



**Universidade de Brasília**

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

## Ajuste dinâmico de dificuldade em jogos a partir de variáveis do jogo e do usuário

João Victor Santos Aguiar  
Lucas Vanderlei Fernandes

Monografia apresentada como requisito parcial  
para conclusão do Bacharelado em Ciência da Computação

Orientadora  
Prof.<sup>a</sup> Dr.<sup>a</sup> Carla Denise Castanho

Brasília  
2017

Universidade de Brasília — UnB  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Rodrigo Bonifácio

Banca examinadora composta por:

Prof.<sup>a</sup> Dr.<sup>a</sup> Carla Denise Castanho (Orientadora) — CIC/UnB

Prof. Dr. Edson Alves da Costa Júnior — FGA/UnB

Prof. Dr. Tiago Barros Pontes e Silva — DIN/UnB

### **CIP — Catalogação Internacional na Publicação**

Aguiar, João Victor Santos.

Ajuste dinâmico de dificuldade em jogos a partir de variáveis do jogo e do usuário / João Victor Santos Aguiar, Lucas Vanderlei Fernandes. Brasília : UnB, 2017.

62 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2017.

1. Ajuste dinâmico de dificuldade, 2. jogos eletrônicos, 3. fluxo, 4. dados fisiológicos, 5. EDA

CDU 004

Endereço: Universidade de Brasília  
Campus Universitário Darcy Ribeiro — Asa Norte  
CEP 70910-900  
Brasília-DF — Brasil



# Dedicatória

Dedicamos ao nossos pais, amigos e colegas de curso que sabem a longa e difícil jornada enfrentada para conseguir nosso diploma em Ciência da Computação.

# Agradecimentos

Agradecemos a nossa orientadora Profa. Carla Castanho, aos professores Tiago Barros e Edson Alves por participarem da banca avaliadora, ao mestrando em Design Marcos Akira pelo auxílio nas questões relativas ao tratamento das emoções e na elaboração do questionário de estado do fluxo. Agradecemos também ao doutorando em Ciência da Computação Elton Sarmanho pela atenção e auxílio prestado durante a execução deste trabalho.

# Resumo

A percepção do nível de dificuldade em jogos eletrônicos é algo pessoal e à medida que este mercado evolui surgem novos mecanismos de ajuste de dificuldade para se adaptar a cada jogador. Atualmente, a maioria dos jogos alteram sua dificuldade com base no desempenho do jogador. Contudo, diversas pesquisas têm sido feitas em modelos que ajustam a dificuldade em função do estado afetivo do jogador, com auxílio de sensores. Este trabalho tem como objetivo investigar se um mecanismo de Ajuste Dinâmico de Dificuldade (ADD) híbrido, isto é, que utiliza medidas de desempenho do jogador em conjunto com dados sobre o estado afetivo do mesmo, faz o jogador ter uma melhor percepção do estado de *flow* em comparação com modelos que utilizam apenas uma destas duas variáveis. Para tanto, um jogo existente foi adaptado com o ADD híbrido, utilizando o sensor EDA (*Electrodermal Activity*) para coleta dos dados fisiológicos do jogador. Foram conduzidos testes com dezesseis voluntários, nos três modelos de ADD, ou seja, ADD baseado em dados de desempenho, ADD baseado em dados fisiológicos e ADD híbrido. Além de jogar o jogo com um dos modelos de ADD, cada participante também respondeu a um questionário que quantifica o quanto o estado de *flow* foi percebido. A média das respostas daqueles que jogaram o modelo híbrido foi maior que daqueles que jogaram os outros dois modelos baseados em apenas uma variável.

**Palavras-chave:** Ajuste dinâmico de dificuldade, jogos eletrônicos, fluxo, dados fisiológicos, EDA

# Abstract

The perception of the level of difficulty in electronic games is something personal and as the market evolves new mechanisms of adjustment of difficulty arise to adapt to each player. Currently, most games change their difficulty based on the player's performance. However, several researches have been done on models that adjust the difficulty according to the affective state of the player, with the aid of sensors. This paper aims to investigate whether a hybrid Dynamic Difficulty Adjustment (DDA), that is, a mechanism that uses measures of player performance in conjunction with data regarding the affective state of the player, makes the player have a better perception of the state of flow compared to models that make use of only one of these two variables. To do so, an existing game was adapted with the hybrid ADD, using the EDA (Electrodermal Activity) sensor to collect the physiological data of the player. Trials were conducted with sixteen volunteers playing one of three models of DDA, ie DDA performance data based, DDA physiological data based and DDA hybrid. In addition to playing the game with one of the DDA models, each participant also responded to a questionnaire that quantifies how much the state of flow was perceived. The average responses of those who played the hybrid model were higher than those who played the other two models based on only one variable.

**Keywords:** Dynamic difficult adjustment, games, flow, physiological data, EDA

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Definição do problema . . . . .	2
1.2	Objetivo . . . . .	2
1.3	Metodologia . . . . .	2
1.4	Estruturação . . . . .	3
<b>2</b>	<b>Embasamento Teórico</b>	<b>4</b>
2.1	Jogo . . . . .	4
2.2	Taxonomia de jogos . . . . .	5
2.3	Jogador . . . . .	6
2.3.1	Classificação de jogadores . . . . .	6
2.4	Emoções . . . . .	6
2.5	Atividade Eletrodermal . . . . .	7
2.6	Fluxo . . . . .	8
2.7	Ajuste de dificuldade em jogos . . . . .	10
2.8	Ajuste de Dificuldade Dinâmica . . . . .	11
2.9	Maximizando Satisfação do Jogador . . . . .	13
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>14</b>
3.1	Avaliação do nível de dificuldade . . . . .	14
3.2	Controle da dificuldade da IA de um oponente . . . . .	15
3.3	ADD e Inteligência Artificial . . . . .	15
3.4	Estudo de caso em ADD . . . . .	16
3.5	Boa experiência em jogos adaptativos . . . . .	17
3.6	Efeitos da adaptação na imersão do jogador . . . . .	17
3.7	Criação de inimigos através do uso de ADD . . . . .	18
3.8	Análise da afetividade no ADD . . . . .	19
3.9	ADD afetivo . . . . .	20

<b>4</b>	<b>O Jogo</b>	<b>21</b>
4.1	Conceito . . . . .	22
4.2	Detalhamento . . . . .	23
4.2.1	Jogo Original . . . . .	23
4.2.2	Jogo Adaptado . . . . .	24
4.2.3	Problemas e Soluções . . . . .	27
4.2.4	Controles . . . . .	27
4.2.5	Dificuldade . . . . .	27
4.3	Implementação . . . . .	28
4.3.1	Engine . . . . .	28
4.3.2	Componentes . . . . .	29
4.3.3	Dificuldade . . . . .	32
4.3.4	Código . . . . .	33
<b>5</b>	<b>Ajuste Dinâmico de Dificuldade Híbrido</b>	<b>34</b>
5.1	Unity . . . . .	35
5.2	BITalino . . . . .	36
5.3	MATLAB . . . . .	36
5.4	ADD . . . . .	37
<b>6</b>	<b>Testes</b>	<b>40</b>
6.1	Teste para balanceamento de dificuldade . . . . .	41
6.2	Comparação de <i>flow</i> entre ADDs . . . . .	44
<b>7</b>	<b>Considerações Finais</b>	<b>46</b>
	<b>Referências</b>	<b>48</b>
	<b>Apêndice</b>	<b>50</b>
<b>A</b>	<b>Questionário</b>	<b>51</b>
A.1	Objetivo . . . . .	51
A.2	Desafio . . . . .	51
A.3	Imersão . . . . .	52
A.4	Dados sociodemográficos . . . . .	52

# Lista de Figuras

2.1	Gráfico com as diferentes emoções geradas pela combinação de excitação e valência [1] . . . . .	8
2.2	Diagrama de canal de fluxo [2] . . . . .	9
2.3	Jogo de plataforma Super Mario World . . . . .	12
3.1	Exemplo do algoritmo de <i>Mont Carlo Tree Search</i> . . . . .	19
3.2	Jogo criado para testar Ajuste dinâmico afetivo . . . . .	20
4.1	Asteroids [3] . . . . .	21
4.2	Jogo Original . . . . .	24
4.3	Componentes do jogo . . . . .	26
4.4	Movimentação da Nave do Jogador . . . . .	28
4.5	Interface da Engine Unity . . . . .	29
4.6	Diagrama de estados do jogo adaptado . . . . .	31
4.7	Mapa da Fase 5 . . . . .	31
5.1	Placa <i>BITalino</i> ajusta para o sensor EDA [4] . . . . .	37
5.2	Interface do MATLAB . . . . .	37
6.1	Exemplo de um participante realizando os testes com o sensor posicionado na palma da mão esquerda . . . . .	40
6.2	Seção Desafio do Questionário . . . . .	44

# Lista de Tabelas

2.1	Exemplo de taxonomia dos jogos eletrônicos (adaptada de [5]). . . . .	5
4.1	Densidade na dificuldade baixa e na Fase 0 . . . . .	32
4.2	Densidade na dificuldade média . . . . .	32
4.3	Densidade na dificuldade alta . . . . .	32
4.4	Velocidade por dificuldade . . . . .	32
5.1	ADD Híbrido: Dificuldade Alta . . . . .	38
5.2	ADD Híbrido: Dificuldade Média . . . . .	38
5.3	ADD Híbrido: Dificuldade Baixa . . . . .	39
6.1	Adaptação de dificuldade baseada em mortes: Nível 0 . . . . .	42
6.2	Adaptação de dificuldade baseada em mortes: Nível 1 e 2 . . . . .	42
6.3	Adaptação de dificuldade baseada em mortes: Nível 3 e 4 . . . . .	43
6.4	Definição de estados baseada no sinal tônico sensor EDA, usando a diferença da media dos dez menores valores com a media total . . . . .	43
6.5	Média das respostas do Questionário . . . . .	45

# Capítulo 1

## Introdução

A intensidade de prazer que um jogador sente na maioria das vezes é dependente do nível de habilidade dele em relação à dificuldade do jogo. Um jogo muito difícil para um jogador casual leva a uma experiência frustrante, enquanto um jogador experiente pode sentir tédio se o jogo for fácil demais.

Para mitigar essa distância entre a dificuldade do jogo e o nível de habilidade do jogador surgiu o conceito de ajuste dinâmico de dificuldade (ADD) [6]. A ideia principal do ADD consiste em obter meios no jogo de avaliar a performance e habilidade do jogador e modificar a dificuldade em função desta avaliação.

Enquanto o objetivo principal do ADD é maximizar a diversão do jogador, isso também pressupõe um modelo de diversão existente que pode ser usado para maximizar a satisfação do jogador [7]. Para verificar o nível de satisfação do jogador é usado a teoria de fluxo oriunda da Psicologia, que consiste em um estado psicológico no qual o indivíduo está altamente concentrado em uma dada tarefa, cujo desafio está à altura das habilidades do indivíduo [8].

Quando um ADD é implementado num jogo, é fundamental que o jogador não perceba que a dificuldade está sendo manipulada, pois estudos [9] mostram que há uma queda na satisfação do jogador ao perceber tais ajustes. Além disso, também existe por parte do usuário o abuso das mecânicas do ADD para facilitar o jogo.

Os primeiros ADDs construídos utilizavam como base de mudança a performance do jogador por meio da coleta de dados *in-game*, como por exemplo a adaptação do jogo *Half-life* [6] que verificava a performance e fazia uma economia de itens de vida.

Estudos mais recentes começaram a investigar o uso de afetividade como base para o ADD [9], verificando que estas podem representar com mais precisão a dificuldade sentida pelo jogador. Em outras palavras, cria-se a ideia de um ADD afetivo [10] que utiliza sensores para captar os dados fisiológicos do usuário, os quais serão interpretados por um sistema, que irá manipular a dificuldade do jogo em tempo de execução.

## 1.1 Definição do problema

Os estudos analisados sobre ajuste de dificuldade dinâmica em jogos costumam levar em conta somente as variáveis dos jogadores ou aquelas que podem ser medidas pelo jogo. Individualmente esses ADDs, que levam em conta somente dados *in-game* [11] [12] [13], ou apenas dados fisiológicos [10] [9], se mostraram eficientes em manter o jogador em um estado de fluxo. Contudo não encontrou-se, nas bases de dados da *IEEE Xplore Digital Library*<sup>1</sup>, da ACM (*Association for Computing Machinery*)<sup>2</sup> e do *Google Scholar*<sup>3</sup>, um trabalho que busque uma combinação das duas abordagens de ADD, tão pouco uma investigação sobre sua eficiência em fazer o jogador se sentir em um estado fluxo.

## 1.2 Objetivo

Neste trabalho, o objetivo é verificar se um ADD híbrido faz o jogador se sentir mais no estado de *flow*, em comparação aos modelos de ADD comumente usados. Um ADD híbrido se baseia na combinação de dados *in-game*, isto é, dados de desempenho coletados do próprio jogo, e de dados fisiológicos do jogador. Os ADDs usuais, ou não híbridos, são aqueles que utilizam apenas umas dessas fontes de dados. Objetiva-se implementar o ADD híbrido, onde os dados fisiológicos serão obtidos por meio de um sensor de condutância da pele, cuja confiabilidade já foi testada em estudos anteriores [10]. Os dados recebidos pelos sensores serão usados como base para identificar o nível de frustração ou ansiedade do jogador, e assim ajustar a dificuldade do jogo de modo que o jogador tenha uma melhor percepção do estado de fluxo, garantindo uma melhor imersão [9].

## 1.3 Metodologia

Para atingir o objetivo proposto, será feita a adaptação de um jogo *open-source* chamado *Asteroids* [14], desenvolvido em *Unity*. Trata-se de um jogo em 2D que consiste em uma nave espacial que precisa achar uma pilula azul no meio de um campo de asteroides. A adaptação consiste da adição de 5 fases com três níveis de dificuldade, diversos tipos de asteroides e uma mudança de objetivo, sendo necessário passar pelo campo de asteroides e chegar no fim dele para avançar à próxima fase.

O ADD híbrido proposto será implementado nesse jogo usando tanto a performance do jogador como a avaliação da afetividade do jogador. A combinação destas informações determinará qual a dificuldade da próxima fase. A performance será medida por meio

---

<sup>1</sup><http://ieeexplore.ieee.org/>

<sup>2</sup><http://www.acm.org/>

<sup>3</sup><https://scholar.google.com.br/>

de dados extraídos do jogo e para avaliação da afetividade será utilizado um sensor de atividade eletrodermal (EDA) com tratamento dados feito externamente no *MATLAB* e enviado para o jogo por meio de mensagens na porta UDP.

Após essa implementação serão conduzidos testes com jogadores, além da verificação, por meio de um questionário adaptado, se o jogador se sente em um estado de fluxo. [15]

## 1.4 Estruturação

Segue como foi feita a estruturação do restante deste trabalho:

**Capítulo 2:** para melhor compreensão do assunto abordado neste trabalho, os fundamentais conceitos teóricos são apresentados e definidos com base em publicações passadas.

**Capítulo 3:** os trabalhos diretamente correlacionados com ADD e de importância para a realização desta pesquisa são descritos neste capítulo.

**Capítulo 4:** Apresenta detalhes sobre a escolha do jogo no qual os ADDs foram inseridos bem como os detalhes de implementação desse jogo.

**Capítulo 5:** introduz o conceito de ADD híbrido e apresenta o detalhamento de como o mesmo foi construído e implementado no jogo adaptado.

**Capítulo 6:** tem como foco a descrição dos testes realizados com o ADD proposto, bem como a apresentação dos resultados dos mesmos.

