

**Balanceamento Dinâmico de Dificuldade  
em Jogos Eletrônicos**

*um subtítulo*

Henrique Cerquinho

DISSERTAÇÃO APRESENTADA AO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
DA UNIVERSIDADE DE SÃO PAULO  
PARA OBTENÇÃO DO TÍTULO DE  
MESTRA EM CIÊNCIAS

Programa: Ciência da Computação

Orientador: Prof. Dr. Marco Dimas Gubitoso

São Paulo

10 de agosto de 2017



**Balanceamento Dinâmico de Dificuldade  
em Jogos Eletrônicos**

*um subtítulo*

Henrique Cerquinho

Esta é a versão original da dissertação  
elaborada pela candidata Henrique Cerquinho,  
tal como submetida à Comissão Julgadora.

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

*Esta seção é opcional e fica numa página separada;  
ela pode ser usada para uma dedicatória ou epígrafe.*



[illegible]





# Resumo

Henrique Cerquinho. **Balanceamento Dinâmico de Dificuldade em Jogos Eletrônicos: *um subtítulo***. Dissertação (Mestrado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2017.

O objetivo deste trabalho foi estudar e implementar o balanceamento dinâmico de dificuldade em jogos eletrônicos. Foi desenvolvido um protótipo de um jogo no estilo bullet-hell. Dada a infame e elevada dificuldade encontrada neste estilo de jogo, o foco principal do estudo foi de torná-lo acessível a todos os públicos, independente da habilidade do jogador. Foram adotadas funções matemáticas que levam em consideração diversas variáveis do jogo e definem uma métrica de dificuldade a ser usada no balanceamento de inimigos e recursos em tempo real. Foi enfim criado um ciclo de jogabilidade capaz de se adaptar às ações do jogador e dirigir o nível de dificuldade do jogo, procurando otimizar o entretenimento proposto.

**Palavras-chave:** Dificuldade. Balanceamento dinâmico. Lógica matemática. Jogos eletrônicos.



# Abstract

Henrique Cerquinho. **Dynamic Difficulty Balancing in Video Games: *a subtitle***.  
Thesis (Masters). Institute of Mathematics and Statistics, University of São Paulo, São  
Paulo, 2017.

The main goal of this thesis was to study and implement the dynamic balancing of difficulty in video games. With that purpose in mind, a bullet-hell style game prototype was developed. Given the infamous high difficulty in this game genre, the main focus of the study was to make it as accessible to all publics as possible, regardless of the players ability. Mathematical functions were used to consider many game variables and define a difficulty metric to be later used in the balancing of enemy difficulty and resource abundance, all in gameplay time. Thus, a full cycle of playability was developed, capable of adapting to the player's actions and steer the level of difficulty of the game, aiming to optimize the enjoyment.

**Keywords:** Difficulty. Dynamic balancing. Mathematical logic. Video games.



## Lista de Abreviaturas

ABNT	Associação Brasileira de Normas Técnicas
URL	Localizador Uniforme de Recursos ( <i>Uniform Resource Locator</i> )
IME	Instituto de Matemática e Estatística
USP	Universidade de São Paulo

## Lista de Símbolos

$\omega$	Frequência angular
$\psi$	Função de análise <i>wavelet</i>
$\Psi$	Transformada de Fourier de $\psi$

## Terminologia

Console	Aparelho dedicado a rodar um determinado grupo de jogos, com hardwares de int
Singleplayer	Modo/gênero de jogo em que se joga sozinho
Local Multiplayer	Modo/gênero de jogo em que se joga acompanhado de outros jogadores, no mesm
Online Multiplayer	Modo/gênero de jogo em que se joga acompanhado de outros jogadores, estes jog

## Lista de Figuras

## Lista de Tabelas

## Lista de Programas

A.1	Máximo divisor comum (arquivo importado). . . . .	7
A.2	Máximo divisor comum (em português). . . . .	7

