UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUCAS CORDEIRO MARQUES
LUCAS CLEMENTE MIRANDA BRAGA

ELABORAÇÃO DE UM JOGO MULTIJOGADOR ASSIMÉTRICO: DESIGN, DESENVOLVIMENTO E PUBLICAÇÃO

RIO DE JANEIRO
2021

LUCAS CORDEIRO MARQUES
LUCAS CLEMENTE MIRANDA BRAGA

ELABORAÇÃO DE UM JOGO MULTIJOGADOR ASSIMÉTRICO: DESIGN, DESENVOLVIMENTO E PUBLICAÇÃO

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Geraldo Bonorino Xexéo, D. Sc

RIO DE JANEIRO
2021

M357e

Marques, Lucas Cordeiro
Elaboração deum jogo multijogador assimétrico: design, desenvolvimento e publicação / Lucas Cordeiro Marques, Lucas Clemente Miranda Braga. – Rio de Janeiro, 2021.

67 f.

Orientador: Geraldo Bonorino Xexéo.
Coorientador: Leandro Ouriques Mendes de Carvalho.
Coorientador: Eduardo Freitas Mangeli de Brito.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Bacharel em Ciência da Computação, 2021.

1. Matemática. 2. Jogos assimétricos. 3. Multijogador. 4. Design. 5. Desenvolvimento. I. Braga, Lucas Clemente Miranda. II. Xexéo, Geraldo Bonorino (Orient.). III. Carvalho, Leandro Ouriques Mendes de (Coorient.). IV. Brito, Eduardo Freitas Mangeli de (Coorient.). V. Universidade Federal do Rio de Janeiro, Instituto de Matemática. VI. Título.

LUCAS CORDEIRO MARQUES
LUCAS CLEMENTE MIRANDA BRAGA

ELABORAÇÃO DE UM JOGO MULTIJOGADOR ASSIMÉTRICO: DESIGN, DESENVOLVIMENTO E PUBLICAÇÃO

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em: 15 de junho de 2021

BANCA EXAMINADORA:

Prof. Geraldo Bonorino Xexéo

D.Sc

participação por vídeo conferência

Prof. Leandro Ouriques Mendes de Carvalho

M.Sc

participação por vídeo conferência

Prof. Eduardo Freitas Mangeli de Brito

D.Sc

participação por vídeo conferência

Prof. Rafael Studart Monclar

M.Sc

participação por vídeo conferência

Profa. Adriana Santarosa Vivacqua

D.Sc

AGRADECIMENTOS

Lucas Cordeiro Marques

Quero agradecer aos meus familiares, que me incentivaram e me deram apoio quando precisei. Em especial, meus pais, Cristiane e Roberto, sem os quais não existiria essa combinação de átomos e ansiedade que eu me acostumei a chamar de mim; e Louise, minha prima, que me ajudou com a revisão de texto e com conselhos.

Agradeço também a meu orientador, Xexéo, por nos indicar material de pesquisa e a seus orientandos, Mangeli, que contribuiu com discussões importantes, e Leandro, que deu feedback e dicas sem as quais eu não sei como esse trabalho teria terminado, além de ter conseguido muitos voluntários para testar o projeto.

AGRADECIMENTOS

Lucas Clemente Miranda Braga

Quero agradecer aos meus pais, que me deram toda a base para enfrentar a graduação e chegar até aqui, apesar de todas as barreiras. Em especial a minha mãe, Ercilaide, que em muitos momentos me ajudou com o lado emocional e proporcionou leveza e calma para continuar seguindo em frente.

Aos meus amigos, que em momentos de estresse me proporcionaram um pouco de descontração que me fizeram recuperar as energias para conseguir retomar aos trilhos.

Agradeço também ao meu orientador Xexéo, que contribuiu bastante com conselhos, diretrizes e material de apoio para que esse projeto se tornasse possível. Agradeço também ao Mangeli, que contribui com opiniões e debates que enriqueceram o projeto como um todo, e ao Leandro, que mesmo sem muito contato, nos ajudou na revisão da monografia e também com voluntários para testar o projeto.

RESUMO

Este trabalho aborda o *design*, o desenvolvimento e a publicação de um jogo multiplayer assimétrico. O design utilizou os *frameworks* MDA (*Mechanics, Dynamics and Aesthetics*) e 6-11 (seis emoções e onze instintos) como possibilidades para análises das ações permitidas aos jogadores e da estética do jogo. O desenvolvimento do jogo foi realizado com a *engine* de jogos Unity e foi utilizado o serviço Photon para prototipagem do serviço multijogador. Em seguida, a biblioteca Telepathy foi utilizada para comunicar as aplicações por meio de mensagens. Por fim, relata-se a publicação do jogo, resultados obtidos e possíveis pontos de melhorias.

Palavras-chave: Jogo; multijogador; assimétrico; design; desenvolvimento.

ABSTRACT

This work addresses the design, development and publishing of an asymmetric multiplayer game. Its design uses the frameworks MDA (Mechanics, Dynamics and Aesthetics) and 6-11 (six emotions and eleven instincts) which are utilized as possible means of analyzing the game's aesthetics. For its development the Unity game engine was used, along with the Photon service for the multiplayer routine's prototyping. The project was then adapted to use a message-based communication between applications with the Telepathy library. Lastly, the text mentions the game's publication, obtained results and possible improvements.

Keywords: Game; multiplayer; asymmetric; design; development.

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS

MDA – *Mechanics, Dynamics and Aesthetics*, em português: Mecânicas, Dinâmicas e Estéticas.

NES – *Nintendo Entertainment System*

SNES – *Super Nintendo Entertainment System*

SUMÁRIO

1 INTRODUÇÃO

Jogos se mostram cada vez mais populares tanto como mídia de consumo, mas também como uma forma de negócio viável para pequenos desenvolvedores. Aqueles que são desenvolvidos de forma independente, conhecidos como *indie games*, vêm crescendo em popularidade desde meados da década de 2000 até os dias atuais (COBBETT, RICHARD, 2017). Braid (Jonathan Blow, 2008) e Super Meat Boy (Team Meat, 2010) são exemplos de jogos feitos por equipes pequenas que se tornaram muito populares nesta época. O desenvolvimento desses dois jogos foi abordado no filme Indie Game: The Movie (SWIRSKY, JAMES, 2012).

Entre os jogos, existem aqueles que são multijogadores, ou seja, requerem ou incentivam que o jogo tenha a participação de mais de um jogador, em geral também humanos, podendo ou não serem controlados por inteligências artificiais. Estes jogadores tomam decisões significativas para, eventualmente, atingir, ou não, um objetivo pré-definido. Ainda entre os jogos multijogadores existem os chamados "assimétricos", que são jogos que adotam uma filosofia de design voltada a estabelecer, incentivar e aproveitar diferenças entre os jogadores (HARRIS, JOHN; SCOTT, STACEY; HANCOCK, MARK, 2016). Estas diferenças se manifestam, por exemplo, em quais mecânicas estão estabelecidas para cada jogador, qual objetivo cada um precisa atingir, entre outros aspectos.

Um bom exemplo de um jogo multijogador clássico são os jogos da série Mario Kart (NINTENDO, 1992). Nele, cada jogador controla um personagem fictício que pilota algum veículo terrestre, geralmente um kart, em um mundo lúdico. Os personagens participam em uma corrida na qual almejam chegar ao fim do percurso em primeiro lugar, utilizando-se de itens encontrados pela pista para atrapalhar outros corredores e obter vantagens. Este jogo possui diversas características que são marcantes no gênero. Entre elas temos a possibilidade de se jogar contra outros humanos ou contra uma inteligência artificial, a existência de um objetivo claro e similar para todos e, também, ser jogado de forma igual para todos os participantes. Todos os jogadores possuem exatamente os mesmos controles, os mesmos objetivos e as mesmas possibilidades de interagir com o mundo virtual.

Em contrapartida, um exemplo de jogo assimétrico é o Keep Talking and Nobody Explodes (Steel Crate Games, 2015). Este é um jogo cooperativo que exige dois participantes. Um deles tem o papel de desarmar uma bomba, a qual é exibida em uma tela, e possui diversos módulos que precisam ser desativados de formas específicas para cada modo, determinadas em um manual. Já o outro, que não pode observar a tela do primeiro jogador, tem acesso ao manual que explica como cada módulo deve ser operado. Ambos os jogadores precisam se comunicar

de forma verbal, sem compartilhar o que cada um consegue ver com o outro, de forma a atingir um objetivo comum, que é desarmar a bomba. Entretanto, é interessante observar que as ações que cada jogador desempenha para chegar a esse objetivo são distintas.

O clássico Lobisomem é um outro exemplo de jogo multijogador assimétrico. Ele foi baseado no jogo Mafia (DAVIDOFF, DIMITRY, 1986), comumente conhecido no Brasil como "Cidade Dorme", mas também por outros nomes. Lobisomem é jogado apenas com um grupo de pessoas que conseguem se comunicar, seja fisicamente ou por meios digitais. Neste jogo, um narrador é responsável por designar secretamente papéis diferentes aos jogadores, tais como lobisomem, detetive, ou aldeão. O jogo se passa em dois turnos: a noite, em que os lobisomens votam, em conjunto e sem o conhecimento dos outros jogadores, em um jogador para ser eliminado; e o dia, em que os aldeões tomam conhecimento de quem foi eliminado e devem votar por expulsar um jogador que suspeitam ser o lobisomem. O detetive tem, ainda, a chance de perguntar ao narrador a cada turno da noite, sem que os outros saibam, se um jogador específico é o lobisomem, ganhando mais informação para argumentar nos turnos de dia sobre quem deve ser linchado. Estes exemplos são interessantes para ressaltar que a assimetria pode ser uma opção para o design tanto de jogos cooperativos como competitivos, sejam eles analógicos ou digitais.

Mafia ou Lobisomem pertencem a um gênero de jogos chamado de Dedução Social, caracterizado pelos diversos papéis secretos associados aos jogadores em cada partida. Diversos jogos com características muito semelhantes se popularizaram, como The Resistance (DON ESKRIDGE, 2009). Em 2020, impulsionado pelo isolamento causado pandemia de COVID-19, um jogo digital, multijogador online e assimétrico, *Among Us* (INNERSLOTH, 2018), cresceu muito em popularidade (GOSLIN, AUSTEN, 2020), demonstrando o potencial deste tipo de jogo, especialmente como forma de aproximar jogadores através do desempenho de funções distintas, mas que acabam por conversar entre si.

1.1 MOTIVAÇÃO

Os autores deste trabalho entendem que, embora não haja nada de errado com jogos multijogador clássicos e simétricos, há um potencial não explorado nos jogos assimétricos. Este tipo de jogo pode levar a interações incomuns ou inusitadas entre os jogadores, além de ser um exemplo importante de design experimental de jogos. Assim sendo, a motivação para este projeto é, em grande parte, pessoal.

1.2 OBJETIVOS

O objetivo deste trabalho é implementar um jogo multijogador assimétrico, estabelecendo antes uma metodologia de design e desenvolvimento. Também pretende-se publicá-lo em uma plataforma de distribuição para que fique disponível na internet.

1.3 METODOLOGIA

A metodologia deste projeto segue a linha descrita a seguir. Primeiro há o planejamento, ou seja, a ideia inicial e organização de etapas da confecção do jogo; em seguida é feito o design baseado em princípios pré-estabelecidos; depois o desenvolvimento, realizado com as tecnologias escolhidas, para se obter um protótipo, que será testado e iterado para obter um produto final. Por fim, publica-se o jogo em uma plataforma de distribuição online. Nos capítulos a seguir estes passos serão abordados e detalhados individualmente.

2 DESIGN DE JOGOS

Antes do desenvolvimento propriamente dito de um jogo, é conveniente ou até mesmo necessário, de acordo com o escopo em questão, utilizar ferramentas de design para realizar o planejamento e o projeto do mesmo. Essas ferramentas contribuem para garantir que as funcionalidades elaboradas para o jogo sejam mais sólidas e conversem melhor entre si, além de formalizar o design do jogo e se comunicar mais claramente a respeito do projeto. Existem diversas abordagens que podem ser utilizadas para o design de um jogo. Neste trabalho foram utilizados os *frameworks* MDA (ZUBEK, ROBERT; LEBLANC, MARC, 2004) e 6-11 (DILLON, ROBERTO, 2011), como descrito na metodologia *On the Way to Fun* (DILLON, ROBERTO, 2010), os quais serão detalhados a seguir.

2.1 O FRAMEWORK MDA

O *framework* MDA (*Mechanics, Dynamics and Aesthetics*) foi elaborado por Robin Hunicke, Marc LeBlanc, Robert Zubek e descrito no artigo *MDA: A Formal Approach to Game Design and Game Research*. A sigla MDA pode ser traduzida literalmente por Mecânicas, Dinâmicas e Estéticas.

Segundo o método do MDA, um jogo pode ser composto por Mecânicas, que são suas regras e componentes fundamentais. As Dinâmicas são geradas a partir das interações que ocorrem entre as Mecânicas. As Dinâmicas provocam e são provocadas pelas Estéticas, que descrevem as respostas emocionais causadas no jogador ao interagir com o jogo. Do ponto de vista do designer, ele age na construção das Mecânicas do jogo, que vão gerar um sistema de diferentes Dinâmicas que, então, fará o jogador reagir com a Estética previamente esquematizada, se o design funcionar como esperado. Do ponto de vista do jogador, ele experimenta a Estética, que é proporcionada por dinâmicas, sendo causada pela atuação do jogador no jogo e pelas regras do mesmo, descritas pelas mecânicas. O objetivo desta metodologia é pensar em quais respostas emocionais o designer pretende despertar no jogador. Então, confeccionam-se mecânicas, que habilitam dinâmicas. Estas, por sua vez, provocam as respostas emocionais pretendidas no jogador.

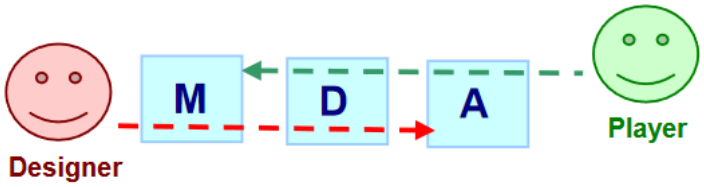


Figura 1 - As perspectivas opostas do jogador ao experienciar o jogo e do designer ao confeccioná-lo (ZUBEK, ROBERT; LEBLANC, MARC, 2004)

É interessante notar que a Estética é um artefato abstrato no modelo. As respostas emocionais que um ser humano pode ter são subjetivas e diversas. Assim, o modelo aborda uma resposta emocional pretendida, mas que pode não ocorrer em parte dos jogadores. Além disso, é comum adotar um conjunto bem definido dessas respostas para se falar da Estética ao utilizar esse modelo, formando um vocabulário próprio para esta discussão.

2.1.1 Estética no MDA

No artigo original são listadas as seguintes respostas emocionais, mas também esclarece que não são únicas,: Sensação, ou seja, jogos como experiências sensoriais e/ou prazerosas; Fantasia, jogos como faz de conta; Narrativa, jogos como dramatização; Desafio, jogos como experiências com obstáculos a serem ultrapassados; Comunhão, jogos como experiências sociais; Descoberta, jogos como ambientes a serem explorados; Expressão, jogos como autodescoberta e, por fim, Submissão, jogos como passatempo ou distração. Adotando-se este vocabulário, pode-se então debater com mais facilidade a estética de um jogo. Um jogo de aventura e investigação em geral terá maior foco em Narrativa, Fantasia e Descoberta, enquanto um de tiro multijogador terá como elementos estéticos centrais de seu design o Desafio, a Sensação, e até mesmo a Expressão, através da customização de personagens, por exemplo. Estes diferentes tipos de Estética permitem discutir não sobre a “diversão” ou o “entretenimento” de um jogo, mas sobre as Experiências que o jogador sente ao jogar.

2.1.2 Dinâmicas no MDA

Neste modelo, as Dinâmicas são o que proporcionam as experiências Estéticas. Quando se evidencia uma determinada Estética, deve-se estimular que as Dinâmicas que a suportam se manifestem durante o jogo. Por exemplo, caso o foco do jogo seja a Expressão, é proveitoso fazer com que surjam dinâmicas de criação e exposição no jogo. Os jogos da série *Animal Crossing* (NINTENDO, 2001) são bons exemplos. Neles, o jogador possui uma casa, que pode ser exposta publicamente ou para amigos, e pode decorá-la com diferentes objetos e mobília em diversas posições. As Dinâmicas de exposição e decoração colaboram fortemente para o surgimento da Estética de Expressão, e observe que não foi preciso mencionar quais Mecânicas fazem essas Dinâmicas acontecerem, apenas que essas Dinâmicas existem durante o jogo. No entanto, é claro que deve-se elaborar Mecânicas compatíveis para que ocorra o desenvolvimento das Dinâmicas corretas.

2.1.3 Mecânicas no MDA

As mecânicas são parte crucial do comportamento do jogo. As mecânicas constituem as regras, incluindo o conjunto de possíveis ações que podem ser disparadas pelos controles disponíveis ao jogador. Seguindo o mesmo exemplo da série *Animal Crossing*, as Mecânicas que promovem Dinâmicas de arrumação e decoração são, de forma resumida: posicionamento de objetos, remoção de objetos, rotação e movimentação de objetos posicionados, coloração de objetos, alternação do modo do objeto (como ligar ou desligar uma televisão), troca do papel de parede e piso. É possível perceber que estas Mecânicas colaboram em conjunto para gerar as Dinâmicas de exposição e decoração. Seria difícil imaginar que estas Mecânicas conseguissem gerar as Dinâmicas de forma efetiva isoladamente.

2.2 O FRAMEWORK 6-11

Como foi dito na seção 2.1, para se falar de Estética dos jogos é importante adotar um conjunto de respostas emocionais possíveis. Na seção 2.1.1 foram mostradas as respostas emocionais listadas pelo artigo original sobre MDA, no entanto, pode-se adotar outros conjuntos de respostas emocionais. Roberto Dillon, em seu livro *On the Way to Fun: An Emotion-Based Approach to Successful Game Design* (DILLON, ROBERTO, 2010) e em seu artigo *THE 6-11 FRAMEWORK: A NEW METHODOLOGY FOR GAME ANALYSIS AND DESIGN* (DILLON,

ROBERTO, 2011), detalha o *framework* 6-11. Neste modelo, ele lista seis emoções e onze instintos básicos que podem surgir ao jogar e os utiliza como conjunto das Estéticas que um jogo pode querer invocar.

2.2.1 As emoções no Framework 6-11

Entre as emoções que podem surgir ao jogar, o modelo foca em seis delas: medo, raiva, felicidade (alegria), orgulho, tristeza e excitação. O medo está fortemente presente não só em jogos de terror, mas também de sobrevivência ou espionagem, por exemplo. A raiva está usualmente associada à frustração de uma derrota que gera vontade de jogar novamente. A felicidade (alegria) é fortemente associada a vencer um desafio ou cumprir alguma tarefa que o jogo propõe. O orgulho é relevante, por exemplo, para jogos com tabelas de pontuação, em que se quer manter uma posição alta no ranking. A tristeza é menos comum, mas está associada a um desenrolar trágico na narrativa do jogo, porém mais complicado de se introduzir por meio de Dinâmicas. Por fim, a excitação, que Dillon (DILLON, ROBERTO, 2010) argumenta ser o resultado final que a maioria dos jogos querem obter. Estas emoções disparam e são disparadas por diferentes instintos, e são essenciais para manter o engajamento do jogador. Nesse sentido, é importante pensar em quais momentos e após quais interações as emoções devem ser despertadas.