**Capítulo 7:** são apresentados algumas considerações finais em face aos resultados obtidos bem como sugestões de trabalhos futuros.

# Capítulo 2

## Embasamento Teórico

Neste capítulo são discutidos as diversas definições e conceitos que servirão de base para este trabalho, tais como jogo, sua taxonomia e classificação de seus jogadores, emoções e a atividade eletrodermal, que pode medir afetividade, e por fim os tipos de ajuste de dificuldade.

### 2.1 Jogo

A definição de jogo já foi trabalhada por muitos autores. Essas definições foram compiladas por Jasper Juul [16] e a partir disso ele chegou a definição que usaremos como base no nosso trabalho. Segundo Jasper, um jogo é um sistema formal baseado em regras com um resultado variável e quantificável, onde a diferentes resultados são assinalados diferentes valores, o jogador exerce um esforço para influenciar o resultado, o jogador se sente emocionalmente ligado ao resultado e as consequências da atividade são opcionais e negociáveis [16].

Apesar de alguns autores citarem problemas nessa definição, ela é adequada para jogos eletrônicos, pois esses se consistem de um sistema cujas regras não são negociáveis, considerando também as *cheats*, cujo uso já é levado em consideração na implementação do jogo.

Dentro de jogo, também são levados em consideração dois fatores importantes: jogabilidade e *gameplay*. Segundo Xavier [17], a jogabilidade é uma decorrência natural do lidar com o jogo, como o termo usabilidade o é para com produtos e processos de manipulação. Jogabilidade como característica intrínseca ao conjunto de ações que são esperados do jogador para com todos os seus componentes, sejam eles audiovisuais, sejam eles emocionais ou mesmo puramente cognitivos.

*Gameplay* é definido por Salen e Zimmerman [18] como um processo reativo, delimitado por regras e emergente de sequências de ações pertencentes a um ou mais jogadores.

Essencialmente, as regras governando a exploração do espaço de possibilidades do jogo por esse processo são as mecânicas de jogo: mecânicas tem uma grande influência sobre a experiência do jogador.

## 2.2 Taxonomia de jogos

Para este trabalho é importante destacar a questão da taxonomia de jogos, pois a alteração de dificuldade de cada jogo é particular ao seu gênero, resultando na implementação de um ADD diferente para cada jogo. Para uma adaptação de *Half-life*, por exemplo, foi feito um ADD baseado em economia de itens, tendo mais ou menos itens de vida espalhado pela fase dependendo do desempenho do jogador [6], enquanto que para uma adaptação de *Super Mario* um ADD focou na adaptação da distância entre as plataformas do jogo em função do número de mortes do jogador [19]. A Tabela 2.1 mostra algumas taxonomias com base na jogabilidade, gênero e quantidade de jogadores.

Tabela 2.1: Exemplo de taxonomia dos jogos eletrônicos (adaptada de [5]).

Classificação		Exemplos
Quanto o estilo		
Jogos de ação e habilidade	Luta Tiro Plataforma	Street Fighter, Mortal Kombat Space Invaders, Quake 3 Super Mario Bros, Castlevania
Jogos de estratégia	Turnos RPG Explore, expand, exploit, exterminate	Civilization Diablo, Final Fantasy Starcraft, Civilization
Simuladores	Transportes Militares Construção Sociabilização	Flight Simulator America's Army Sim City, Zoo Tycoon The Sims, Second Life
Esportes	Corrida Coletivos Individuais	Pole Position, F1 2010 FIFA Soccer, NBA Jam Virtua Tennis, Chessmaster
Educativos	Matérias escolares Conhecimento Treinamento	Coelhinho Sabido Carmen Sandiego, Show do Milhão
Quanto ao numero de jogadores		
	<i>Single player</i> <i>Multiplayer</i>	Uncharted, Sonic Contra, World of Warcraft

## 2.3 Jogador

Uma vez definido o conceito de jogo, é possível então conceituar o termo jogador. Essa conceito é importante, pois buscar-se neste trabalho melhorar a experiência do usuário durante o jogo. Logo, é importante saber como é seu relacionamento com o jogo, ou seja, o que lhe dá prazer ao jogar, qual nível de desafio ele procura, o que o motiva a jogar, o que ele busca no jogo e quais os fatores no jogo que pode ser amplificados para melhorar sua experiência de jogo.

Quando nos referimos a jogador, pode-se dizer que é o participante de um jogo qualquer. Como na seção anterior, há jogos diferentes para jogadores diferentes. Contudo, deve-se considerar também que existem tipos de jogadores diferentes.

### 2.3.1 Classificação de jogadores

É comum os sites de entretenimento, empresários e acadêmicos dividirem os jogadores em grandes categorias comportamentais. Essas categorias são, por vezes, separadas por nível de dedicação ao jogo, ou pelo tipo primário do jogo, ou por uma combinação desses e mais fatores. Não há um consenso sobre as definições ou nome dessas categorias, embora existam muitas tentativas de formalizá-las. Para este trabalho, são considerados dois tipos de jogadores, os casuais e os experientes, como é comumente adotado pela comunidade de desenvolvedores [20].

Jogadores casuais são aqueles que jogam com menos frequência, preferem jogos rápidos que não exigem tanta dedicação e proporcionam uma distração rápida. Este nicho tem preferência por jogos de celulares ou via *web*, comumente referidos como jogos casuais.

Já os jogadores experientes, mais conhecidos como jogadores *hardcore*, são aqueles que dedicam bastante tempo em jogos, escolhendo por vezes jogos mais complexos, que tem narrativas ou mecânicas mais elaboradas. Como dito por Fortugno, os jogadores casuais não abordam o mesmo conjunto de habilidade dos jogadores experientes, portanto, possuem níveis diferentes de exploração do jogo e paciência com falhas [21].

Por estarem mais acostumados a explorar e experimentar, os jogadores dedicados são então mais tolerantes a frustrações com falhas e com dificuldade de assimilar conhecimento. Já os casuais não seguem estas convenções, nem estão condicionados a tolerar falhas.

## 2.4 Emoções

Os sentimentos formam uma parte importante na identidade humana. Eles definem nossas interações sociais, comportamento, processos de decisão e de outra forma moldam o jeito

de como percebemos e interagimos com o mundo ao nosso redor.

Uma teoria amplamente aceita no mundo acadêmico desenvolvida por Lang [22], sugere que as emoções são classificadas melhores em base de dois componentes inter-relacionados: a excitação e a valência. Em termo simples, a excitação descreve a intensidade de uma emoção sentida, que vai desde de calma até a excitação. Valência identifica a positividade ou a negatividade inerente de uma emoção. Baixa valência está associada a emoções desagradáveis, como tristeza ou depressão, enquanto que a felicidade e euforia são exemplos de emoções com alta valência. Estes dois componentes não existem de forma independente um do outro. Para descrever com precisão um estado emocional, é preciso identificar a excitação e valência da emoção sentida. Como tal, existe uma relação entre os dois conceitos de modo que diferentes combinações de valência e excitação produzem emoções diferentes (Figura 2.1).

Obviamente, as emoções não são simplesmente abstratas ou estados invisíveis da mente. Mudanças na emoção se manifestam por meio de alterações fisiológicas no corpo humano. Algumas dessas mudanças são claramente visíveis a olho nu: transpiração, fluxo sanguíneo cutâneo (corar ou empalidecer) e pilo ereção (elevação involuntária de pelos no corpo) podem todas ser observadas por meio da visão a olho nu. Outras respostas, tais como alterações na frequência cardíaca e condutância da pele, ocorrem dentro do corpo humano e só podem ser avaliadas por meio da análise empírica de medições de dispositivos e sensores especializado. Facilmente observáveis ou não, todas as manifestações fisiológicas de emoção são acionadas pelo sistema nervoso autônomo, um ramo do sistema nervoso que é responsável por controlar a resposta de fuga ou luta do corpo e de funções corporais involuntárias, como a respiração e a frequência cardíaca. Como tal, é amplamente aceito que o estado emocional de uma pessoa pode ser inferido por meio da medição de alterações fisiológicas no corpo. As emoções também são uma parte vital de nossa experiência com entretenimento interativo, ajudando a formar decisões feita durante o tempo de jogo. [23] Para fazer uma medição do estado afetivo podemos fazer o uso de sensores, como por exemplo o EDA(Atividade Eletrodermal).

## 2.5 Atividade Eletrodermal

A secreção das glândulas sudoríparas é um processo que permite ao nosso corpo regular sua temperatura, mas também está associada à atividade do sistema nervoso simpático. Sempre que nos tornamos excitados (por exemplo, nervosos) ou relaxados, esse estado é parcialmente traduzido na produção de suor ou inibição nas glândulas em nossas mãos, palmas e pés. Isso muda a resistência da nossa pele. O monitoramento da atividade eletrodermal (EDA) permite a tradução dessas mudanças de resistência em valores numéricos,



Figura 2.1: Gráfico com as diferentes emoções geradas pela combinação de excitação e valência [1]

permitindo seu uso em uma ampla gama de aplicações. Usos conhecidos deste sensor incluem mapeamento afetivo, o teste de polígrafo (também conhecido como detector de mentiras), e também *biofeedback* de estresse/relaxamento [4]. Este foi o sensor escolhido para ser usado na implementação deste trabalho, como será explicado no Capítulo 5.

## 2.6 Fluxo

Jogos de videogames que tendem a fazer sucesso tem como um fator comum a capacidade de trazer o foco do jogador para o jogo. Eles permitem as pessoas se abstrair do mundo real para dentro da realidade do jogo, provendo uma distração dos problemas e preocupações do dia a dia. Algumas pessoas se envolvem tanto em um jogo que não percebem as coisas ao seu redor, como o tempo passado desde que ela começou a jogar, ou até mesmo outra pessoa chamando por ela. Nesses momentos, sua atenção está quase totalmente voltada para o jogo, resultando na sensação de que o personagem controlado é o próprio jogador.

Essa experiência é referida como imersão, um termo que é bastante usado por jogadores e pela mídia de jogos. Imersão é comumente vista como crítica para o aproveitamento do jogo, sendo ela o resultado de uma boa experiência de jogo.

O ser humano entra em um estado de completa imersão em uma determinada atividade quando está o proporciona com um desafio suficiente para que ele não perca o interesse por tédio e não seja além das suas habilidades, evitando que ele se sinta frustrado pela dificuldade desproporcional. Este estado foi definido pelo psicólogo e filósofo Mihaly Csikszentmihalyi, e foi nomeado como fluxo [8]. A consequência usual de um usuário entrar no estado de fluxo é um aumento do autocontrole, concentração focada na atividade em questão, uma clara visão dos objetivos, perda de autoconsciência, sensação alterada do tempo, aumento na motivação pessoal e uma fusão de ação e percepção. Quanto maior a

habilidade do usuário maior precisa ser o desafio para manter o usuário no estado de fluxo. A Figura 2.2 representa um modelo do estado de fluxo, representado por um gráfico onde um dos eixos é o desafio da tarefa e o outro eixo a habilidade do jogador. O estado de fluxo acontece quando existe um equilíbrio entre os dois. Mihaly em seu trabalho também identificou oito componentes necessário para se atingir um estado de fluxo:

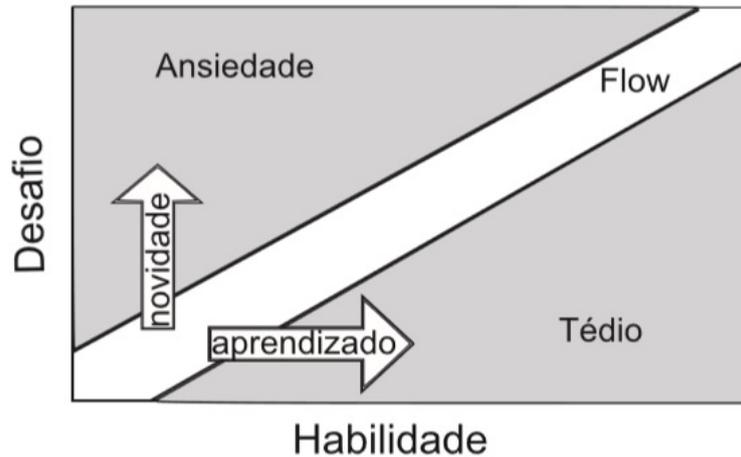


Figura 2.2: Diagrama de canal de fluxo [2]

- Uma atividade desafiadora que requer habilidade;
- Uma mistura de ação e percepção;
- Objetivos claros;
- Um *feedback* direto e imediato;
- Concentração na atividade sendo realizada;
- Sensação de controle;
- Perda da consciência própria;
- Sensação de tempo alterada.

Quando esses critérios são atingidos, uma pessoa entra em um estado de fluxo. Ela fica completamente imersa na atividade e perde o senso de tempo. No domínio de jogos, fluxo pode ser melhor visualizado como um balanceamento entre o desafio e a habilidade. Na Figura 2.2, essa relação é melhor demonstrada e o estado de fluxo pode ser facilmente percebido nela. Novidades no jogo levam a ansiedade, enquanto que o aprendizado leva ao tédio. O balanceamento entre novidade e aprendizado leva então ao estado de fluxo.

O conceito de fluxo aplicado no consumo de mídia é fundamental, pois o aproveitamento de um produto se dá devido ao resultado de experiências no estado fluxo. Em

concordância com Sherry [24], o estado fluxo pode ser atingido especialmente quando estamos jogando jogos digitais, pois estes possuem os requisitos ideais para criar e manter o estado fluxo com o equilíbrio entre a habilidade do jogador e o desafio do jogo [24]. Para conseguir esse equilíbrio, pode ser usado um ajuste de desafio, para manter a habilidade proporcional ao desafio.

## 2.7 Ajuste de dificuldade em jogos

Videogames servem para uma gama de propósitos e casos de uso, mas sua função geral é entreter pessoas. Obviamente, não existe um jeito pré-definido de fazer um jogo divertido. Durante o processo de desenvolvimento de jogo, um número grande de aspectos do design devem ser considerados. Um aspecto essencial é a dificuldade. Um jogo é entediante quando é fácil demais e frustrante quando é difícil demais. Dificuldade é subjetiva e varia de acordo com o nível de habilidade do jogador. De um modo geral, há duas maneiras de um jogo modular o grau de dificuldade que ele oferece ao jogador: por meio da seleção de dificuldade, ou do ajuste dinâmico de dificuldade.

O conceito de dificuldade em jogos é derivado da relação entre habilidade e desafio, quanto maior a habilidade necessária para resolver o desafio, maior a dificuldade. Para os jogadores cotidianos, pode ser difícil separar esse conceito essencial aos jogos e dissociá-lo da separação de dificuldade comumente aceita como *easy*, *normal* e *hard*.

Jogos como *pong* e *Space War!*, sendo alguns dos primeiros jogos eletrônicos criados, por serem simples não usavam o conceito de dificuldade, pois o desafio era único e a habilidade constante. Já *Mario* e *Mega Man* da NES (*Nintendo Entertainment System*) possuem uma escala de desafio que aumenta gradativamente a cada fase, enquanto a série *Elder Scrolls* faz essa escala com o nível do personagem. Vale ressaltar também a série *Mario Kart* que possui um nível de desafio que é adaptado à habilidade do jogador.

O ajuste de dificuldade poder ser essencialmente separado em 3 categorias [2]:

**Estáticos:** Não oferecem um mecanismo dinâmico de ajuste de dificuldade. Possuem um único nível de dificuldade ou permitem ao jogador escolher qual o nível que o satisfaz. Como exemplos de jogos temos o *Pong* dificuldade única e a série *God Of War* que no início do jogo deixa o jogador escolher a dificuldade desejada.