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Objetivos . . . . .	1
<b>2</b>	<b>Dificuldade e Gêneros de Jogos</b>	<b>3</b>
2.1	Arcades e Consoles . . . . .	3
2.2	Cenário Atual . . . . .	4
2.3	Gênero Escolhido . . . . .	5
 <b>Apêndices</b>		
<b>A</b>	<b>Código-Fonte e Pseudocódigo</b>	<b>7</b>
 <b>Anexos</b>		
<b>A</b>	<b>Perguntas Frequentes sobre o Modelo</b>	<b>9</b>
 <b>Referências</b>		
		<b>11</b>
 <b>Índice Remissivo</b>		
		<b>13</b>





# Capítulo 1

## Introdução

### 1.1 Motivação

A dificuldade em um jogo tende a ser um elemento extremamente visível e importante para o seu aproveitamento. Jogos mais desafiadores são conhecidos por serem mais frustrantes, e os menos desafiadores, mais relaxantes. Porém, esse pensamento pode levar à falsa idéia de impossibilitação do aproveitamento por parte da dificuldade elevada. No artigo de Jesper [JUUL \(2009\)](#) é discutido o verdadeiro papel da dificuldade em jogos, de forma enxuta e didática. Quando apresentamos o conceito de vitória em um jogo, não faria sentido termos este sem o conceito de derrota. A derrota entra na experiência do jogo de forma a induzir os jogadores a reajustarem sua perspectiva perante ao jogo, como visto no estudo de Juul, levando o jogador a sempre pensar em novas estratégias e, consequentemente, adicionando conteúdo ao jogo.

O nível de dificuldade de um jogo pode ser visto como o responsável pelo desenvolvimento nas habilidades do jogador em experimentos como o de [APONTE \*et al.\* \(2011\)](#), porém, neste mesmo experimento, vemos ela também influenciando fortemente o aproveitamento geral do jogo, e podendo prejudicá-lo se não definida apropriadamente. Segundo [CRANDALL e SIDAK \(2020\)](#), jogos são uma das formas de entretenimento mais consumidas atualmente, totalizando uma receita de 159.3 bilhões de dólares no mundo inteiro em 2020. Portanto é natural associarmos jogos a entretenimento e, consequentemente, imaginarmos que um dado jogo é divertido, porém adequar a dificuldade de um jogo à experiência desejada prova-se um trabalho difícil.

### 1.2 Objetivos

Considerando a tese apresentada, é necessário manter em mente que um jogador pode se sentir "violado" pelo jogo se este sofre mudanças drásticas em sua dificuldade durante ou entre as sessões jogadas ([HUNICKE, 2005](#)). Com isso, foi almejado um sistema onde ocorrem ajustes dinâmicos de dificuldade em um jogo, porém de forma a se adaptar ao nível da habilidade do jogador. Isso é, se o jogador se adapta bem à dificuldade atual do jogo, esta tende a aumentar, propondo um desafio adequado à habilidade do jogador. Em

contrapartida, se o jogador é muito prejudicado pela dificuldade atual do jogo, esta tende a diminuir, com o fim de manter o jogador em um estado mais favorável à sua habilidade, mas que também o força a adaptar suas estratégias e seu estilo de jogo, permitindo-o "avançar" na dificuldade de jogo.

Dar uma tunada na introdução ainda

"aspa"

## Capítulo 2

# Dificuldade e Gêneros de Jogos

### 2.1 Arcades e Consoles

*Arcades*, mais comumente conhecidos como *Fliperamas*, foram introduzidas na década de 1930, com jogos mecânicos ainda hoje famosos, como *Pinball*. As máquinas eram projetadas com uma entrada de moeda, que quando inserida permitia que fossem jogados determinado número de vezes ("rounds" ou "vidas"). Não se afastando desse sistema, os jogos eletrônicos em gabinetes foram introduzidos apenas na década de 60 às *Arcades*. Como relatado por GREEN (2017), jogos como *Pac-Man*, *Donkey Kong*, *Space Invaders*, *Asteroids* e diversos outros títulos dominavam as *Arcades* no mundo inteiro.