2.2.2 Os Instintos no Framework 6-11

Como dito, o Framework também descreve instintos despertados no jogador. Estes instintos são agrupados em três categorias: Primeira Pessoa, Terceira Pessoa e Mundo. Cada categoria de instintos possui suas particularidades quanto à maneira com a qual o jogador se envolve com eles.

2.2.2.1 Instintos de Primeira Pessoa

Os instintos de Primeira Pessoa são os voltados para a manutenção do próprio bem estar do jogador. Estes instintos são: Sobrevivência, que representa o ímpeto de manter a si mesmo vivo e seguro, e pode desmembrar-se em uma luta ou fuga; Identificação, fortemente ligado à apreciação e empatia por e/ou representatividade do personagem controlado no jogo; Coleta, que remete à atividade de adquirir (coletar) itens como forma de preparo, por exemplo, sendo

recompensada com vidas extras e melhorias, e Ganância, associada à coleta além do necessário, ou a manutenção de recursos em detrimento de outros jogadores ou oponentes.

2.2.2.2 Instintos de Terceira Pessoa

Estes são os instintos ligados a terceiros, sejam outros jogadores ou personagens e objetos presentes no mundo do jogo. Eles são: Proteção, muito semelhante à sobrevivência, porém voltada a outro personagem ou jogador, cujo bem-estar e segurança quer-se manter, por exemplo em jogos de bichos virtuais; Agressividade, em oposição à Proteção, presente em jogos mais violentos ou competitivos; Vingança, apesar de ser menos comum, normalmente é associada narrativamente à necessidade de volta por cima e triunfo do personagem controlado; Comunicação, mais comum em jogos online ou que promovam a criação de comunidades, mas que permite designs interessantes, como o de *Journey* (THATGAMECOMPANY, 2012), onde jogadores podem se encontrar aleatoriamente e precisam se comunicar com apenas um botão, e Competição, fortemente presente em jogos de pontuação ou multijogador, inclusive, como será mostrado, no jogo desenvolvido para este trabalho.

2.2.2.3 Instintos em Relação ao Mundo

Os instintos em Relação ao Mundo não possuem necessariamente uma conexão com outro objeto, personagem ou indivíduo, mas sim com os cenários e situações do jogo. São apenas dois: Curiosidade, muito comum em jogos de exploração ou que possuam segredos a serem encontrados, e Apreciação de Cor. Entende-se "Apreciação de Cor", na verdade, como apreciação visual ou artística do jogo, ou seja, o quão atraente ou intrigante é a direção visual do mesmo ou de cenas e situações específicas. Este é claramente o instinto mais subjetivo do modelo proposto por Dillon no artigo em que detalha o framework 6-11 (DILLON, ROBERTO, 2011).

2.2.3 Relações entre Instintos e Emoções

Os instintos e as respostas emocionais estão interligados no modelo proposto. Desta forma, uma emoção pode levar a outra ou a um instinto. Igualmente, um instinto pode despertar outro ou causar outra resposta emocional. Estas relações podem ser visualizadas de forma mais prática

na elaboração de diagramas para jogos, que será abordada a seguir. Um gráfico para a interações possíveis entre respostas emocionais e instintos foi sugerido por Dillon.

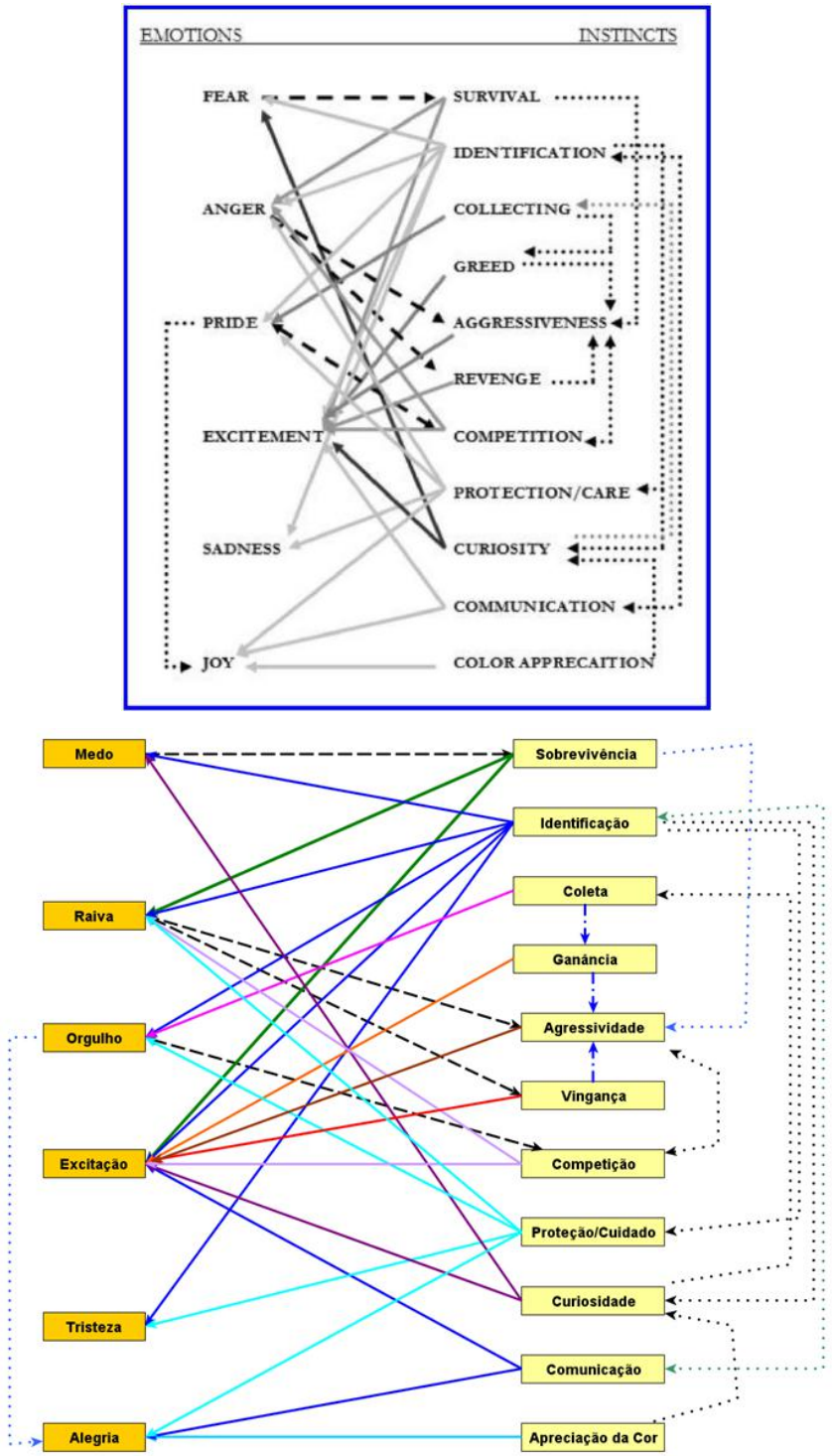
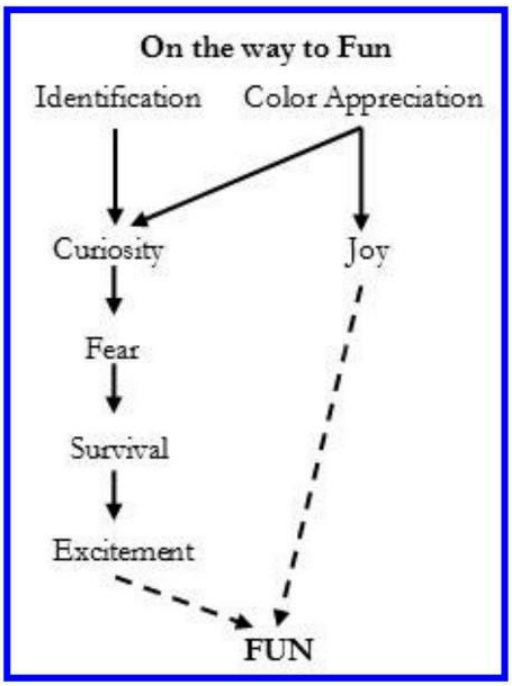


Figura 2 - Diagrama proposto sobre como as emoções e instintos do framework 6-11 interagem entre si (DILLON, ROBERTO, 2010)

2.3 DIAGRAMAS ON THE WAY TO FUN

Dillon contextualiza o uso do MDA e do 6-11 através de diagramas que seguem o método que ele denomina *On the Way to Fun*, homônimo de seu livro (DILLON, ROBERTO, 2010). O método consiste em Explicitar as Mecânicas e Dinâmicas pretendidas para o jogo e mostrar como elas resultam nas Estéticas de Excitação e/ou Alegria que, em geral, são as duas Estéticas mais comuns de se querer obter ao desenvolver um jogo, resultando na noção proposta por Dillon de *Fun*, ou seja, Diversão. Ao propor diagramas para análise do design de jogos, Dillon primeiro associa exemplos de Instintos e Emoções de acordo com as relações da Figura 2. Em seguida, expande esse diagrama para incluir Dinâmicas e Mecânicas, como pode ser visto nas Figuras 3 e 4. Nas seções seguintes estes diagramas serão exemplificados.



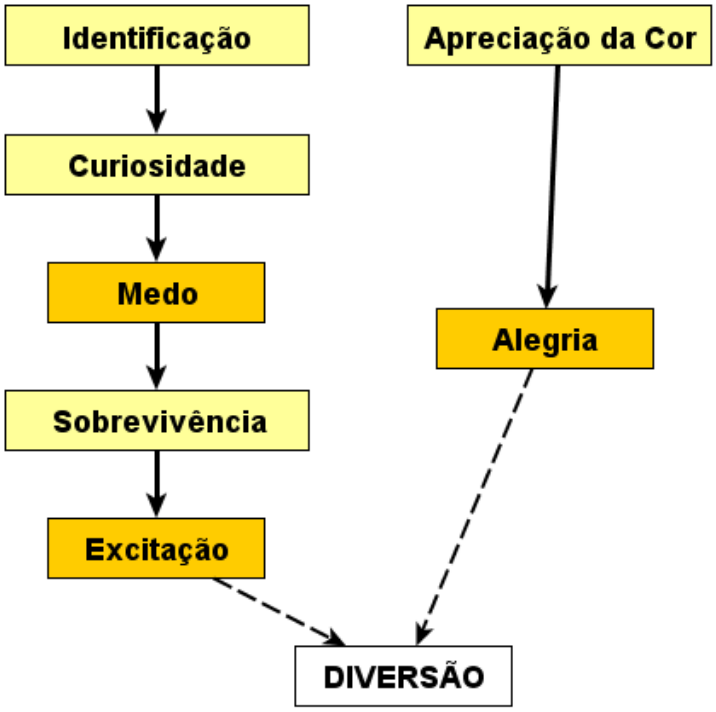
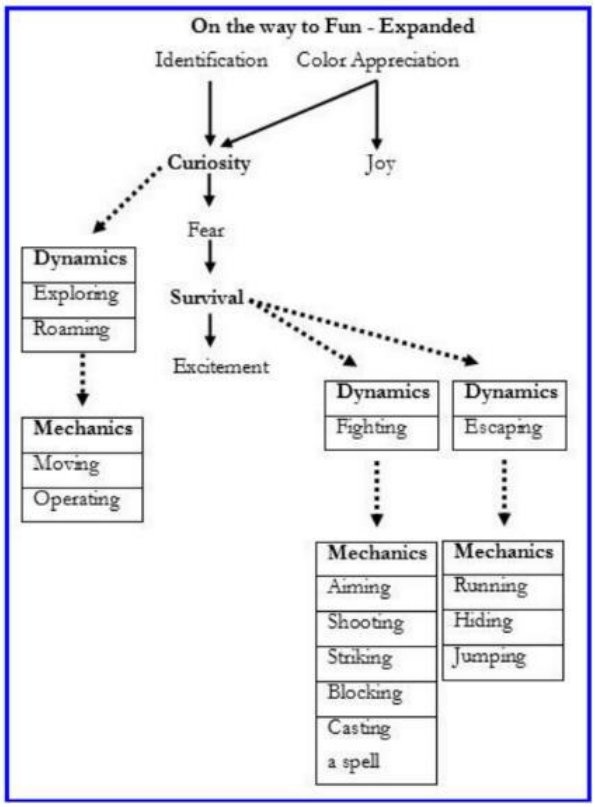


Figura 3 - Diagrama associando apenas emoções e instintos (DILLON, ROBERTO, 2010)



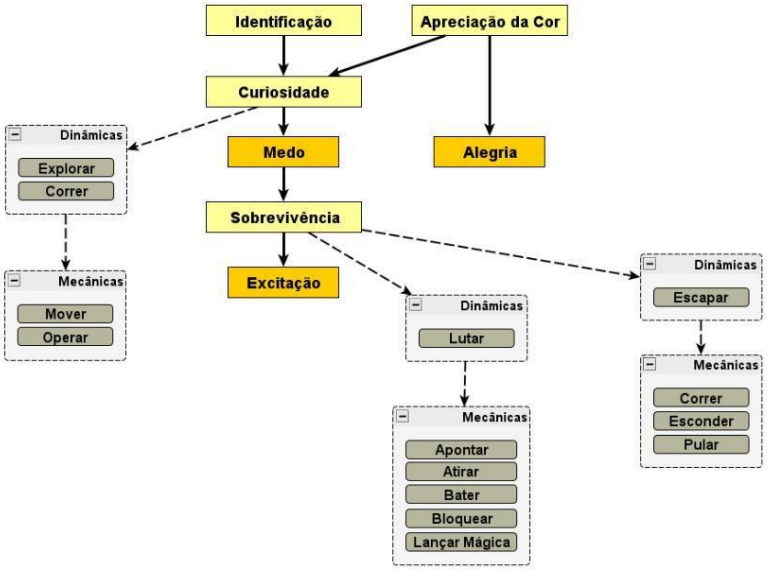


Figura 4 - Diagrama expandido com a inserção de Dinâmicas e Mecânicas (DILLON, ROBERTO, 2010)

2.3.1 Diagrama para Downwell

Downwell (DEVOLVER DIGITAL, 2015) é um jogo em que um personagem deve descer um poço. No caminho ele passa por diversos cenários, evitando ou destruindo inimigos e coletando tesouros, que podem ser utilizados em lojas espalhadas pelos níveis para obter itens. Este é um dos jogos que serviram de inspiração para o jogo desenvolvido por parte deste projeto. Na figura 5, pode-se ver o personagem principal do jogo e o cenário em preto e branco, com os inimigos, as balas e o *HUD* (Heads-Up Display, informações do jogo que se sobrepõem à tela, como a barra de vida do personagem ou quantidade de gemas coletadas) possuindo tons de vermelho.

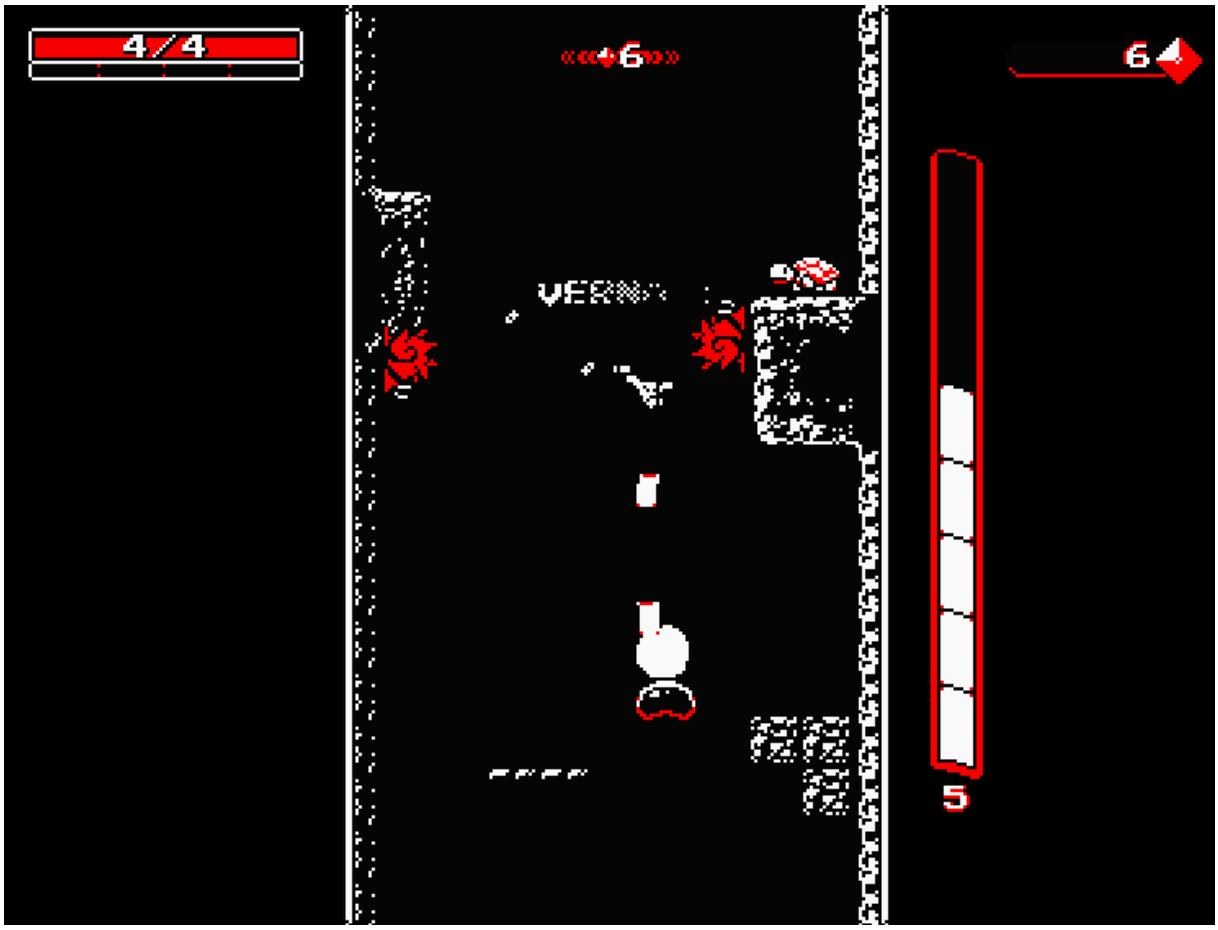


Figura 5 - Uma captura de tela do jogo *Downwell*

Um diagrama possível para este jogo seria o seguinte:

