

WILLIAN MAGNO PEREIRA REIS

COMPUTAÇÃO HUMANA APLICADA À  
GERAÇÃO DE CONTEÚDO EM JOGOS DE  
PLATAFORMA 2D

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA  
MINAS GERAIS – BRASIL  
2015

**Ficha catalográfica preparada pela Biblioteca Central da Universidade  
Federal de Viçosa - Câmpus Viçosa**

T

R375c  
2015  
Reis, Willian Magno Pereira, 1990-  
Computação humana aplicada à geração de conteúdo em  
jogos de plataforma 2D / Willian Magno Pereira Reis. – Viçosa,  
MG, 2015.  
xii, 52f. : il. (algumas color.) ; 29 cm.

Inclui apêndice.

Orientador: Levi Henrique Santana de Lelis.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f.47-51.

1. Inteligência artificial. 2. Interação homem-máquina.  
3. Sistemas de computação interativos. 4. Jogos eletrônicos.  
I. Universidade Federal de Viçosa. Departamento de Informática.  
Programa de Pós-graduação em Ciência da Computação.  
II. Título.

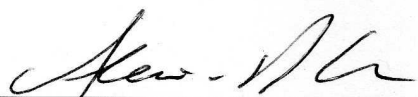
CDD 22. ed. 006.3

**WILLIAN MÁGNO PEREIRA REIS**

**COMPUTAÇÃO HUMANA APLICADA À GERAÇÃO DE  
CONTEÚDO EM JOGOS DE PLATAFORMA 2D**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 06 de novembro de 2015.



Alcione de Paiva Oliveira



Alex Fernandes da Veiga Machado



Levi Henrique Santana de Lelis  
Orientador

*À Deus, porque dele e por ele, e para ele, são todas as coisas, à minha família, à minha namorada Júlia que ainda não é da família, mas quase, e aos meus amigos que me ajudaram, e muito, a chegar até aqui.*

*“Educação é uma descoberta progressiva de nossa própria ignorância”*  
(Voltaire)

# Agradecimentos

A vida é feita de diversas jornadas e em cada uma destas, muitas pessoas fazem parte. Algumas ficam, outras vão, mas todas tem sua importância na formação da nossa história. Deixo aqui, meus sinceros agradecimentos a todos que fizeram parte dessa história.

Primeiramente, agradeço a **Deus** por minha existência e saúde. Por ser meu fundamento, apoio e luz no fim do túnel. Pois apesar das tribulações, houve a paciência, a experiência, e a esperança.

Agradeço aos meus pais **Marcos, Marlene** e IRMÃOS **Alex, Jonatas, Vitor** e **Ester**, que me auxiliaram em tudo, me apoiaram e possibilitaram, com muito esforço, que eu chegasse até aqui.

Agradeço a minha namorada **Júlia**, que esteve ao meu lado e presenciou grande parte desta minha jornada. Ainda me auxiliou e deu-me doses de ânimo para continuar.

Agradeço a todos meus familiares, sem exceções, maternos e paternos: avó **Lourdes**, tios, tias e primos.

Agradeço imensamente ao meu orientador **Levi Lelis**, que na chegada ao departamento me “adotou”. Obrigado por acreditar até mesmo quando eu não acreditei, por me dar “injeções” de ânimo, por me auxiliar nos momentos bons e ruins.

Agradeço à “Família DPI”, com quem compartilhei momentos muito bons e alguns ruins. Agradeço pelos grupos de estudo, passeios, brincadeiras, distrações e as muitas e muitas esperas para o almoço. Agradeço a todos que fizeram parte do meu dia-a-dia na UFV, em especial a **Angelica Ribeiro, Cristiane Lana, Gerardo Zárate, Italo Oliveira, Jaqueline Alvarenga, Katia Borges, Lucas Gonçalves, Marcos Montanari, Marques Sousa** e **Paôla Cazetta** que marcaram positivamente esta jornada.

Ao meu amigo **Marques Souza**, agradeço novamente, pois além de ter feito parte da família DPI, foi meu companheiro desde o primeiro algoritmo até a ajuda com a utilização do Latex. Meu companheiro de viagens e concursos.

Aos colegas do futebol DPI, meu agradecimento pelas ótimas horas semanais, pela ajuda na fuga do sedentarismo, já que para a maioria era a única hora de atividade física na semana.

Aos professores do DPI, em especial dos quais fui aluno ou com os quais desempenhei algum tipo de atividade. Contribuíram, e muito, para minha formação acadêmica e pessoal.

Aos funcionários do DPI, que contribuíram de alguma forma nesta jornada. Em especial ao **Altino**, que mais do que um secretário, dispunha-se como um amigo.

Aos meus professores durante a graduação no IFSEMG, que contribuíram com toda base da minha formação. Estes que além de professores, tornaram-se amigos. Agradecimento especial aos professores **Alex Machado, Gustavo Reis e João Paulo Lamas**, que mais que professores, foram amigos e contribuíram com mais do que com a formação acadêmica.

Aos meus colegas de república, que foram como uma família durante o tempo que estive em Viçosa. Muito obrigado por todos os momentos proporcionados.

Agradeço a CAPES pelo financiamento fornecido para que esta jornada fosse possível.

Agradeço a Universidade Federal de Viçosa e ao Departamento de Informática pela condição e suporte concedidos para que aperfeiçoasse minha parte acadêmica.

Agradeço a todos que não foram citados e que não se consideram inclusos nos grupos acima, mas que tiveram participação no meu dia-a-dia, no período que estive cursando o mestrado.

A todos meu muito obrigado.

# Sumário

<b>Lista de Figuras</b>	<b>viii</b>
<b>Lista de Tabelas</b>	<b>x</b>
<b>Resumo</b>	<b>xi</b>
<b>Abstract</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Descrição do Problema . . . . .	2
1.2 Objetivos . . . . .	3
1.3 Estrutura da Dissertação . . . . .	3
<b>2 Revisão Bibliográfica</b>	<b>5</b>
2.1 Geração Procedimental de Conteúdo . . . . .	5
2.2 Sistemas PCG para o Jogo IMB . . . . .	5
2.3 Avaliação de Sistemas PCG em IMB . . . . .	6
2.4 O Super Mario Bros . . . . .	7
2.5 Computação Humana . . . . .	12
2.6 Arcos de Tensão . . . . .	13
<b>3 Human Computation for Procedural Content Generation in Plat-</b>	
<b>form Games</b>	<b>15</b>
3.1 Introduction . . . . .	15
3.1.1 Our Contributions . . . . .	16
3.2 Related Work . . . . .	17
3.3 The Problem Domain of Mario Bros . . . . .	18
3.4 The Human Computation Approach to PCG . . . . .	18
3.5 Basic System for Level Generation . . . . .	20

3.6	Human-Annotated Levels . . . . .	20
3.6.1	Representative Annotated Levels . . . . .	21
3.6.2	Relation of Level Size and Enjoyment . . . . .	24
3.7	Tension-Arc Based Small Level Combination . . . . .	25
3.8	Empirical Evaluation . . . . .	27
3.8.1	Methodology . . . . .	27
3.8.2	Hypotheses . . . . .	29
3.8.3	Quantitative Results . . . . .	30
3.8.4	Discussion . . . . .	31
3.8.5	Qualitative Results . . . . .	32
3.9	Conclusions . . . . .	33
<b>4</b>	<b>Considerações Finais</b>	<b>34</b>
	<b>Referências Bibliográficas</b>	<b>35</b>
<b>A</b>	<b>Os sistemas utilizados para coleta de dados</b>	<b>40</b>
A.1	Avaliação dos pequenos níveis . . . . .	40
A.2	Avaliação dos níveis completos . . . . .	41

# Lista de Figuras

2.1	Inimigos presentes no Infinite Mario Bros. . . . .	9
2.2	Cascos das Koopas . . . . .	9
2.3	Itens do jogo. . . . .	10
2.4	Estados ou Modos do Mario . . . . .	10
2.5	Imagem de uma lacuna ou buraco . . . . .	11
2.6	Power-Ups . . . . .	11
3.1	Screen shot of the game of Infinite Mario Bros. . . . .	19
3.2	A few representative human evaluations for levels in $\Gamma$ . The numbers in parenthesis show the value of enjoyment, aesthetics, and difficulty, respectively in a scale from 1 to 9. For example, a level with values (1, 1, 1) mean that the level is not challenging, it has bad aesthetics, and it is not enjoyable. . . . .	22
3.3	Tension arc used in our experiments. The difficulty values are provided by human annotated levels, and the values on the $x$ -axis denote the tension arc's ordering. . . . .	26
A.1	Fluxograma HCTA . . . . .	41
A.2	Tela sobre o sistema. . . . .	42
A.3	Tela de comandos. . . . .	42
A.4	Tela de instruções nº 1. . . . .	43
A.5	Tela de instruções nº 2. . . . .	43
A.6	Tela de instruções nº 3. . . . .	44
A.7	Tela de instruções nº 4. . . . .	44
A.8	Tela de instruções nº 5. . . . .	45
A.9	Tela de instruções nº 6. . . . .	45
A.10	Tela de instruções nº 7. . . . .	46
A.11	Tela de instruções nº 8. . . . .	46