A partir desse ponto, foram apenas introduzidas novas tecnologias aos jogos de *Arcade*. *Duck Hunt* foi considerado revolucionário por usar um controle no formato de um revólver, usado para mirar diretamente na tela, simulando um tiro-ao-alvo. Alguns anos depois, jogos de corrida como *Grand Prix* levaram desenvolvedores à idéia de um controle no formato de um volante e um pedal, simulando, similarmente ao *Duck Hunt*, um cenário de pilotagem mais próximo da realidade.

Por mais atraentes que os jogos de *Arcade* estavam se tornando, sua dificuldade tendia a surpreender mais ainda o jogador. Até este momento, o papel principal da dificuldade era de se aproveitar ao máximo do sistema de moedas do jogo. Quanto mais difícil o jogo, mais moedas seriam gastas para completá-lo, e poucos desenvolvedores deixavam de ver essa vantagem.

Com a entrada dos consoles, principalmente do *Nintendo Entertainment System*, também conhecido como *NES* e *Nintendinho*, jogos começaram a ser reestruturados, deixando de ter apenas uma fase e seu objetivo de arrecadar o maior número de pontos possível, e adquirindo uma estrutura mais linear, com diversas fases e, conseqüentemente, tendo seu objetivo principal deslocado para terminar todas as fases, ou derrotar o último chefe. Com isso, o papel da dificuldade sofreu uma leve mudança, deixando de buscar as moedas do jogador, e passando a ser a sustentação principal do tempo de jogo. Quanto mais difícil, mais tempo o jogador levaria para terminar o jogo, portanto maior aproveitamento seria tirado. Ideologias como essa moldaram a grande maioria dos jogos de *NES* como jogos extremamente difíceis de se completar.

## 2.2 Cenário Atual

Após uma onda de jogos notoriamente difíceis, desenvolvedores começaram a explorar a ideia de dificuldades selecionáveis em jogos, em busca de poderem atender públicos mais abrangentes com seus projetos. Estava então emergindo a mecânica de, ao começar um *novo jogo*, era apresentada a opção de dificuldade, normalmente variando entre *fácil*, *médio* e *difícil*, mas não se limitando a essas opções.

Variáveis do jogo como quantidade de pontos de vida do jogador e dos inimigos oscilavam de acordo com a dificuldade escolhida. Os primeiros grandes jogos de sucesso a explorar mais a fundo a mecânica de diversos níveis de dificuldade, como *Halo*, *Call of Duty*, saga *The Elder Scrolls*, saga *Final Fantasy*, entre outros, moldaram esse padrão para uma grande parcela dos jogos futuros, porém não a perfeccionando instantaneamente. Picos na dificuldade eram comumente encontrados em níveis mais altos no jogo. As últimas dificuldades, normalmente rotuladas como *Impossível* ou termos do gênero, costumavam inflar exageradamente variáveis do jogo como pontos de vida dos inimigos, muitas vezes tornando o jogo praticamente impossível e distanciando o jogador do cenário de entretenimento que os desenvolvedores se propunham a oferecer.

Em paralelo, é possível observar certas estruturas de jogos que permitem ao jogador explorar suas mecânicas de modo a afetar a dificuldade. Por exemplo, jogos no estilo *Online Multiplayer*<sup>1</sup>, jogadores frequentemente participam de partidas e deliberadamente evitam usar certos itens do jogo, de modo a propor um desafio maior a si mesmos, se exibir ou até mesmo estabelecer recordes. Neste último cenário, também vemos a grande presença das *Speedruns* em jogos *Singleplayer*. Estas consistem em o jogador tentar completar determinado jogo no período de tempo mais curto possível. Para isso, o jogo é estudado profundamente e suas estratégias são otimizadas, procurando pular fases, explorar pontos fracos de inimigos ou até mesmo *bugs* nos jogos. Essa cultura é popularmente conhecida por propor tarefas de dificuldade significativamente mais elevadas que o jogo "base", consequentemente demandando muito da habilidade do jogador, ao mesmo tempo que adiciona conteúdo ao jogo.