**Parcialmente adaptativos:** São aqueles que, ou possuem uma forma automática de detectar que a habilidade do jogador é superior à dificuldade atual, mas não realizam a mudança automaticamente, ou possuem um mecanismo que escala a dificuldade somente em uma direção, seja aumentando ou diminuindo, de maneira que a mesma seja compatível com o jogador. A série *Elder Scrolls* está nessa categoria, assim como a série *Mega-man zero*.

**Adaptativos:** Possuem um mecanismo dinâmico de ajuste de dificuldade, aumentando ou diminuindo automaticamente a dificuldade de forma a balancear o jogo à habilidade do jogador. Essa categoria será mais expandida na seção seguinte.

## 2.8 Ajuste de Dificuldade Dinâmica

O Ajuste de Dificuldade Dinâmica (ADD) em jogos eletrônicos é uma técnica emergente que procura adaptar o nível de dificuldade de um jogo enquanto este é jogado para poder se adaptar às habilidades de cada jogador específico. Devido ao custo de implementação e complexidade, os mecanismos de ajuste de dificuldade dinâmica ainda não evoluíram para englobar todos os jogos [25]. Essa técnica foi desenvolvida para ir contra os níveis de dificuldade pré-determinados. A dificuldade pode ser alterada de diversos meios diferentes, dependendo do gênero e do objetivo do designer do jogo. Assim como o mecanismo pode se basear em diversas variáveis para poder adaptar seu jogo.

A medida que os videogames continuam a fazer parte do cotidiano, seu público-alvo se expande e diversifica. Por essa razão, é difícil se esperar que um jogo com níveis de dificuldade estática atendam um público com uma variedade de habilidade e traços emocionais. Além de que, o uso desse método estático pode causar tédio ou estresse indevido em jogadores porque o jogo não tem nenhuma forma de avaliar o nível de desafio que melhor corresponda às habilidades do jogador.

Para abordar a rigidez de níveis de dificuldade fixa, o corpo acadêmico vem realizando pesquisas que tem desenvolvido o ajuste de dificuldade dinâmica, o qual, acima de tudo, busca adaptar dinamicamente a dificuldade de um jogo para um nível compatível de habilidade do jogador atual. Em um nível básico, o ajuste dinâmico funciona mudando fatores no jogo e variáveis em tempo real para adequar o desafio ao jogador. Para ajudar nesse trabalho, implementações de ADD tipicamente usam três módulos interligados:

- uma *engine* de monitoração que grava métricas brutas da performance do jogador (e.g tempo levado pra completar o nível, vida, número de mortes), e passa para a *engine* de análise. Pode coletar também dados de sensores psicológicos especializado para determinar o nível de tensão do jogador, entre outras coisas.
- A *engine* de análise, que computa os dados brutos que ela recebe para determinar quais elementos do jogo devem ser ajustados para prover uma melhor experiência para o jogador.
- E a *engine* de controle que ajusta os componentes de jogo como ditado pela *engine* de análise.

Com o design certo, uma grande porção dos elementos do jogo pode ser ajustado dinamicamente, esses elementos podem ser classificados em três grupos principais:

- Atributos do personagem principal: como vida, distância de pulo e dano infligido no inimigo.
- Atributos de NPC (especialmente inimigos): como personagens não controlados pelo jogador são controlados diretamente pelo jogo, um grande número de fatores pode ser ajustado dinamicamente. Esses incluem não somente atributos fundamentais, como vida, mas também processo de decisões, como algoritmos de procura de caminho (*path-finding*). Logo, quão melhor o jogador joga, mas inteligente fica o inimigo.
- Variáveis de mundo e de nível: nos trabalhos relacionados que serão apresentados no próximo capítulo, pode ser visto que a estrutura e design dos níveis de jogo tem provado ser impactante na dificuldade do jogo. Com isso em mente, os desenvolvedores podem projetar seu jogo de maneira que o mundo seja alterado sutilmente e transparente de maneira a ajudar o jogador ao longo de sua jornada. Por exemplo, num jogo de plataforma como *Super Mario Bros* (Figura 2.3) que depende bastante de pular plataformas, a distância entre essas pode ser ajustada para reduzir a ênfase na habilidade de pulo e precisão de tempo do pulo.



Figura 2.3: Jogo de plataforma Super Mario World

No capítulo a seguir, que trata dos trabalhos relacionados, será enfatizado que o sistema de ajuste de dificuldade dinâmica deve ser invisível ao jogador, pois ao perceber que o jogo está sendo mudado pode haver abuso do jogador através da manipulação das

mudanças ou pode haver uma perda de imersão no jogo, interferindo de forma negativa na experiência de jogo.

Exemplos do uso de mecânicas de ajuste de dificuldade dinâmica em jogos também serão explorados no Capítulo 3.

## **2.9 Maximizando Satisfação do Jogador**

As diversas abordagens usadas para maximizar a satisfação do jogador podem ser classificadas em duas categorias: implícitas e explícitas [7]. Abordagens implícitas consistem em alterar parâmetros que são periféricos à satisfação do jogador. Abordagens explícitas baseiam-se na maximização do valor de uma função que representa a satisfação do jogador. Para maximizar a satisfação do jogador, será usado neste trabalho um mecanismo de dificuldade dinâmico, logo a abordagem usada faz parte da categoria implícita. Entretanto, a maioria de trabalhos nessa categoria focam apenas em variáveis derivadas do jogo. Neste trabalho serão utilizados sensores fisiológicos para avaliar o estado emocional do jogador e com base nesses dados criar uma relação com os dados provenientes do jogo, para deste modo alterar a dificuldade do jogo.

# Capítulo 3

## Trabalhos Relacionados

Neste capítulo são apresentados alguns artigos correlatos cujas investigações e resultados foram úteis no âmbito deste trabalho. Cada seção foca num assunto relevante e destaca os pontos principais das pesquisas estudadas.

### 3.1 Avaliação do nível de dificuldade

Com o intuito de entender o conceito de dificuldade no ambiente de jogos de plataforma com níveis gerados automaticamente, os autores de [26] fizeram um estudo nas questões principais relacionadas com dificuldade em jogos de plataforma. Mais precisamente, o trabalho propôs um método que mede a dificuldade em um jogo de plataforma se baseando na probabilidade de sucesso.

Citando também um uso prático do princípio de dificuldade proposto em alguns níveis de jogos comerciais existentes, o conceito de jogos de plataforma usado pelos autores do trabalho acima é o conceito abordado por Compton e Matias [27], onde os níveis são definidos com um ponto de início e um de fim, e alguns possuem objetos que machucam o jogador.

O artigo usa duas considerações sobre a experiência do jogador: (1) o usuário tem consciência do que está fazendo e cada desafio é reconhecido e (2) tarefas triviais não são consideradas para a dificuldade do nível.

Para medir a dificuldade de uma fase de um jogo de plataforma os autores consideraram a probabilidade de o jogador obter sucesso em um pulo, de desistir e de tentar novamente. Calculando a combinação das probabilidades de sucesso de cada situação de pulo em uma fase vai resultar em uma estimativa para a dificuldade daquela fase em particular. Além disso, alterações na posição das plataformas, posições de inimigos, alturas e distâncias entre plataformas podem ser usadas para facilitar ou dificultar uma fase.

O mapeamento da dificuldade em jogos comerciais e o teste com jogadores foi a próxima etapa do trabalho que fez uso do jogos *Super Mario* e *Little Big Planet*. Usando a fórmula proposta pelos autores foi possível estimar se uma fase é mais difícil que a outra, entretanto, vale ressaltar que isso não representa a probabilidade real de sucesso da fase.

Esta pesquisa fornece uma melhor compreensão sobre como determinar a dificuldade de um jogo de plataforma, e conseqüentemente fornece subsídios para a criação de um mecanismo de ajuste de dificuldade dinâmica.

## 3.2 Controle da dificuldade da IA de um oponente

Esta pesquisa relatada em [28] propõe usar uma fórmula *Challenging Rate* para descobrir a dificuldade de jogos de Estratégia em Tempo Real (*Real Time Strategy*). Tal fórmula pode ser usada para descobrir se o balanceamento do jogo está adequado ou não. O jogo usado é baseado em defesa de linha, onde cada lado quer destruir a torre do outro. Neste jogo existem três dificuldades pré-determinadas: fácil, médio e difícil, implementadas na inteligência artificial do oponente. Tais dificuldades são baseadas nos recursos recebidos, quanto maior a dificuldade mais recursos para construir suas unidades o oponente recebe por tempo.

A fórmula proposta pelos autores analisará os pontos de vida total de todas as unidades, o ataque total das unidades, quantas unidades e quantos recursos ele possui. Para validar a fórmula, foram realizados testes onde ambos os lados eram controlados pela AI previamente programada e definida no artigo. As seguintes combinações de dificuldade foram testadas: fácil e fácil, médio e médio, difícil e difícil, fácil e médio, fácil e difícil, médio e difícil.

Os autores do trabalho concluíram que a fórmula proposta consegue medir a dificuldade, e mostra que o equilíbrio existe entre os níveis de dificuldade determinados e quanto mais balanceado estão ambos lados, mais tempo vai demorar a partida.

Para o gênero de *Real Time Strategy*, o artigo estudado mostrou que utilizando a inteligência artificial do oponente é possível controlar a dificuldade e possivelmente medi-la com a fórmula proposta na pesquisa. Isso viabiliza uma possível adaptação para outros gêneros de jogos.

## 3.3 ADD e Inteligência Artificial

Em [13], foi testado uma nova abordagem para o ajuste de dificuldade dinâmica em jogos, isto é, uma IA(Inteligência Artificial) foi evoluída ortogonalmente, onde o jogador recebe suporte de agentes colaboradores que são evolvidos junto com agentes oponentes

(onde os colaboradores e oponentes tem incentivos ortogonais). A vantagem é que a dificuldade do jogo pode ser ajustada de forma mais granular por meio da manipulação de dois eixos independentes, ou seja, por ter colaboradores mais ou menos adeptos, e por ter adversários mais ou menos adeptos. Além disso, a interação humana pode ser informada pelo desempenho e comportamento dos agentes colaboradores. Desta forma, as IAs ortogonalmente evoluídas tanto facilitam um ajuste mais suave da dificuldade quanto permitem novas experiências de jogo.

Os testes realizados mostram que tal evolução ortogonalizada em inteligências artificiais é uma área promissora no ajuste de dificuldade dinâmica.

### 3.4 Estudo de caso em ADD

Publicado em 2005, os autores de [6] argumentam que jogos com dificuldades estáticas muitas vezes não são capazes de manter a atenção do jogador, pois se este for muito experiente e o jogo não for difícil nem desafiador, ele perderá o interesse por tédio. Por outro lado, se o jogo for muito difícil para aquele jogador, este perderá o interesse por frustração. Jogos precisam deixar o jogador se sentir no estado de fluxo [8].

Mecanismos de ajuste dinâmico de dificuldade eram pouco explorados. O artigo examina requerimentos básicos de design para um ajuste dinâmico de dificuldade efetivo e apresenta um sistema interativo do mesmo e resultados de uma avaliação preliminar.

O mecanismo para ajuste dinâmico da dificuldade do jogo desenvolvido regula a economia do jogo, manipulando-a para verificar o progresso do jogador. O artigo discute como as mudanças afetam a percepção da dificuldade do jogo e a experiência do jogador, sendo que essas mudanças devem manter o balanceamento do jogo sem obstruir o progresso do jogador.

Na realização deste experimento os autores desenvolveram um jogo utilizando o *framework Mechanics Dynamics Aesthetics* para manter objetivos fundamentais do design e respeitar a experiência do jogador. A mecânica é baseada em diagramas de estados de um jogo do gênero *First-person Shooter*.

A dinâmica vai de fase para fase, com encontros variados e prêmios para desafios vencidos. Em jogos deste gênero o desafio e a variabilidade aumentam com o tempo e ajuda demora mais para vir.

Para criação do jogo foram feitas mudanças na SDK de *Half-Life*, que consistiram em alterações para controlar a economia do jogo, baseado no desempenho do jogador (probabilidade de morte) e mudanças de acordo com uma política de ajustamento. Estima-se a performance do jogador, se ele teve um melhor desempenho, o jogo fica mais difícil, caso contrário, fica mais fácil.

Os participantes jogaram por 15 minutos e responderam a um questionário sobre o jogo e sua experiência. O resultado demonstrou que jogadores morriam menos e jogadores experientes reportaram maior aproveitamento do jogo. A percepção de dificuldade como descrita pelos autores foi de 3 até 5 em uma escala de 1 a 5.

Esse resultado não mostrou correlação entre avaliação da dificuldade e a dificuldade ajustada. Jogadores novos assumem que não vão ser muito bons no jogo sem pensar na própria habilidade enquanto jogadores mais experientes vão assumir que terão um bom rendimento pela experiência. O artigo aponta que não foi conclusivo o suficiente para dizer se a dificuldade ajustada melhora a experiência do jogo ou não.

O artigo contribuiu para criar uma noção do início do estudo em mecanismos de ajuste de dificuldade dinâmica em jogos, bem como o entendimento do modo como tais mecanismos foram idealizados e suas concepções iniciais.

### **3.5 Boa experiência em jogos adaptativos**

O trabalho apresentado em [7] usa modelos de satisfação derivados de processos de modelagem de entretenimento para ajustar parâmetros do jogo *Bug-Smasher* e propõem a construção de uma função que mapeia o jogo e o usuário examinando o nível de satisfação reportado. Baseando-se em estudos prévios e modelos para determinar a preferência de entretenimento de crianças com uma acurácia de 77.77% foi feito o controle do jogo. A adaptação do jogo tem como base uma função que mapeia a satisfação do jogador com variáveis do próprio jogador e do jogo, mas não leva em consideração os sinais fisiológicos. As variáveis utilizadas foram: tempo de resposta, pressão do pé, número de interação (variáveis do jogador), curiosidade e diversidade de inimigos (variáveis do jogo).

Para determinar se o jogo adaptado obteve sucesso em otimizar a satisfação do jogador, foi realizado um teste com o público-alvo de *Bug Smasher*, no qual após jogar uma versão adaptada e uma versão não adaptada, os mesmos eram questionados sobre qual era mais divertida. Com base nas respostas, foi observado uma preferência estatística pelo jogo adaptado, indicativo que o modelo obteve sucesso em adaptar o jogo ao jogador.

### **3.6 Efeitos da adaptação na imersão do jogador**

O estudo apresentado por Denisova [12] busca o uso de ADD para adaptar o desafio do jogo à habilidade do jogador. Para tanto, os autores fizeram uso da manipulação de tempo como método de adaptação simples para explorar os efeitos na experiência do jogador. Durante o estudo eles consideram o risco do jogador não sentir o desafio. Além disso, foi

de suma importância que o jogador não soubesse sobre a manipulação, pois isto poderia prejudicar sua imersão no jogo.

A tese que os autores do trabalho buscaram comprovar foi de que o balanceamento do jogo de acordo com a habilidade do jogador levaria a uma melhora potencial na experiência do jogador o que acarretaria em períodos maiores de jogo.

No estudo foram feitas sessões com 2 grupos de jogadores, um com manipulação de tempo e outro sem. O jogo usado nos testes foi com a temática zombies e adaptado de um tutorial de *Unity*. Os jogadores eram encorajados a conseguir um score de 300 em 90 segundos. No grupo experimental o tempo foi baseado na habilidade do jogador e calculado num intervalo de 20 segundos com um fator de 1.4 resultando num máximo de 108 segundos se ele está tendo um desempenho pior, e mínimo de 72 segundos se ele está jogando melhor.