A.12 Tela com informações sobre o ranking. . . . .	47
A.13 Tela do pré-jogo. . . . .	47
A.14 Tela do jogo. . . . .	48
A.15 Tela de Avaliação. . . . .	48
A.16 Tela de solicitação de algumas informações sobre o jogador. . . . .	49
A.17 Tela exibindo as teclas a serem utilizadas para jogar. . . . .	49
A.18 Pequeno nível para que os jogadores possam treinar os controles. . . . .	50
A.19 Tela pré-jogo. . . . .	50
A.20 Tela com o nível gerado por um dos geradores avaliados neste trabalho. . . . .	51
A.21 Tela para avaliar o nível jogado. . . . .	51
A.22 Tela final que permite ao jogador fazer comentários sobre o experimento. . . . .	52

# Lista de Tabelas

- 3.1 Empirical evaluation of different PCG systems. Lower values of enjoyment and aesthetics indicate levels which are more enjoyable to play and have better aesthetics; larger values of Turing indicate levels which participants were more prone to believe that were generated by humans. 30

# Resumo

REIS, Willian Magno Pereira, M.Sc., Universidade Federal de Viçosa, Novembro de 2015. **Computação Humana Aplicada à Geração de Conteúdo em Jogos de Plataforma 2D**. Orientador: Levi Henrique Santana de Lelis.

Um dos grandes desafios na geração procedimental de conteúdo em jogos de computador é avaliar de forma automática se o conteúdo gerado tem boa qualidade. Essa dissertação descreve um sistema que utiliza computação humana para avaliar pequenos níveis gerados por um sistema existente para o jogo de plataforma Infinite Mario Bros. Esses pequenos níveis são então combinados em um nível completo do jogo. A composição desses pequenos níveis em um nível completo é feita de acordo com as informações coletadas através da computação humana e de acordo com o modelo matemático de arco de tensão comumente utilizado em narrativas como filmes, romances e novelas. O sistema foi testado com pessoas e os resultados sugerem que a abordagem proposta é capaz de gerar níveis com estética visual melhor que níveis gerados por outras abordagens. Os resultados também sugerem que, dentre todas as abordagens testadas, os níveis gerados pelo sistema proposto são os mais agradáveis de se jogar.

# Abstract

REIS, Willian Magno Pereira, M.Sc., Universidade Federal de Viçosa, November of 2015. **Human Computation for Procedural Content Generation in Platform Games**. Advisor: Levi Henrique Santana de Lelis.

One of the major challenges in procedural content generation in computer games is to automatically evaluate whether the generated content has good quality. In this dissertation we describe a system which uses human computation to evaluate small portions of levels generated by an existing system for the game of Infinite Mario Bros. Several such evaluated portions are then combined into a full level of the game. The composition of the small portions into a full level is done by accounting for the human-annotated information and the mathematical model of tension arcs used in interactive drama and storytelling. We tested our system with human subjects and the results show that our approach is able to generate levels with better aesthetics and that are more enjoyable to play than other existing approaches.

# Capítulo 1

## Introdução

A indústria dos jogos vem apresentando um grande crescimento nos últimos anos. Além da quantidade de jogos, a procura dos jogadores por jogos ricos em conteúdo aumenta cada vez mais. Isto faz com que o processo de desenvolvimento de jogos fique cada vez mais complexo, levando a indústria de jogos de computador a enfrentar a perspectiva do aumento crescente e contínuo dos custos com artistas, programadores e designers [Togelius et al., 2011]. A etapa que envolve o projeto e desenvolvimento do conteúdo de um jogo é uma atividade que exige criatividade e consome recursos relacionados a tempo e dinheiro [Shaker et al., 2012a]. A utilização de algoritmos capazes de produzir, de forma automática, conteúdo desejável, pode colaborar para uma redução significativa nas despesas neste sentido [Togelius et al., 2011].

Esta, entre outras causas, tem colaborado para o aumento no interesse pela geração automática de conteúdo do jogo, por meio de técnicas que envolvem ou não a interação humana do designer. Técnicas computacionais permitem, não somente reduzir o custo do desenvolvimento, como também gerar uma grande variação no conteúdo, proporcionando uma experiência única a cada repetição do jogo [Shaker et al., 2012a].

A geração automática de conteúdo de boa qualidade é um objetivo de longo prazo na área de Inteligência Artificial. Em jogos de computador, ao dizer conteúdo, pode-se estar fazendo referência às fases, regras, texturas, histórias, comentários, entre outros. A área da pesquisa que trabalha com a geração automática de conteúdo é conhecida como Geração de Conteúdo Procedimental (PCG - sigla do inglês, *Procedural Content Generation*). A utilização técnicas PCG facilitam o processo de desenvolvimento, possibilitando um foco maior em pontos de maior relevância para o produto final. Togelius et al. [2011] menciona alguns exemplos famosos de jogos

comerciais que utilizam PCG, como a geração de masmorras em *Rogue* (Projeto IA 1980) e *Diablo* (Blizzard 1996), a geração de mapas em *Civilization* (MicroProse 1991), a geração das armas em *Borderlands* (Gearbox 2009) e para a geração de vegetação pelo software *SpeedTree* (Interactive Data Visualization 2003). Apesar de existir casos mais antigos de sua utilização, apenas recentemente abordagens de Inteligência Artificial e Computacional começaram a ser exploradas no contexto da criação de elementos centrais do jogo, tais como níveis e mapas [Shaker et al., 2011].

Um dos principais desafios em PCG é avaliar automaticamente se o conteúdo gerado tem boa qualidade. Por exemplo, no contexto de jogos de computador, é importante ser capaz de avaliar se o conteúdo gerado tem boa estética visual e se o conteúdo torna o jogo mais divertido de se jogar. Essa avaliação é importante porque ela orienta os sistemas de PCG baseados em algoritmos de Inteligência Artificial a gerar conteúdo de boa qualidade.

A fim de resolver problemas que computadores não podem resolver, Computação Humana surge como um paradigma que utiliza o poder de processamento humano, para auxiliar os computadores [Von Ahn, 2009]. Este trabalho apresenta um gerador PCG, que utiliza computação humana com o objetivo de obter conteúdo de qualidade.

Durante a criação de histórias, escritores procuram utilizar obstáculos como fonte de tensão entre o protagonista e seus objetivos. A variação da tensão no decorrer do tempo é chamado arco de tensão. Com o objetivo de estudar o efeito do arco de tensão nos jogos, neste trabalho utiliza-se um arco de tensão para geração automática de fases de um jogo de plataforma 2D.

Levando em consideração a importância que tem tido a geração automática de conteúdo e avaliação desse conteúdo, este trabalho focou-se nessas duas direções, com intuito de propor um gerador procedimental de conteúdo capaz de produzir conteúdo de boa qualidade do ponto de vista do jogador.

## 1.1 Descrição do Problema

Conforme mencionado, a indústria tem se deparado com gastos crescentes no desenvolvimento de jogos, principalmente quando relacionado ao conteúdo. A fim de minimizar estes custos, a aplicação de técnicas de geração procedimental surge como um bom recurso. Entretanto, o conteúdo gerado precisa ter uma boa qualidade. E a geração automática de conteúdo de boa qualidade é um objetivo de longo prazo, assim como o desafio de avaliar automaticamente este conteúdo.

A proposta apresentada neste trabalho é de um gerador capaz de produzir como saída conteúdo de qualidade do ponto de vista do jogador.