Com o decorrer das décadas de 2000 e 2010, desenvolvedores começaram a aprofundar a complexidade do balanceamento da dificuldade de seus jogos (BYTE, 1981). A introdução de *checkpoints*<sup>2</sup> tornou jogos extensos muito mais acessíveis (MORAN, 2010), por impedir que a derrota levasse o jogador de volta para o primeiro nível do jogo, e reduzindo o quanto do jogo teria que ser rejogado. Jogos mais datados são conhecidos por praticamente não possuírem checkpoints, o que contribuía fortemente para sua dificuldade elevada. É importante, porém, manter em mente que um uso excessivo de *checkpoints* pode trivializar as conquistas do jogador, uma vez que este não será apropriadamente punido por seus erros.

Neste mesmo período, com o avanço das Inteligências Artificiais (IAs), jogos *Single-*

---

<sup>1</sup>Em contrapartida ao *Singleplayer*, é um estilo de jogo onde diversas pessoas jogam uma partida juntas, em tempo real, através de seus respectivos aparelhos ou *consoles* e uma conexão à *internet*.

<sup>2</sup>Quando encontrado, garante que o jogador possa voltar a este ponto do jogo após ser derrotado. Funciona normalmente como uma "volta no tempo" para não punir tão severamente o jogador.

*player*<sup>3</sup> introduziram novas dinâmicas, não apenas de dificuldade, mas de jogabilidade no geral. Um dos gêneros que mais se beneficiou desse avanço foi o de jogos de luta. Originados da geração dos *arcades*, jogos de luta eram projetados no estilo *versus* para dois ou mais jogadores invariavelmente. Porém, com a entrada das IAs, foi possibilitado o modo de jogo para um único jogador, onde este enfrentaria um ou mais personagens controlados pelo próprio jogo, e, em muitos casos, com níveis de dificuldade selecionáveis. Com isso, uma grande gama de habilidades poderia ser atendida pelo jogo, ao mesmo tempo que era reduzida a necessidade da presença de outro jogador para que o jogo pudesse ser aproveitado ao máximo.

O recém-lançado *Risk of Rain 2*, desenvolvido pela *Hopoo Games*, tornou-se uma referência interessante para os estudos de dificuldade de jogos eletrônicos. Trata-se de um *Rogue-Like*<sup>4</sup> onde a dificuldade é aumentada com o decorrer do tempo, independente das ações do jogador, que por sua vez fica encarregado de "acompanhar" a dificuldade do jogo conforme esta avança. A progressão da dificuldade é baseada em valores como tempo de jogo, quantidade de jogadores, fase atual do jogador, entre outros valores, cuja fórmula será aprofundada em um capítulo mais adiante.

## 2.3 Gênero Escolhido

Com os avanços não apenas na tecnologia disponível para desenvolvimento de jogos, mas também na criatividade dos desenvolvedores, jogos eletrônicos começaram a tomar uma complexidade que tornou inevitável uma segmentação destes em gêneros. Atualmente, o gênero de um jogo não necessariamente definirá seu nível de dificuldade. Em grande parte dos gêneros de jogos podemos encontrar tanto jogos difíceis como fáceis. Porém, existe um gênero que se destaca entre os outros por englobar alguns dos jogos mais difíceis conhecidos, portanto o protótipo de jogo desenvolvido neste estudo seguiu este modelo de jogo.

O *Bullet Hell*, também conhecido como *Shoot 'em Up*, ou apenas *Shmup*, é um subgênero de jogos eletrônicos de tiro em 2D<sup>5</sup>, onde o jogador controla um único personagem, normalmente uma espaçonave ou avião, este disparando projéteis em diversos inimigos, enquanto foca paralelamente em desviar de uma quantidade demasiada de projéteis inimigos (DAVIES, 2008). O gênero teve sua origem com o famoso *Space Invaders* em 1978 e seus sucessores *Galaxian* e *Defender*, em 1980, mas sua emergência foi com a série *Radiant Silvergun* em 1998 (PICARD, 2013). Possibilitado pelo avanço na capacidade computacional tanto de computadores pessoais como de *consoles*, novas mecânicas e visuais foram introduzidos ao gênero rapidamente.