O resultado do estudo mostrou que os jogadores no grupo experimental acharam o jogo mais imersivo, o que contribui para área de ADD, fortalecendo o argumento de que mecanismos de ajuste dinâmico de dificuldade em jogos fazem com o jogador se sinta mais imerso no jogo, chegando a um estado de fluxo.

### 3.7 Criação de inimigos através do uso de ADD

Na época em que [11] foi publicado, a maioria das inteligências artificiais eram implementadas por uma FSM (máquina de estados finitos) que tem como desvantagem uma participação intensiva do designer, cujas estratégias e dificuldade não podem ser ajustadas dinamicamente por limitação das FSM.

Assim, o autor propôs duas técnicas para contornar essa questão:

- *Time constrained CI (computer intelligence)*
- *Knowledge based Time constrained CI*

O primeiro usa o algoritmo *Mont Carlo Tree Search* (Figura 3.1) que é um método de aprendizado usado para estimação, aproximação e escolha da estratégia mais adequada. É recomendado para *standalone-PC's* por fazer um uso mais intensivo do recursos do Sistema.

O segundo usa uma rede neural artificial baseada nos dados coletados por um *CI-controlled-NPC*. Serve para jogos *multiplayers* online tanto quanto *standalone-PC's*

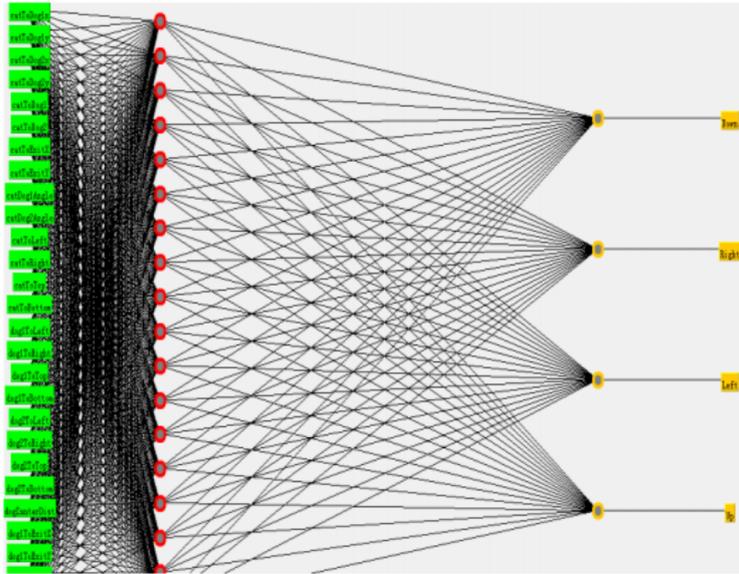


Figura 3.1: Exemplo do algoritmo de *Mont Carlo Tree Search*

### 3.8 Análise da afetividade no ADD

Na ocasião da publicação de [9], a maioria dos mecanismos de ajuste de dificuldade dinâmica se baseavam no desempenho do jogador para poder determinar a dificuldade adequada. A motivação do jogador não era considerada um fator crítico para essa determinação.

O artigo propõe o reconhecimento de estados afetivos do jogador e a investigação de como usá-los para ajustar automaticamente o nível de dificuldade do jogo em tempo real para melhorar a experiência de jogo.

O experimento do trabalho consistiu de duas etapas. A primeira para reconhecer o estado afetivo de ansiedade dos voluntários, usando dois jogos e diversos sensores fisiológicos capturando dados durante a sessão de jogo. Os jogos usados na primeira etapa foram *Pong* e um jogo de anagramas. Além disso, questionários foram aplicados após certo intervalo de tempo.

A segunda etapa consistiu em analisar o impacto da sensibilidade da afetividade na experiência de jogo. Nesta etapa o jogo utilizado foi *Pong*, mas usando mecanismos de ADD diferentes, um baseado no desempenho e outro baseado nos níveis de ansiedade do jogador. Questionários foram respondidos durante e no final das sessões de jogo.

Após a condução do experimento os autores chegaram aos seguintes resultados: os modelos de afetividade tiveram um índice alto de acurácia; a performance da maioria foi melhorada usando modelos baseados em afetividade; a atividade foi percebida como mais desafiadora e mais satisfatória pela maioria e o nível de ansiedade foi reduzido na maioria dos participantes nos mecanismos de ajuste dinâmico de dificuldade de afetividade.

Este artigo fortalece o argumento, também usado nesta monografia, de que a experiência de jogar um jogo eletrônico pode ser melhorada usando variáveis do jogador para ajustar a dificuldade do jogo em tempo real.

### 3.9 ADD afetivo

A tese apresentada em [10] teve como foco a área de ajuste dinâmico de dificuldade baseado em afetividade, uma das muitas áreas de exploração dentro do campo emergente dos jogos afetivos. Trata-se de um conceito de design de alto nível que se destina a alavancar os indicadores de afetividade do jogador para manipular a dificuldade de um jogo em tempo real.

O autor provê uma avaliação compreensiva dos métodos de modulação de dificuldade em jogos e avalia a confiabilidade da atividade eletrodermal como forma de medir o estresse e a carga cognitiva. O trabalho discute um novo método de ajuste dinâmico de dificuldade em tempo real que funciona ajustando fatores dentro do jogo baseado nas medidas de atividade eletrodermal do jogador. Para demonstrar se a condutância da pele pode ser usada como uma variável de modulação dificuldade num sistema de ADD afetiva, um jogo de pequena escala utilizando ADD afetiva (Figura 3.2), apelidado *electroderma*, foi desenvolvido e testado. Em conclusão, o autor por meio da análise de resultados conseguiu comprovar sua tese e abrir novas possibilidades no futuro dessa área.

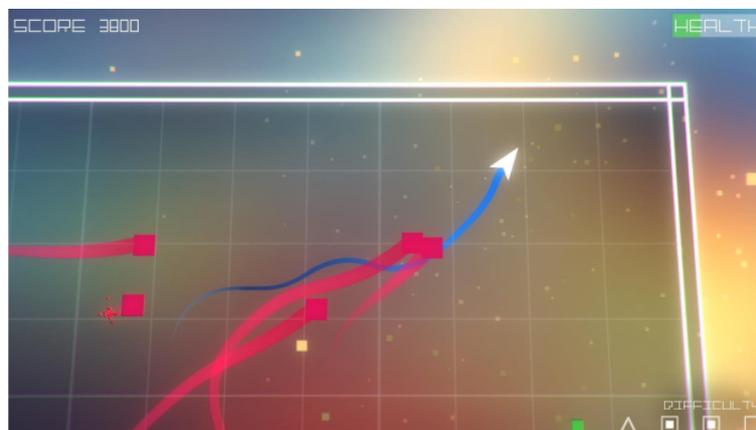


Figura 3.2: Jogo criado para testar Ajuste dinâmico afetivo

# Capítulo 4

## O Jogo

Antes da implementação de um mecanismo de ADD é necessário um jogo para aplicá-lo. Contudo, a implementação de um jogo a partir do zero demanda tempo e dedicação [29]. Como o propósito principal deste trabalho não se concentra no desenvolvimento de um jogo eletrônico propriamente dito, optou-se pela utilização de um previamente programado. Dessa forma, a equipe pode dedicar mais esforços no desenvolvimento dos mecanismos de ajuste dinâmico de dificuldade. Ainda assim, o jogo escolhido, "*Asteroids: in the 2nd and 1/2th Dimension*", criado por Matthew Renze [14], necessitou de uma adaptação para viabilizar a implantação de um modelo de ADD. O jogo é semelhante à *Asteroids* de 1979 [30], cuja tela pode ser observada na Figura 4.1. Em "*Asteroids: in the 2nd and 1/2th Dimension*", o jogador controla uma nave e tem como objetivo atravessar um campo de asteroides. Este capítulo detalha o jogo escolhido, como foi feita sua implementação e como ele irá usar modelos de ADD.

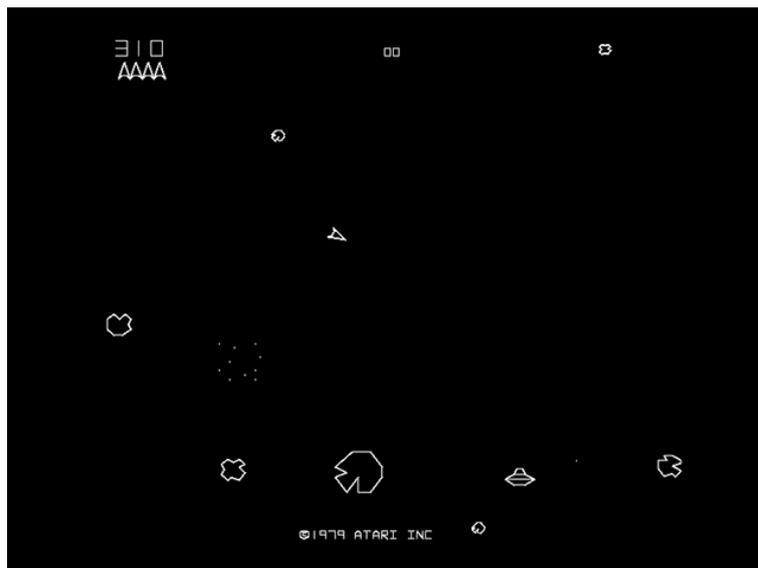


Figura 4.1: Asteroids [3]

## 4.1 Conceito

Para viabilizar sua utilização neste trabalho, foi determinado que o jogo escolhido deve cumprir diversos requisitos que foram estabelecidos considerando que o mesmo receberá um modelo de ADD e será utilizado em testes:

- O jogo deve rodar em um computador com o sistema operacional *Windows*, pois este é o sistema operacional instalado nos computadores que serão usados nesta pesquisa. Assim como deve possibilitar sua execução em diversos outros computadores, como, por exemplo, em computadores de participantes do teste. Essa possibilidade é maior para o sistema operacional *Windows*, pois este é o sistema operacional mais comum em *desktops* [31].
- A mecânica principal de jogo deve ser simples. Essa simplicidade possibilita que jogadores menos experientes com o gênero se familiarizem com o jogo mais rapidamente, não resultando em uma dificuldade e frustração desnecessários.
- O jogo deve ser uma experiência para apenas um jogador. As variáveis que afetam a dificuldade e sua percepção sofrem interferências quando a experiência inclui outros jogadores jogando o mesmo jogo e, como essas interferências não fazem parte deste estudo, o jogo não pode ser multiusuário.
- O jogo deve possuir meios para alteração de dificuldade, possibilitando o uso de mecanismos para mudança da mesma. Como o trabalho foca em modelos de ADD, este requisito possibilita o uso do jogo na implementação de um ADD.
- O jogo deve permitir a avaliação da performance do jogador. Em suas mecânicas e implementação deve existir alguma variável ou algum meio para quantificar o desempenho do jogador. Este requisito está diretamente relacionado com o modelo ADD baseado em performance e por consequência com o modelo híbrido.
- O jogo deve ser completado em menos de 10 minutos para possibilitar várias sessões de jogo em curto período de tempo. Outra razão para exigência deste requisito é diminuir as chances do jogador ficar entediado por longas sessões de jogo, o que provocaria alterações no ADD baseado em afetividade, não relacionadas com a dificuldade do jogo.
- O jogo deve possuir a capacidade de alteração para criação de jogos derivados do jogo origem. Permitindo que o mesmo possa acoplar modelos de ADD, *inputs* de sensores, modificações para execução de testes e armazenamento de dados para análises.

- O jogo deve possuir código fonte aberto e sua licença deve permitir seu uso não comercial. Como o jogo será usado neste trabalho, o autor do mesmo deve permitir seu uso em pesquisas.

O jogo "*Asteroids: in the 2nd and 1/2th Dimension*", criado por Matthew Renze [14], atende aos requisitos mencionados acima, e portanto, foi o escolhido para ser utilizado nesta pesquisa.

## 4.2 Detalhamento

Nesta sessão será primeiramente mostrado o jogo *Asteroids* original apontando os elementos que foram descartados na adaptação. Em seguida detalhada a adaptação, isto é, os componentes do jogo, as fases, os problemas e suas soluções, como o jogador controla seu avatar e detalhes sobre a dificuldade.

### 4.2.1 Jogo Original

O jogo "*Asteroids: in the 2nd and 1/2th Dimension*" coloca o jogador para controlar uma nave, cujo objetivo é encontrar um item azul no meio de um campo de asteroides e após encontrá-lo, atravessar o campo e ativar o hiperespaço para completar o jogo. A Figura 4.2 mostra a tela do jogo em execução, indicando também um contador de pontos, um item azul e um campo de força ao redor da nave. O jogo possui uma fase e completá-la significa completar o jogo. Para a implementação dos ADDs, será desenvolvida uma mecânica de múltiplas fases, onde a dificuldade será ajustada após a finalização de uma fase. O controle da nave foi mantido na adaptação e será detalhado na Subseção 4.2.4.

Na interface visual, canto inferior esquerda da tela, o jogo original exibe um contador de pontuação que aumenta quanto mais asteroides são destruídos pelo jogador e diminui quanto mais tiros não acertam asteroides. Entretanto sua exibição pode gerar um novo objetivo para alguns jogadores e conseqüentemente poderia causar divergência na percepção da dificuldade. Por essa falta de certeza em relação ao contador, foi decidido que ele não será usado no jogo adaptado. Semelhante ao contador, o item azul necessário para ativação do hiperespaço foi retirado do jogo, com o intuito de eliminar possíveis fatores de dificuldades que afetam alguns jogadores positivamente, outros negativamente e que não são objetos de estudo desta pesquisa. Esses fatores podem incluir o jogador decidir jogar para conseguir uma pontuação maior, criando um desafio a mais e com suas próprias dificuldades, uma dificuldade nova que os ADDs que serão implementados não ajustarão, desviando assim do escopo dessa pesquisa. Três visões são incluídas no jogo original,

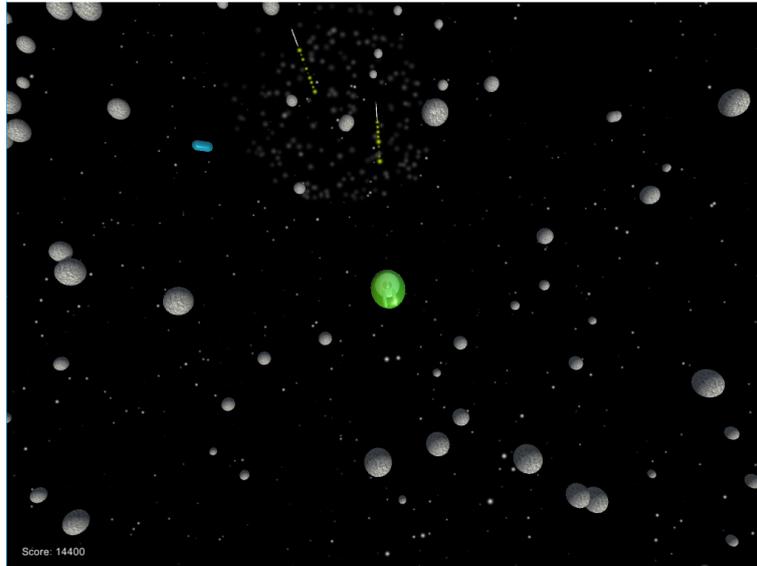


Figura 4.2: Jogo Original

entretanto para padronizar a experiência do jogador duas opções foram retiradas, logo a visão do jogo se manteve na visão padrão.