## 1.2 Objetivos

O principal objetivo desta dissertação é a proposta de uma técnica para geração procedimental de conteúdo, onde este possua boa qualidade. Para esta proposta, será utilizado computação humana para avaliação do conteúdo gerado. Esta técnica será implementada em um sistema para que possa ser avaliada. Foi escolhido o *Infinite Mario Bros* (IMB), uma variante do *Super Mario Bros*, para o qual serão gerados níveis de forma automaticamente. A escolha por IMB se deu por este ter recebido muita atenção de pesquisadores de IA recentemente. A razão por esta popularidade entre os pesquisadores de IA é que IMB é um excelente teste para sistemas PCG: o jogo é simples o suficiente para permitir aos pesquisadores experimentar rapidamente novas abordagens PCG e ainda é bastante divertido. Para isso, deverão ser executados os seguintes objetivos específicos:

- Investigar técnicas de geração de conteúdo procedimental na literatura;
- Investigar técnicas de avaliação de conteúdo gerado por técnicas PCG;
- Apresentar uma metodologia para uma técnica PCG, capaz de gerar conteúdos de qualidade;
- Aplicar a metodologia através de uma implementação de um gerador PCG;
- Avaliar o conteúdo gerado com a opinião de voluntários;
- Discutir os resultados.

## 1.3 Estrutura da Dissertação

Os resultados apresentados nessa dissertação foram publicados nos anais da *IEEE International Conference on Computational Intelligence and Games (CIG)* [Reis et al., 2015]. Essa dissertação está em formato de artigo.

Este documento está organizado da seguinte forma: o Capítulo 1 faz uma introdução ao problema e apresenta os objetivos. O Capítulo 2 apresenta um breve referencial teórico, onde explica-se os principais tópicos abordados neste trabalho. Nele estão os principais trabalhos relacionadas com esta dissertação. O Capítulo 3

é uma reprodução *ipsis litteris* de Reis et al. [2015]. Capítulo 4 são apresentadas considerações gerais sobre o trabalho.

# Capítulo 2

## Revisão Bibliográfica

Neste capítulo é apresentada uma revisão da literatura relacionada à proposta apresentada nesta dissertação. São apresentados definições relacionadas aos termos usados nesta.

### 2.1 Geração Procedimental de Conteúdo

Togelius et al. [2014] referem-se a conteúdo como elementos contidos em um jogo como níveis, mapas, regras do jogo, texturas, histórias, itens, quests, música, armas, veículos, personagens, dentre outros. Excluem da definição o motor de jogo em si e o comportamento do personagem não-jogador (*non-player character* NPC).

A utilização dos termos “procedimental” e “geração” implica na utilização de procedimentos computacionais, que criam algo. Um método PCG pode ser executado por um computador, com ou sem ajuda humana, e produzir algo como saída. Um sistema PCG, refere-se a um sistema que incorpora um método PCG como uma de suas partes. Por exemplo, um jogo adaptativo ou uma ferramenta de design de jogos assistida por Inteligência Artificial [Togelius et al., 2014]. Os métodos PCG podem ser usados não só para criar conteúdo de um dado jogo, mas também para criar novas mecânicas de jogo. Ou também para criar conteúdo de acordo com a preferência do usuário [Togelius et al., 2014].

### 2.2 Sistemas PCG para o Jogo IMB

Essa seção revisa os principais sistemas PCG para geração de níveis de IMB.

Shaker et al. [2010] descrevem um sistema para gerar níveis IMB adaptados para jogadores específicos. O sistema em questão pergunta diretamente aos jogadores sobre suas preferências. Dahlskog & Togelius [2013] apresentam um estudo comparando diferentes níveis de IMB. Bakkes et al. [2014] descrevem um sistema para balancear o desafio dos níveis de IMB. Pedersen et al. [2009] apresentam um sistema que modela a experiência do jogador. Essa modelagem é feita através de dados coletados em um estudo com usuários. Eles utilizam modelos estatísticos de aprendizagem para prever, dado um nível IMB  $L$ , o desafio que  $L$  oferece ao jogador. Da mesma forma, em diferentes trabalhos, Shaker et al. [2011, 2012b, 2013] mostraram como extrair características para modelos preditivos da experiência do jogador em IMB. Pederson et al. e Shaker et al. tem o objetivo de utilizar esses modelos para guiar a busca por níveis IMB de boa qualidade adaptados a jogadores específicos.

Smith et al. [2010] apresentaram Tanagra, um sistema para gerar níveis para jogos de plataforma 2D. Mais tarde, Smith et al. [2011b] apresentaram Launchpad, um sistema que usa *rhythm groups* para gerar níveis de jogos de plataforma. Shaker et al. [2012a] usaram uma gramática para codificar de forma concisa restrições de design para a evolução de níveis de IMB. Em outro trabalho Shaker et al. [2012c] avalia o conteúdo personalizado gerado por um sistema PCG baseado em gramática com agentes artificiais. Em um trabalho recente Shaker & Abou-Zleikha [2014] usam fatoração de matrizes para gerar níveis com base em padrões aprendidos a partir de níveis gerados por outros sistemas.

Dahlskog et al. [2014] utilizaram *n-grams* criados a partir de níveis originais do IMB para gerar novos níveis do jogo. Em outros dois trabalhos Dahlskog & Togelius [2014a,b] apresentaram sistemas que utilizam padrões para gerar níveis de IMB. Sorenson et al. [2011] apresentaram um sistema que usa a ideia de *rhythm groups* introduzidas por Smith et al. [2008, 2011b] para definir um modelo computacional de prazer do jogador para evoluir níveis de IMB. Este modelo também é utilizado para avaliar os níveis resultantes.

## 2.3 Avaliação de Sistemas PCG em IMB

Smith & Whitehead [2010] e Horn et al. [2014] introduziram várias métricas computacionais para avaliar, o que os autores chamaram de a expressividade dos sistemas PCG. Essas métricas foram usadas em vários trabalhos como uma forma de avaliação de sistemas PCG. Mariño et al. [2015] mostraram que mais importante que um

estudo com as métricas introduzidas por Smith et al. e Horn et al. é realizar um estudo com usuários do sistema. Nesse trabalho o sistema apresentado é avaliado como sugerido por Mariño et al.

Alguns sistemas PCG são avaliados sem estudos de usuários ou métricas computacionais. Por exemplo, o sistema PCG *Occupancy-Regulated Extension* (ORE) é avaliado pelos próprios autores através de uma análise crítica dos níveis gerados [Mawhorter & Mateas, 2010]. Kerssemakers et al. [2012] também apresentam uma auto crítica da abordagem proposta, além de uma análise empírica do tempo de execução do sistema. O trabalho seminal de Compton & Mateas [2006] na geração de conteúdo para jogos de plataforma não apresenta uma avaliação formal dos níveis gerados.

Recentemente, Canossa & Smith [2015] introduziram várias métricas baseadas em teorias de design para avaliar níveis de IMB. As métricas foram obtidas através de discussões com estudantes de design. No entanto, em contraste com as métricas computacionais introduzidas por Smith & Whitehead [2010] e Horn et al. [2014], algumas das métricas de Canossa e Smith não são formais o suficiente para serem implementadas como um procedimento de computador.

## 2.4 O Super Mario Bros

Nesta seção é descrito o Super Mario Bros, jogo que inspirou o Infinite Mario Bros, utilizado como caso de estudo neste trabalho.

Super Mario Bros é um famoso jogo de plataforma da Nintendo. O jogo se passa no Reino dos Cogumelos. Este é um reino pacífico que é invadido por uma tribo de tartarugas, denominadas Koppas. Essas são famosas por sua magia negra. Os habitantes do reino são transformados em pedras e tijolos levando o reino ao caos. A princesa do Reino dos Cogumelos, Toadstool, é única capaz de desfazer o feitiço de sobre seu reino. Entretanto ela está presa com o rei das Koopas, chamado Bowser. Mario, que é um encanador e candidato a herói da história, sai em uma missão para libertar a princesa do Reino do Cogumelo e restaurar o Reino dos Cogumelos [Nintendo, 1985].

Em Super Mario Bros joga-se com o Mario, que é o personagem principal. A jogabilidade consiste em mover o Mario através de níveis bidimensionais, que são vistos lateralmente. O principal objetivo do jogo é controlar o Mario para que ele atinja a meta no final de cada nível. A meta está localizada na parte mais à direita dos níveis. É possível movimentar o Mario para a direita ou esquerda, correr, saltar

e atirar bolas de fogo, quando o Mario está no modo fogo. Há ação de gravidade no jogo, o que faz com que seja necessário saltar sobre lacunas e outros desafios. Além de evitar as lacunas (Figura 2.5), é possível se movimentar em plataformas no decorrer do jogo, saltando entre elas.