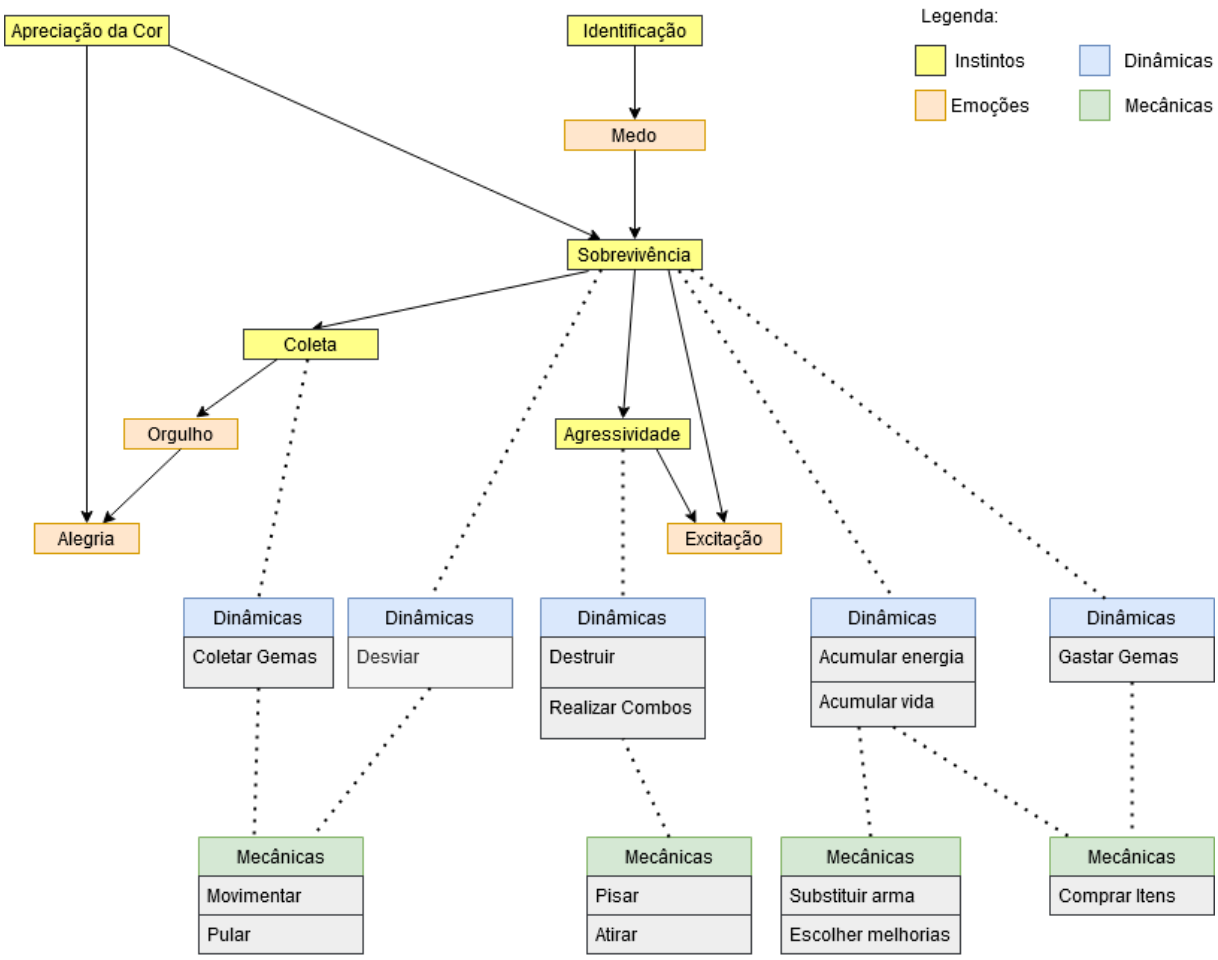


Figura 6 - Diagrama *On the Way to Fun* para Downwell

No topo do diagrama, estão as Estéticas que surgem inicialmente ao se jogar *Downwell*: a Identificação com o personagem que está sendo controlado e a Apreciação de Cor com os visuais minimalistas do jogo, com cores neutras contrastando com o vermelho. Como relatado, este instinto é associado mais à apreciação estética e não à presença de cores em si. O jogo possui apenas três cores possíveis em uma mesma paleta, e o mesmo vale para todas as paletas de cores colecionáveis, estilo visual que irá inspirar o design visual do jogo deste projeto. É interessante notar que o jogo também usa a Apreciação de Cor de maneira a indicar partes perigosas dos inimigos com a cor vermelha, auxiliando a identificar quais elementos se deve evitar em uma partida e colaborando com o instinto de Sobrevivência.

Estas Estéticas principais dão lugar a outras que, eventualmente, são suportadas e realimentadas pelas Dinâmicas do jogo. Através do diagrama, também é possível notar quais conjuntos de Mecânicas geram estas Dinâmicas. Nota-se que as emoções e os instintos resultam em Alegria e Excitação, como provavelmente foi desejado pelo desenvolvedor. As associações entre

Dinâmicas e Estéticas são de certa forma intuitivas. Por exemplo, a Estética de Agressividade é despertada e suportada pelas dinâmicas de Destruir e Realizar Combos. O termo Combo refere-se a capacidade de destruir inimigos sucessivamente sem aterrissar no solo, resultando em benefícios extras para o jogador quando é terminado. Sobrevivência está associada a desviar dos inimigos quando necessário, mas também está associada a Acumular Vida e Energia (que funciona como quantidade máxima de munição) para durar mais tempo na partida.

2.3.2 Diagrama para Tetris

Tetris (ACADEMYSOFT, 1984) é um dos jogos mais conhecidos e populares da história. A primeira versão surgiu em 1984 e seu *design* foi realizado por Alexey Pajitnov na União Soviética. Até os dias atuais, o jogo teve diversas alterações visuais e tecnológicas em diferentes edições, mas a sua jogabilidade permaneceu muito próxima à versão original. Assim, as Estéticas variaram entre as versões, mas as mecânicas e dinâmicas originais estiveram presentes em todas elas. Houve pequenas variações entre as versões, mas podemos destacar as inovações produzidas por *Tetris Effect* (ENHANCE GAMES, 2018). O jogo original consiste em um espaço dividido em uma matriz com dez espaços de largura por vinte de altura, com peças chamadas Tetraminós formadas por 4 blocos conectados, que são escolhidas aleatoriamente e caem do topo da tela, fixando sua posição ao atingir o chão. Cada vez que uma linha de blocos é formada, esta linha é eliminada e o jogador obtém pontos relativos às linhas que foram eliminadas simultaneamente. O jogador pode movimentar horizontalmente e rotacionar estas peças enquanto caem. O objetivo do jogo é durar o mais tempo possível na partida sem que a tela seja preenchida por blocos.

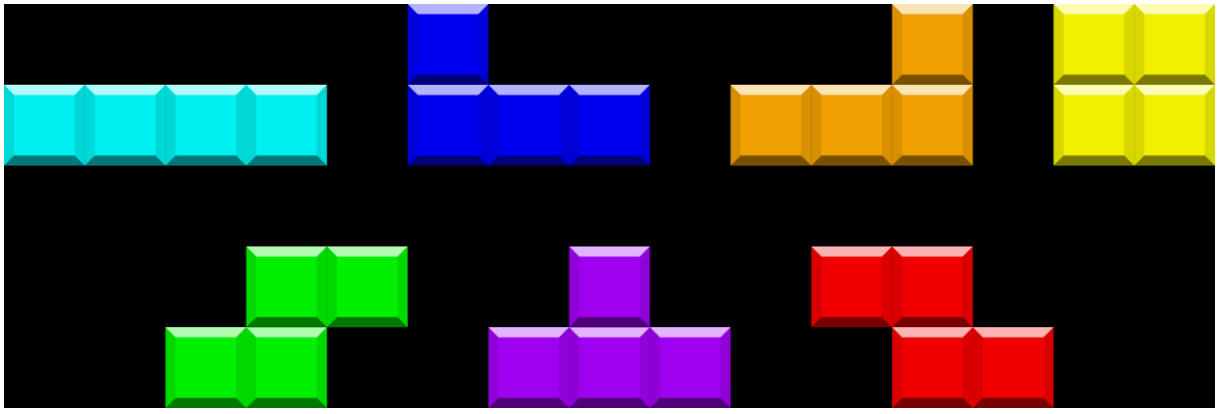


Figura 7 - Os Tetraminós que são gerados em um jogo de Tetris (YERRICK, DAMIAN, 2016)

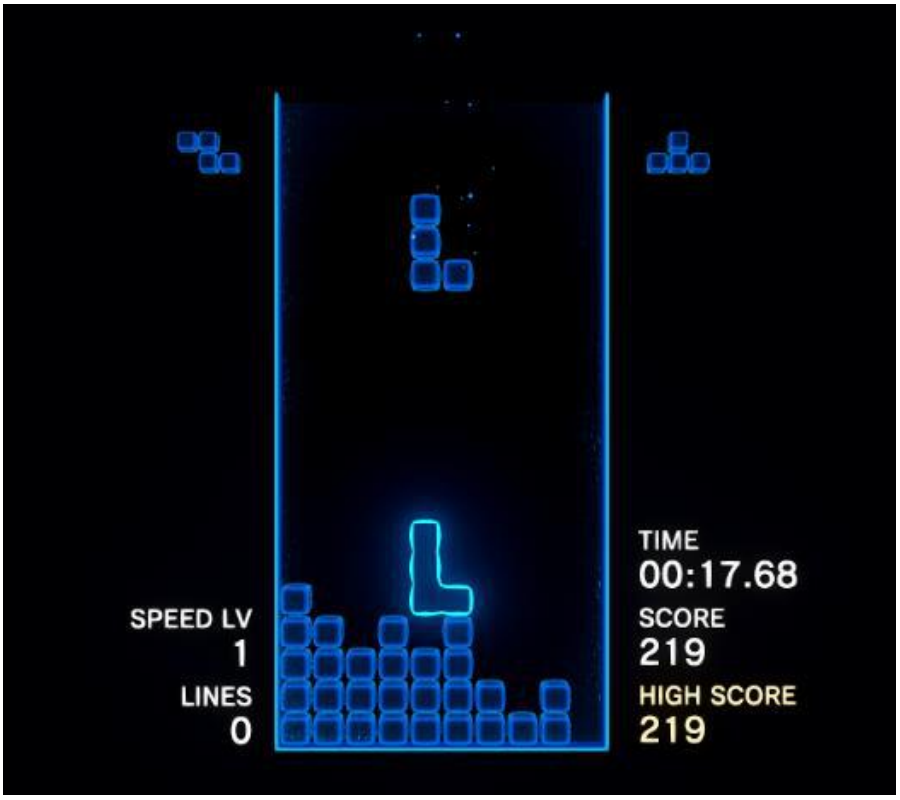


Figura 8 - Área de jogo em *Tetris Effect*

Um diagrama para *Tetris* poderia ser o seguinte:

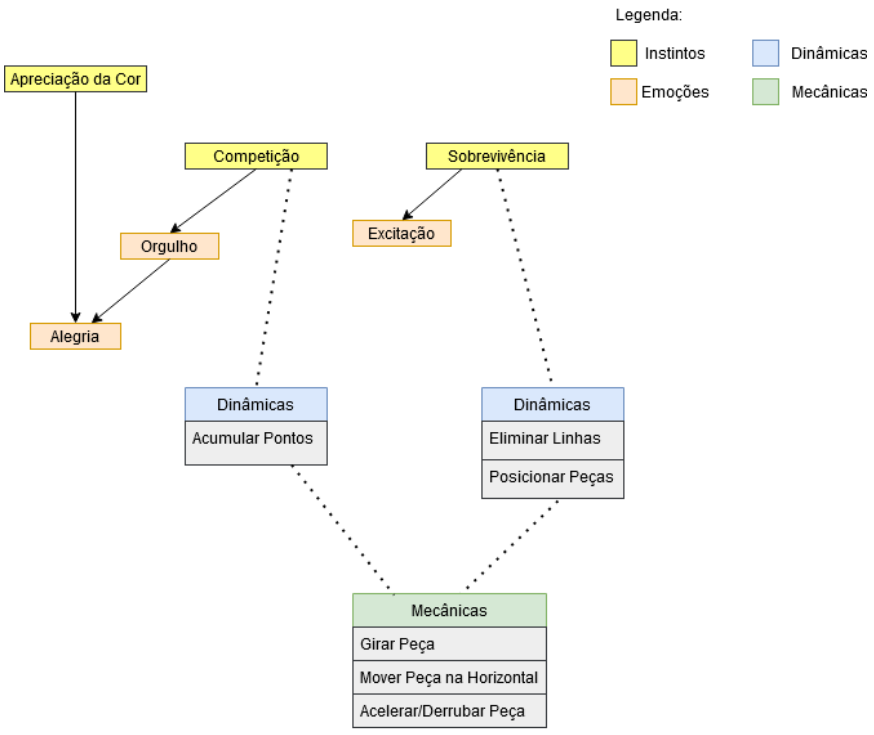


Figura 9 - Diagrama *On the Way to Fun* para *Tetris*

Nota-se uma lacuna no diagrama quanto ao instinto de Identificação. Tetris é um jogo considerado “abstrato”, já que ele não possui um tema específico ou representa alguma atividade realizada por um ator ou personagem facilmente distinguível. Assim, não há como identificar claramente os instintos de primeira ou terceira pessoa, já que não há outras entidades às quais se identificar, como personagens, ou mesmo lugares, presentes durante a partida. É possível relacionar o jogo aos Instintos de Competição, presente no registro de pontos, de forma a induzir a superação de pontuações anteriores, bem como o de Sobrevivência, já que o jogador precisa manter o ambiente de jogo com menos peças para continuar jogando. Entretanto, não é evidente qual instinto despertaria estes, o que pode ser uma falha do modelo ao se trabalhar com jogos abstratos. Para o Instinto de Sobrevivência, é incomum não associar ele a um personagem que precisa de fato sobreviver a um ambiente hostil, o que caracterizaria uma exceção para se adequar ao modelo. A Competição também não está presente em um sentido tradicional, já que não há elementos multijogador estabelecidos formalmente no jogo. Isso tornaria a sugerida “Competição” um metajogo que se resume à obtenção de pontuações mais elevadas, de forma a superar as pontuações que foram obtidas anteriormente pelo próprio jogador ou por outros jogadores, em partidas diferentes. Algumas versões posteriores do jogo possuem diversos modos de multijogador dedicados, além de jogos derivados incluírem modos de jogo semelhantes, como *Puyo Puyo* (COMPILE, 1991) ou *Dr. Mario* (NINTENDO, 1990).

3 DESIGN DE UM JOGO ASSIMÉTRICO

Partindo dos princípios e exemplos do capítulo anterior, pode-se elaborar o design do jogo que é o objetivo deste trabalho, com algumas adaptações, decorrentes do fato de ser um jogo assimétrico. Em relação a estas adaptações, é notável a necessidade de se estabelecer respostas emocionais desejadas não somente para um, mas sim para dois jogadores, já que ambos interagem de maneiras distintas com o jogo. Assim, serão obtidos dois diferentes diagramas *On the Way to Fun*, um para cada componente do jogo. Para um jogo assimétrico, podemos adotar um componente "Principal" e um componente "Secundário", que são definições arbitrariamente escolhidas pelos autores de acordo com o dispositivo no qual cada componente do jogo será jogado. Neste caso, convencionou-se que o componente Principal é jogado em um computador de mesa e o componente Secundário é jogado em um dispositivo móvel.

3.1 PLANEJAMENTO DO JOGO

O planejamento de um jogo costuma começar por suas inspirações. Se tomarmos o início do desenvolvimento e distribuição comercial de jogos digitais como o ano de 1971, quando foi lançado um dos primeiros jogos comerciais, o *Computer Space* (SYZYGY ENGINEERING, 1971), passaram-se, até o momento da escrita deste trabalho, cerca de 50 anos desde o início da indústria de jogos digitais. Embora extremamente recente em comparação com outras indústrias do ramo do entretenimento, a indústria de jogos digitais vem sofrendo mudanças consideráveis nessas décadas de existência. Isso implica que, ao considerar a concepção de um design novo de jogo, muito do que será aplicado nele é derivado de outros jogos e/ou inspirado em outras mídias.

Por exemplo, como inspiração para o jogo deste trabalho, foram utilizados os jogos abordados nas seções 2.3.1 e 2.3.2. A partir dos elementos destes jogos, como mecânicas e dinâmicas, pode-se elaborar um novo jogo. É proveitoso elaborar cada um dos componentes de um jogo assimétrico tendo como inspiração elementos de jogos já lançados.

Servindo como inspiração o *Downwell*, abordado na seção 2.3.1, pode-se imaginar um *Scrolling Shooter* com foco em sobrevivência. *Scrolling Shooter* é a denominação usual para um gênero de jogos em duas dimensões, em que a tela de jogo se move automaticamente, controla-se uma nave espacial ou um player-toke, e sobrevive-se atirando em obstáculos e inimigos ou desviando dos mesmos. Apesar dos elementos de jogo de plataforma, o Downwell tem muito em comum com os jogos do gênero *Scrolling Shooter*. A diferença mais expressiva

é a progressão da tela, que não é automaticamente contínua. No *Downwell*, a progressão depende da posição do jogador, que pode parar em certos lugares para explorar cavernas, adentrar em lojas para comprar melhorias, ou simplesmente esperar para planejar melhor seus próximos movimentos. Com exceção destes fatores, *Downwell* é um bom representante do gênero, já que apresenta uma progressão rápida pelas fases e a destruição ou desvio de obstáculos e inimigos.

Neste trabalho optou-se, para o componente Principal do jogo, pelo gênero *Scrolling Shooter* com progressão automática da tela. Outros aspectos do *Downwell* foram aplicados ao segundo componente do jogo, e serão abordados em seguida.

O Tetris, jogo pertencente ao gênero *Puzzle*, foi outro que inspirou este trabalho, e é abordado na seção 2.3.2. Grande parte da identidade do Tetris é o fato de que o jogador recebe as peças de jogo em ordem aleatória. O Tetris, inclusive, utiliza algoritmos diferentes em cada versão do jogo para a escolha das próximas peças. A partir da mecânica de distribuição aleatória de peças, pode-se imaginar um cenário em que o jogador do componente Secundário recebe itens de forma aleatória para utilizar contra o outro jogador..

Algumas versões de Tetris possuem uma mecânica que permite armazenar uma peça quando ela surge, caso ela não seja efetiva perante ao estado atual do jogo. Assim, ela poderá ser utilizada em outro momento mais oportuno. Ao tentar utilizar novamente essa funcionalidade, se já houver uma peça armazenada, esta será imediatamente utilizada no jogo em substituição à nova peça que se deseja armazenar. O jogo assimétrico deste trabalho possui uma mecânica semelhante no componente Secundário, que permite armazenar uma peça e trocar uma peça atual por uma já armazenada, caso ela não seja efetiva para o estado do jogo observado no componente principal.

Unindo ambos os componentes temos, então, um único jogo que funciona da seguinte forma: o jogador do componente Principal joga, em uma tela grande e visível a ambos os jogadores, um jogo semelhante a um *Scrolling Shooter* clássico, com o objetivo de sobreviver por uma quantidade de tempo estabelecida antes do início da partida. Já o jogador do componente Secundário joga um jogo semelhante a um “tiro ao alvo”, usando itens em uma tela própria não visível ao jogador do componente Principal. Estes obstáculos são então transportados para a tela maior e podem acertar o personagem do jogador Principal, com o intuito de reduzir seus pontos de vida a zero, o que resulta na vitória do jogador Secundário.