## 4.2.2 Jogo Adaptado

Para o uso dos modelos de ADDs foram feitas diversas adaptações: criação de uma estrutura de múltiplas fases, mudanças na interface visual do usuário e componentes do jogo.

### Componentes Essenciais do Jogo

No jogo adaptado foram acrescentados três tipos diferentes de asteroides, tal mudança foi motivada pela necessidade de manter a atenção do jogador. Com uma estrutura de múltiplas fases, a falta de elementos novos nelas pode resultar em tédio, o que afetaria tanto o ADD afetivo quanto o ADD híbrido. Os diferentes tipos de asteroides serão os elementos novos em cada nova fase. Como mencionado no detalhamento do jogo original, foi retirado o item azul necessário para completar a fase, o jogador entra no hiperespaço assim que sair do campo de asteroides. Substituindo o contador de pontos do jogo original, foi acrescentado à interface visual do jogador um contador de segundos desde o início da primeira fase e um contador da distância até o final do campo de asteroides. Ambos contadores têm como propósito dar uma resposta ao jogador sobre seu desempenho, há quanto tempo ele está jogando e quanto falta para completar a fase que ele está jogando.

Cada componente será detalhada a seguir:

- Nave: componente controlado pelo jogador.

- **Míssil:** componente que quando entra em contato com asteroides causa a explosão de ambos. O jogador atira os mísseis que vão na direção para a qual a frente da nave estava direcionada quando o jogador acionar o comando de atirar mísseis.
- **Asteroide Pequeno:** asteroides desse tipo são pequenos e destruídos ao encontro de um míssil.
- **Asteroide Grande:** asteroides desse tipo são maiores que os asteroides pequenos e são destruídos com um míssil, entretanto, após sua destruição, dois asteroides pequenos apareceram no lugar da destruição do grande.
- **Asteroide Gigante:** asteroides gigantes são os maiores asteroides do jogo e não podem ser destruídos com mísseis.
- **Asteroide Explosivo:** estes asteroides quando encontram com um míssil geram uma área de impacto que destrói a nave do jogador se ela estiver nessa área ou destrói apenas o campo de força da nave, se ativado. Este asteroide possui uma textura especial para diferenciação.
- **Asteroide com Campo de Força:** esses asteroides possuem um campo de força semelhante ao que o jogador pode ter, entretanto para diferenciar os campos de força, a cor deste é vermelha. Para destruir este asteroide o jogador deve atirar no campo, destruindo-o, e depois atirar no asteroide que estava dentro, este podendo ser pequeno ou grande.
- **Campo de Força:** No início de cada fase existe um item que ativa um campo de força ao redor da nave do jogador. Esse item sempre está na mesma localização, na frente do jogador assim que a fase é iniciada. O campo de força protege a nave do jogador, evitando que ela seja destruída ao colidir com um asteroide, mas o campo de força é perdido depois do primeiro asteroide que colidir com o mesmo. O campo de força do jogador é verde, uma diferença com relação ao campo de um dos tipos dos asteroides.
- **Limites laterais:** Como o campo de asteroides tem limites laterais, foram criados dois cilindros vermelhos para evitar que o jogador controle sua nave para além desses limites, resultando em uma percepção de dificuldade mais baixa.

A Figura 4.3 contém um quadro com a representação gráfica de cada elemento do jogo.

## Fases

Para inclusão dos modelos de ADDs serão feitas seis fases, começando com a Fase 0 que será uma fase introdutória, seguidas da Fase 1 até a Fase 5. A Fase 1 até a Fase

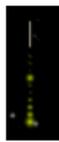
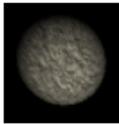
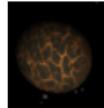
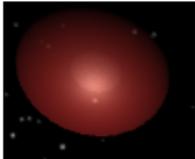
NAVE	MÍSSIL	PEQUENO	GRANDE	GIGANTE	EXPLOSIVO	BARREIRA	ESCUDO	LIMITE
								

Figura 4.3: Componentes do jogo

4 introduzem um novo tipo de asteroide. Essa introdução é feita no meio da fase para possibilitar que o jogador esteja familiarizado com a fase antes que um novo desafio seja introduzido. Limites laterais existem em todas as fases. Mencionado anteriormente, toda fase possui um item para ativação do campo de força do jogador, este estará sempre posicionado no início de todas as fases.

- Fase 0: fase introdutória, possui apenas os asteroides pequenos, presentes em toda extensão do campo de asteroides. Essa fase tem apenas o tipo mais simples dos asteroides, pois a mesma foi feita para o jogador se familiarizar com os controles e como o jogo funciona.
- Fase 1: é introduzido o tipo de asteroide grande, com asteroides pequenos presentes na extensão completa do campo. A inclusão apenas do tipo grande e do pequeno nesta fase foi feita para não deixar a segunda fase muito mais complexa que a primeira, mas ainda incluir um novo tipo de desafio para o jogador.
- Fase 2: o tipo introduzido é o asteroide gigante, com asteroides pequenos e grandes presentes em toda extensão do campo. Na terceira fase é esperado que o jogador já esteja familiarizado com o jogo e suas mecânicas, assim desde o início, asteroides pequenos e grandes estão presentes e no meio da fase asteroides gigantes aparecem. Os asteroides gigantes provocam a estratégia de desviar ao invés de atirar em todos os asteroides.
- Fase 3: Semelhante a fase anterior, esta fase possui asteroides pequenos e grandes em toda sua extensão com a introdução dos asteroides explosivos no meio da fase. Estes novos asteroides provocam uma economia no lançamento de mísseis pela razão de se atingidos geram uma grande área de impacto, podendo resultar na destruição da nave mais facilmente.
- Fase 4: Assim como nas duas fases anteriores, os asteroides pequenos e grandes estão sempre presentes no campo de asteroides. O novo tipo de asteroide é aquele com campo de força, necessitando mais de um míssil para destruição completa de

cada asteroide. Os campos de força podem estar ativados em asteroides pequenos e grandes.

- Fase 5: Para a última fase, todos os tipos de asteroides estão presentes desde o começo. Como o jogador já passou pela introdução de todos os asteroides, nessa fase ele é desafiado com todos juntos. Ao completar esta fase, o jogador consegue vencer o jogo.

### 4.2.3 Problemas e Soluções

Durante a adaptação do jogo foram encontrados alguns problemas. Inicialmente, a Fase 5 conteria um inimigo controlado por inteligência artificial no lugar de asteroides, essa ideia foi descartada pois criaria um novo desafio não condizente com o resto do jogo, logo no final dele. Então foi decidido que seriam utilizados todos os tipos de asteroides desde o começo da última fase, como o jogador já teria encontrado todos os asteroides nas fases anteriores o novo desafio seria encará-los juntos. Mencionados anteriormente, os cilindros vermelhos definidos como limites laterais foram implementados para impedir que a nave ultrapasse o campo pelas laterais e siga em frente sem nenhum desafio. Um último problema que surgiu na adaptação foi que alguns asteroides, que foram definidos como pequenos, estavam gerando dois asteroides minúsculos que podiam destruir a nave do jogador, então foi programado que apenas os asteroides grandes gerariam dois pequenos.

### 4.2.4 Controles

O jogador pode controlar a direção para onde a nave está direcionada virando a nave para direita ou esquerda. A nave pode seguir na direção em que se encontra ou ir à direção contrária; esse movimento segue uma aceleração até uma velocidade limite. Essa movimentação está representada na Figura 4.4. Existe também a possibilidade de atirar um míssil na direção que a nave se encontra; não existem limites de mísseis, entretanto, existe um pequeno intervalo de tempo entre disparos.

### 4.2.5 Dificuldade

Para poder executar um ajuste na dificuldade foram programadas três variações de cada fase. Estas variações serão denominadas Dificuldade Baixa, Dificuldade Média e Dificuldade Alta. Entretanto a Fase 0 não possui essas variações tendo apenas uma dificuldade padrão. Assim foi feito para que a primeira fase seja feita para determinar a futura dificuldade baseada no jogador. Por essa razão a dificuldade padrão da Fase 0 é mais fácil que a dificuldade média das outras fases, para possibilitar que todos tipos de jogadores

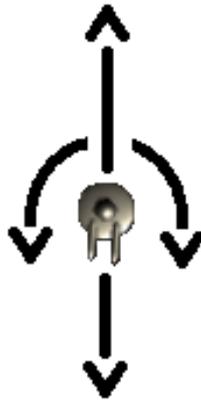


Figura 4.4: Movimentação da Nave do Jogador

possam concluí-la. As variáveis escolhidas para determinar a dificuldade de cada fase são: a margem de velocidade dos asteroides e a densidade de asteroides. Os valores serão detalhados na Seção Implementação.

Após a realização dos primeiros testes foi incluída uma nova alternativa de dificuldade para os casos de ADDs. Isto é, dentro de cada variação de fase é feita uma modificação na margem de velocidade.

## 4.3 Implementação

Nesta seção são detalhados: a ferramenta utilizada para adaptar o jogo "*Asteroids: in the 2nd and 1/2th Dimension*", como funcionam os *inputs*, como os componentes do jogo estão programados, assim como seus estados, como foi programado cada asteroide, os elementos do jogo e o mapa do jogo.

### 4.3.1 Engine

O jogo foi implementado na plataforma *Unity* 5.3.5 [32], um Ambiente de Desenvolvimento Integrado, assim como um motor de jogo 3D. A Figura 4.5 mostra o jogo rodando na interface do *Unity*. Os códigos usados dentro da plataforma foram feitos na linguagem C#, uma linguagem de programação orientada a objeto. Cada objeto do jogo tem atribuído a ele um ou mais componentes e funções. Para adaptação foram reutilizadas diversas classes e objetos do jogo original, assim como criado um objeto ADD que fará o ajuste de dificuldade.

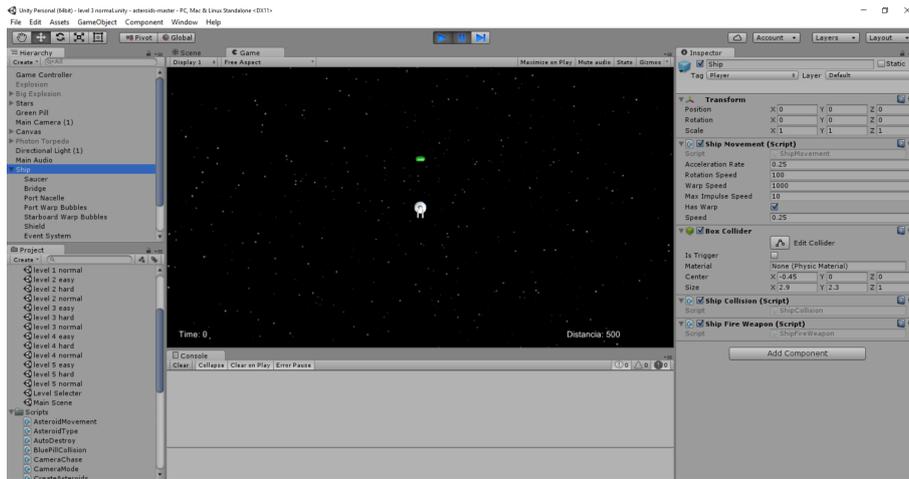


Figura 4.5: Interface da Engine Unity

## Controle de Inputs

Para os controles do jogador, os seguintes inputs foram usados do jogo original: as setas direcionais do teclado foram usadas para movimentação do jogador, as setas verticais para mudança de direção da nave e as setas horizontais para fazer a nave acelerar na direção da mesma ou contrária; a barra de espaço foi usada para disparar os mísseis e mudar de fase, uma vez que a mesma foi concluída. Durante os testes alguns jogadores pediram que fosse desenvolvido o uso das teclas W, A, S e D para movimentação, com o W acelerando a nave na direção da mesma, S acelerando para direção contrária, A mudando a direção para esquerda e D mudando para direita. Para realização de testes foi implementado um meio para encerrar a fase que está sendo jogada, a tecla Z era usada para este meio. Para o funcionamento do modelo de ADD baseado em afetividade, sinais de EDA (*Electrodermal Activity*) foram usados como *inputs* do jogador.

### 4.3.2 Componentes

Dentro do *script* do controlador do jogo foi implementado o sistema de fases, onde após completar uma fase, o sistema inicia a cena da próxima fase na dificuldade que o ADD ditar, se o mesmo não estiver ativado, a dificuldade será mantida.

Asteroides explosivos foram programados para destruir a nave do jogador dentro de um raio. Para visualização desse raio foi usado uma explosão vermelha. Uma textura diferente dos outros asteroides foi utilizada para identificação deste tipo de asteroide.

Cada asteroide pequeno e grande possui um campo de força, entretanto estes são somente renderizados e seus códigos de colisão são ativados quando o asteroide é criado com o tipo de campo de força.

Asteroides gigantes foram feitos para serem iguais aos asteroides pequenos e grandes, entretanto ao colidirem com um míssil, apenas o míssil é destruído.

O objeto ADD foi criado para contabilizar as mortes do jogador, o tempo em segundos gasto no jogo e é usado para ajustar a dificuldade. Como cada morte ou nova fase inicia uma nova cena, o objeto deve permanecer como na cena anterior, então o objeto ADD foi feito para ser persistente, entretanto esse fato faz com que para o ADD funcionar a primeira cena deve conter o objeto ADD, mas as outras não.

Para cada fase, um objeto com um *script* de geração de asteroides coloca todos os asteroides no campo de asteroides, em posições e velocidades randômicas dentro da margem de velocidade da dificuldade da fase. Esse *script* cria cópias dos modelos de asteroides existentes com opções dos tipos de asteroides programados. Um objeto é usado para cada tipo de asteroide.

## Elementos do Jogo

Após disparados, mísseis são deletados do jogo se os mesmos atingirem algum asteroide ou caso saírem do campo de asteroides.

A nave do jogador está programada para não atravessar o campo de asteroides pelas laterais, entretanto para o jogador não encontrar uma barreira invisível foram colocados os limites laterais mencionados na Subseção Componentes Essenciais do Jogo da Seção 4.2.2.

O campo de força está sempre presente em volta da nave, entretanto ele não é renderizado e não possui o código de colisão ativo enquanto o jogador não pegar o item de campo de força. Quando o campo de força entrar em contato com algum asteroide, uma explosão ou outro campo de força, ambos são destruídos e o campo de força volta a não ser renderizado assim como seu código de colisão não é ativado.

## Estados do Jogo

Assim que o jogo é iniciado o jogador entra na primeira fase, Fase 0, se ele chegar no final do campo de asteroides o jogo verifica no ADD qual deve ser a próxima dificuldade e inicia a próxima fase, se o jogador vencer a Fase 5, o jogo é encerrado. Em cada fase, se o jogador morrer a fase é reiniciada. A Figura 4.6 mostra a representação de um diagrama de estados do jogo, onde as letras A, M e B representam a dificuldade de cada fase, com A significando dificuldade alta, M significa dificuldade média e B significa dificuldade baixa.

## Mapa

O mapa completo do jogo com seus elementos está presente na Figura 4.7.