No decorrer do jogo o Mario deve evitar ou matar os inimigos até alcançar o objetivo. Há vários tipos de inimigos. Cada um dos inimigos tem uma periculosidade diferente para o Mario. A Figura 2.1 mostra diferentes inimigos do Super Mario Bros presentes no Infinite Mario Bros. A seguir são listadas características dos inimigos de acordo com [Nintendo, 1985; Tsay et al., 2011]:

- **Goomba:** pequena e semelhante a um cogumelo marrom. Na história é considerada traidora do reino. Pode ser morta com um salto do Mario sobre ela, por bolas de fogo ou com o auxílio de um casco removido de uma Koopa. Veja Figura 2.1a.
- **Koopa Troopa:** são criaturas semelhantes a tartarugas. Podem ser verdes, denominadas Green Koopas (Figura 2.1c), ou vermelhas, chamadas de Red Koopas (Figura 2.1e). Green Koopas e Red Koopas possuem comportamentos diferentes. Quando estão em plataformas, por exemplo, as Greens Koopas descem das plataformas mais altas para as mais baixas. As Reds Koopas não ultrapassam o fim das plataformas. A ação do Mario saltar sobre elas, faz com liberem o casco como na Figura 2.2. Este pode ser usado como arma para matar outros inimigos. Entretanto, os cascos também podem machucar o Mario caso colidam contra ele. Assim como a goomba, podem ser mortas por bolas de fogo ou com o casco retirado delas.
- **Koopa Paratroopa:** é uma Koopa Troopa com asas e que pode voar - veja Figura 2.1d e Figura 2.1f. Saltar sobre elas com o Mario faz com que percam as asas e se tornem uma Koopa Troopa. Elas tem a mesma vulnerabilidade das Troopas, sendo resistentes a um salto a mais.
- **Spiny:** semelhante a uma tartaruga com espinhos no casco - veja Figura 2.1g. É uma Koopa quadrúpede. É resistente a saltos do Mario e a bolas de fogo. Podem ser mortos arremessando os cascos das Koopas contra eles.
- **Plantas Piranha:** na história vivem em vasos de flores que alimentam-se de homens. No jogo surgem dos tubos verdes no sentido vertical, de baixo para cima, e retornam ao tubo - veja Figura 2.1i. Podem ser mortas por bolas de

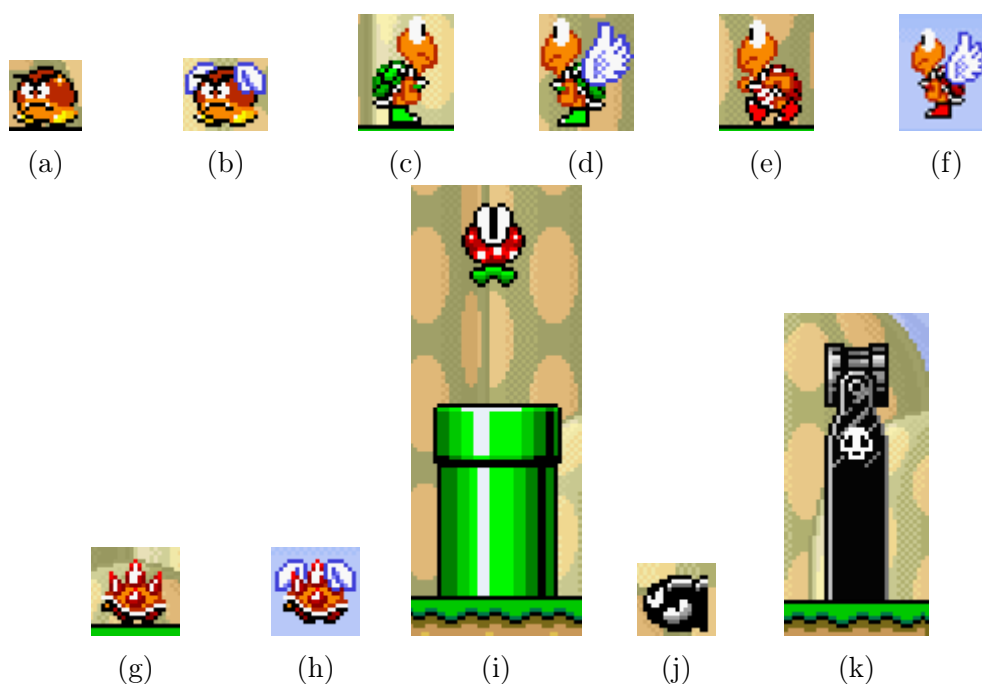


Figura 2.1: Inimigos presentes no Infinite Mario Bros.



Figura 2.2: Cascos das Koopas

fogo ou com cascos retirados das Koopas. Se o Mario estiver sobre o tubo quando a planta piranha está dentro dele, ela não sai.

- **Bullet Bill:** é um tiro de canhão - veja Figura 2.1j. Movimenta-se na horizontal no sentido do personagem e não é afetado pela gravidade. É imune contra bolas de fogo, mas podem ser mortas com um salto sobre elas ou com conchas.
- **Bill Blaster:** Um canhão indestrutível, que dispara uma quantidade infinita de Bullet Bills. Entretanto, não faz nada ao personagem mesmo este estando perto dele. Veja Figura 2.1k.

Além das Koopas, a goomba (Figura 2.1b) e o spiny (Figura 2.1h) tem uma versão alada. Assim como as Paratroopa, eles tem a mesma vulnerabilidade das versões não aladas.

No decorrer do jogo é possível encontrar com diversos itens que são exibidos na Figura 2.3. Dentre os itens estão:

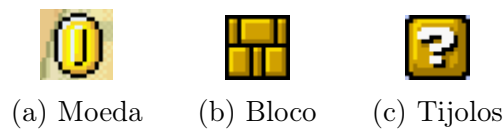


Figura 2.3: Itens do jogo.

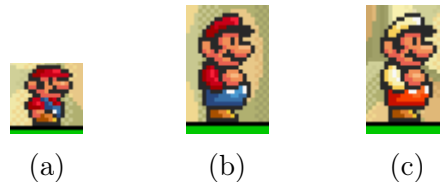


Figura 2.4: Estados ou Modos do Mario

- **Moedas:** adiciona pontos ao jogador. A cada cem moedas coletadas, o Mario ganha uma vida. Podem estar espalhadas a céu aberto ou escondidas dentro de blocos ou tijolos. Veja Figura 2.3a.
- **Blocos:** esses blocos sempre possuem um item escondido - veja Figura 2.3c. Podem ser moedas ou *power-ups*. Os itens irão aparecer quando o Mario salta de baixo para cima e bate com a cabeça contra o bloco ou quando arremessa um casco contra ele.
- **Tijolos:** esses tijolos podem ser quebrados pelo Mario quando está no modo Grande ou Fogo - veja Figura 2.3b. Algumas vezes pode conter *power-ups*.

O Mario pode estar em um dos três estados que são mostrados na Figura 2.4: Pequeno (no início de um jogo)(Figura 2.4a), Grande (pode esmagar alguns objetos usando a cabeça ou o salto) (Figura 2.4b), e Fogo (pode disparar bolas de fogo)(Figura 2.4c) [Shaker et al., 2012a; Karakovskiy & Togelius, 2012; Tsay et al., 2011].

Além do objetivo principal de chegar ao fim do nível, atravessando-o da esquerda para a direita, existem alguns objetivos auxiliares. Estes incluem fazer uma pontuação elevada, recolhendo o maior número possível de moedas que estão espalhadas em torno do nível, limpando-o o mais rápido possível, e matando o maior número possível de inimigos [Shaker et al., 2012a; Karakovskiy & Togelius, 2012].

A presença das lacunas, como a da Figura 2.5, e dos inimigos que se deslocam pelo nível, são os principais desafios do jogo. Se Mario cair em uma lacuna, ele perde uma vida. Se ele tocar um inimigo, sendo que este toque não é um salto sobre o inimigo, ele se machuca; isso significa perder uma vida se ele estiver no estado

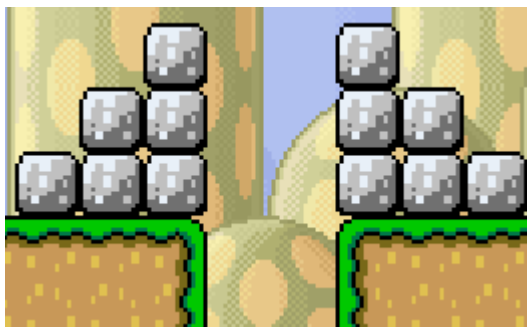


Figura 2.5: Imagem de uma lacuna ou buraco

(a) Cogumelo  
Mágico(b) Flor de  
Fogo

Figura 2.6: Power-Ups

Pequeno. Caso esteja no estado Grande, ele retorna para Pequeno e se estiver no estado Fogo se desloca para o estado Grande [Shaker et al., 2012a; Tsay et al., 2011].