Scrolling Shooter, como o nome sugere, é um gênero associado à destruição constante de obstáculos e inimigos que surgem na tela de jogo. Assim, as temáticas mais comuns associadas a esse gênero envolvem veículos aéreos ou personagens poderosos que têm a

capacidade de atirar em obstáculos e inimigos. Isso caracteriza o gênero como sendo bem dinâmico, assim, presume-se que o jogador deve tomar muitas decisões rapidamente. O ritmo do jogo deste trabalho já é bem menos acelerado, uma vez que o objetivo do jogador do componente Principal não é obter pontos ou destruir um inimigo final, mas sim sobreviver aos ataques do segundo jogador. Este aspecto se relaciona melhor com um tema mais calmo do que o habitual para jogos do gênero.

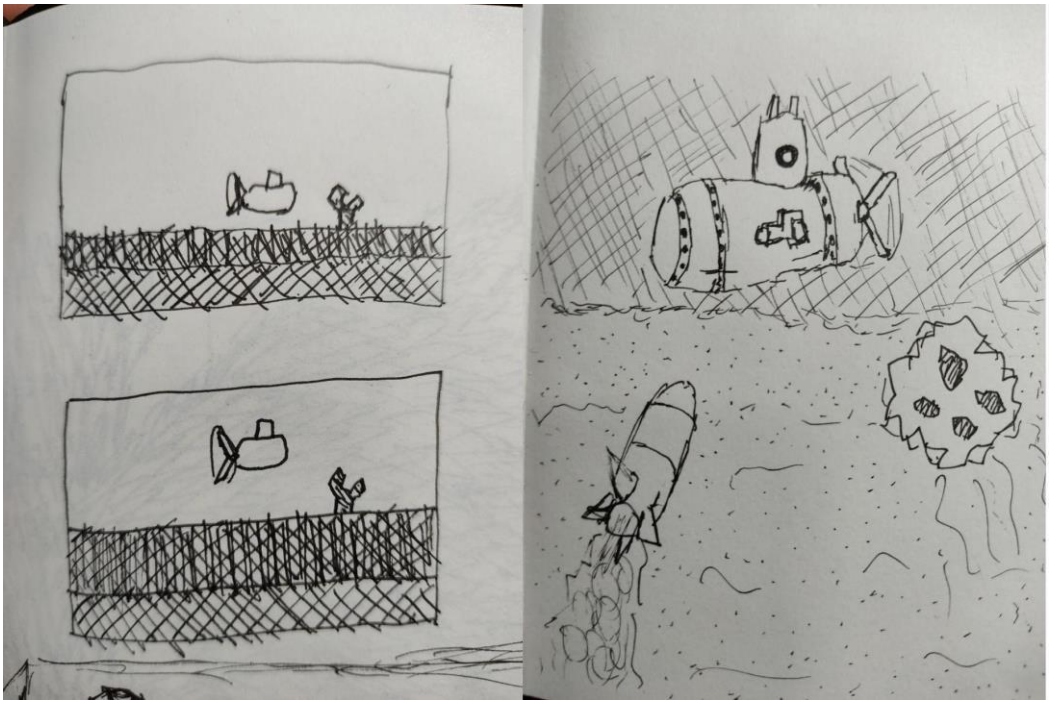


Figura 10 - Esboços iniciais do jogo

3.2 TEMÁTICA DO JOGO

A escolha da temática é um ponto crucial do design do jogo e está fortemente associada à Estética de "Apreciação de Cor" mencionada no capítulo anterior. Por meio desta Estética o jogador irá assimilar o contexto do jogo e o mundo fictício no qual a ação se desenrola. Existem *Scrolling shooters* com as mais diversas temáticas, de jogos de naves espaciais, como *Ikaruga* (TREASURE, 2001), até jogos com bruxas voando em vassouras e realizando magias, como *Cotton: Fantastic Night Dreams* (SUCCESS, 1991). Considerando que o jogo deste trabalho caracteriza-se por um ritmo mais calmo e um objetivo mais defensivo, adotou-se para o componente Principal uma temática de fundo do mar, onde o jogador controla um submarino tentando resistir aos ataques de um inimigo misterioso e buscando um tesouro escondido. O cenário do jogo deve exibir claramente o ambiente em que o jogo se passa, de forma que o jogador associe com facilidade a temática ao jogo. Assim, o cenário escolhido foi o de uma caverna profunda, com estalactites, corais e algas.

3.3 APARÊNCIA DO JOGO

Recentemente, observa-se um ressurgimento dos visuais 1-bit para jogos, ou seja, com gráficos preto e branco em *pixel art* (artes minimalistas em resoluções pequenas), que tanto remetem a jogos clássicos com displays limitados, como o *Pong* (ATARI, 1972), como também incorporam elementos mais recentes do design que são impulsionados, também, pelo avanço da tecnologia, como animações mais expressivas, cenários com mais elementos, personagens bem caracterizados e HUDs mais intuitivas e organizadas. Alguns jogos da Editora Devolver, como o *Minit* (DEVOLVER DIGITAL, 2018) e *Gato Roboto* (DEVOLVER DIGITAL, 2019) são bons exemplos desta tendência. O próprio *Downwell,* embora não seja realmente 1-bit por padrão, apresenta uma paleta limitada de apenas três tons e, inclusive, tem a opção de desbloquear uma paleta 1-bit para substituir a inicial. Estes jogos também apresentam direção sonora que remete a jogos clássicos, como os dos consoles NES e SNES. Embora seja um tipo de design simples de se elaborar, a implementação de *pixel art* 1-Bit deve ser pensada com cuidado, pois a falta de cores pode facilmente fazer com que os elementos na tela sejam confundidos, dificultando o discernimento do jogador do que é um obstáculo, cenário ou personagem, por exemplo. Isso implica em desenhos mais compactos e com movimentos mais suaves, de forma que a tela consiga ser facilmente compreendida por quem está jogando e as reações dos jogadores consigam acompanhar o que acontece no ambiente de jogo. A

simplicidade para confeccionar este estilo foi um fator decisivo para ele ter sido escolhido para o jogo deste trabalho.

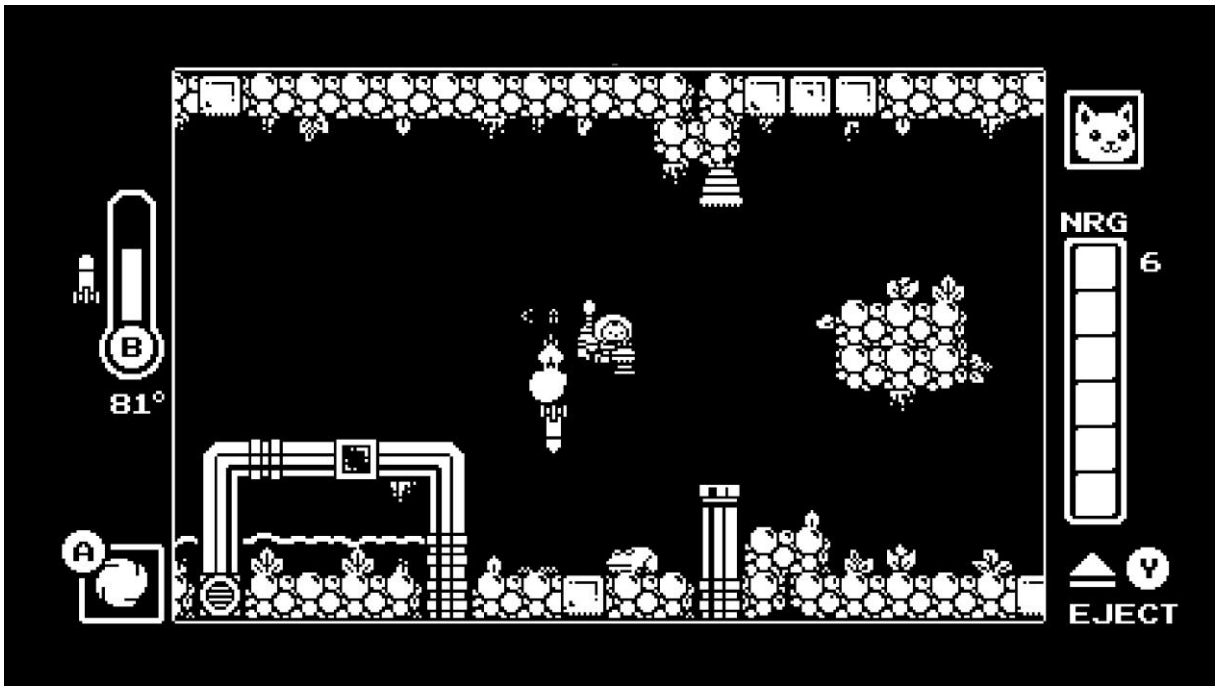


Figura 11 - Captura de tela de *Gato Roboto* (Devolver Digital, 2019)

3.4 ESCOLHAS DE DESIGN

Ao elaborar um jogo é necessário realizar algumas escolhas sobre os elementos que estarão presentes no seu design. Estas decisões são o caminho para se chegar até as mecânicas, dinâmicas e estéticas finais, o que motivou os criadores do jogo a fazer o design de uma forma e não de outra. Essas escolhas podem ser arbitrárias, por exemplo, surgindo das preferências particulares dos autores. Entretanto, em muitos casos, elas se baseiam em aspectos práticos, como facilidade de implementação ou clareza de identificação pelo jogador do que está sendo proposto. Nas seções a seguir, serão explicadas algumas escolhas que foram tomadas ao elaborar o jogo.

Um exemplo de situação que demanda uma escolha seria pensar como o espaço de jogo, ou seja, o ambiente que se observa ao jogar, se movimenta: Poderia-se escolher que a movimentação fosse exclusivamente diagonal, indo de um canto da tela a outro, em vez de horizontal ou vertical, como é de costume na maioria dos jogos. Nada impede que, se bem executada, essa decisão possa ter utilidade genuína na experiência do jogo. No entanto, a

princípio, é fácil observar que ela vai na direção oposta a diversos aspectos que seriam desejáveis para o jogo, como facilidade de entendimento do jogador de quais elementos estão na tela e de onde eles surgem ou, ainda, a intuitividade da movimentação do personagem pelo jogador.

Grande parte destas decisões, porém, podem ter como propósito apenas tornar o jogo mais divertido ou não, o que é bastante subjetivo e atrelado às visões particulares dos criadores do jogo. Nesse ponto, a elaboração dos diagramas *On the Way to Fun* podem ser de grande ajuda na avaliação do design elaborado, avaliando se o jogo que será obtido faz sentido de acordo com o modelo.

3.4.1 Escolhas de Design para o componente Principal

O componente Principal do jogo é constituído principalmente por quatro elementos: o **submarino** controlado pelo jogador; o **cenário**, que é constituído pelos limites definidos pela parte superior e inferior da caverna; o ***background*** e o **HUD**. Além destes elementos, surgem eventualmente os obstáculos gerados pelo próprio cenário ou lançados pelo jogador do componente Secundário.

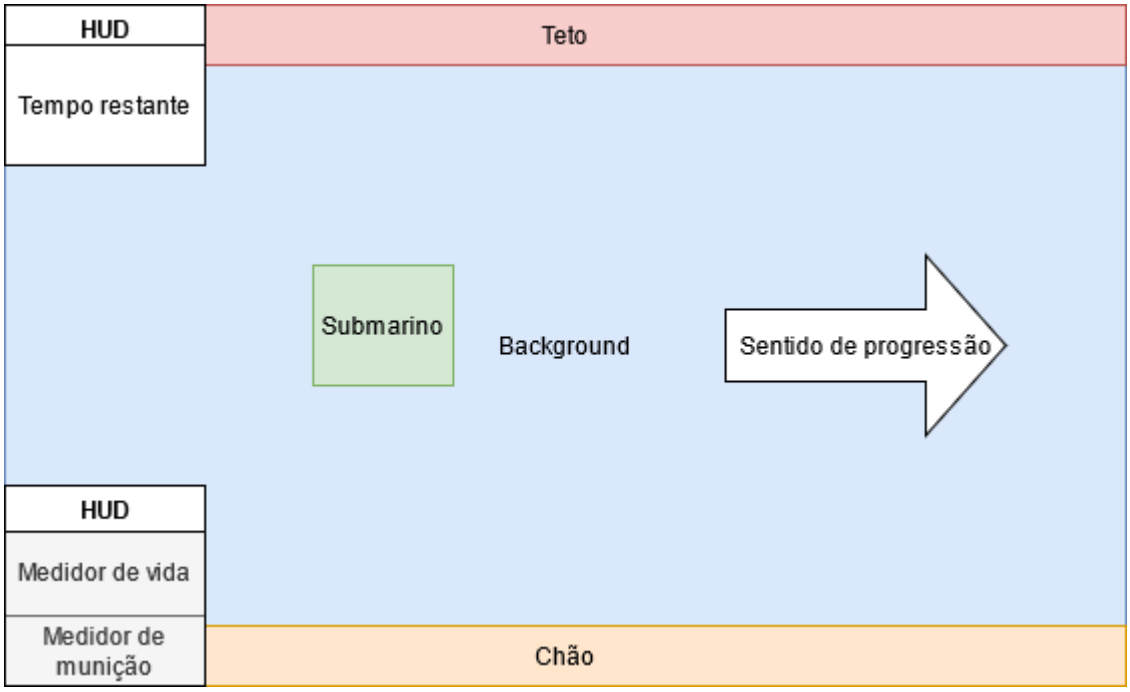


Figura 12 - Esboço da área de jogo da componente Principal

O submarino tem a possibilidade de se mover livremente pelos eixos horizontal, denotado por X, e vertical, denotado por Y, do cenário. Estes eixos respeitam os limites

impostos pelo teto e chão do cenário. Além dos limites verticais, também há limites nas laterais do cenário, que não são explicitados na tela, sendo meramente uma restrição para que o submarino não saia do campo de visão do jogador. Assim, o submarino é impedido de sair da tela de jogo em todas as direções. Conforme explicitado no Capítulo 3, o modelo de jogo segue uma linha de *Scrolling*, em que, além do movimento pelo eixo X pelo próprio usuário, há uma movimentação automática do cenário e do background na horizontal. Esta movimentação resulta na ilusão de que o submarino progride pelo mundo de jogo, quando na verdade não há uma movimentação de fato. Isso ocorre com os *tiles* (unidades do mundo de jogo) que compõem o cenário sendo criados no lado direito da tela, movendo-se e sendo destruídos no lado esquerdo.

O submarino também possui uma arma, cujo sentido é controlado pelo cursor do mouse. Seus projéteis são disparados com o botão esquerdo do mouse. Estes projéteis atuam sobre alguns dos obstáculos do jogo, podendo destruí-los ou não.

3.4.2 Escolhas de Design para o componente Secundário

O componente Secundário possui duas partes: uma **área livre**, onde se pode escolher onde os itens serão arremessados; e um **painel** ao lado direito da tela, que exibe uma lista contendo os próximos itens que serão disponibilizados para serem arremessados.

Figura 13 - Esboço da tela de jogo do componente Secundário

Foram pensados três itens possíveis que o jogador pode receber: uma **pedra**, com tamanho maior que os outro itens, porém mais lenta e destrutível; um **torpedo**, com mais velocidade e que causa mais dano ao submarino, porém com uma área de atuação bem menor; e **bombas**, que explodem alguns momentos após surgir no componente Principal e podem criar explosões em sequência na vertical ou horizontal, dependendo do tipo da bomba.

3.5 DIAGRAMAS *ON THE WAY TO FUN*

A partir do que foi elaborado para o design do jogo, é possível criar diagramas para avaliar se o design faz sentido para obter as respostas emocionais desejadas nos jogadores. Neste trabalho, deseja-se despertar os instintos de Sobrevivência no jogador do componente Principal, Agressividade no jogador do componente Secundário, e Competição em ambos os jogadores. O jogo tem como propósito provocar Alegria e Excitação (emoções) em ambos os jogadores, conforme evidenciado nos diagramas das Figuras 14 e 15.

3.5.1 Diagrama para o componente Principal

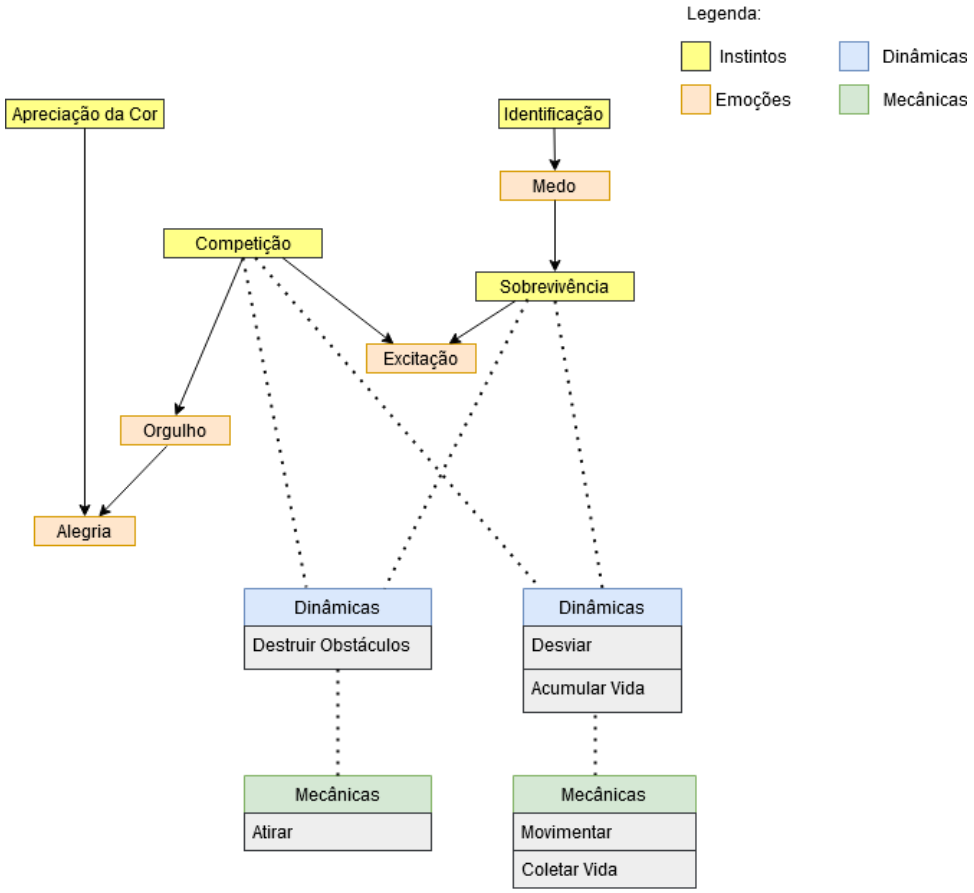


Figura 14 - Diagrama *On the Way to Fun* para o componente Principal

O diagrama de jogo referente ao componente Principal é bem simples, com apenas três mecânicas, com o intuito de desencadear as estéticas propostas de Alegria e Excitação. É possível observar que, como esperado, o fluxo entre Identificação e as mecânicas do diagrama é bem semelhante ao fluxo no diagrama do *Downwell*. Diante do escopo reduzido do projeto do jogo, o diagrama contém uma menor quantidade de mecânicas em relação a *Downwell*. Uma diferença importante entre os diagramas é o instinto de competição. Tendo em vista o objetivo de despertar as emoções Alegria e Excitação, é preciso que o desenvolvimento do jogo implemente as mecânicas escolhidas para que a partir delas as dinâmicas planejadas sejam desencadeadas e, por consequência, as Estéticas desejadas sejam refletidas no jogador.