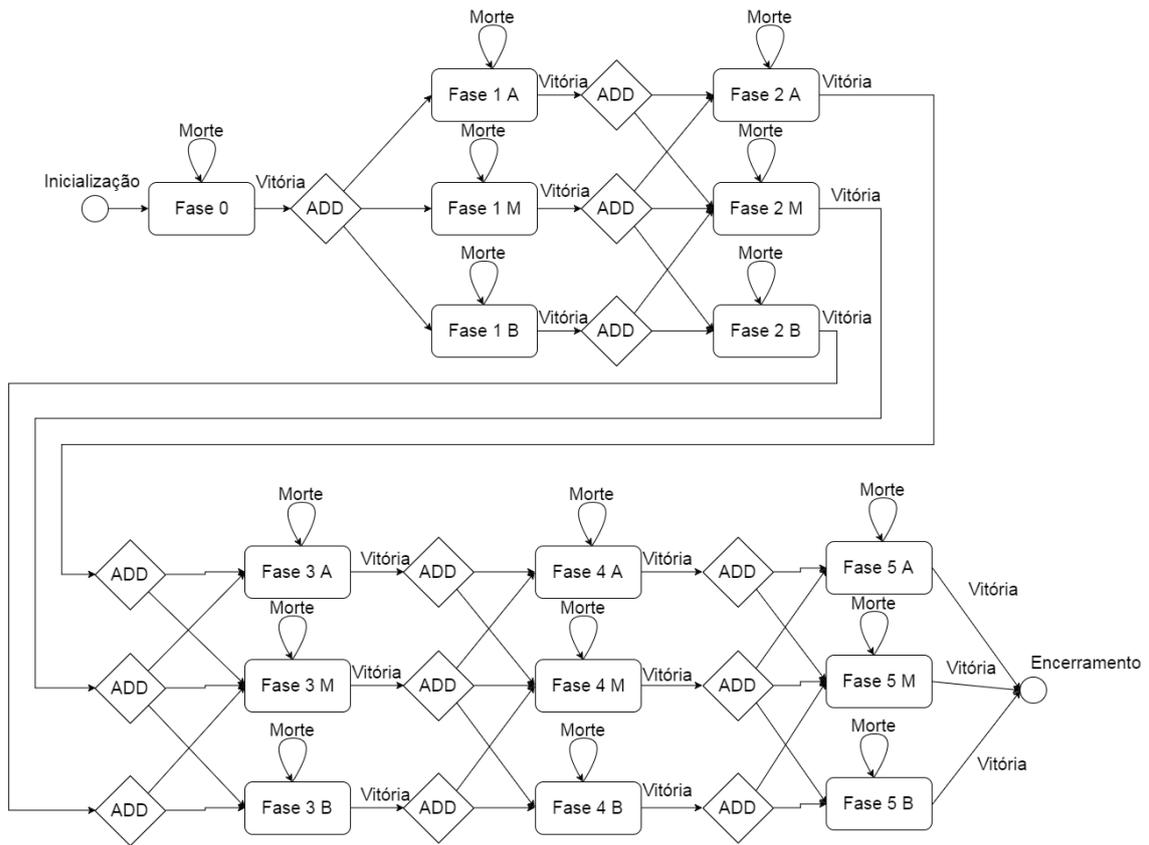


Figura 4.6: Diagrama de estados do jogo adaptado

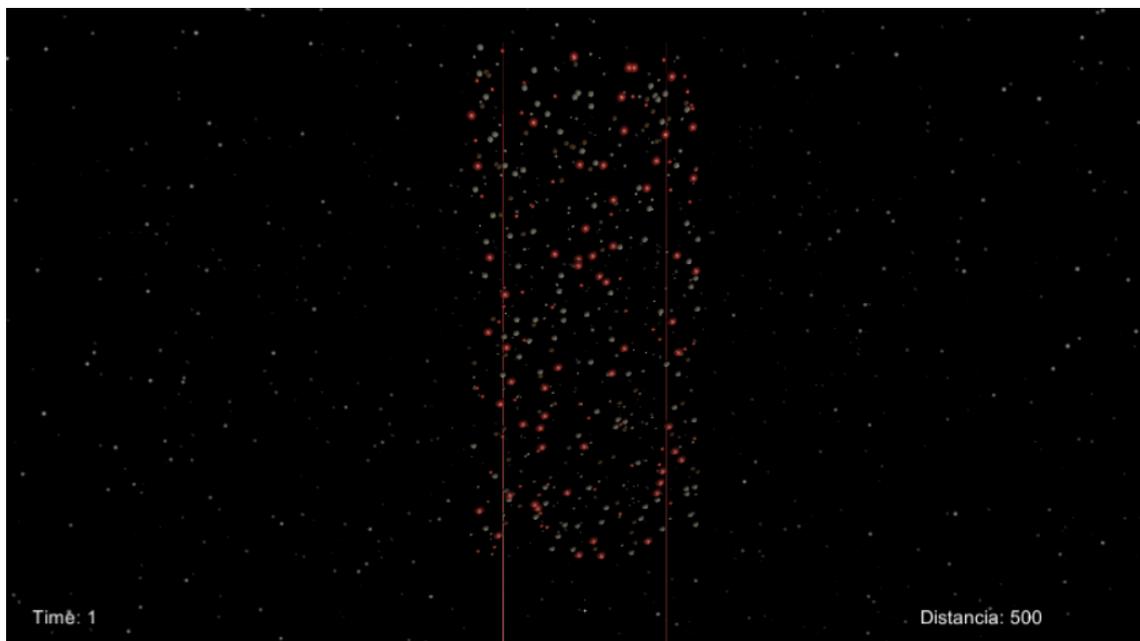


Figura 4.7: Mapa da Fase 5

### 4.3.3 Dificuldade

A densidade de cada fase assim como a margem da velocidade de cada asteroide está presente nas tabelas seguintes. A Tabela 4.1 mostra a quantidade de cada tipo de asteroides na Fase 0 e nas fases 1 até 5 da dificuldade baixa, a Tabela 4.2 mostra as fases 1 até 5 na dificuldade média e a Tabela 4.3 mostra as mesmas fases na dificuldade alta. A Tabela 4.4 mostra a velocidade de cada dificuldade para todos asteroides.

Tabela 4.1: Densidade na dificuldade baixa e na Fase 0

	<b>Fase 0</b>	<b>Fase 1</b>	<b>Fase 2</b>	<b>Fase 3</b>	<b>Fase 4</b>	<b>Fase 5</b>
<b>Pequeno</b>	650	350	200	180	200	100
<b>Grande</b>	0	150	200	180	200	110
<b>Gigante</b>	0	0	100	0	0	110
<b>Explosivo</b>	0	0	0	0	100	60
<b>Barreira</b>	0	0	0	140	0	120

Tabela 4.2: Densidade na dificuldade média

	<b>Fase 1</b>	<b>Fase 2</b>	<b>Fase 3</b>	<b>Fase 4</b>	<b>Fase 5</b>
<b>Pequeno</b>	470	220	230	250	150
<b>Grande</b>	200	220	230	250	150
<b>Gigante</b>	0	130	0	0	150
<b>Explosivo</b>	0	0	0	170	70
<b>Barreira</b>	0	0	200	0	140

Tabela 4.3: Densidade na dificuldade alta

	<b>Fase 1</b>	<b>Fase 2</b>	<b>Fase 3</b>	<b>Fase 4</b>	<b>Fase 5</b>
<b>Pequeno</b>	550	270	270	215	160
<b>Grande</b>	240	270	270	215	160
<b>Gigante</b>	0	170	0	0	140
<b>Explosivo</b>	0	0	0	200	70
<b>Barreira</b>	0	0	250	0	140

Tabela 4.4: Velocidade por dificuldade

	<b>Maxíma</b>	<b>Miníma</b>
<b>Baixa</b>	4	0
<b>Média</b>	2	0
<b>Alta</b>	7	4

#### 4.3.4 Código

A implementação completa do jogo adaptado neste trabalho, inclusive com os ADDs inseridos, está disponível em repositório público na URL <https://www.dropbox.com/sh/udzpfi9qgvvd3i5/AAAHgtTtMKjNEy1Nxd-cZQJna?dl=0>.

# Capítulo 5

## Ajuste Dinâmico de Dificuldade Híbrido

De acordo com Daniel Imre em sua tese [10], o futuro da área de ajuste dinâmica de dificuldade será um processo gradual que consiste em três fases cronologicamente ordenadas:

- Para os próximos anos, a grande maioria dos títulos com implementação de ADD continuarão a usar de implementações baseadas em desempenho.
- Em um futuro próximo, como sensores de afetividade tornaram-se menores e mais baratos e os jogos utilizarão uma mistura de métodos de ADD.
- Como a capacidade de medir afetividade se torna mais onipresente e está integrada em dispositivos de jogos comumente usadas (tais como controles), a avaliação da afetividade se tornará a forma dominante para adaptação das experiências de jogo ao jogador.

A proposta apresentada neste trabalho é derivada parcialmente no segundo item acima. Como visto nos trabalhos relacionados, geralmente é feito um ADD, pode-se dizer, *tradicional*, baseado exclusivamente nas variáveis de jogo ou nas variáveis do estado afetivo do jogador.

O objetivo desta investigação é verificar se um ADD híbrido, cuja dinâmica será baseada na correlação de dados fornecidos tanto pelo jogo como pelo jogador, é melhor em fazer o jogador se sentir no estado *flow*, em comparação aos modelos de ADD comumente usados.

No intuito de realizar o objetivo desta pesquisa, foi implementado um ADD híbrido, que foi posteriormente aplicado em um jogo existente e por fim comparado com os outros modelos de ADD por meio de testes com usuários. Pela limitação de um ano de trabalho

e recursos, foi utilizado apenas uma métrica para definição de performance do jogador e um sensor EDA para realizar a análise do estado afetivo do jogador.

Para fins de teste e comparação de resultados, os modelos de ADD, tanto o híbrido quanto o tradicional, isto é, aquele baseado em apenas uma fonte (como dados do próprio jogo, por exemplo) devem ser aplicados no mesmo jogo.

Neste trabalho, o ADD foi implementado em um jogo existente que possibilitou alterações e ajustes para tanto. O jogo foi modificado em sua plataforma de implementação (Unity [32]) para inclusão dos ADDs e comunicação com o sensor EDA. O sensor EDA utilizado foi o BITalino [33]. A comunicação com o sensor EDA foi feita no MATLAB [34], que interagiu com a ferramenta Unity para obtenção do estado afetivo do jogador.

Nas próximas sessões será fornecido o detalhamento das fases de implementação dos ADDs.

## 5.1 Unity

Para aquisição dos sinais do sensor EDA, foi inicialmente usado uma Interface de Programação de Aplicativos, ou API, fornecida no site oficial da placa de aquisição de sinais usada, *BITalino* [33]. A inclusão da API foi feita no jogo, mas não foi descoberto um meio de tratar os dados recebidos dentro da plataforma. Após uma pesquisa sobre tratamento de dados derivados de um sensor eletrodermal, foi encontrado um código aberto em MATLAB [34] que calcula o valor tônico da saída do EDA. Para que esse código fosse usado em conjunto com o jogo foi necessário uma comunicação entre *Unity* e *MATLAB*. Essa comunicação foi feita através de portas UDP (*User Datagram Protocol*). O código para realizar uma comunicação de portas UDP a partir do *Unity* foi retirado do site: <https://msdn.microsoft.com/de-de/library/bb979228.aspx>, com alterações feitas por Sandra Fang. Com esse código, adaptado para funcionar em conjunto com o jogo, foi possível estabelecer a comunicação de mensagens do *Unity* para o *MATLAB* e a recepção de mensagens no *Unity* originadas do *MATLAB*.

Para a realização da primeira bateria de testes, foi programado no jogo o registro da quantidade de mortes e o tempo em segundos que o jogador demorou em concluir a fase. Este registro foi feito dentro de um arquivo em formato *.txt*, para cada fase concluída, com a data, hora, nível e dificuldade registrada como nome do arquivo. A segunda bateria de testes foi realizada com a adição de três modelos de ADDs. O jogo com o modelo de ADD baseado apenas em performance, isto é, em número de mortes, foi compilado, através do *Unity*, em um formato executável para possibilitar que os testes, com este modelo, fossem realizados em outros computadores diferentes daquele no qual o jogo foi programado. Os

modelos com o ADD híbrido e com o ADD baseado apenas nos dados dos sensores foram executados diretamente a partir da ferramenta *Unity* em conjunto com o *MATLAB* [34].

## 5.2 BITalino

O *BITalino* [33] é um dispositivo de baixo custo capaz de ler três sinais fisiológicos comuns: Eletromiográfico (EMG), Eletrocardiográfico (ECG) e Eletrodermal (EDA). Pela sua disponibilidade e abundância de códigos e APIs com livre distribuição, este foi o dispositivo usado nesta pesquisa para colher os dados do EDA [4] da placa do *BITalino*. Pela sua configuração física e falta de recursos não foi possível usar mais de um sensor da mesma placa para aplicar no ADD, resultando apenas no uso do sensor EDA para este trabalho. A posição que deve ser colocada os eletrodos do EDA pode ser vista na Figura 5.1. O *BITalino* conecta com o computador através do sinal *bluetooth* e possui sua própria bateria. A conexão exclusiva por *bluetooth* trouxe dificuldade em momentos, devido as desconexões e interferências no sinal, o que causava a necessidade de reiniciar o jogo. Outra dificuldade foi a degradação do gel dos eletrodos com o frequente uso. Pela limitação de tempo e recursos, não foi possível adquirir novos eletrodos, resultando no uso de esparadrapo para manter a adesão do eletrodo.

## 5.3 MATLAB

O código usado para extrair o valor tônico da saída do EDA está disponível com distribuição aberta no site da *MathWorks* [35]. Para aquisição dos dados do sensor do *BITalino* foi utilizado uma *toolbox* desenvolvida pela equipe *MathWorks Instrument Control Toolbox Team* [36]. Foi programada a comunicação através de portas UDP em conjunto com os códigos citados anteriormente para compor apenas um programa. O programa espera receber um sinal do *Unity* para começar a aquisição de dados do sensor EDA, outro sinal para encerrar a captura de dados uma vez que o jogador complete a fase. Após encerrar a captura, o programa usa o código *SimpleEDA* [35] para adquirir os valores tônicos e após a análise desses valores, o mesmo enviará uma mensagem para o *Unity* com o resultado da análise que informará se o jogador está estressado, desafiado ou entediado. O programa retorna à espera do sinal para iniciar uma nova captura na próxima fase. A Figura 5.2 mostra a interface do *MATLAB* e o começo do código usado.

Assim como realizado na *Unity*, para aquisição dos resultados da primeira bateria de testes foi programado a impressão dos valores tônicos de cinco em cinco segundos em um arquivo de formato *.txt* com a mesma identificação da data, hora, nível e dificuldade no nome do arquivo.