Assim como pode se machucar, o Mario pode passar a um estado a frente. Isso é possível através de itens denominados *Power-Ups*. Estes power-ups podem ser o “Cogumelo de Energia“ (Figura 2.6a) ou a “Flor de fogo“ (Figura 2.6b) [Shaker et al., 2012a; Tsay et al., 2011].

No jogo original da Nintendo, o personagem é controlado com um D-pad (abreviação de *Directional Pad* - Controle Direcional) que permite movimentá-lo para a direita ou esquerda e para cima ou para baixo quando se faz necessário. Também é possível utilizar dois botões (A, B). O botão A permite dar saltos. A altura do salto é influenciada, em parte, pelo tempo que A é pressionado. O botão B permite ao Mario correr e se o Mario estiver em modo de Fogo, além de correr, ele atira uma bola de fogo.

Shaker et al. [2012a] afirma que não é possível expressar propriamente a jogabilidade de um jogo de forma textual. Assim como no trabalho de Shaker et al., essa seção descreve apenas algumas das principais regras e elementos de Super Mario Bros. Ainda segundo Shaker et al., o jogo original é um dos jogos mais vendidos do mundo, e ainda é jogado por milhares de fãs, há aproximadamente três décadas de seu lançamento. O design do jogo foi grandemente influente e inspirou inúmeros outros jogos, justificando ser uma boa plataforma para trabalhos relacionados.

Para se ter uma ideia da fama do Super Mario, em 28 de setembro de 2015 o sítio de notícias do [Yahoo, 2015] publicou um texto relacionado a um vídeo do

[Youtube, 2015] de um jovem japonês jogando um nível de Super Mario Bros. O vídeo mostra as ultimas tentativas do jovem de passar a fase *Pit of Panga: P-Break*, do Super Mario Maker. Super Mario Maker foi desenvolvido pela Nintendo [Nintendo, 2015], é um jogo que permite criar e jogar níveis customizados baseados no Super Mario Bros e seus sucessores. Segundo a notícia, *Pit of Panga: P-Break* é a fase mais difícil já criada no game. O desenvolvedor da fase demorou mais de cinco horas para projetá-la e cerca de dez horas para conseguir passar por ela. Na data da notícia, apenas seis pessoas haviam conseguido passar por esta fase e 467.326 pessoas já haviam tentado. O vídeo mostra manobras complexas que o japonês teve de fazer para conseguir terminar a fase. Ao final, ele mesmo parece não acreditar em seu triunfo.

Na próxima seção é feita uma breve descrição sobre computação humana.

## 2.5 Computação Humana

Há certos problemas que ainda não podem ser resolvidos por pessoas sem o auxílio de um computador. Outros são difíceis de serem resolvidos por computador sem o auxílio humano. A computação humana estuda formas de aproveitar o poder da combinação humano-computador para resolver esses tipos de problemas [Von Ahn, 2009].

Von Ahn [2009] cita o jogo ESP Game [Von Ahn & Dabbish, 2004] como um bom exemplo de utilização da computação humana, onde os jogadores ajudam determinar o conteúdo de imagens, fornecendo a elas rótulos significativos. Este é um jogo jogado online criado para ser jogado por um grande número de pessoas de uma vez. Ele é jogado em pares, ou seja, cada jogador tem seu parceiro. Os parceiros não são escolhidos, são distribuídos aleatoriamente entre todos jogadores. Não há comunicação entre os parceiros e não se sabe quem é o parceiro de quem durante o jogo. Para cada dupla são mostradas imagens e o objetivo é que o jogador tente adivinhar o que o parceiro está escrevendo para a imagem. Quando ambos digitarem a mesma sequência de caracteres para a imagem, eles avançam para a próxima imagem. A sequência que ambos jogadores concordam é considerada como um bom rótulo para a imagem [Von Ahn & Dabbish, 2004].

Segundo Von Ahn [2009] a abordagem utilizada é considerada incomum mas mostrou-se eficaz, pois incentiva as pessoas a fazerem o trabalho que seria feito por técnicas computacionais que ainda não funcionam muito bem. E a forma de recompensa dessas pessoas é o entretenimento. O ESP Game já coletou mais de 50

milhões de rótulos. A utilização de rótulos adequados para imagens na Web permite pesquisar com mais precisão imagens, melhorando a acessibilidade dos sítios web a indivíduos com deficiência visual. O Google licenciou o ESP game e passou a utilizá-lo para aperfeiçoar seu sistema de pesquisa de imagens.

A computação humana está associada a área da computação por máquina, desde de a década de 1950, quando iniciaram as tentativas de emular nas máquinas as capacidades humanas como linguagem, processamento e raciocínio. Apenas recentemente o foco a esta área tornou-se mais sério e atualmente tem tido o foco de pesquisadores de diversas áreas como Inteligência Artificial, Administração, Criptografia, Artes, Interface Homem Máquina, entre outras [Quinn & Bederson, 2011]. Em seu trabalho, Quinn & Bederson [2011] procuram dar uma definição concreta e um posicionamento das ideias e técnicas relacionadas a computação humana. Os autores ainda trazem uma breve descrição de termos como *crowdsourcing*, computação social, mineração de dados e inteligência cognitiva para auxiliar no entendimento dos limites da computação humana.

Não é um objetivo neste trabalho o desenvolvimento de um sistema capaz de adquirir conhecimento automaticamente. Como não é de interesse que o sistema tome decisões para classificar o conteúdo gerado quanto à qualidade, e sim que ele seja capaz de apresentar como saída conteúdo de boa qualidade, optou-se por utilizar computação humana ao invés de outras abordagens como Aprendizado de Máquina (AM). Optando pela computação humana, não há a necessidade de se preocupar em fornecer uma base de conhecimento - exemplos fornecidos ao sistema por um processo externo - geralmente utilizados em sistemas de AM.

Na próxima seção é feita uma breve descrição sobre arcos de tensão.

## 2.6 Arcos de Tensão

A fim de criar histórias interessantes, escritores procuram utilizar obstáculos entre o protagonista e seus objetivos. São estes obstáculos a fonte de tensão de uma história e esta geralmente é, cuidadosamente, controlada pelos autores. A medida que a trama se desenrola, eles aumentam ou diminuem a tensão de acordo com o que desejam que o público sintam. O nível de tensão de uma história em função do tempo é denominado arco de tensão. Apesar de existirem variações, o arco de tensão utilizado com maior frequência é composto por tensão crescente, clímax e desenlace [Barros & Musse, 2008].

Mateas e Stern [Mateas & Stern, 2003] procuraram gerar histórias consistentes

para o drama interativo *Façade*, procurando seguir um determinado arco de tensão.

Neste trabalho não há a criação de histórias. Entretanto, em uma parte deste, utiliza-se os obstáculos para controle da tensão nos níveis gerados. O objetivo é realizar um estudo do efeito da utilização dos arcos de tensão na construção do nível. Os obstáculos influenciam diretamente na dificuldade do nível gerado. Então nesta parte do trabalho a dificuldade do nível varia de acordo com o arco de tensão proposto.

O próximo capítulo é uma reprodução *ipsis litteris* de Reis et al. [2015].

## Capítulo 3

# Human Computation for Procedural Content Generation in Platform Games

### 3.1 Introduction

In procedural content generation (PCG) one is interested in using computer systems to automatically generate content for specific problem domains. For example, when applied to computer games, PCG systems automatically produce levels, rules, textures, and other contents traditionally generated by human professional designers. PCG in computer games has drawn a lot of attention in the recent years—see Togelius et al. [2011]; Hendrikx et al. [2013]; Shaker et al. [2015] for excellent surveys. One of the reasons PCG systems have attracted so much attention in the games community is that such systems can be used to reduce the cost and time required for producing computer games. Moreover, PCG can be a way of generating content tailored to specific players Shaker et al. [2010]; Smith et al. [2011a] and also to increase the replayability of the games Compton & Mateas [2006].