3.5.2 Diagrama para o componente Secundário

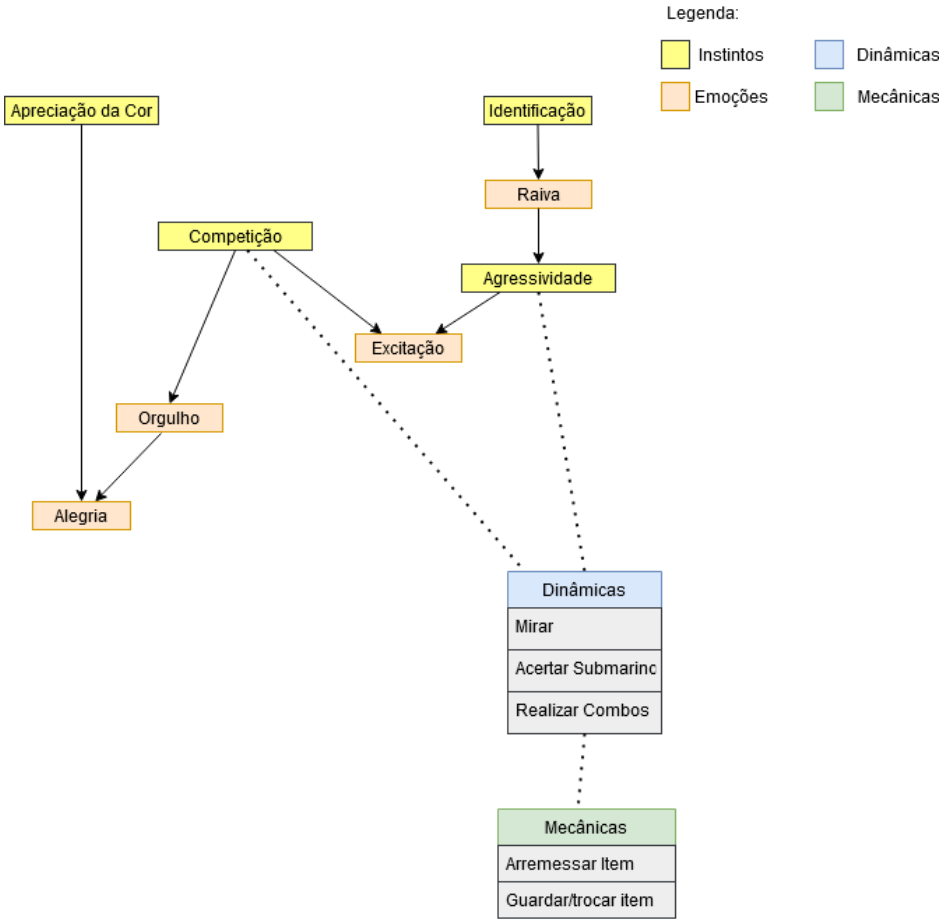


Figura 15 - Diagrama *On the Way to Fun* para o componente Secundário

O componente secundário é representado por um diagrama ainda mais simples, mas não necessariamente menos efetivo. Destaca-se uma quantidade reduzida de Mecânicas presentes, o que implica em uma jogabilidade mais simples para o jogador. Um ponto cuja discussão é importante é o instinto de Identificação. No componente Principal, é evidente que a Identificação surge porque o jogador possui um personagem controlável, que é o Submarino. É visível que o mesmo sofre ataques e tem sua vida reduzida, podendo até mesmo ser destruído. Contudo, esta situação não é observada no componente Secundário: não há um personagem controlado visível, nem um objetivo claramente perceptível para o segundo jogador, assim como sobreviver é para o primeiro. Mesmo que sutil, a identificação está presente e ocorre pelas consequências das ações do jogador do componente Secundário.

Isso ocorre da seguinte forma, assumindo que o jogador não saiba como o jogo funciona. Ao começar o jogo e ver apenas um painel vazio e uma lista de itens, o jogador só tem uma opção clara: testar os comandos do jogo. Fazendo isso, o jogador arremessará os itens pela tela ao

tocá-la. Os itens eventualmente sairão da tela secundária e aparecerão na tela principal. Quando um item acerta o submarino, este sofrerá um dano e a sua vida diminuirá. Tendo apenas uma linha de ação, fica claro que o papel do jogador do componente Secundário é derrotar o jogador do componente Principal. Logo, a Identificação ocorre como uma força em oposição ao jogador do componente Principal. A Dinâmica de acertar o submarino auxilia nesta oposição. Acertos em sequência fazem o nível de combo do jogador do componente Secundário aumentar e, consequentemente, diminuem o tempo para se obter itens. Desta forma, desperta-se o instinto de Agressividade, que colabora para a identificação.

3.5.3 União dos Diagramas

A característica mais evidente de um jogo multijogador é justamente a presença de outros jogadores na mesma partida. Qualquer jogo com mais de um jogador poderia ter os outros jogadores substituídos por Inteligências Artificiais controladas pelo jogo e boa parte de sua jogabilidade se manteria intacta. Em particular, no caso do jogo tema deste trabalho, é possível fazer o jogo ser para um único jogador, fazendo os obstáculos serem arremessados aleatoriamente e automaticamente, no caso de um jogo focado no componente Principal; ou fazendo um submarino que se move automaticamente de acordo com uma estratégia programada, no caso de um jogo focado no componente Secundário. Entretanto, nos jogos que oferecem a opção de jogar tanto contra outros jogadores como contra uma IA, os jogadores frequentemente preferem a primeira opção.
Ao unir os diagramas para ambos os lados do jogo, tem-se o seguinte diagrama:

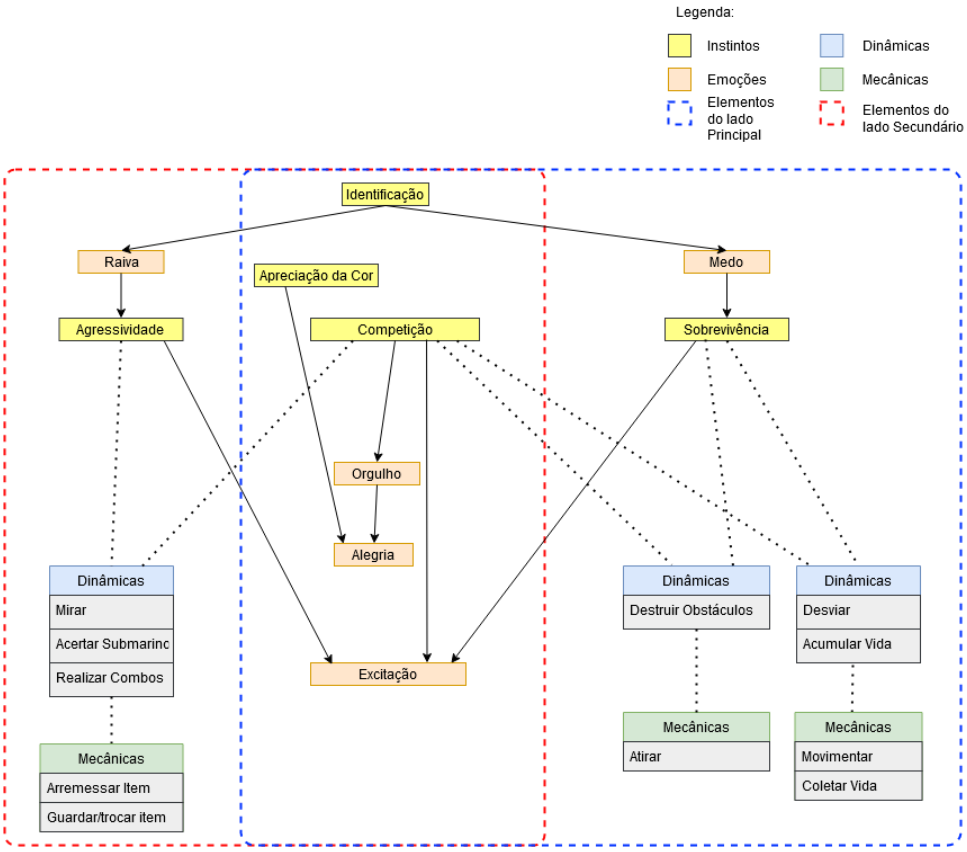


Figura 16 - Diagrama unindo elementos dos componentes Principal e Secundário

Esta disposição permite compreender melhor os elementos comuns à experiência dos jogadores de cada lado. Ela é uma adaptação para dois jogadores dos diagramas *On the Way to Fun*, e cada jogador interage apenas com os elementos do seu lado do diagrama.

É notável que as respostas emocionais de Alegria e Excitação são comuns a ambos os jogadores, como é desejado. Em especial, destaca-se o Instinto de competição como a estética que une os jogadores de um jogo multijogador. Na ausência deste elemento, uma boa parte dos outros elementos do diagrama ainda estaria válida, feitas as devidas adaptações (por exemplo, incluindo elementos para representar uma possível inteligência artificial), porém a experiência seria prejudicada. A interação com outros jogadores, mesmo que mínima, isto é, envolvendo apenas saber que as ações alheias ao jogador são realizadas por outros jogadores, é responsável por uma grande parcela da resposta emocional que é obtida ao jogar. No caso especial de um jogo assimétrico, essa é a Estética que une as partes do jogo, fazendo as interações de um dos jogadores com Mecânicas diferentes terem significado para o outro jogador.

Outro detalhe interessante é que, apesar de não haver outras semelhanças diretas entre os diagramas, é possível observar que ambos herdam propriedades dos diagramas do capítulo anterior. No diagrama de *Downwell (Figura 5)*, por exemplo, a Dinâmica de realizar combos

é encontrada no diagrama do componente Secundário, enquanto as Dinâmicas de desviar e destruir são vistas no diagrama do componente Principal. A partir de dois jogos que inspiraram o design do jogo, foi possível combinar elementos em duas partes (componentes) distintas de um mesmo jogo assimétrico.

4 DESENVOLVIMENTO DO JOGO

Tendo elaborado o design do jogo, o passo seguinte é seu desenvolvimento, ou seja, de fato criar o software que vai, idealmente, corresponder ao design proposto e despertar no jogador as reações planejadas. Para gerar toda a experiência esperada pelos modelos de design definidos pelos diagramas seguindo o framework MDA, também foi necessário definir o processo de desenvolvimento de um protótipo. Protótipos permitem visualizar concretamente o jogo em funcionamento antes de partir para o desenvolvimento do produto final.

4.1 FERRAMENTAS UTILIZADAS

Para o desenvolvimento de um jogo, é importante escolher os softwares e ferramentas mais apropriados para o resultado que se quer obter. Em geral, é conveniente o uso de alguma *Game Engine*, ou Motor de Jogos, além de softwares de edição de imagens e sons para criar o conteúdo que é exibido para o jogador. Os critérios adotados para escolher as ferramentas foram o quanto cada uma atenderia aos objetivos estabelecidos e o custo de aquisição ou utilização das mesmas.

4.1.1 Escolha da Game Engine

Após um levantamento das ferramentas que poderiam ser utilizadas no desenvolvimento do protótipo, optou-se pela Unity *(Unity Technologies)* como *engine* usada para o desenvolvimento. Em suma, uma *Game Engine* consiste em um software que possui um conjunto de ferramentas com a finalidade de facilitar o desenvolvimento de um jogo. Geralmente, esses softwares possuem desde recursos para criação de efeitos gráficos até opções para acrescentar física aos objetos, trilhas sonoras, entre outros recursos. Através da *Game Engine*, os recursos visuais, musicais e de design são estruturados e transformados em um software que poderá então ser distribuído. Todo desenvolvimento dentro da Unity é feito através de Scripts que compõem um objeto com função pré-definida dentro do jogo, seja ele um personagem jogável, uma câmera, um elemento de colisão ou um simples objeto que guarda a lógica para a geração do cenário em que ocorre o jogo em questão. Todos os scripts são desenvolvidos utilizando a linguagem de programação C#, que recebe suporte da Microsoft. A Unity possui uma licença de uso gratuito para projetos sem fins comerciais ou com baixo retorno financeiro, sendo assim uma opção viável para o projeto. Outras *Game Engines*

gratuitas também se mostraram boas candidatas, como a Godot e a Unreal *(Epic Games)*. Entretanto, optou-se pela Unity pela familiaridade dos autores com o ambiente de desenvolvimento e com a linguagem C#.

4.1.2 Escolha de Softwares Auxiliares

Uma *Game Engine* precisa de recursos para exibir para o jogador. Estes recursos incluem, por exemplo, imagens, ou *sprites*, animações e áudio, e precisam ser confeccionados em softwares de terceiros e então importados para uso na *Engine*.

Como definido na seção 3.3, o jogo tem como identidade visual o estilo 1-Bit de *pixel art*. Embora seja possível elaborar este tipo de arte na maioria de softwares de edição de imagens, a escolha para esse projeto foi o software Aseprite (IGARA STUDIO, 2014), programa com foco na produção de *pixel art*, com diversas funcionalidades que tornam sua confecção mais simples. Apesar do software ser pago, é de baixo custo e seu uso foi recomendável.

Para os sons do jogo, foi utilizado o site beepbox.co (NESKY, JOHN, 2019), que permite criar áudios simples que podem ser baixados e importados em qualquer *Game Engine*. O site é de fácil utilização e permite usar padrões de som que se enquadram na estética de um jogo retrô, além de ser gratuito para uso.

4.2 ORGANIZAÇÃO DO PROJETO

A organização do projeto envolveu a criação de uma estrutura de diretórios para organizar os arquivos criados e utilizados, bem como o controle de versão do código fonte, que permite acompanhar mais facilmente a implementação de diferentes funcionalidades para o jogo.

4.2.1 Estrutura de Diretórios

Todos os recursos utilizados no jogo estão organizados dentro do diretório *Assets*. Dentro dela destacam-se os seguintes diretórios: *Scenes*, onde ficam as cenas do jogo; *Sprites*, onde se situam todos os elementos visuais do jogo, incluindo animações, imagens dos personagens e elementos da interface do usuário; *Áudio*, que contém os efeitos sonoros e músicas do jogo; *Resources*, onde ficam os *prefabs*, objetos pré-configurados que podem ser

replicados através de cópias que são instanciadas durante a execução e, por fim, *Scripts*, que contém todos os arquivos C# com os códigos criados ou importados para o projeto. Este último diretório, por sua vez, está dividida em *ObjectBehaviours*, que possui os scripts de comportamento dos objetos do jogo, como obstáculos e o personagem; *Communication*, que inclui todos os arquivos necessários à comunicação; e *UI*, onde estão os scripts relacionados ao comportamento da interface visual do jogo.

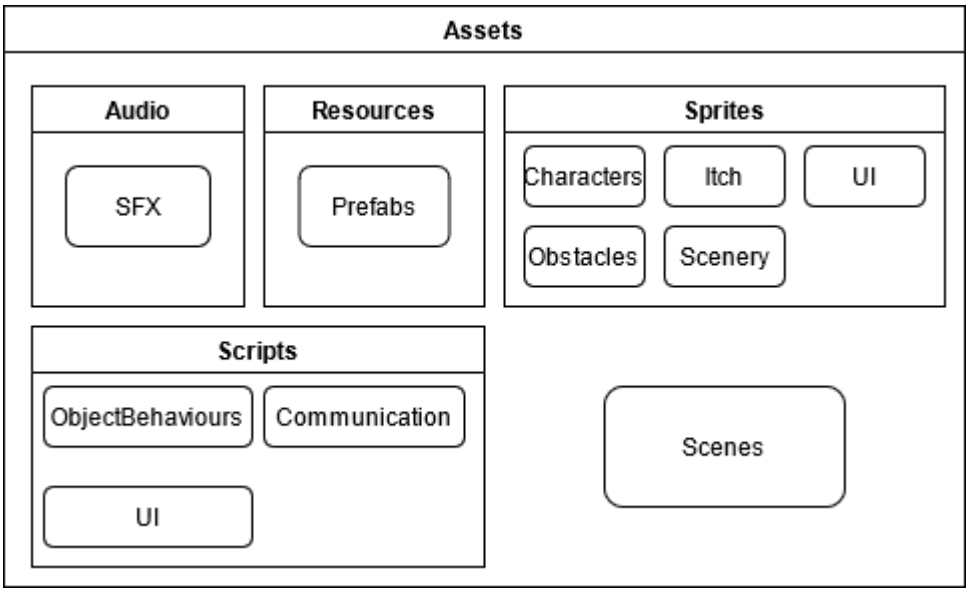


Figura 17 - Estrutura geral de diretórios do projeto

4.2.2 Versionamento de Código e Recursos

Para manter o controle das versões do projeto e de quais funcionalidades foram implementadas em cada versão é importante usar um sistema de versionamento. A ferramenta escolhida para realizar o versionamento do projeto foi o Git (TORVALDS, LINUS, 2005). No Git, cada versão é identificada com um código hash, que então pode ser usado para se alternar entre as versões. Os arquivos versionados devem ser armazenados em um repositório ao qual os desenvolvedores tenham acesso. Este repositório deve servir de backup e permitir a colaboração entre os participantes do projeto. O serviço escolhido para este fim foi o Github (https://github.com/). Cada nova funcionalidade do projeto foi implementada em uma branch própria. Uma *branch* funciona como um ambiente de desenvolvimento separado do original, para que possam ser feitas alterações sem afetar o código original. Posteriormente, ao fim das atualizações, é criado um *pull request*, isto é, um pedido para incorporar as mudanças à *branch* original. O *pull request* permite visualizar quais arquivos foram alterados, criados ou excluídos

desde que a *branch* foi criada, comparando-os com os arquivos originais. Depois de criado, pode-se revisar estas mudanças e, caso não haja problemas, o *pull reques*t é aceito e os novos arquivos são incorporados à *branch* destino. A utilização do *pull request* e a criação de uma nova *branch* a cada alteração são de suma importância para um versionamento organizado e para que mais de um programador consiga trabalhar no mesmo projeto sem conflitos entre as versões do código. O repositório do projeto pode ser acessado no seguinte endereço: https://github.com/lucascmgs/Tcc

4.3 ELEMENTOS DO JOGO

Aqui destaca-se a implementação de alguns elementos importantes para o jogo. Os **controláveis**, isto é, os objetos que os jogadores controlam; os **obstáculos**, que são os objetos com os quais o jogador do lado Principal interage e o **cenário**, que é meramente ilustrativo, servindo apenas para efeito visual e não apresentando de fato interatividade com o personagem ou com outros objetos.

4.3.1 Controláveis

Os elementos controláveis do jogo são o submarino, no componente Principal; e a área de arremesso de itens, ou lançador, no componente Secundário.

4.3.1.1 Controles no componente Principal

O submarino possui movimento na horizontal, controlado com as teclas A (esquerda), D (direita) e na vertical, com as teclas W (cima) e S (baixo). Este movimento é realizado trocando a velocidade do submarino pelo seu script de movimento de acordo com qual tecla está pressionada, zerando-a caso todas as teclas sejam soltas. É possível combinar teclas para realizar movimentos na diagonal, como W e D para mover o submarino na diagonal superior.