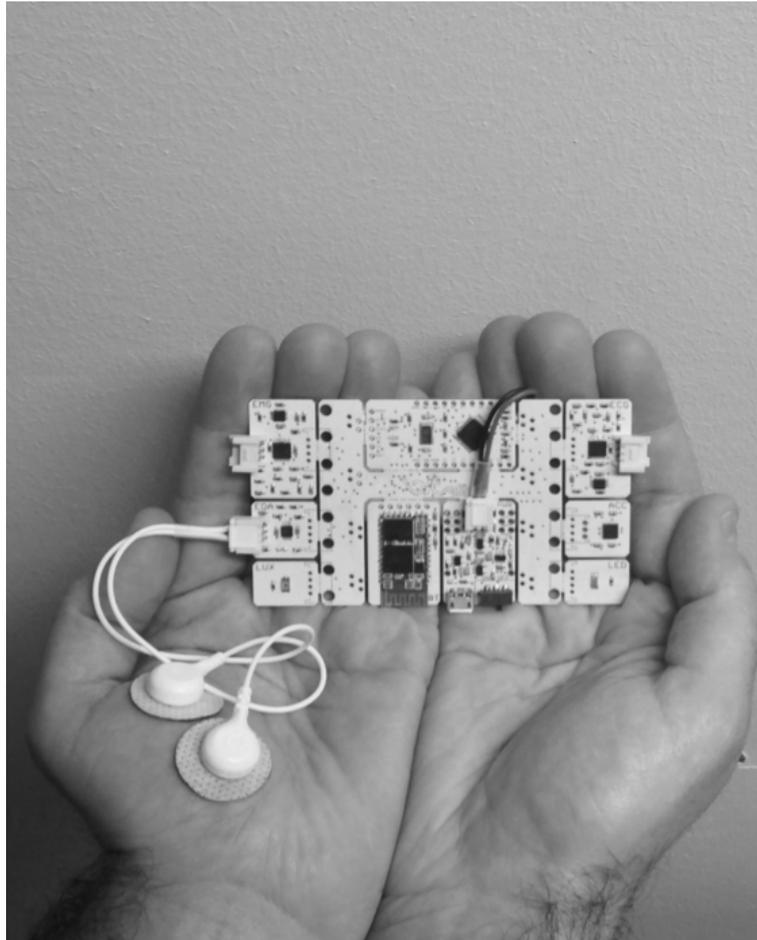


Figura 5.1: Placa *BITalino* ajusta para o sensor EDA [4]

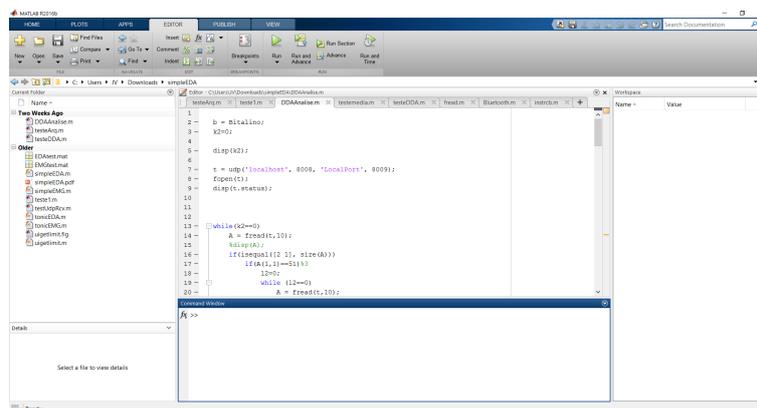


Figura 5.2: Interface do MATLAB

## 5.4 ADD

O ADD atua na seleção da dificuldade da fase seguinte uma vez que uma fase tenha sido completada. Para fins de comparação foram feitos três modelos de ADD. Com os resultados da primeira bateria de testes foram definidos limites de mortes para cada fase

e cada dificuldade, assim como os limites do sinal do sensor que ditam se o jogador esta estressado, estimulado ou entediado.

- **Performance:** De acordo com os limites definidos pela análise do resultado da primeira bateria de testes, se o jogador morrer além do limite de uma dificuldade, o ADD diminuirá a dificuldade para a próxima fase, caso ele não morra o suficiente para manter a dificuldade, o ADD aumentará a dificuldade da próxima fase. Se o jogador morrer acima de um dos limites definidos, os asteroides terão suas velocidades diminuídas.
- **Estado Afetivo:** Se a média total dos valores tônicos subtraída da média dos dez menores valores, fórmula considerada efetiva apos os testes para verificar os dados vindos do sensor EDA, for abaixo dos limites definidos pelo resultado da primeira bateria de testes, a dificuldade será aumentada. Se for maior que a margem definida, a dificuldade será abaixada. Caso contrário, permanecerá a mesma.
- **Híbrido:** Assim como o primeiro modelo, se o jogador morrer acima de um dos limites definidos, os asteroides terão suas velocidades diminuídas. Para alterar a dificuldade da próxima fase, este ADD seguirá o estabelecido nas Tabelas 5.1, 5.2 e 5.3. As letra D e V são para dificuldade e velocidade, respectivamente. Se temos um D+ devemos aumentar a dificuldade, D- para diminuir e D para manter a dificuldade. O V significa manter a velocidade dos asteroides, V+ para aumentar a velocidade dos asteroides e V- para diminuir a velocidade dos asteroides.

Tabela 5.1: ADD Híbrido: Dificuldade Alta

	<b>Mortes no limite</b>	<b>Muitas Mortes</b>
<b>Estressado</b>	D-/V+	D-/V
<b>Desafiado</b>	D/V	D-/V+
<b>Entediado</b>	D/V+	D/V+

Tabela 5.2: ADD Híbrido: Dificuldade Média

	<b>Poucas Mortes</b>	<b>Mortes no limite</b>	<b>Muitas Morreu</b>
<b>Estressado</b>	D/V-	D- G	D-
<b>Desafiado</b>	D/V+	D	D-/V+
<b>Entediado</b>	D+	D+/V-	D/V+

Tabela 5.3: ADD Híbrido: Dificuldade Baixa

	<b>Poucas Mortes</b>	<b>Mortes no limite</b>
<b>Estressado</b>	D/V-	D/V-
<b>Desafiado</b>	D/V+	D
<b>Entediado</b>	D+	D+/V-

# Capítulo 6

## Testes

Foram realizadas duas baterias de testes. A primeira bateria teve o propósito de encontrar as médias de mortes por fase e a dificuldade, assim como os valores do sensor EDA que foram posteriormente utilizados para definir uma métrica que determina o estado afetivo do jogador. A segunda bateria de testes foi feita com objetivo de comparar os diferentes modelos de ADD, isto é, ADD híbrido, ADD somente com dados de desempenho e ADD somente com dados do sensor. Na segunda bateria de testes também foi aplicado um questionário aos jogadores após a sessão de jogo. Para ambos testes o participante jogou em um *laptop* com o sensor EDA posicionado na palma da mão esquerda, como mostra na Figura 6.1

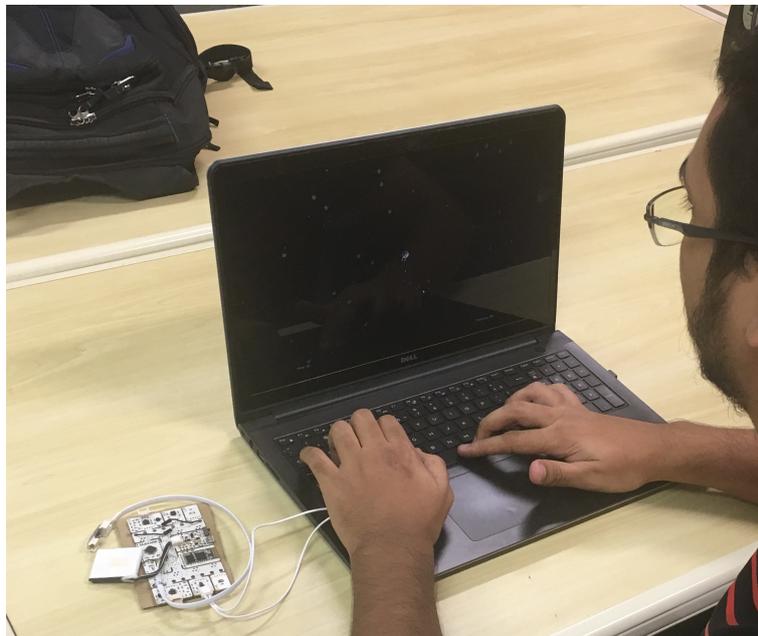


Figura 6.1: Exemplo de um participante realizando os testes com o sensor posicionado na palma da mão esquerda

## 6.1 Teste para balanceamento de dificuldade

Catorze candidatos participaram da primeira bateria de testes, sendo quatro mulheres e dez homens com idade de 20 a 30 anos. Com o propósito de funcionar para qualquer tipo de jogador e independente da habilidade do mesmo, os candidatos possuíam diversos níveis de experiência com jogos, variando de jogadores casuais, que jogam com menos frequência e tem menos habilidade, a experientes, que jogam com mais frequência e tem mais habilidade. Dentre estes, seis participaram duas vezes dos testes, pois consideramos também pessoas que jogam mais de uma vez. O nível de experiência dos participantes foi considerado com base em um autorrelato.

Os participantes jogaram as seis fases em apenas uma das três possíveis dificuldades. Foram realizados:

- Seis testes na dificuldade baixa: três jogadores experientes e três jogadores casuais;
- Sete testes na dificuldade média: quatro jogadores experientes e três casuais;
- Sete testes na dificuldade alta: cinco jogadores experientes e dois casuais.

Após o fim de cada fase, o participante foi questionado se ele estava se sentindo desafiado pelo jogo, se o desafio do mesmo estava abaixo ou acima de suas habilidades. O participante tinha a opção de desistir da fase ou do jogo, alegando tédio, desinteresse, cansaço ou estresse.

Durante os testes foi observado que os participantes desistiam de jogar após muitas mortes na mesma fase. Para evitar tal desistência, a velocidade dos asteroides foi diminuída gradualmente até a velocidade da dificuldade baixa, no caso do participante morrer acima de um limite definido para a fase em que está jogando. Esse limite foi definido com base nos dados da primeira bateria de testes de modo que o jogo comece a diminuir a velocidade dos asteroides antes que o jogador desista.

Unindo os dados provenientes do jogo, do sensor e das respostas dos participantes, foram definidas as margens usadas para alterar a dificuldade com base nas mortes e no estado afetivo do jogador. A margem de mortes foi calculada com base nas médias dos valores em que o jogador relatava estresse, desafio justo ou tédio. Esta margem é apresentada nas Tabelas 6.1, 6.2 e 6.3. As linhas destas tabelas indicam a dificuldade da próxima fase, para o modelo de ADD baseado em dados de performance (mortes no jogo). As colunas mostram cada nível e cada possível dificuldade deles, exceto o nível 5, pois no jogo não existe um nível seguinte que possibilite a mudança de dificuldade. O cruzamento entre linha e coluna indica a margem de mortes no nível da coluna que enviará o jogador para a dificuldade da linha no nível seguinte. Essas margens foram deduzidas a partir do

resultado dos primeiros testes. As letras B, M e A das Tabelas 6.2 e 6.3 representam as dificuldades baixa, média e alta, respectivamente.

Se o jogador jogou uma fase na dificuldade Baixa, ele não será direcionado para dificuldade Alta independente de seu estado afetivo ou número de mortes. Da mesma forma que se a dificuldade jogada for Alta, o mesmo não será direcionado para dificuldade Baixa. O ADD foi feito desta maneira para evitar que o jogador note uma mudança muito grande na dificuldade, algo que prejudicaria sua imersão e por consequência o afastando do estado de *flow*. Por essa razão as Tabelas 6.2 e 6.3 não apresentam margens do número de mortes para estas situações descritas acima.

É possível notar que a margem de mortes é menor para o aumento da dificuldade e que para a manutenção da dificuldade essa margem é menor do que para a redução da dificuldade. Tais números mostram que jogadores com mais habilidade tendem a morrer menos, logo a margem de mortes é menor quanto maior a habilidade do jogador. O limite superior das margens que sugerem um aumento de dificuldade são semelhantes, com erro de no máximo 2 mortes. As margens dos níveis mais altos sugerem que jogadores com mais habilidades possuem mais perseverança, pois morrem mais antes de relatarem que estão estressados.

Como exemplo de adaptação baseada em morte, na tabela 6.1 que é relativa ao nível 0 se o jogador tiver no máximo uma morte, ele irá para o próximo nível na dificuldade alta, se tiver de 2 à 6 mortes para dificuldade média e se tiver 7 ou mais mortes vai para a dificuldade baixa.

Tabela 6.1: Adaptação de dificuldade baseada em mortes: Nível 0

	<b>Nível 0</b>
<b>Alta</b>	0 - 1
<b>Média</b>	2 - 6
<b>Baixa</b>	7+

Tabela 6.2: Adaptação de dificuldade baseada em mortes: Nível 1 e 2

	<b>Nível 1B</b>	<b>Nível 1M</b>	<b>Nível 1A</b>	<b>Nível 2B</b>	<b>Nível 2M</b>	<b>Nível 2A</b>
<b>Alta[A]</b>	- - - - -	0 - 3	0 - 10	- - - -	0 - 2	0 - 10
<b>Media[M]</b>	0 - 3	4 - 8	11+	0 - 2	3 -6	11+
<b>Baixa[B]</b>	4+	8+	- - - -	3+	6+	- - - -

Observou-se que os valores derivados do sensor EDA variavam em ordens de grandeza nos primeiros testes, alguns dados vinham na ordem de centenas, enquanto outro de dezenas. A origem dessa variação não foi descoberta, mas a troca dos eletrodos, por outros que vieram juntos do kit do BITalino, estabilizou o restante dos testes. Entretanto

Tabela 6.3: Adaptação de dificuldade baseada em mortes: Nível 3 e 4

	Nível 3B	Nível 3M	Nível 3A	Nível 4B	Nível 4M	Nível 4A
<b>Alta</b> [A]	-----	0 - 3	0 - 7	-----	0 - 3	0 - 7
<b>Media</b> [M]	0 - 1	4 - 6	8+	0 - 1	4 -7	8+
<b>Baixa</b> [B]	3+	7+	-----	2+	8+	-----

usar uma margem para os valores do EDA se tornou inviável. Realizando uma média dos menores valores tônicos [37] e subtraindo-a da média total, foi possível notar uma margem entre os valores de estresse, tédio ou desafio. Essa margem, exibida na Tabela 6.4, mostra a distância média dos menores valores da média total, medida que foi usada para definição dos estados afetivos usados nos ADD híbrido e no ADD baseado somente no estado afetivo. Os números usados para calcular essas médias foram os valores adquiridos através do sensor EDA do *BITalino* [33] após a finalização de qualquer fase.

Tabela 6.4: Definição de estados baseada no sinal tônico sensor EDA, usando a diferença da media dos dez menores valores com a media total

	Margem
<b>Estresse</b>	maior que 0,2
<b>Desafio</b>	entre 0,1 e 0,2
<b>Tédio</b>	menor que 0,1

Entretanto não foram todos os resultados que se encaixaram nesta medida adotada. Quatro valores relatados como estresse estavam na margem de desafio usada, porém três valores estavam na mesma margem mas foram relatados como tédio. Esses resultados podem ser atribuídos ao erro da margem que não engloba todas possibilidades pela pequena amostragem ou ao erro humano, que pode ter falhado em reconhecer seu estado afetivo. Foram 10 resultados relatados como "estressados", 13 resultados relatados como "desafiados", 14 resultados como "entediados". Uma limitação observada durante o teste foi a falha do sensor em sua comunicação *bluetooth* com o computador, ocasiões onde o *MATLAB* encerrava o programa de captura de dados antes do jogador terminar o jogo e momentos onde o elétrodo descolava da mão do participante e o mesmo não notava. Essas limitações tiveram como consequência a perda de recolhimento de dados para esta análise.

Os números presentes na Tabela 6.4 mostram que a diferença entre os estados afetivos, com base nos valores tônicos do sensor EDA, é pequena, algo que esta de acordo com o comportamentos desse tipo de dado [37]. Assim é possível perceber que se a média for muito maior que os menores valores da fase, houve um aumento considerável nos valores tônicos, indicando um estado de estresse. Considerando o caso contrário, se a média dos menores valores estiver muito perto da média total, o valor tônico não alterou o

suficiente para manter a atenção do jogador, baseando-se nos relatos de estado. Valores mais confiáveis e precisos podem ser obtidos com uma amostragem maior e o uso de outros sensores em conjunto com o EDA para não depender somente do relato dos participantes.

