texto texto texto texto



## 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 estrategicamente.

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 inovadores 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 chefe 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, m 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

Falar que consegui terminar um prototipo beta em dezembro de 2016. Falar do prazer de lançar um jogo e ver pessoas jogando e tendo experiencias próprias

Falar da importancia de feedback

### 5.1 Conteúdo

Falar que fiz 2 fases com 3 musicas com a ajuda de amigos. Mostrar varias fotos do jogo. Falar dos x tipos de inimigos e 2 chefoes.

### 5.2 Eventos

Falar dos eventos que levei (lets play 1 (dezembro de 2016), lets play 2 (2017) com melhorias depois de feedback). Mostrar umas fotos de pessoas jogando, e como eu quis colocar a mecanica de highscore graças ao feedback. Falar como nesses eventos achavam bugs que pude consertar depois (ou na hora)

### 5.3 Como Jogar

Colocar links onde da pra baixar e jogar o jogo.





## Capítulo 6

# Conclusões

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

### 6.1 Matérias Utilizadas

Falar de quais materias 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