A arma do submarino aponta sempre para a posição do mouse, não importa a posição do submarino. Os disparos são feitos com o botão esquerdo do mouse, atirando um projétil que percorre um caminho linear em direção à posição clicada, partindo do submarino. A arma possui quatro munições, e cada disparo remove uma do total. Cada munição é reposta automaticamente assim que é destruída, o que acontece ao colidir com um objeto ou ao sair da tela. Isso significa

que, em um dado momento, é possível haver no máximo quatro projéteis na tela de jogo simultaneamente, tendo que aguardar para efetuar mais disparos.

4.3.1.2 Controles no componente Secundário

O jogador no componente secundário tem acesso à lista de itens que serão usados em seguida por ele. Essa lista fica disponível do lado direito da tela. A todo momento, o jogador possui duas opções: arremessar o item no início da lista ou guardá-lo, tocando no botão HOLD. Toda vez que um item da lista é removido sem ser substituído, seja porque foi arremessado ou guardado, o item seguinte da lista passa a ser o próximo e um novo item é gerado e adicionado ao fim da lista.

Há duas possibilidades ao guardar um item. Caso seja o primeiro item a ser guardado na partida, ele simplesmente é removido da fila e armazenado no espaço apropriado. Caso contrário, o item é removido da fila, o item que foi guardado anteriormente torna-se o próximo da fila e o item removido passa a ser armazenado como item guardado. Os itens não podem ser guardados sucessivamente, isto é, depois de guardar um item, o próximo item deve ser necessariamente arremessado para que se possa guardar um item seguinte novamente.

Um item é removido da lista após ser arremessado. Para evitar que os itens sejam arremessados de forma pouco criteriosa, há um tempo que deve ser aguardado entre um arremesso e outro. Tem-se um arremesso de sucesso quando o item acerta o submarino. Quando sucessivos arremessos resultam em sucesso em um curto período de tempo, o medidor de combo se eleva. A cada nível de combo, o jogador Secundário recebe um bônus que torna o tempo entre arremessos mais curto. Os itens são arremessados de duas formas. Tocando rapidamente na tela, o item será arremessado para o lado esquerdo do ponto tocado, saindo da tela e indo para o ambiente do componente Principal. Outra maneira para arremessar o item é arrastar o dedo pela tela, assim, ao soltá-lo, o item será arremessado na direção correspondente à linha formada entre as posições de início e fim do toque. Isso permite, por exemplo, arremessar o item em uma diagonal, fazendo com que ele apareça no componente Principal também movendo-se na diagonal.

4.3.2 Obstáculos

Existem dois tipos de obstáculos. Há os obstáculos gerados junto ao cenário do jogo, que são as estalactites presas ao teto da caverna do mundo do jogo, e os itens arremessados pelo jogador no componente Secundário. Os obstáculos causam danos ao submarino, seja ao entrar em contato com o mesmo ou ao explodir, no caso das bombas. Após receber um dano, o

submarino tem um curto período de invencibilidade para evitar que continue recebendo danos continuamente e assim o jogador consiga se recuperar.

As estalactites são criadas junto ao cenário do jogo, cuja geração será detalhada em seguida. Elas são anexas ao teto e, com alguma chance aleatória, podem se desprender do mesmo após algum tempo, caindo até o chão e prendendo-se ao aterrissar. Enquanto estiverem na tela, elas podem danificar o submarino caso entrem em contato com ele. Elas também podem ser destruídas pelos disparos do jogador do lado Principal, caso consiga atingi-las.

Os itens arremessados a partir do componente Secundário tornam-se obstáculos ao serem transportados para o componente Principal. Eles, então, se comportam de maneira diferente de acordo com cada item, como descrito na Seção 3.4.2.

As pedras e o torpedo funcionam de modo mais simples. Ambos apenas se movem pela tela até saírem do campo de visão ou, no caso das pedras, serem atingidas por um projétil do jogador. As bombas, após surgirem, iniciam uma contagem de tempo regressiva pré-definida e explodem ao fim da contagem, gerando uma fileira horizontal ou vertical de explosões em sequência. Essas explosões são feitas utilizando uma co-rotina (UNITY TECHNOLOGIES, 2021), de forma que, antes de ser destruída, a bomba chama o método de criação de explosões. A cada explosão criada, a co-rotina espera pelo tempo de explosão correspondente à animação da explosão desaparecendo. Quando retoma o controle, as posições das próximas explosões são calculadas e estas são criadas, esperando-se novamente pelo tempo de explosão e recomeçando o procedimento.

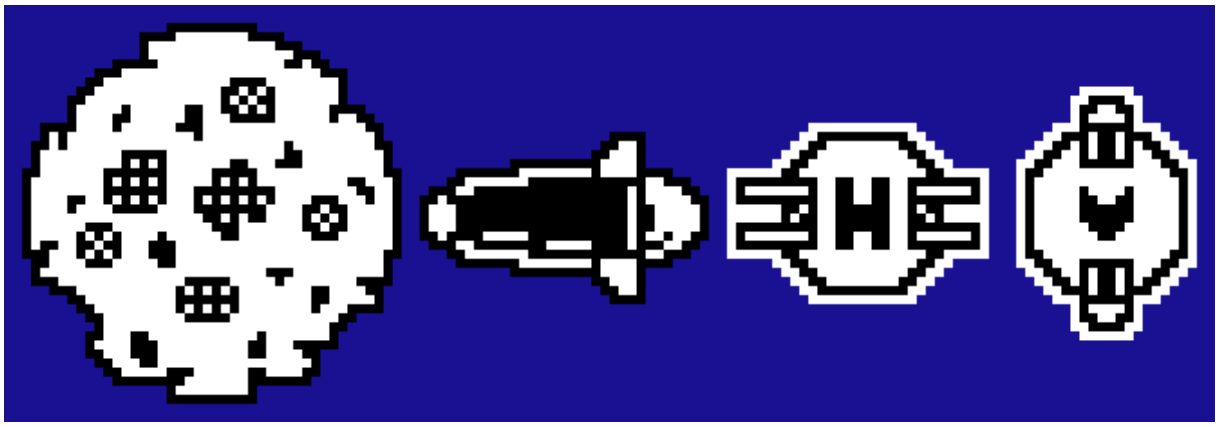


Figura 18 - Sprites para os obstáculos do jogo. Da esquerda para a direita: Pedra, Torpedo, Bomba Horizontal e Bomba Vertical

4.3.3 Cenário

O cenário do jogo, como explicitado na Seção 3.2, é uma caverna na qual o submarino se encontra, tendo que chegar ao final da mesma para atingir seu objetivo. A caverna é constituída por 3 camadas. A camada frontal é onde estão presentes o teto e o chão, que são os mesmos que limitam o movimento do jogador, apresentados na seção 3.4.1. Esta camada delimita a área de jogo em si, onde o submarino e outros objetos se encontram. Já as camadas do meio e de fundo são apenas visuais, servindo para dar efeito de profundidade ao jogo. A cada camada, os *sprites* são mais escuros, dando a impressão que estão mais distantes.

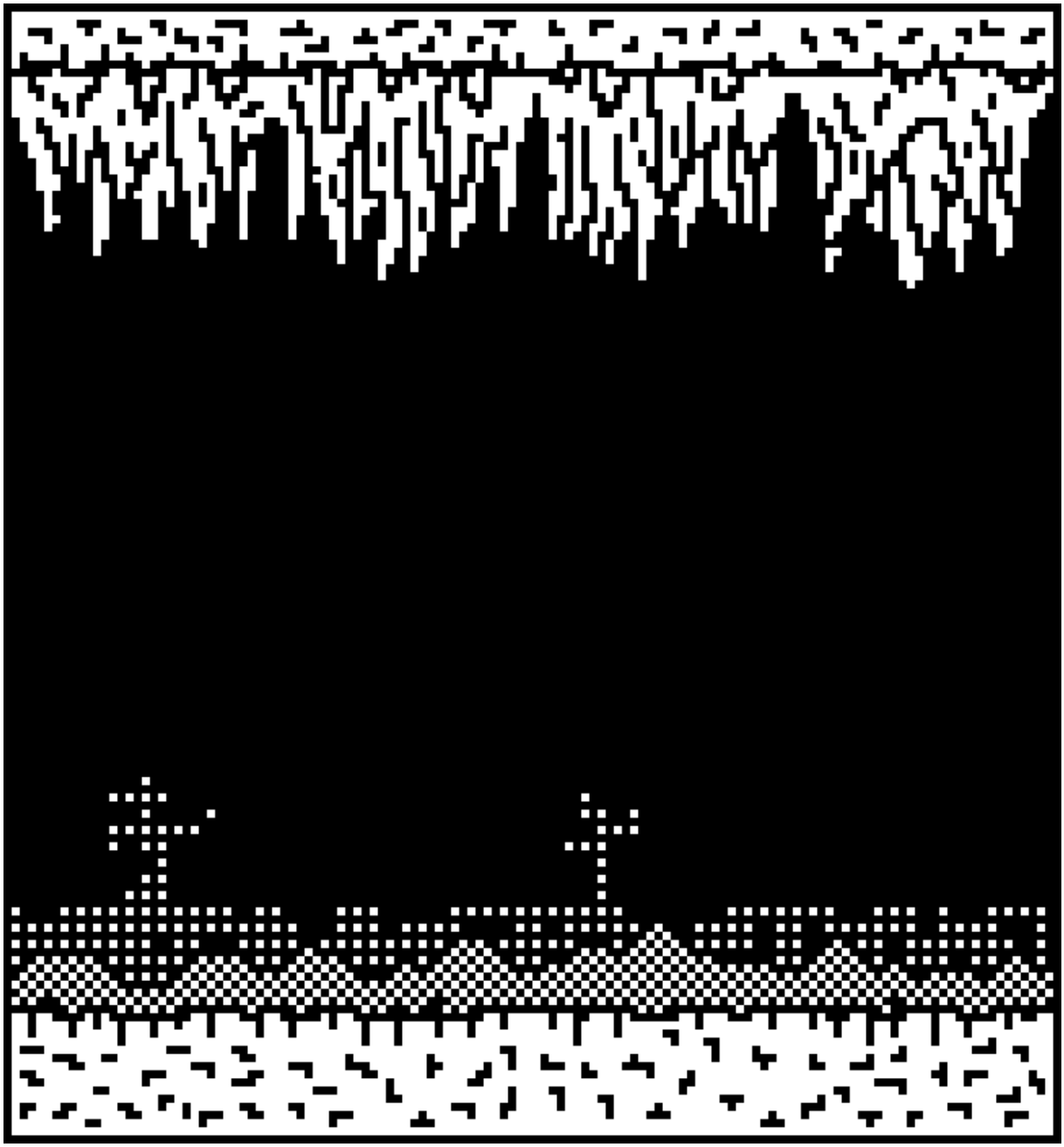


Figura 19 - Exemplo com os *sprites* utilizados para o cenário

Os *sprites* das camadas do cenário se movimentam da direita para a esquerda em velocidades diferentes. Os *sprites* da camada frontal são mais rápidos que os da camada do meio e estes que os da camada de fundo, contribuindo com a impressão de profundidade. Este é um efeito de paralaxe, presente em muitos jogos 2D com rolagem lateral da tela.

4.3.3.1 Geração do Cenário

A geração do cenário é feita de forma procedural: são utilizados quatro *sprites* para definir o teto da caverna e outros quatro *sprites* para o solo. O controle dos *sprites* gerados é feito por um objeto fixo. A cada iteração, o objeto gera um *sprite* aleatório entre os quatro possíveis para o teto e um para o solo. Assim que criados, os *sprites* passam a se movimentar e, em seguida, adentram o campo de visão do componente Principal. Ao aparecerem na tela, inicia-se uma nova iteração do gerador, que gera outros dois *sprites* ao lado dos anteriores, dando a impressão de uma única superfície contínua se movimentando. Este processo é repetido durante todo o jogo, e quando o *sprite* gerado chega ao fim da tela, tem-se o teto e o solo formados por completo. Quando um *sprite* sai por completo da tela, o objeto correspondente é destruído para liberar a memória utilizada por ele. O mesmo modelo é utilizado para a geração das camadas de meio e fundo, porém com uma velocidade de geração um pouco menor, visto que os *sprites* de cada camada possuem sua própria velocidade de movimentação, para assim garantir o efeito de *parallax*.

Para o efeito de scrolling ocorrer, esses objetos são gerados a uma certa velocidade inicial e, conforme o tempo do jogo passa, essa velocidade aumenta de acordo com uma aceleração pré-estabelecida, para assim tornar o ritmo do jogo mais dinâmico. É importante frisar que as estalactites que funcionam como obstáculos também são geradas nesse passo, sendo que a velocidade que estas adquirem ao se desprender do teto acompanha a velocidade com que os *sprites* estão se movendo no momento.

4.4 COMUNICAÇÃO ENTRE DISPOSITIVOS

Para que o jogo tenha duas instâncias que interagem entre si, foi necessário elaborar um esquema de comunicação entre essas duas, que ficam conectadas por uma rede local. Por exemplo, se um objeto sai da instância do componente Secundário, a ocorrência desse evento deve ser transmitida para o componente Principal para que o objeto seja recriado neste lado e possa de fato interagir com o Submarino.

A arquitetura para realizar a comunicação é simples: Ela tem seu início em uma instância no componente Principal, que abre a conexão para que uma instância do componente Secundário possa se conectar. Uma vez conectadas, ambas as instâncias podem comunicar eventos uma à outra.

Com a arquitetura definida, é necessário realizar sua implementação, ou seja, decidir que tecnologia irá possibilitar a confecção dos scripts de comunicação. Existem diversas bibliotecas e serviços de comunicação disponíveis para C# e Unity, com níveis de abstração diversos. Pode-se optar, por exemplo, por um serviço que faça a gestão da comunicação em alto nível, com objetos automaticamente criados e sincronizados entre todas as instâncias do jogo; assim como é possível decidir utilizar uma biblioteca que faça a comunicação em um nível de abstração mais baixo, permitindo apenas a troca de mensagens serializadas em bytes e deixando a cargo dos desenvolvedores a sincronização entre as instâncias. Ambas as possibilidades foram exploradas neste trabalho. É importante notar que a própria Unity disponibiliza serviços próprios de comunicação para o desenvolvimento, porém os mesmos estavam depreciados nas versões mais recentes sem outros serviços disponíveis como substitutos na data do desenvolvimento deste trabalho. Isso levou os autores a procurarem outras alternativas, que serão abordadas abaixo.

4.4.1 Prototipagem com Photon

Para elaborar um protótipo para o Jogo foi utilizado o PUN 2 (*Photon Unity Networking 2) (Exit Games, 2018),* que pode ser obtido no site do Photon. Ele consiste em um pacote para Unity com diversos serviços que facilitam o desenvolvimento de jogos multijogador. Estes serviços permitem a comunicação em alto nível com objetos sincronizados automaticamente entre as instâncias de jogo. Isso significa que apenas é necessário adicionar um script a cada objeto que deve ser sincronizado e o próprio Photon se encarrega de criar os objetos em ambas as instâncias do jogo simultaneamente, por exemplo.

A comunicação neste serviço é feita por meio do acesso a um servidor proprietário. O time de desenvolvimento precisa criar uma conta de acesso ao Photon e obter um número de identificação, chamado de Id, para a aplicação. Este Id é preenchido na configuração do pacote dentro da Unity para vincular o projeto à conta criada. Então, cada instância da aplicação precisa se conectar através da internet a um servidor do Photon para poder se comunicar com outra instância. O modelo de negócios do Photon oferece planos de acordo com a utilização do

serviço. O plano básico e gratuito, entre outras restrições, permite apenas até 20 jogadores acessando o jogo simultaneamente.

Assim que a aplicação se conecta ao servidor do Photon é criada uma Sala, uma divisão lógica do serviço, na qual outras instâncias podem entrar. Clientes conectados na mesma sala são considerados como participantes do mesmo jogo. A biblioteca do Photon possui diversos *Callbacks*, ou seja, funções que são chamadas automaticamente pelo sistema quando certos eventos ocorrem. Um dos *Callbacks* é acionado quando um jogador se conecta a uma sala. Assim, pode-se definir que, ao executar este *Callback*, a instância da aplicação consulta o servidor e verifica se a sala possui exatamente dois jogadores, que é o número de jogadores do jogo deste trabalho. Caso afirmativo, carrega-se a Cena correspondente para dar início ao jogo, dependendo de qual componente do jogo está sendo executado.

O esquema de comunicação do jogo é bem simples, além de contar apenas com dois jogadores interagindo entre si no mesmo ambiente. Assim, não se justificou o uso do Photon para a versão final do projeto, tanto por conta da eventual necessidade de contratar um plano do serviço que permitisse mais jogadores, quanto pela natureza do Jogo de ser jogado por duas pessoas em um mesmo local, ou seja, não sendo viável requisitar o acesso à internet para que ambas possam jogar. Apesar destas questões, o uso do Photon foi útil para a elaboração de um protótipo: suas funções de comunicação se mostraram intuitivas e facilitaram o início do desenvolvimento.

4.4.2 Refatoração do Projeto com a Biblioteca Telepathy

A escolha final para implementar a comunicação do projeto foi a biblioteca *Telepathy* (VIS2K, 2018), que pode ser obtida no repositório do projeto no Github. Ela é desenvolvida em C# e é focada em comunicação a um nível mais baixo de abstração que o Photon, permitindo a conexão entre um servidor e um cliente, através do protocolo TCP, que então podem trocar mensagens entre eles. Esta biblioteca foi escolhida por ser compatível com Unity, além de fornecer funções simples e baixa quantidade de código.

4.4.2.1 Funcionamento da Biblioteca Telepathy

Nesta biblioteca, a comunicação começa com o servidor sendo iniciado em uma porta pré-determinada. Clientes podem, então, se conectar utilizando o endereço IP do Servidor, bem como a porta escolhida previamente, de forma semelhante à utilização de *sockets*.

Uma vez conectados, ambos os lados podem enviar mensagens com o tipo *Telepathy.Message*. As propriedades mais relevantes destas mensagens são *EventType*, que descreve o tipo da mensagem como, por exemplo, sendo de início de conexão ou de envio de dados; além de *data*, que são os dados da mensagem serializados em bytes. Estes dados devem ser convertidos para um tipo comum a ambos os lados da conexão, em geral *Strings*, que permitem uma boa flexibilidade de manipulação e de transmissão de informação. Em todos os tipos de mensagem, os dados devem ser convertidos em bytes para envio e reconvertidos para esse tipo após o recebimento para que sejam usados.

4.4.2.2 Aplicação da Biblioteca Telepathy ao Projeto

Foi criada uma classe base chamada *ManagerBase*, responsável por funções comuns a ambos os lados do jogo, bem como as classes *ServerManager*, para o Servidor (lado Principal), e *ClientManager*, para o Cliente (lado Secundário), que herdam da classe base.

Como os eventos definidos para ambos os lados são simples, a comunicação em alto nível ocorre com a transmissão de qual evento deve ter seu resultado sincronizado entre ambos os lados, bem como os dados da instância deste evento.