## 6.2 Comparação de *flow* entre ADDs

Dezesseis voluntários participaram desta bateria de testes, sendo cinco mulheres e onze homens, com idade de 14 a 31 anos, dos quais onze pessoas haviam participado dos testes anteriores. Os participantes foram divididos entre os três modelos de ADD. Seis participantes jogaram o modelo híbrido, dos quais dois eram jogadores casuais e quatro eram experientes. Cinco jogaram o modelo de ADD baseado na performance do jogador dentro do jogo, sendo quatro experientes e um casual. Cinco jogaram o modelo baseado no estado afetivo percebido pelo sensor EDA, com três jogadores experientes e dois casuais.

Para fins de comparação entre os três modelos de ADD, após uma sessão de jogo, o voluntário respondeu a um questionário, feito no *Google Docs* [38] e fundamentado nos aspectos de *flow* presentes na experiência de jogos [15]. O questionário encontra-se em sua totalidade no Apêndice A. Este foi respondido no mesmo computador que o jogo foi jogado e era iniciado pelos pesquisadores após o participante encerrar a sessão de jogo.

**Flow Index**

\*Obrigatório

**O desafio**

As afirmações a seguir se referem ao nível de dificuldade do jogo. Leia-as e indique o grau com que você concorda com cada uma delas. Quanto maior a nota que der, maior o nível de concordância com a afirmação.

**Eu fui desafiado pelo jogo, mas achei que era capaz de superar todos os desafios. \***

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	Concordo totalmente				

**Jogar este jogo me desafiou. \***

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	Concordo totalmente				

**Eu me senti confortável com os controles do jogo. \***

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	Concordo totalmente				

Figura 6.2: Seção Desafio do Questionário

O questionário foi separado em quatro seções: Objetivo, Desafio, Imersão e Dados Sociodemográficos. Nesta pesquisa o foco para análise foi o Desafio do jogo, assim como a Imersão. As perguntas usadas nestas seções citam cada aspecto de *flow* e questionam o quanto cada aspecto foi verdadeiro para a experiência, em uma escala de 1 a 5, onde 5 significa concorda e 1 discorda completamente. A Figura 6.2 mostra a tela vista pelo participante ao começar a responder a Seção de Desafio, mostrando três afirmações e o grau de concordância que o participante sentiu com cada afirmação. A Seção de Objetivo, feita para saber se os participantes compreenderam o que devia ser feito no jogo e busca entender o comportamento dos mesmos durante o *gameplay*, mostrou que os jogadores tinha em todos os modelos prioridade alta em atravessar o campo de asteroides e levar o menor dano possível, indicando que os objetivos do jogo estavam claros. Dados Sociodemográficos priorizam o conhecimento sobre os participantes e a comprovação da diversidade de indivíduos que fizeram partes dos testes, com propósito de que o resultado possa ser mais amplo e não restrinja a um grupo de indivíduos, o que foi demonstrado nas respostas dessa seção, que se mostraram bastante diversificadas. A Tabela 6.5 mostra a média das respostas das Seções Desafio e Imersão do questionário para cada um dos três modelo de ADD testados.

Tabela 6.5: Média das respostas do Questionário

	<b>Performance</b>	<b>Híbrido</b>	<b>Estado Afetivo</b>
<b>Desafio</b>	3,8	4,08	3,867
<b>Imersão</b>	3,46	3,73	3,18

Observando os valores da Tabela 6.5, as médias pertencentes ao ADD híbrido são superiores a ambas médias dos outros modelos, tanto na Seção Desafio como na Seção Imersão. Este fato pode indicar que o modelo híbrido de ADD faz o jogador se sentir no estado de *flow* melhor que os outros, ou faz com que o jogador perceba o estado de *flow* mais facilmente. Outras observações podem ser feitas com os dados resultantes do questionário. O valor superior das Seções Imersão e Desafio do modelo híbrido pode indicar que o desafio afeta a imersão positivamente, pois este era o único fator diferente nos três modelos. Entretanto as médias da Seção de Desafio foram maiores em todos os modelos, indicando que outros fatores afetam a imersão, não somente o desafio. A média da Seção Desafio do modelo de performance foi inferior ao modelo afetivo por uma margem muito pequena, ao contrário do que aconteceu com a imersão que foi superior ao modelo afetivo por uma margem considerável.

# Capítulo 7

## Considerações Finais

Existem vários estudos em ajuste dinâmico de dificuldade (ADD) que levam em conta vários fatores, dentre eles variáveis de performance do jogador ou estado afetivo. Não foi encontrado na literatura pesquisas que utilizam estes fatores em conjunto para definir a dificuldade adequada durante um jogo. Este trabalho teve, portanto, como objetivo principal investigar se um ADD híbrido (baseado em dados tanto do jogo como do usuário) mantém o jogador no estado *flow* em comparação com os ADDs que levam em consideração apenas um fonte de dados.

Para isso foi implementado uma adaptação de um jogo *open source* feito em *Unity* chamado "*Asteroids: in the 2nd and 1/2th Dimension*" [14] e nela foram construídas três versões de um ADD, ou seja, (1) um baseado em dados de performance, derivados do jogo, (2) outro que utiliza o *feedback* afetivo do jogador, coletado por meio de sensores e (3) um híbrido, que combina os dois anteriores. Para a captura dos dados relativos à afetividade do jogador, foi utilizado um sensor EDA (*Electrodermal Activity*) com tratamento de dados no *MATLAB*. Também foram conduzidos testes e aplicados questionários, com jogadores de habilidades e frequência de horas jogadas diferentes.

Através da análise das respostas do questionário aplicado na segunda bateria de testes foi possível observar uma média superior do modelo híbrido proposto nos quesitos relacionados a desafio e imersão, em comparação com os outros dois modelos de ADD testados. Ambos aspectos estão relacionados ao estado de *flow*, e portanto, há indícios de que a hipótese desta investigação pode estar correta. Contudo, observa-se que a diferença entre as médias apresentadas no capítulo anterior não foi o suficiente para comprovar que o ADD híbrido é melhor em relação aos outros modelos.

A limitação de um ano de trabalho e dos recursos disponíveis não permitiram que os resultados desta pesquisa pudessem ser generalizados para uma gama mais ampla de jogos e indivíduos. Sendo assim, como trabalhos futuros neste campo de pesquisa sugere-se a condução de mais testes com um número mais abrangente de indivíduos. Também

será útil a exploração do uso de múltiplos sensores para uma melhor definição de estados afetivos, assim como a utilização de mais variáveis derivadas da performance do jogador. Com os sensores cada vez menores e menos intrusivos, essa é uma área que pode ser cada vez mais explorada e usada em futuros trabalhos.

# Referências

- [1] Ju Sousa. O cosmo das emoções: a neurociência. <http://bebrainiac.blogspot.com.br/2014/05/o-cosmo-das-emocoes-neurociencia.html>. Acessado em 05/03/2017. x, 8
- [2] BBPL DE ARAUJO e B FEIJ O. *Um estudo sobre adaptatividade dinâmica de dificuldade em jogos*. PhD thesis, Dissertação de mestrado, Pontifícia Universidade Católica do Rio de Janeiro-PUC-Rio, Rio de Janeiro, Brasil, 2012. x, 9, 10
- [3] Wikipedia. Artigo sobre asteroids na wikipedia. [https://en.wikipedia.org/wiki/Asteroids\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Asteroids_(video_game)). Acessado em 17/03/2017. x, 21
- [4] BITalino. Eda sensor datasheet, 2015. original document from BITalino. x, 8, 36, 37
- [5] D. Gularte. *Jogos eletrônicos: 50 anos de interação e diversão*. 2AB Editora, 2010. xi, 5
- [6] Robin Hunicke. The case for dynamic difficulty adjustment in games. pages 429–433, 2005. 1, 5, 16
- [7] Georgios N. Yannakakis e John Hallam. Real-time game adaptation for optimizing player satisfaction. *IEEE Transactions on Computational Intelligence and AI in Games 1.2*, pages 121–133, 2012. 1, 13, 17
- [8] Mihaly Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. New York: Harper & Row, 1992. 1, 8, 16
- [9] Changchun Liu, Pramila Agrawal, Nilanjan Sarkar, e Shuo Chen. Dynamic difficulty adjustment in computer games through real-time anxiety-based affective feedback. *International Journal of Human-Computer Interaction*, pages 506–529, 2009. 1, 2, 19
- [10] Daniel Imre. *Real-time Analysis of Skin Conductance for Affective Dynamic Difficulty Adjustment in Video Games*. PhD thesis, Algoma University, 2016. 1, 2, 20, 34
- [11] Lingdao Sha. Creating appropriate challenge level game opponent by the use of dynamic difficulty adjustment. 2010. 2, 18
- [12] Alena Denisova e Paul Cairns. Adaptation in digital games: The effect of challenge adjustment on player performance and experience. *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, pages 97–101, 2015. 2, 17

- [13] Arend Hintze, Randal Olson, e Joel Lehman. Orthogonally evolved ai to improve difficulty adjustment in video games. In *Applications of Evolutionary Computation*, pages 525–540. Springer, 2016. 2, 15
- [14] Matthew Renze. Asteroids: in the 2nd and 1/2th dimension. <http://www.matthewrenze.com/software.html>. Acessado em 02/07/2016. 2, 21, 23, 46
- [15] Xiaowen Fang, Jingli Zhang, e Susy S. Chan. Development of an instrument for studying flow in computer game play. *Intl. Journal of Human–Computer Interaction*, 29:456–470, 2013. 3, 44
- [16] Jasper Juul. The game, the player, the world: Looking for a heart of gameness. *Level Up: Digital Games Research Conference Proceedings*, pages 30–45, 2003. 4
- [17] Guilherme Xavier. *Condição Eletrolúdica - Cultura Visual Nos Jogos Eletrônicos*. Editora Novas Idéias, 2010. 4
- [18] Katie Salen e Eric Zimmerman. *The game design reader: a rules of play anthology*. MIT press, Cambridge Mass, 2006. 4
- [19] Martin Jennings-Teats, Gillian Smith, e Noah Wardrip-Fruin. Polymorph: dynamic difficulty adjustment through level generation. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, page 11. ACM, 2010. 5
- [20] Jeannie Novak. *Desenvolvimento de Games*. Cengage Learning, Brazil, 2011. 6
- [21] Nick Fortugno. *The strange case of the casual gamer*. 2008. 6
- [22] Peter J Lang. The emotion probe: studies of motivation and attention. *American psychologist*, 50(5):372, 1995. 7
- [23] Guillaume Chanel, Cyril Rebetez, Mireille Bétrancourt, e Thierry Pun. Emotion assessment from physiological signals for adaptation of game difficulty. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(6):1052–1063, 2011. 7
- [24] John L. Sherry. Flow and media enjoyment. *Communication Theory*, pages 328–347, 2004. 10
- [25] David Michael Jordan Chang. Dynamic difficulty adjustment in computer games. acessado em 02/07/2016. 11
- [26] Fausto José Mourato e Manuel Próspero dos Santos. Measuring difficulty in platform videogames. *4ª Conferência Nacional Interação humano-computador*, pages 1–8, 2009. 14
- [27] Kate Compton e Michael Mateas. Procedural level design for platform games. *Proceedings of the 2nd Artificial Intelligence and Interactive Digital Entertainment Conference*, pages 109–111, 2006. 14

- [28] Christyowidiasmoro, Ramadhany Candra Arif Putra, e Supeno Mardi Susiki. Measuring level of difficulty in game using challenging rate (cr) on 2d real time strategy line defense game. *2015 International Electronics Symposium*, pages 218–222, 2015. 15
- [29] Jonathan Blow. Game development: Harder than you think. pages 29–37, 2004. 21
- [30] Atari. Números de produção da atari inc. <http://www.atarigames.com/atarinnumbers90s.pdf>. Acessado em 17/03/2017. 21
- [31] Mercado de sistemas operacionais. <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0>. Acessado em 17/03/2017. 22
- [32] Unity - game engine. <http://unity3d.com/pt/>. Acessado em 02/07/2016. 28, 35
- [33] Bitalino. <http://www.bitalino.com/>. Acessado em 24/02/2017. 35, 36, 43
- [34] Matlab. <https://www.mathworks.com/products/matlab.html>. Acessado em 24/02/2017. 35, 36
- [35] Robert Schleicher. Simpleeda/emg. <https://www.mathworks.com/matlabcentral/fileexchange/9223-simpleeda-emg>. Acessado em 26/02/2016. 36
- [36] MathWorks Instrument Control Toolbox Team. Bitalino support from matlab. <http://www.mathworks.com/matlabcentral/fileexchange/53983-bitalino-support-from-matlab>. Acessado em 26/02/2016. 36
- [37] What should i know to use eda data in my experiment. <https://support.empatica.com/hc/en-us/articles/203621955-What-should-I-know-to-use-EDA-data-in-my-experiment->. Acessado em 02/03/2017. 43
- [38] Google. Google docs. <https://www.google.com/docs/about/>. Acessado em 09/03/2017. 44

# Apêndice A

## Questionário

Para as Seções 1, 2 e 3 o participante deve responder de 1 (discordo totalmente) a 5 (concordo totalmente)

### A.1 Objetivo

1. No jogo, quão importante para você foi cada uma das ações listadas a seguir?
  - Atravessar o campo de asteroides.
  - Explorar todo o espaço do campo de asteroides.
  - Evitar que a nave sofresse danos.
  - Destruir o máximo de asteroides possível.
  - Melhorar meu desempenho no jogo.
  - Dominar 100% do controle sobre a nave.
  - Completar a fase o mais rápido que pudesse.
2. Você tinha noção clara de quão bom era seu desempenho durante a sessão de jogo?

### A.2 Desafio

1. Jogar este jogo me desafiou.
  2. Este jogo pode ser um bom teste das minhas habilidades.
  3. Achei que esse jogo exigiu o limite das minhas habilidades.
  4. Eu fui desafiado pelo jogo, mas achei que era capaz de superar todos os desafios.
  5. Quando estava jogando o jogo, me senti sob controle sobre o que estava fazendo.
  6. Eu me senti confortável com os controles do jogo.

### A.3 Imersão

1. Minha atenção estava inteiramente focada no jogo.
2. Quando estava jogando, estava completamente concentrado no que estava fazendo.
3. Jogar esse jogo é, por si, satisfatório.
4. Eu adorei a sensação do meu desempenho e quero de senti-la novamente.
5. Eu gostei da experiência de jogar esse jogo.
6. Em vários momentos da partida, eu me vi fazendo coisas automaticamente, sem que tivesse que pensar.
7. Eu senti dentro do mundo criado pelo jogo enquanto o jogava.
8. Eu meio que perdi a noção de mim mesmo(a) enquanto jogava.
9. Quando joguei o jogo, às vezes senti como se coisas se movessem em câmera lenta.
10. Eu perdi a noção do tempo enquanto jogava o jogo.

### A.4 Dados sociodemográficos

1. Com que frequência você joga nas diferentes plataformas? Responda de 1 (diariamente) à 4 (não jogo).
  - Celular / tablet
  - Console de mesa (e.g., PlayStation 4, XBOX One, Nintendo Wii)
  - Console portátil (e.g., Nintendo 3DS, Playstation Vita)
  - Computador pessoal (desktop ou laptop)
2. Quantos anos você tem? Responda de 16 à 50.
3. Qual o seu sexo? Responda masculino ou feminino.
4. Qual a sua impressão geral sobre jogos do tipo de Asteroids? Responda de 1 (não gosto) à 5 (é meu tipo de jogo favorito).