<sup>3</sup>Estilo de jogo projetado para ser jogado por uma única pessoa simultaneamente.

<sup>4</sup>Gênero de jogo onde os mapas são, normalmente, gerados aleatoriamente ou proceduralmente. A derrota do jogador o força a recomeçar o jogo desde o primeiro mapa, porém permitindo que sejam desbloqueados elementos novos para o jogo neste processo.

<sup>5</sup>Em um jogo 2D, o jogador controla a posição do personagem em apenas duas dimensões. No caso dos *Shmups*, o personagem se move apenas para cima, para baixo, esquerda e direita em um plano.



# Apêndice A

## Código-Fonte e Pseudocódigo

Com a *package listings*, programas podem ser inseridos diretamente no arquivo, como feito no caso do Programa ??, ou importados de um arquivo externo com o comando `\lstinputlisting`, como no caso do Programa A.1.

---

**Programa A.1** Máximo divisor comum (arquivo importado).

---

```

1  FUNCTION euclid( $a, b$ )  $\triangleright$  The g.c.d. of  $a$  and  $b$ 
2       $r \leftarrow a \bmod b$ 
3      while  $r \neq 0$   $\triangleright$  We have the answer if  $r$  is 0
4           $a \leftarrow b$ 
5           $b \leftarrow r$ 
6           $r \leftarrow a \bmod b$ 
7      end
8      return  $b$   $\triangleright$  The g.c.d. is  $b$ 
9  end
```

---

Trechos de código curtos (menores que uma página) podem ou não ser incluídos como *floats*; trechos longos necessariamente incluem quebras de página e, portanto, não podem ser *floats*. Com *floats*, a legenda e as linhas separadoras são colocadas pelo comando `\begin{program}`; sem eles, utilize o ambiente `programruledcaption` (atenção para a colocação do comando `\label{}`, dentro da legenda), como no Programa A.2<sup>1</sup>:

---

**Programa A.2** Máximo divisor comum (em português).

---

```

1  FUNCAO euclides( $a, b$ )  $\triangleright$  O máximo divisor comum de  $a$  e  $b$ 
2       $r \leftarrow a \bmod b$ 
3  enquanto  $r \neq 0$   $\triangleright$  Atingimos a resposta se  $r$  é zero
4           $a \leftarrow b$ 
5           $b \leftarrow r$ 
```

*cont*  $\longrightarrow$

---

<sup>1</sup>listings oferece alguns recursos próprios para a definição de *floats* e legendas, mas neste modelo não os utilizamos.

```

→ cont
6       $r \leftarrow a \bmod b$ 
7      fim
8  devolva  $b$  ▷ O máximo divisor comum é  $b$ 
9  fim

```

---

Além do suporte às várias linguagens incluídas em listings, este modelo traz uma extensão para permitir o uso de pseudocódigo, útil para a descrição de algoritmos em alto nível. Ela oferece diversos recursos:

- Comentários seguem o padrão de C++ (`//` e `/* ... */`), mas o delimitador é impresso como “▷”.
- “:=”, “<>”, “<=”, “>=” e “!=” são substituídos pelo símbolo matemático adequado.
- É possível acrescentar palavras-chave além de “if”, “and” etc. com a opção “`morekeywords={pchave1,pchave2}`” (para um trecho de código específico) ou com o comando `\lstset{morekeywords={pchave1,pchave2}}` (como comando de configuração geral).
- É possível usar pequenos trechos de código, como nomes de variáveis, dentro de um parágrafo normal com `\lstinline{blah}`.
- “`$....$`” ativa o modo matemático em qualquer lugar.
- Outros comandos LaTeX funcionam apenas em comentários; fora, a linguagem simula alguns pré-definidos (`\textit{}`, `\texttt{}` etc.).
- O comando `\label` também funciona em comentários; a referência correspondente (`\ref`) indica o número da linha de código. Se quiser usá-lo numa linha sem comentários, use `/// \label{blah}`; “`///`” funciona como `//`, permitindo a inserção de comandos LaTeX, mas não imprime o delimitador (▷).
- Para suspender a formatação automática, use `\noparse{blah}`.
- Para forçar a formatação de um texto como função, identificador, palavra-chave ou comentário, use `\func{blah}`, `\id{blah}`, `\kw{blah}` ou `\comment{blah}`.
- Palavras-chave dentro de comentários não são formatadas automaticamente; se necessário, use `\func{}`, `\id{}` etc. ou comandos LaTeX padrão.
- As palavras “Program”, “Procedure” e “Function” têm formatação especial e fazem a palavra seguinte ser formatada como função. Funções em outros lugares *não* são detectadas automaticamente; use `\func{}`, a opção “`functions={func1,func2}`” ou o comando “`\lstset{functions={func1,func2}}`” para que elas sejam detectadas.
- Além de funções, palavras-chave, strings, comentários e identificadores, há “`specialidentifiers`”. Você pode usá-los com `\specialid{blah}`, com a opção “`specialidentifiers={id1,id2}`” ou com o comando “`\lstset{specialidentifiers={id1,id2}}`”.