Por exemplo, para o evento que ocorre quando um obstáculo sai da tela no componente Secundário e deve ser transportado para o componente Principal, o fluxo do jogo se dá desta forma:

1. O objeto em questão é destruído no componente lado Secundário;
2. O componente Secundário monta uma mensagem com o tipo String contendo o tipo de evento, no caso "*Spawn*", seguido pelo índice do objeto no vetor de Objetos que podem ser transmitidos, bem como sua altura em relação à tela e sua velocidade nos eixos horizontal e vertical. Estes dados são separados por ';'. Um exemplo de mensagem, para o caso de um obstáculo "Armadilha" registrado com o índice 4, que saiu da tela na metade de sua altura e com vetor velocidade (-3, -2) poderia ser "Spawn;4;0.5;-3;-2";
3. O componente Secundário transforma a mensagem em bytes e a envia;
4. O componente Principal recebe a mensagem em bytes e realiza a serialização para o tipo String;
5. O componente Principal divide a string com o separador ";" e armazena cada elemento separado em um vetor. Avalia-se então o que deve ser realizado com os valores da mensagem. O primeiro campo da mensagem indica sempre o tipo

de evento. Lendo o primeiro elemento do vetor, sabe-se que é um evento de spawn, logo os elementos seguintes são referentes a criação de um objeto;

6. O componente Principal executa a ação correspondente à mensagem

Há outros eventos definidos para carregar cenas de vitória e derrota para o componente Principal, bem como para os casos de conexão e desconexão, quando se deve carregar as cenas do jogo e as do menu inicial, respectivamente.

4.5 RESULTADOS DO DESENVOLVIMENTO

Após o fim do desenvolvimento, foram obtidas versões jogáveis correspondentes aos componentes Principal e Secundário. As versões são funcionais, correspondem aos elementos propostos na etapa de design do jogo e são jogáveis do início ao fim de uma partida. Com estas versões obtidas, o jogo pode ser publicado e testado por outros jogadores.

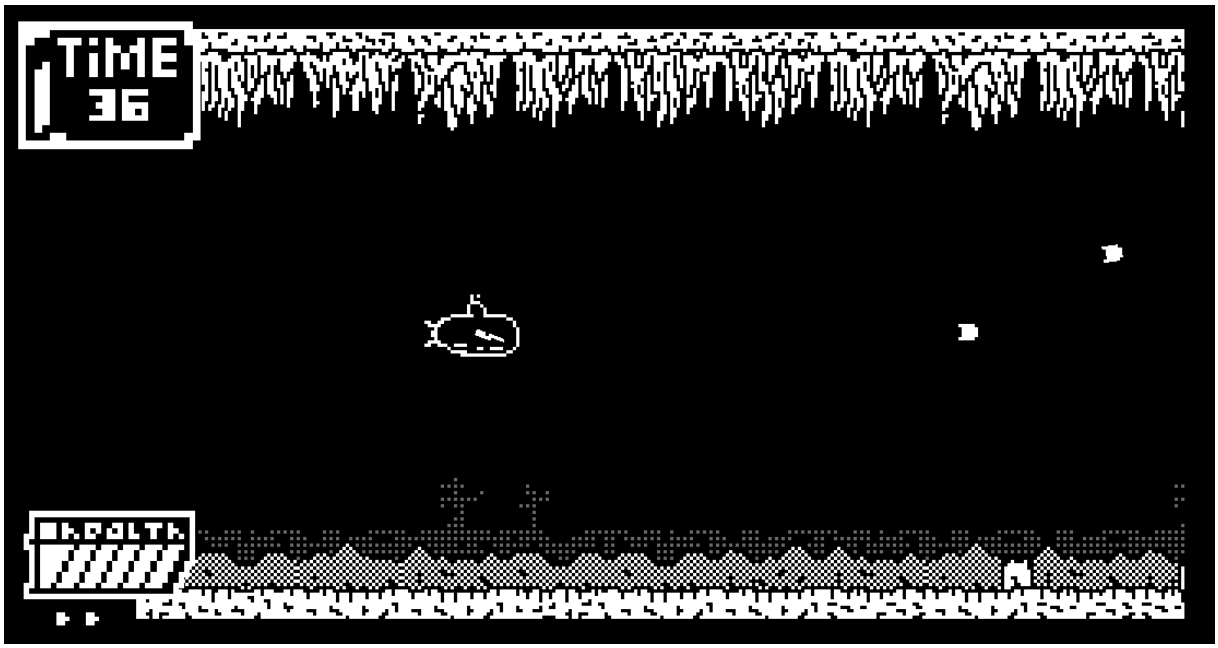


Figura 20 - Captura de tela do componente Principal

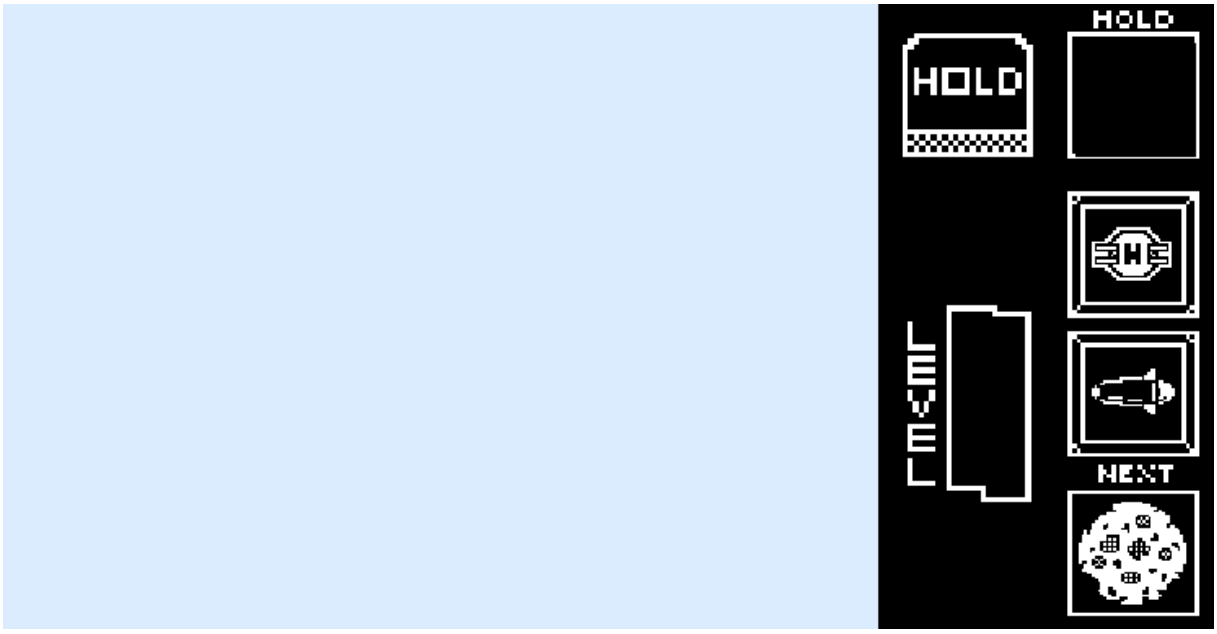


Figura 21 - Captura de tela do componente Secundário

4.6 PUBLICAÇÃO

A partir da elaboração de uma versão inicial jogável, é possível disponibilizar o jogo para que se possa obtê-lo e jogá-lo.

4.6.1 GERANDO VERSÕES EXECUTÁVEIS DO JOGO

As engines de jogos conseguem gerar arquivos executáveis para as mais diversas plataformas. No caso deste projeto, é necessário gerar uma versão para PC, que representa o componente Principal, e outra para dispositivos móveis, para o componente Secundário. Isso é possível com a Unity. Ao selecionar a opção para realizar a compilação, é pode-se escolher para qual plataforma o executável deve ser gerado, bem como quais cenas devem estar presentes na versão.

A Figura 22 apresenta as opções disponíveis para gerar uma *build* ou versão executável do jogo. Em particular, no topo há a marcação de cenas presentes na versão, e na esquerda estão as plataformas-alvo possíveis. Então, deve-se demarcar as cenas necessárias para a *build* que se quer obter e clicar no botão “Build”.

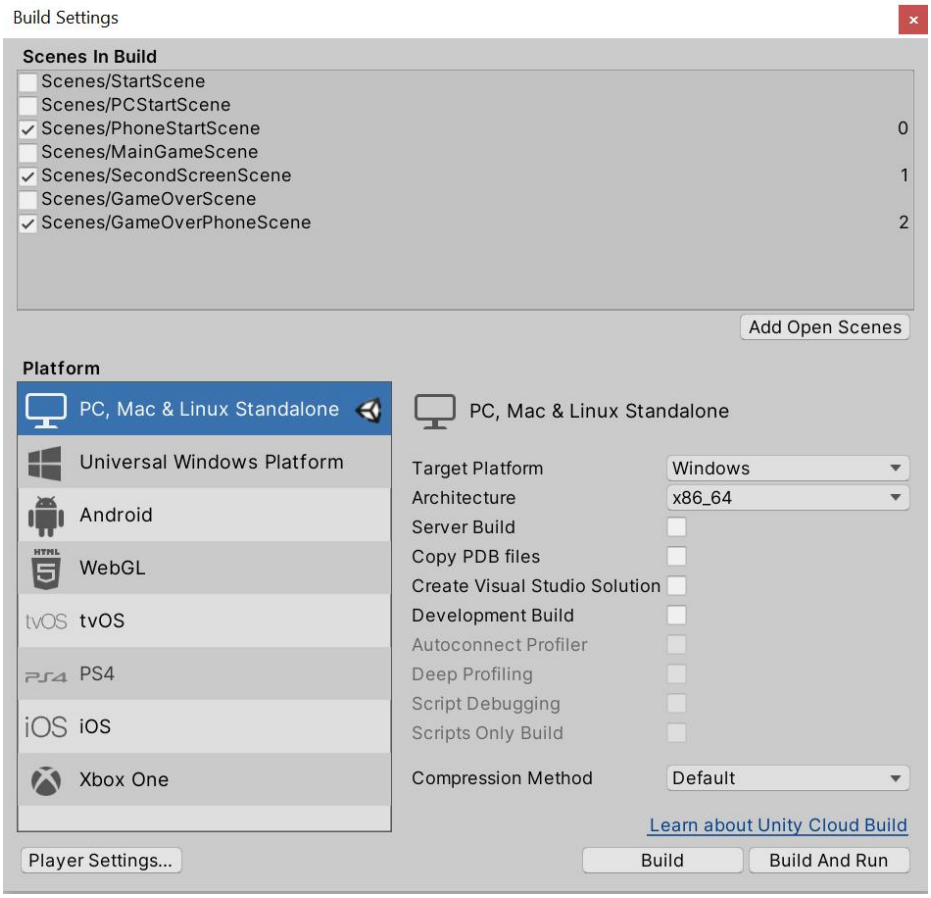


Figura 22 – Configurações de *builds* na Unity.

4.6.2 PUBLICAÇÃO DOS EXECUTÁVEIS

Após obtidas as *builds* a serem distribuídas, é necessário disponibilizá-las em alguma plataforma de distribuição para que os jogadores tenham como obter o jogo e usufruir dele.

Atualmente existem diversas plataformas de distribuição de jogos digitais. Desde o lançamento do serviço *Steam (2003, Valve)*, diversos outros serviços surgiram, como o GOG *(CD Projekt, 2008), Origin (Electronic Arts, 2011)* e *Epic Games Store (Epic Games, 2018)*. Entretanto, estas plataformas possuem um processo de publicação burocrático e por vezes custoso, favorecendo médios e grandes times de desenvolvimento.

Com a popularidade dos jogos *indie* cada vez maior, surgiu a necessidade de plataformas com menos limitações para os criadores. A plataforma escolhida para a publicação deste projeto é o *Itch* (itch corp, 2013). Ele não requer taxas para criação de projetos, além de possuir um fluxo de publicação simples e sem burocracia, ideal para jogos independentes e protótipos pequenos.

4.6.2.1 Publicação na Plataforma Itch

A publicação na plataforma é feita de maneira simples. A Figura 23 apresenta a interface inicial para criação de um novo projeto na plataforma Itch.

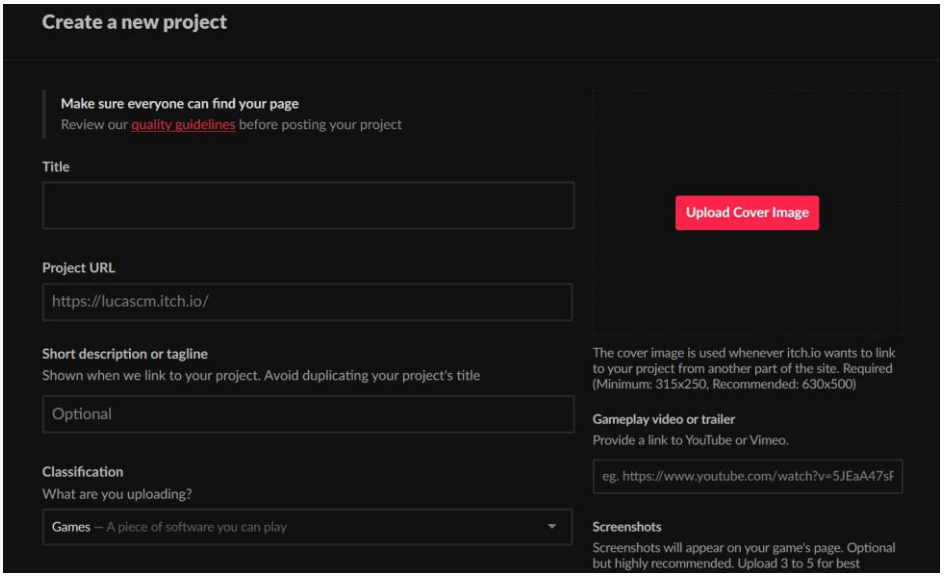


Figura 23 - Menu de criação de novo projeto no Itch.

Após selecionar a opção para criar um novo projeto, o Itch dá diversas opções para customizar a aparência da página do jogo, bem como categorizações que podem ser escolhidas, como gêneros ou dispositivos compatíveis. Pode-se, também, atrelar o projeto a outros sites de distribuição, como a *Steam*, e escolher se o projeto é pago ou gratuito. No caso de ser gratuito, é possível escolher um valor sugerido para ser doado ao tentar acessar o conteúdo em questão. Esta foi a opção escolhida para este projeto.

É importante escolher um nome que seja facilmente identificável para o jogo. Remetendo ao submarino Nautilus do livro Vinte Mil Léguas Submarinas (VERNE, JULIO, 1870) e à assimetria do jogo, optou-se pelo nome Nautilus Diverge. Diverge significa "Divergir" em português, relacionado aqui à disparidade da interação dos jogadores com o jogo.

Tendo customizado todas as opções disponíveis, é necessário fazer o upload dos arquivos do jogo. Isso pode ser feito pelo próprio menu de criação de projeto ou através do gerenciador de arquivos do *Itch*, o *butler* (WENGER, AMOS, 2016). O uso do *butler* é recomendado, pois facilita o envio de correções e atualizações para o jogo, permitindo a automatização do envio de versões novas para os servidores do *Itch*. Sendo assim, o *butler* foi escolhido para fazer o upload de arquivos deste projeto.

O jogo publicado pode ser acessado em https://lucascm.itch.io/nautilus-diverge. O download é gratuito. O download do componente Principal pode ser feito para as plataformas Windows e Linux. O download do componente Secundário é feito para o Android. Para fins de teste, também foram disponibilizadas *builds* do componente Secundário para Windows e Linux.

5 RESULTADOS DE TESTES

Após a publicação da versão final do jogo, foi organizado um experimento para verificar de fato quais emoções ou instintos os jogadores experimentariam ao jogar com um determinado componente. Para isso, foi elaborado um formulário com duas seções. A primeira contendo perguntas básicas como Nome e Gênero, além de uma pergunta para identificar qual componente o jogador jogou.

Além dessas perguntas, na primeira seção também havia uma questão para perguntar se o jogador de fato conseguiu jogar, onde haviam três opções: A primeira para caso ele tenha conseguido jogar, a segunda para algum problema de conexão e a terceira sobre outro erro qualquer. Tivemos como resultado que todos os participantes conseguiram realizar uma partida.

Para a segunda seção e finalizando o formulário, foram duas questões. A primeira pediu para marcar as emoções despertadas durante a partida e a segunda para assinalar os instintos despertados. Antes dessas duas, foi feito um breve resumo com definições das emoções e instintos menos intuitivos do modelo 6-11, como apreciação de cores e comunicação. A reprodução do formulário elaborado pode ser encontrada no Apêndice A.

Por fim, foram obtidas 26 respostas ao final, sendo 6 do gênero feminino e 20 do masculino. Como explicitado anteriormente, todas as 26 pessoas obtiveram sucesso ao tentar jogar uma partida. Com a dificuldade de organização de testes presenciais durante a pandemia de Covid-19, o número de respostas ficou abaixo do necessário para fazer análises mais complexas. Ainda assim os resultados serão apresentados a seguir.

5.1 RESULTADOS COMPONENTE PRINCIPAL

Para o componente principal, foram um total de 16 respostas. As emoções mais despertadas foram a Excitação (10 respostas), Alegria (9 respostas) e Raiva (7 respostas). Já com relação aos instintos, teve-se a Sobrevivência (13 respostas) como o mais escolhido, seguido da Competição (11 respostas).