One of the major challenges in PCG is to automatically evaluate whether the generated content has good quality. In the context of computer games it is important to be able to evaluate whether the generated content is rated highly by users with respect to different measures such as aesthetics, enjoyment, and satisfaction. Researchers have aimed at understanding the concept of enjoyment to develop methods to automatically evaluate content in computer games. For example, Togelius et al. [2006] use a player's behavior to generate race tracks which

are more fun to the player; Liapis et al., Liapis et al. [2013] introduce general evaluation functions which are applicable to different games; Sorenson et al. Sorenson et al. [2011] learn a model of enjoyment based on levels generated by professional designers. Such works are usually motivated by the fact that it is not possible to have humans evaluating content produced by machines. For example, Shaker et al. Shaker et al. [2012a] stated that “because of the large amount of content that can be generated, it is not feasible to humanly judge the results, and automatic evaluation becomes a necessity”.

### 3.1.1 Our Contributions

In this paper we show that human computation Quinn & Bederson [2011] is a valid alternative to evaluate content generated by PCG systems for the game of *Infinite Mario Bros* (IMB) Togelius et al. [2013]. That is, we rely on human workers to measure whether particular content is of good or bad quality. Human evaluations can be quickly obtained in environments such as the Amazon Mechanical Turk (AMT) for a modest price. When using environments such as the AMT we do not need to ask (and perhaps bother) the player about their preferences as we can have human workers evaluating the content generated.

Our system uses human computation to quickly evaluate small portions of the game generated by an existing PCG system for IMB. In this paper we refer to such small portions of the game as *small levels*. Human workers provide annotations about the aesthetics, enjoyment, and difficulty of the small levels. Several such annotated levels are then combined into a full level of IMB (we also refer to the full levels as *larger levels*). The composition of the small levels into larger levels is done by accounting for the human-annotated information and the mathematical model of tension arcs used in interactive drama and storytelling Mateas & Stern [2003].

In this paper we perform two experiments. In the first experiment human workers annotated a collection of almost 2,000 small levels of the game of IMB. In the second experiment human subjects evaluated our proposed approach and other approaches encountered in the literature. Our quantitative and qualitative results on the second experiment show that the levels generated by our method can be more enjoyable to play and can have better aesthetics than the levels generated by other schemes. Our results show that the human-computation scheme is practical and thus a good alternative approach to generate good-quality content for the game of IMB.

## 3.2 Related Work

Here we describe how our work differentiates from a few other PCG works in platform games. For a thorough literature review we refer the reader to Togelius et al. [2011]; Hendrikx et al. [2013]; Shaker et al. [2015].

Smith et al. [2010] presented Tanagra, a system for developing levels for 2D platform games such as IMB. Tanagra allows the game designer to specify parts of the level and the system completes the level while respecting the designer’s decisions. Sorenson et al. [2011] presented a system which uses the idea of rhythm groups to define a computational model of player enjoyment to evolve levels of IMB.

Our idea of applying tension arcs to generate full levels of IMB is somewhat similar to the rhythm groups used by Sorenson et al. The main difference between our tension arcs and their rhythm groups is that the former are based on humanly-evaluated content.

Shaker et al. [2010] describe a system for generating player-specific content for IMB which directly asks questions to the players about their preferences. By contrast, as our system is not designed to generate player-specific content, we do not ask questions to players directly, we ask questions to human workers as a pre-processing step instead. In another work, Shaker et al. [2013] showed how to extract features to learn predictive models of the player’s experience in IMB. By contrast, we use the annotations provided by humans to directly generate levels of IMB—we assume that human annotations can be quickly obtained, therefore we do not learn a model to generalize the annotated data. Another difference between our work and that of Shaker et al. [2013] is that we introduce a novel method for connecting the small levels to generate a large level of the game of IMB.

The ideas and systems presented in all the works mentioned above are orthogonal to the ideas we introduce in this paper. That is, one could use any PCG system for IMB in conjunction with our system by having them generating the small levels which are then evaluated by human workers and combined by our tension arc-based system into a full level of the game.

The mathematical model of tension arcs we use for composing the IMB levels have been successfully used in interactive drama Mateas & Stern [2003] and storytelling Vogler [2007]. We show that such model can also be effective in PCG for platform games.

Recently, algorithms using human computation have become a good alternative for solving tasks which are hard for computers to solve. For example, reCAPT-

CHA Von Ahn et al. [2008] uses human computation to digitalize words which computer programs are not able to accurately recognize. In this paper we show that human computation is a viable approach for evaluating the content generated by computational intelligence methods.

### 3.3 The Problem Domain of Mario Bros

In this paper we are interested in the problem of automatically generating levels of the game of IMB, a game which has been used by several other researchers to evaluate PCG systems—for more details on the use of IMB in research please refer to the work of Togelius et al. [2009]. The advantage of using IMB in our experiments is that we are able to compare the quality of the content generated by our system with that of other systems found in the literature.<sup>1</sup>

A screenshot of IMB is shown in Figure 3.1. The player controls Mario (on the center of screen). Mario’s goal is to reach the rightmost spot of the level. In order to succeed, Mario has to avoid enemies such turtles and shooting canons. The levels of IMB are grid spaces containing a set of objects such as mountains, shooting cannons, turtles, pits, boxes, and others. Figure 3.1 depicts part of a level which contains four mountains of different widths and heights, several enemies, and a few boxes which Mario can break to collect power-up items. Every object is associated with a location on the grid ( $x$  and  $y$  coordinates) and some of the objects such as mountains and pits can have different heights and widths—boxes, a few enemies, and the small version of Mario himself occupy a single cell on the grid. In this paper all full levels are represented as a grid of size  $160 \times 15$ .

Let  $\mathbf{L} = \{o_1, o_2, \dots, o_n\}$  be a level of IMB where  $o_1, o_2, \dots, o_n$  are the  $n$  objects composing such level. The PCG problem for IMB is to choose the set of objects in  $\mathbf{L}$  as well as the objects’  $x$  and  $y$  coordinates. For some of the objects such as pits and mountains we also need to define their height and width values. Our goal is to generate a level  $\mathbf{L}$  which is both visually appealing and enjoyable to play.

### 3.4 The Human Computation Approach to PCG

In this section we describe our approach for automatically generating levels of IMB. We call our system Human-Computation Tension Arc-Based (HCTA) level generator. A high-level description of HCTA is as follows.

---

<sup>1</sup>The source code IMG is kept by others in the following address <http://www.marioai.org>.

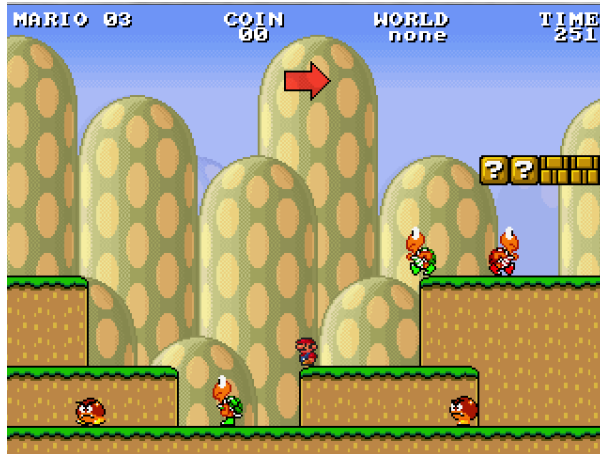


Figura 3.1: Screen shot of the game of Infinite Mario Bros.

1. A PCG system generates a collection  $\Gamma$  of levels of IMB with grid size of  $20 \times 15$ .
2. Workers annotate each  $l \in \Gamma$  with respect to three measures: enjoyment, aesthetics, and difficulty.
3. Different levels of  $\Gamma$  are concatenated to form a full level of larger size.

The central idea behind HCTA is to have human workers quickly annotating a large number of small levels (of size  $20 \times 15$ ) of IMB. Then, such small levels are combined in different ways to generate an even larger number of different IMB levels of larger size. Note that HCTA does not assume that the human workers are professional game designers. HCTA uses the annotation provided by anyone able to play IMB.

A large number of annotated small levels can be quickly obtained for a modest cost in environments such as AMT. Another option is to make our system available online and ask for volunteers to annotate the levels. In this paper we use the latter. HCTA assumes that the workers' perceived aesthetics and enjoyment on the small levels will be similar to the those of the people playing the larger level. Also, while we recognize that different players might enjoy different styles of gameplay, we expect HCTA to produce levels with good aesthetics and which are enjoyable to play on average. Another important observation is that HCTA does not disrupt the gameplay as it asks questions to human workers as a preprocessing step and not to the players. We believe that asking questions to the players could break their gameplay immersion.