## Anexo A

# Perguntas Frequentes sobre o Modelo<sup>2</sup>

- **Não consigo decorar tantos comandos!**

Use a colinha que é distribuída juntamente com este modelo ([gitlab.com/ccsl-usp/modelo-latex/raw/master/pre-compilados/colinha.pdf?inline=false](https://gitlab.com/ccsl-usp/modelo-latex/raw/master/pre-compilados/colinha.pdf?inline=false)).

- **Por que tantos arquivos?**

O preâmbulo  $\text{\LaTeX}$  deste modelo é muito longo; as partes que normalmente não precisam ser modificadas foram colocadas no diretório `extras`, juntamente com alguns arquivos acessórios. Já os arquivos de conteúdo (capítulos, anexos etc.) foram divididos de maneira que seja fácil para você atualizar o modelo (copiando os novos arquivos ou com um sistema de controle de versões) sem que alterações no conteúdo de exemplo (este texto que você está lendo) causem conflitos com o seu próprio texto.

- **As figuras e tabelas são colocadas em lugares ruins.**

Veja a discussão a respeito na Seção ??.

- **Estou tendo problemas com caracteres acentuados.**

Veja a discussão a respeito na Seção ??.

- **Existe algo específico para citações de páginas web?**

Biblatex define o tipo “online”, que deve ser usado para materiais com título, autor etc., como uma postagem ou comentário em um blog, um gráfico ou mesmo uma mensagem de email para uma lista de discussão. Bibtex, por padrão, não tem um tipo específico para isso; com ele, normalmente usa-se o campo “howpublished” para especificar que se trata de um recurso *online*. Se o que você está citando não é algo determinado com título, autor etc. mas sim um sítio (como uma empresa ou um produto), pode ser mais adequado colocar a referência apenas como nota de rodapé e não na lista de referências; nesses casos, algumas pessoas acrescentam uma segunda lista de referências especificamente para recursos *online* (biblatex permite criar múltiplas bibliografias). Já artigos disponíveis *online* mas que fazem parte de uma publicação de formato

---

<sup>2</sup>Esta seção não é de fato um anexo, mas sim um apêndice; ela foi definida desta forma apenas para servir como exemplo de anexo.

tradicional (mesmo que apenas *online*), como os anais de um congresso, devem ser citados por seu tipo verdadeiro e apenas incluir o campo “url” (não é nem necessário usar o comando `\url{}`), aceito por todos os tipos de documento do bibtex/biblatex.

- **Aparece uma folha em branco entre os capítulos.**

Essa característica foi colocada propositalmente, dado que todo capítulo deve (ou deveria) começar em uma página de numeração ímpar (lado direito do documento). Se quiser mudar esse comportamento, acrescente “openany” como opção da classe, i.e., `\documentclass[openany,...]{book}`.

- **É possível resumir o nome das seções/capítulos que aparece no topo das páginas e no sumário?**

Sim, usando a sintaxe `\section[mini-titulo]{titulo enorme}`. Isso é especialmente útil nas legendas (*captions*) das figuras e tabelas, que muitas vezes são demasiadamente longas para a lista de figuras/tabelas.