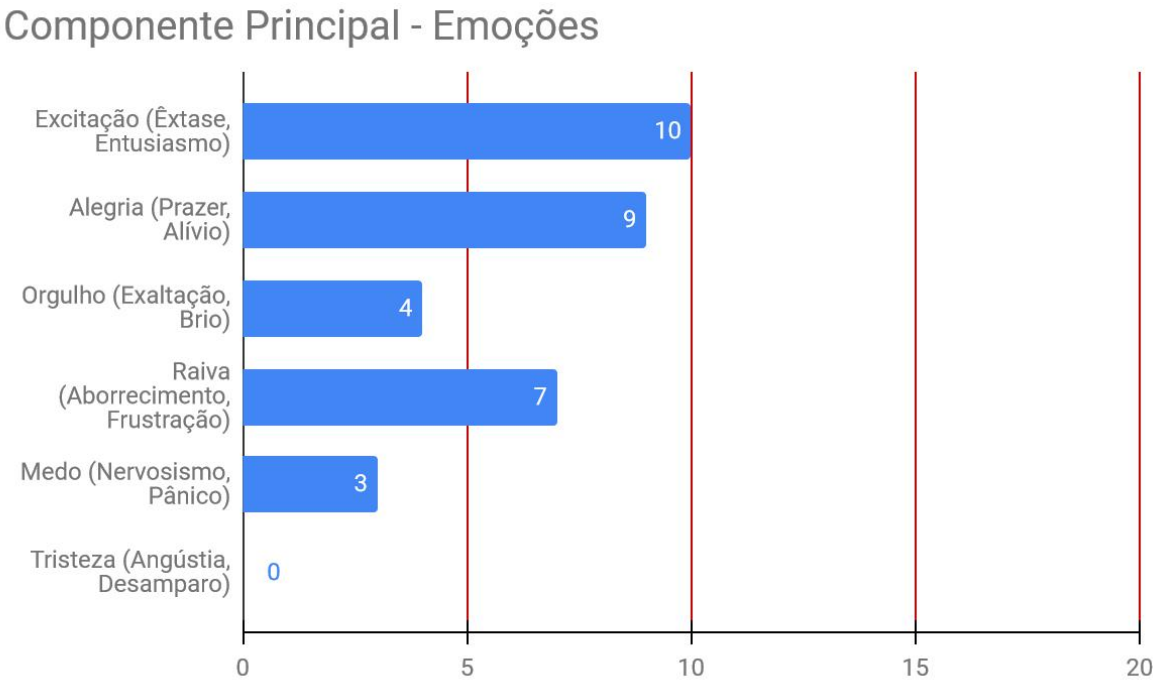


Figura 24 – Emoções despertadas no Componente Principal

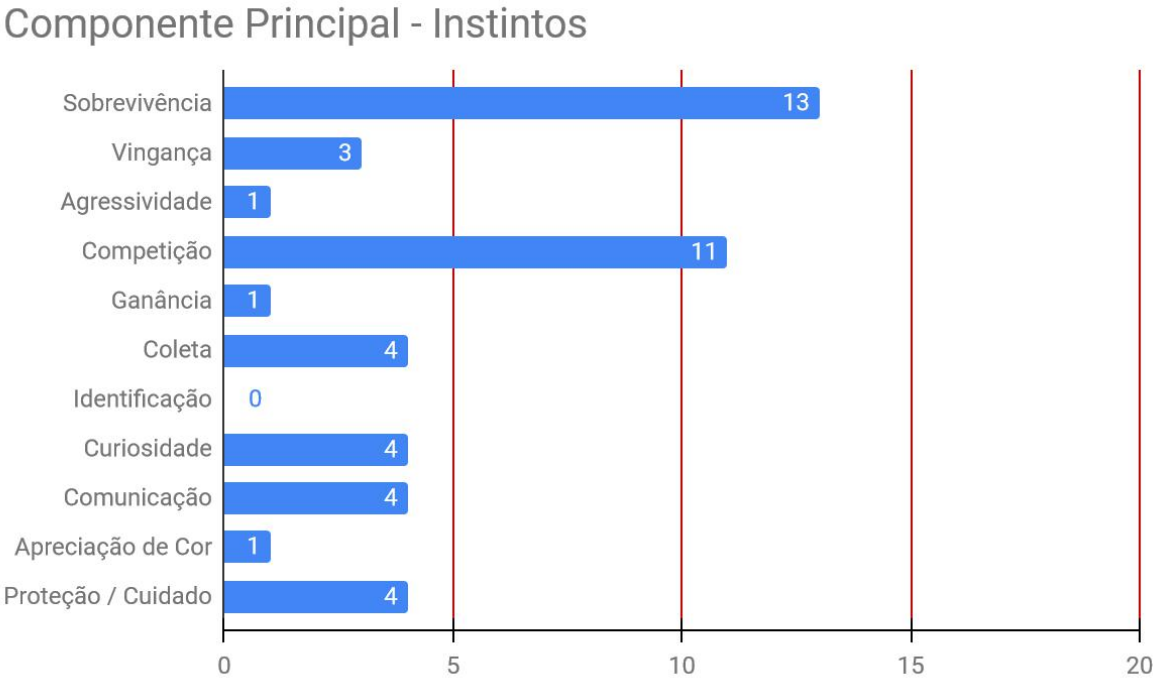


Figura 25 – Instintos instigados no Componente Principal

Realizando um comparativo com o que era esperado, de acordo com o que foi diagramado na Seção 3.5.1, A Excitação e Alegria, emoções que foram pretendidas encontrar através da jogabilidade, foram alcançadas com sucesso na maioria dos jogadores no experimento. O medo e orgulho, outras emoções que também faziam parte do diagrama, aparecem em números menores, três e quatro respectivamente, enquanto obteve-se uma alta recorrência da emoção Raiva, que não era esperada no diagrama inicial.

Com relação aos instintos, observa-se que Sobrevivência e Competição, dois dos instintos previstos no diagrama, são instigados com sucesso na maioria dos jogadores. Em contrapartida, a Identificação não foi instigada em nenhum jogador, talvez pela falta da clareza no que esse instinto realmente significa, enquanto apenas um jogador observou a Apreciação de Cores, que acabou o levando para a emoção Alegria.

5.2 RESULTADOS COMPONENTE SECUNDÁRIO

Para o componente Secundário, foram 10 respostas no total, onde novamente as emoções Alegria e Excitação prevaleceram, alcançando um total de seis respostas, seguida da Raiva, com quatro. Para os instintos, os mais assinalados foram A competição com seis, Vingança com cinco e Agressividade com quatro.

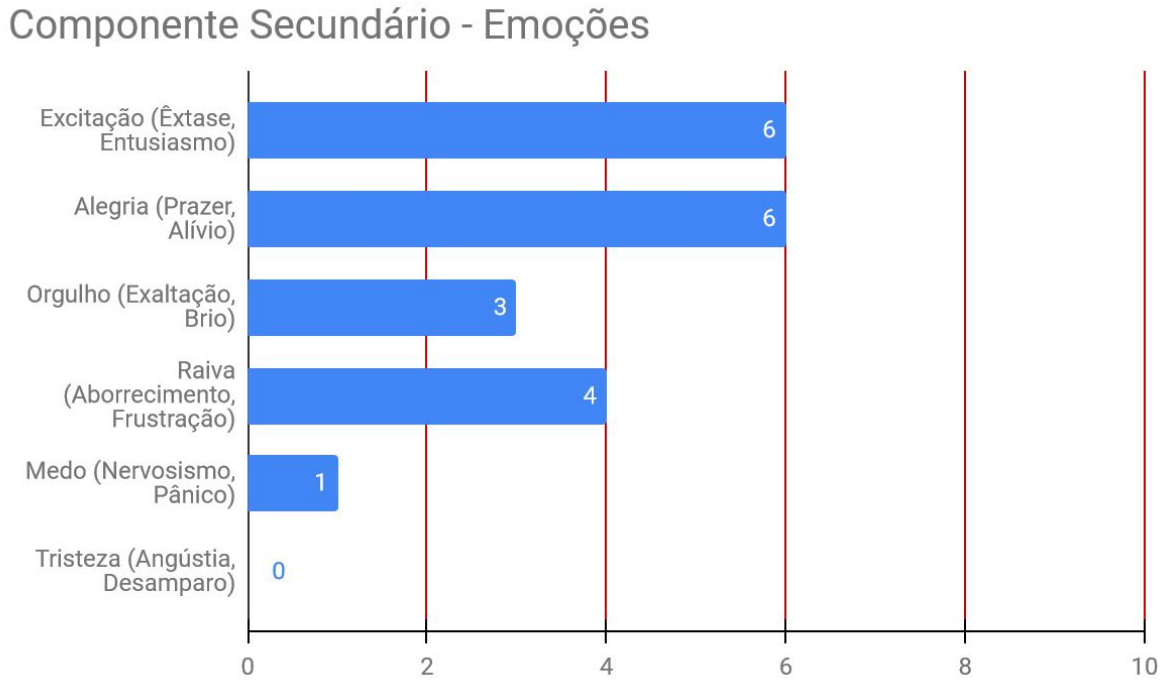


Figura 26 – Emoções despertadas no Componente Secundário

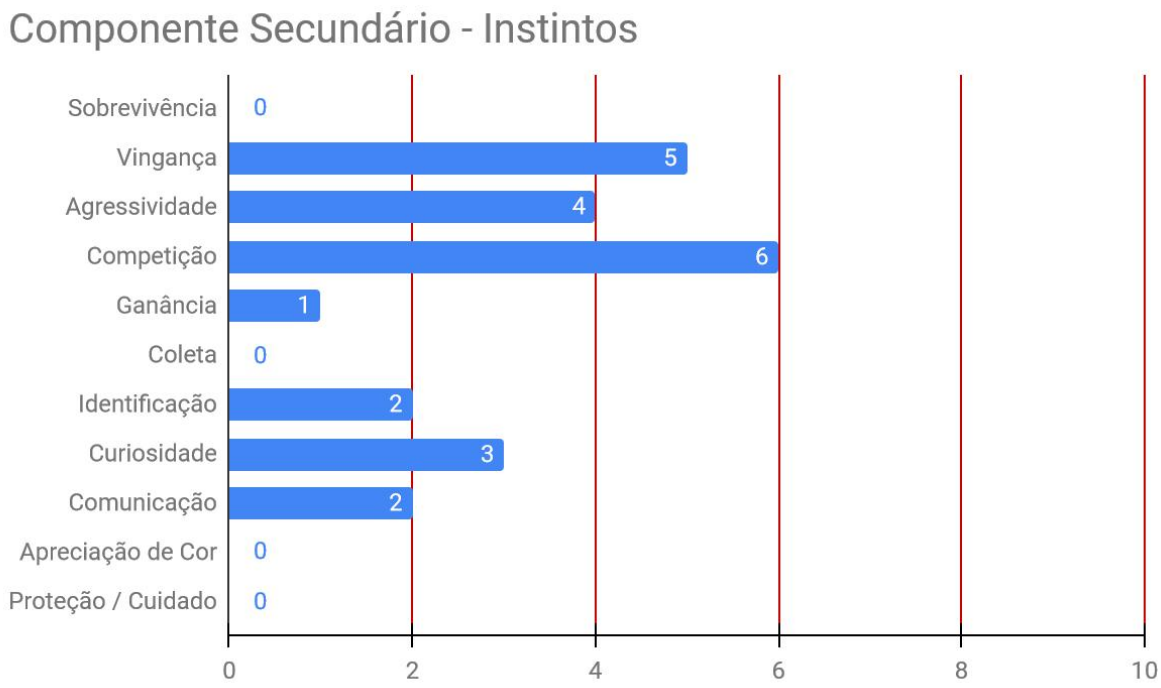


Figura 27 – Instintos instigados no Componente Secundário

A análise junto ao diagrama apresentado na seção 3.5.2 mostra que, de acordo com o resultado do experimento, conseguiu-se despertar as emoções e instintos almejados. Alegria e Excitação, que eram o objetivo final, foram emoções presentes em mais da metade dos jogadores que utilizaram o Componente Secundário, enquanto Raiva e Orgulho, outras duas emoções presentes no diagrama também foram percebidas por algumas pessoas, aparecendo com quatro e três respostas respectivamente.

Para os instintos, a Competição, que é esperado levar a excitação ou ao orgulho, apareceu com o mesmo número de respostas que a primeira, seis. Agressividade, que também era um instinto esperado pelo diagrama, aparece também com uma boa frequência, porém abaixo da metade das respostas. É importante analisar a forte aparência da Vingança com uma quantidade considerável, tendo cinco respostas, possivelmente instigada em pessoas que jogaram outras vezes no lado secundário, após terem perdido para o Principal, ou até mesmo pessoas que perderam no Principal e depois venceram no Secundário.

Diferente do comparativo no Componente Primário, a Identificação no segundo Componente foi de fato notada por alguns jogadores, tendo aparecido duas vezes como resposta. Novamente a Apreciação de Cor apareceu pouco, apenas duas vezes, sendo menos presente do que o pretendido.

6 CONCLUSÃO

6.1 CONSIDERAÇÕES FINAIS

Com a aplicação de técnicas de design e um planejamento adequado, é notório como o processo de desenvolvimento de um jogo está mais acessível do que nunca nos dias atuais. Diversas ferramentas e utilidades estão ao alcance dos desenvolvedores, em grande parte gratuitas ou de baixo custo.

Jogos assimétricos são peculiares no sentido de que não possuem ferramentas adaptadas especificamente para eles. No entanto, usando técnicas consolidadas de design e desenvolvimento para jogos multijogadores comuns e de jogos de apenas um jogador, foi possível desenvolver um jogo multijogador assimétrico, como foi proposto no início deste trabalho.

O design e o desenvolvimento de jogos são atividades consideravelmente iterativas e derivativas, se baseando no que deu certo e no que deu errado nos diversos jogos experimentados pelos autores de um jogo ao longo de suas vidas. Ideias são transmitidas de game designer para game designer, e se manifestam em emoções (idealmente as planejadas pelos designers) que os jogadores irão sentir e compartilhar com outras pessoas enquanto consomem estas obras que compõem uma mídia única e fascinante.

6.2 TRABALHOS FUTUROS

Diversas funcionalidades podem ser acrescentadas ao jogo obtido para torná-lo mais dinâmico e próximo de um jogo comercial. Alguns exemplos seriam novos itens arremessáveis pelo jogador do lado Secundário, outros submarinos selecionáveis com habilidades novas, mais cenários com diferentes obstáculos e mais formas de interação entre os jogadores de cada lado.

Vale destacar que estão presentes alguns bugs no jogo (pequenos defeitos) que não foram solucionados até o término da escrita do trabalho. Em particular, destacam-se problemas de conexão relatados por outros jogadores aos quais foram enviadas versões jogáveis, cuja origem não foi identificada. O ideal seria realizar mais testes em diferentes máquinas, o que não foi possível durante o desenvolvimento do projeto.

REFERÊNCIAS

ACADEMYSOFT. **Tetris**. [s.l: s.n.].

ATARI. **Pong**. [s.l: s.n.].

COBBETT, RICHARD. **From shareware superstars to the Steam gold rush: How indie conquered the PCPC Gamer**, 23 set. 2017. Disponível em: https://www.pcgamer.com/from-shareware-superstars-to-the-steam-gold-rush-how-indie-conquered-the-pc/. Acesso em: 26 jan. 2021

COMPILE. **Puyo Puyo**. [s.l: s.n.].

DAVIDOFF, DIMITRY. **Mafia**Moscou, 1986.

DEVOLVER DIGITAL. **Downwell**. [s.l: s.n.].

DEVOLVER DIGITAL. **Minit**. [s.l: s.n.].

DEVOLVER DIGITAL. **Gato Roboto**. [s.l: s.n.].

DILLON, ROBERTO. **On the Way to Fun: An Emotion-Based Approach to Successful Game Design**. Natick, Massachusetts: A K Peters, Ltd., 2010.

DILLON, ROBERTO. **THE 6-11 FRAMEWORK: A NEW METHODOLOGY FOR GAME ANALYSIS AND DESIGN**, mar. 2011.

DON ESKRIDGE. **The Resistance**, 2009.

ENHANCE GAMES. **Tetris Effect**. [s.l: s.n.].

GOSLIN, AUSTEN. **Among Us saved the world from Zoom fatigue**, 14 dez. 2020. Disponível em: https://www.polygon.com/2020/12/14/22165714/among-us-game-of-the-year-2020. Acesso em: 26 jan. 2021.

HARRIS, JOHN; SCOTT, STACEY; HANCOCK, MARK. Leveraging Asymmetries in Multiplayer Games: Investigating Design Elements of Interdependent Play. 2016.

IGARA STUDIO. **Aseprite**. [s.l: s.n.].

INNERSLOTH. **Among Us**. [s.l: s.n.].

NESKY, JOHN. **beepbox**. [s.l: s.n.].

NINTENDO. **Dr. Mario**. [s.l: s.n.].

NINTENDO. **Série Mario Kart**. [s.l: s.n.].

NINTENDO. **Animal Crossing**. [s.l: s.n.].

SUCCESS. **Cotton: Fantastic Night Dreams**. [s.l: s.n.].

SWIRSKY, JAMES, P., Lisanne. **Indie Game: The Movie**BlinkWorks Media, , 20 jan. 2012.

SYZYGY ENGINEERING. **Computer Space**. [s.l: s.n.].

THATGAMECOMPANY. **Journey**. [s.l: s.n.].

TORVALDS, LINUS. **Git**. [s.l: s.n.].

TREASURE. **Ikaruga**. [s.l: s.n.].

UNITY TECHNOLOGIES. **Coroutines**. Disponível em: https://docs.unity3d.com/Manual/Coroutines.html.

VERNE, JULIO. **Vinte Mil Léguas Submarinas**. [s.l: s.n.].

VIS2K. **Telepathy**. [s.l: s.n.].

WENGER, AMOS. **butler**. [s.l: s.n.].

YERRICK, DAMIAN. **Peças do Tetris**, 14 nov. 2016.

ZUBEK, ROBERT, H., Robin; LEBLANC, MARC. **MDA: A Formal Approach to Game Design and Game Research**, jan. 2004.

APÊNDICE A – REPRODUÇÃO DO FORMULÁRIO DE TESTES UTILIZADO NA SEÇÃO 5

Nautilus Diverge

Este formulário é sobre a experiência de jogar o Nautilus Diverge, jogo de trabalho de conclusão de curso. Ele pode ser obtido gratuitamente em https://lucascm.itch.io/nautilus-diverge

Se possível, peça para a pessoa que jogou contigo responder a pesquisa também, ajudaria bastante o nosso projeto!

Caso não consiga jogar por qualquer motivo, conte um pouco sobre o problema.

*Obrigatório

1. Qual o seu nome? *

2. Qual a sua idade? *

3. Qual seu gênero?

Marcar apenas uma oval.

- ◯ Masculino
- ◯ Feminino
- ◯ Outro: _____

4. Qual componente do jogo você utilizou? *

Marcar apenas uma oval.

- ◯ Componente Primária (Submarino)
- ◯ Componente Secundária (Lançador)

5. Você conseguiu jogar uma partida? *

Marcar apenas uma oval.

- ◯ Sim, consegui conectar e jogar
- ◯ Não consegui fazer a conexão para jogar
- ◯ Não, o jogo apresentou algum outro erro

Respostas emocionais

Ao jogar um jogo, são despertadas algumas emoções e instintos em um jogador. Por exemplo, pode-se sentir raiva ao ser golpeado por outro jogador durante uma partida, o que despertaria um instinto de vingança, agressividade ou competitividade. Alguns desses instintos são intuitivos, já outros são mais complexos. Explicamos um pouco eles aqui para que possa responder de maneira mais adequada:

Sobrevivência (Querer manter o personagem vivo, não deixar ele morrer ou ser atingido pelos obstáculos)

Identificação (Ter algum sentimento de , comportamental ou de acordo com a jogabilidade)

Coleta (Colecionar objetos, obter poderes e itens)

Ganância (Vontade de adquirir determinado poder ou de obter recursos para si)

Vingança (O jogo despertou a vontade de dar o troco em seu adversário ou em algum objeto em específico)

Proteção / Cuidado (Vontade de proteger um personagem secundário ou algum objeto essencial)

Comunicação (Vontade de se comunicar com seu inimigo ou com alguém, por exemplo para pedir auxílio ou provocar um adversário)

Apreciação de Cor (Um dos menos intuitivo, não exatamente tem a ver com cores. Significa que o jogo foi agradável esteticamente, os visuais despertaram a atenção do jogador)

6. Quais emoções você sentiu ou notou enquanto estava jogando? (Entre parênteses estão algumas variações destas emoções) *

Marque todas que se aplicam.

- [ ] Medo (Nervosismo, Pânico)
- [ ] Raiva (Aborrecimento, Frustração)
- [ ] Orgulho (Exaltação, Brio)
- [ ] Excitação (Êxtase, Entusiasmo)
- [ ] Tristeza (Angústia, Desamparo)
- [ ] Alegria (Prazer, Alívio)

7. Quais instintos seus foram instigados durante a partida? *

Marque todas que se aplicam.

- [ ] Sobrevivência
- [ ] Identificação
- [ ] Coleta
- [ ] Ganância
- [ ] Agressividade
- [ ] Vingança
- [ ] Competição
- [ ] Proteção / Cuidado
- [ ] Curiosidade
- [ ] Comunicação
- [ ] Apreciação de Cor

Figura 28 - Reprodução do formulário de testes usado na seção 5