In the next sections we describe each of the steps of HCTA mentioned above. First, we describe the basic PCG system used to generate the library  $\Gamma$ , then the experiment in which volunteers annotated the small level in  $\Gamma$ , and finally we explain the method we introduce to concatenate small levels into full-sized levels of IMB.

### 3.5 Basic System for Level Generation

The system we use for generating  $\Gamma$  is the generator which comes in the game of IMB. In the literature this generator is referred as the Notch Level Generator (which we abbreviate as NLG) after the game designer Markus “Notch” Persson.

NLG receives as input a difficulty value  $d$  for stochastically determining the number of enemies, pits and other challenges to be placed in the level. That is, the levels NLG generates will tend to be harder for larger values of  $d$ . NLG starts with an empty level—in our case an empty grid of size  $20 \times 15$ —and it iteratively adds objects to the grid according to the value of  $d$ . NLG follows simple heuristics for adding objects to the levels. For example, when adding a hill, NLG limits the hill’s height to a value that Mario is able to reach by jumping from the ground or from some other hill. As another example, when adding a shooting cannon, NLG places the object on a hill or on the ground of the level so that the level contains no aloft cannons.

We use NLG to generate more than 2,000 levels of size  $20 \times 15$  with values of  $d$  selected uniformly at random to ensure a collection of levels  $\Gamma$  with different difficulty levels.

### 3.6 Human-Annotated Levels

The NLG system follows a set of hardcoded rules to stochastically create levels of IMB. Although simple, the levels NLG generates can have good aesthetics and be enjoyable to play. Unfortunately, however, due to the stochasticity of the generator and the lack of a systematic evaluation, NLG also produces levels which are not visually appealing and are not necessarily enjoyable to play.

In HCTA human workers play all levels in  $\Gamma$  and annotate each of the levels with respect to three measures: enjoyment, aesthetics, and difficulty. We note that the value of  $d$  provided by the NLG system offers a good indication of the player’s perceived difficulty of the level. However, we noticed in preliminary experiments that sometimes a level  $l$  has a large number of enemies and challenges (determined

by a large value of  $d$ ) but  $l$  is not necessarily a difficult level to play. This happens because there could be alternative paths the player can choose in order to avoid facing all challenges posed by the level. Thus, workers also evaluated the difficulty of the levels in  $\Gamma$ .

We made our system available for download and invited undergraduate and graduate students in the Departamento de Informática, at Universidade Federal de Viçosa, in Brazil to play the levels in  $\Gamma$ . The students voluntarily and anonymously played the levels. Before playing the levels the volunteers were instructed that the levels played would be much smaller than regular levels of the game of IMB. We assumed that most of the volunteers had played the game of Mario before and we did not want them to be disappointed by the reduced size of the levels. After playing each level the volunteers provided a score from 1 to 9 to each of the following criteria.

1. Enjoyment.
2. Aesthetics.
3. Difficulty.

A score of 1 for enjoyment, aesthetics, and difficulty means that the level is not enjoyable, is not visually pleasing, and offers no challenge to the player, respectively. Similarly, a score of 9 for enjoyment, aesthetics, and difficulty means that the level is very enjoyable, is very visually appealing, and is extremely difficult, respectively.

The order in which the volunteers annotated the levels in  $\Gamma$  was random, and every level  $l$  in  $\Gamma$  could be annotated only once by each volunteer. By using such an approach the volunteers annotated most of the levels in  $\Gamma$ , and some of the levels were annotated more than once by different volunteers. We use the average score given by different volunteers in case a level was annotated more than once. We removed from  $\Gamma$  the levels that were not annotated by any volunteer.

Since the levels could be played in only a few seconds, a few minutes of work was worth several annotated levels. We counted 1.928 annotated levels in our server 30 days after we advertised our system in our department's mailing list. We conjecture that this many annotated levels could be obtained much more quickly if the system was deployed for evaluation at AMT.

### 3.6.1 Representative Annotated Levels

We now present a few annotated levels from our experiment with human workers. First we show the scores of a few representative levels (quantitative results), then

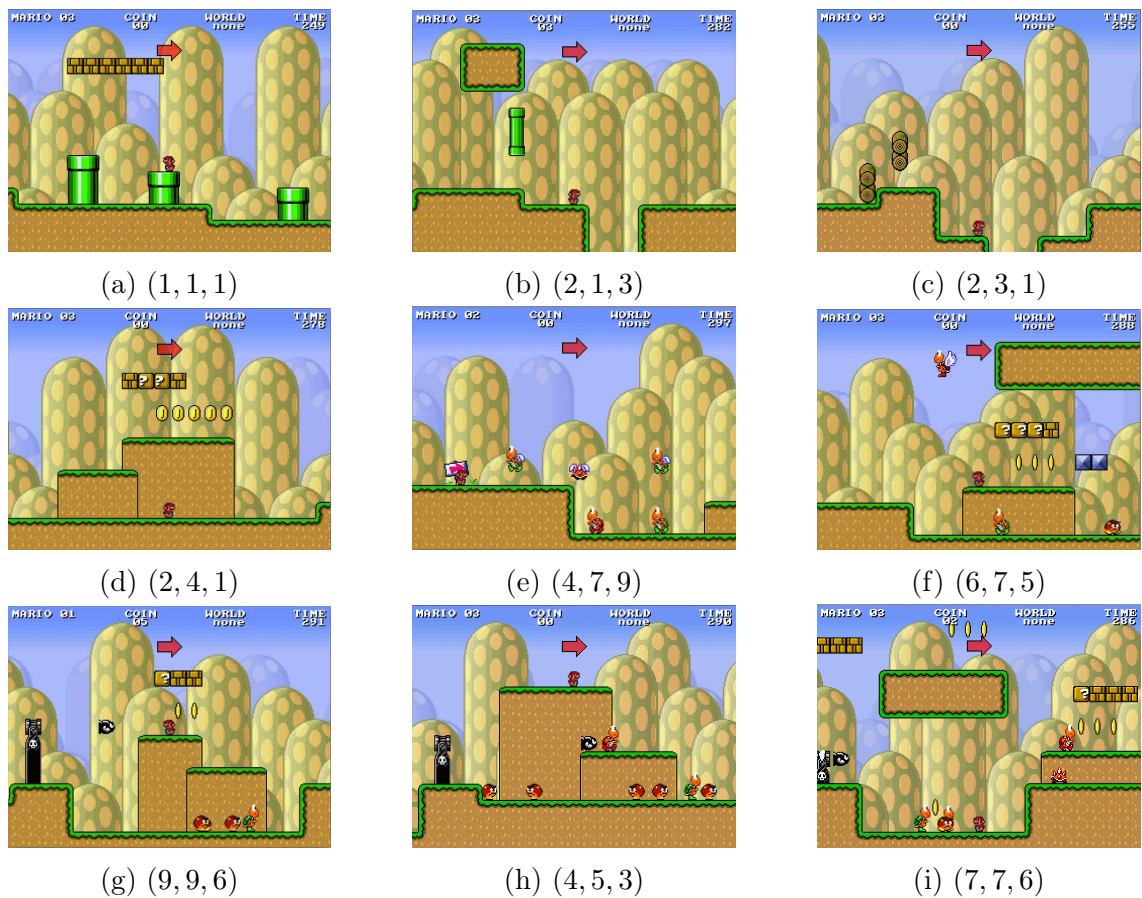


Figure 3.2: A few representative human evaluations for levels in  $\Gamma$ . The numbers in parenthesis show the value of enjoyment, aesthetics, and difficulty, respectively in a scale from 1 to 9. For example, a level with values (1, 1, 1) mean that the level is not challenging, it has bad aesthetics, and it is not enjoyable.

we report a few written comments provided by some of the workers (qualitative results).

### 3.6.1.1 Quantitative Results

Figure 3.2 shows a few representative human annotated levels from our experiment. The numbers in parenthesis show the value in a scale from 1 to 9 of enjoyment, aesthetics, and difficulty, respectively. While the evaluation is subjective and the volunteers do not explain the scores provided, we now try to interpret the evaluation scores of the levels shown in Figure 3.2.

The level shown in Figure 3.2 (a) shows a level with no challenge—there are no enemies or pits. Moreover, the objects are oddly placed on the screen. That is, the platform of blocks on the top of the screen has no purpose since it is not reachable by Mario. These reasons support the score of (1, 1, 1) provided by the worker to the

level.