- **Existe algum programa para gerenciar referências em formato bibtex?**

Sim, há vários. Uma opção bem comum é o JabRef; outra é usar Zotero ou Mendeley e exportar os dados deles no formato .bib.

- **Posso usar pacotes L<sup>A</sup>T<sub>E</sub>X adicionais aos sugeridos?**

Com certeza! Você pode modificar os arquivos o quanto desejar, o modelo serve só como uma ajuda inicial para o seu trabalho.

- **Como faço para usar o Makefile (comando make) no Windows?**

Lembre-se que a ferramenta recomendada para compilação do documento é o latexmk, então você não precisa do make. Mas, se quiser usá-lo, você pode instalar o MSYS2 ([www.msys2.org](http://www.msys2.org)) ou o Windows Subsystem for Linux (procure as versões de Linux disponíveis na Microsoft Store). Se você pretende usar algum dos editores sugeridos, é possível deixar a compilação a cargo deles, também dispensando o make.

- **Como eu faço para...**

Leia os comentários dos arquivos “tese.tex” e outros que compõem este modelo, além do tutorial (Capítulo ??) e dos exemplos do Capítulo ??; é provável que haja uma dica neles ou, pelo menos, a indicação da *package* relacionada ao que você precisa.

## Referências

- [APONTE *et al.* 2011] Maria-Virginia APONTE, Guillaume LEVIEUX e Stéphane NATKIN. “Difficulty in video games : an experimental validation of a formal definition”. Em: (2011), pg. 19 (citado na pg. 1).
- [BYTE 1981] BYTE. *“Zork, The Great Underground Empire.”* 1ª ed. Perspectiva, 1981, pgs. 262–264 (citado na pg. 4).
- [CRANDALL e SIDAK 2020] Robert W. CRANDALL e J. Gregory SIDAK. “Video games: serious business for america’s economy”. Em: (2020), pg. 48 (citado na pg. 1).
- [DAVIES 2008] Jonti DAVIES. “The shooting never stops”. Em: (2008) (citado na pg. 5).
- [GREEN 2017] Katie GREEN. “A short history on arcade gaming”. Em: (2017) (citado na pg. 3).
- [HUNICKE 2005] Robin HUNICKE. “The case for dynamic difficulty adjustment in games”. Em: (2005), pg. 6 (citado na pg. 1).
- [JUUL 2009] Jesper JUUL. “Fear of failing? the many meanings of difficulty in video games”. Em: (2009), pg. 13. URL: <https://www.jesperjuul.net/text/fearoffailing/> (citado na pg. 1).
- [MORAN 2010] Chuk MORAN. “Playing with game time: auto-saves and undoing despite the ‘magic circle’”. Em: (2010). URL: <http://sixteen.fibreculturejournal.org/playing-with-game-time-auto-saves-and-undoing-despite-the-magic-circle/> (citado na pg. 4).
- [PICARD 2013] Martin PICARD. “The foundation of geemu: a brief history of early japanese video games”. Em: (2013). URL: <http://www.gamestudies.org/1302/articles/picard> (citado na pg. 5).



# Índice Remissivo

## B

biblatex, [9](#)

bibtex, [9](#)

## C

Captions, *veja* Legendas

Código-fonte, *veja* Floats

## E

Equações, *veja* Modo Matemático

## F

Figuras, *veja* Floats

Floats

Algoritmo, *veja* Floats, Ordem

Fórmulas, *veja* Modo Matemático

## I

Inglês, *veja* Língua estrangeira

## L

Legendas, [10](#)

## M

Mendeley, [10](#)

## P

Palavras estrangeiras, *veja* Língua estrangeira

## R

Rodapé, notas, *veja* Notas de rodapé

## S

Subcaptions, *veja* Subfiguras

Sublegendas, *veja* Subfiguras

## T

Tabelas, *veja* Floats

## V

Versão corrigida, *veja* Tese/Dissertação, versões

Versão original, *veja* Tese/Dissertação, versões

## Z

Zotero, [10](#)