The levels shown in (b) and (c) are similar to each other: both have a pit and thus offer some challenge to the player. However, the volunteer who annotated (c) found the level very easy and marked its difficulty as a 1, while the volunteer who annotated (b) decided to give a 3 to the level. It is expected to see some variance on the scores provided by the workers. However, we note that the scores for difficulty for both (b) and (c) are somewhat similar. We also notice in these two levels that the score of aesthetics is 1 for (b) and 3 for (c). This is justified by the unreachable mountain above the green pipe in (b) which deteriorates the aesthetics of the level. The level shown in (d) has a better aesthetics score than (a), (b), and (c), which is reasonable since all objects are well distributed on the screen and are all reachable by Mario.

The level shown in (e) is considered more difficult (difficulty of 9) than level (h) (difficulty of 3), despite the fact that (h) has more enemies than (e). This is because in (h) the player has an alternative path going through the top of the mountain while in (e) the player must act quickly before being caught by three flying enemies. Similarly, (f) is considered more difficult than (h) because the flying turtle comes from the top of the mountain after Mario while in (h) the player can safely stay on the top of the mountain. However, (f) is considered easier than (e) probably because of the reduced number of enemies and because in (f) Mario can hide from the flying turtle below the blocks in the center of the level. These examples clearly illustrate the fact that simply counting the number of enemies on the screen can be ineffective in determining the actual difficulty of the level.

Finally, levels (g) and (i) pose interesting challenges which could make the levels more enjoyable to play. In (g) Mario has to dodge bombs while crossing the level and collecting the power-up item available in the question-mark block on the center of the level; in (i) Mario has to face several enemies to reach the other side of the screen.

Interestingly, we noticed a high positive correlation between enjoyment and aesthetics in the levels annotated by the volunteers—coefficient of 0.72 amongst all evaluations. This result suggests an interesting relation of aesthetics and enjoyment: levels with well-placed objects tend to be more enjoyable to play. Enjoyment and difficulty are also correlated (coefficient of 0.67). The relation between difficulty and enjoyment is well known. Namely, Piselli et al. [2009] showed that the Yerkes-Dodson law Yerkes & Dodson [1908] applies to computer games in the sense that pleasure will be maximum somewhere in between the largest and the smallest challenge. That is, enjoyment increases with difficulty up to some point, where the

level gets too difficult to be enjoyable.

The smallest correlation coefficient is between aesthetics and difficulty (coefficient of 0.47). Clearly these two metrics can be very uncorrelated. For example, there can be levels which are too difficult due to a large number of enemies and also have bad aesthetics due to the poor placement of objects on the screen.

### 3.6.1.2 Qualitative Results

We also allowed the volunteers to optionally enter comments on the levels played. One of the volunteers stated that the levels were too small to be enjoyable, specifically they wrote: “*After playing 5 levels I noticed that they were too short to be fun*”. Such a comment is an indication that the enjoyment on the small level could differ from the enjoyment on the same level on a larger context. Nonetheless, other volunteers apparently enjoyed playing some of the small levels as they wrote comments such as “*Perfect!*” and “*This level is very good!*”.

## 3.6.2 Relation of Level Size and Enjoyment

The qualitative results shown above can be worrisome as it shows that people’s enjoyment while playing the small levels (*Fun-Small*) could be different from people’s enjoyment while playing the larger levels (*Fun-Large*). One solution for making *Fun-Small* being closer to *Fun-Large* is by increasing the size of the small levels. For example, instead of using levels of size  $20 \times 15$ , one could use levels of size  $40 \times 15$ , which are closer to the actual full level size. However, using larger levels in the human computation process will imply not only in an increase in the time required to evaluate each individual level, but also in an increase on the number of levels evaluated in order to have a collection of levels with good quality, as we now explain.

Let  $p$  be the probability of the NLG system generating a small level of size  $20 \times 15$  with good aesthetics and which is enjoyable to play. In order to have one level of such size with good aesthetics and enjoyment in our collection of levels, one has to generate and humanly evaluate  $\frac{1}{p}$  levels on average. If the evaluated levels are twice as large (size of  $40 \times 15$ ), then NLG has to generate two small levels of size  $20 \times 15$  in a row with good aesthetics and enjoyment. It is reasonable to assume that NLG generates such a level with probability of  $p^2$ . Thus, one has to generate approximately  $\frac{1}{p^2}$  levels of size  $40 \times 15$  on average to have a single small level with good aesthetics and enjoyment in our collection.

To illustrate the reasoning above, let  $p = 0.01$ . On average we would need to generate 100 levels of size  $20 \times 15$  to have one good level in our collection. By contrast, we would need approximately 10,000 levels on average to have one good level of size  $40 \times 15$  in our collection—we would have to evaluate two orders of magnitude more levels when doubling the size of the levels. We believe there is a compromise between the relation of *Fun-Small* and *Fun-Large* and the number of levels the workers have to evaluate to create a collection of levels containing good-quality levels. We believe that the size of  $20 \times 15$  offers a good tradeoff between the human computation effort and relation of *Fun-Small* with *Fun-Large*. Previous works also used levels of size similar to our small levels. For example, the rhythm groups in Sorenson et al.’s system Sorenson et al. [2011] and Tanagara’s beats Smith et al. [2010] have sizes similar to our small levels.

### 3.7 Tension-Arc Based Small Level Combination

The human computation procedure described above outputs a collection of small annotated levels. In this section we describe how HCTA combines such levels into a full IMB level. For doing so we borrow the idea of tension arcs used in interactive drama [Mateas & Stern, 2003] and story writing [Vogler, 2007]. Vogler [Vogler, 2007] argues that in storytelling, either in books, movies or TV shows, the story usually follows a pattern in which the tension builds up to a climax and it drops before concluding. In HCTA we use similar idea to try to control the player’s tension throughout a full level of IMB as we now explain.

**L**

---

#### Algorithm 1 Tension-Arc Concatenation

---

**Require:** Collection of annotated small levels  $\Gamma$ , tension arc  $T = \{d_1, d_2, \dots, d_M\}$ , quality parameter  $k$ .

**Ensure:**  $\mathbf{L} = \{l_1, l_2, \dots, l_M\}$

- 1: **for**  $i = 1$  to  $M$  **do**
  - 2:   choose at random a small level  $l$  in  $\Gamma$  with difficulty  $d_i$  which is at same time in the set of  $k$  levels with highest aesthetics score and in the set of  $k$  levels with highest enjoyment score.
  - 3:   append  $l$  to  $\mathbf{L}$
  - 4: **end for**
- 

Let a  $\mathbf{L} = \{l_1, l_2, \dots, l_M\}$  be a totally ordered set with  $\mathbf{L} \subseteq \Gamma$  and each level  $l \in \mathbf{L}$  with size  $x \times y$ .  $\mathbf{L}$  represents a level of size  $x \cdot M \times y$  formed by the

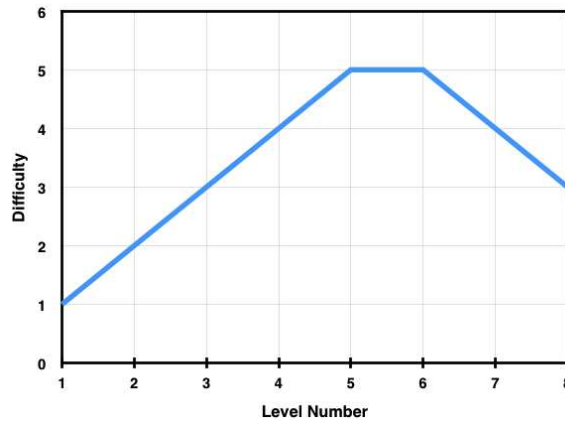


Figure 3.3: Tension arc used in our experiments. The difficulty values are provided by human annotated levels, and the values on the  $x$ -axis denote the tension arc’s ordering.

concatenation of the levels  $l \in \mathbf{L}$  according to some ordering  $\mathbf{O}$ .  $\mathbf{L}$  is the final output of HCTA.

In HCTA a *tension arc* defines the ordering  $\mathbf{O}$ . A tension arc  $T$  is a sequence of difficulty values  $\{d_1, d_2, \dots, d_M\}$  where  $d_1$  is the difficulty of the first small level composing  $\mathbf{L}$ ,  $d_2$  is the difficulty of the second small level, and so on. Figure 3.3 shows a tension arc as a function of the levels in  $\mathbf{L}$  and their difficulty value. The numbers on the  $x$ -axis from left to right define the ordering  $\mathbf{O}$  of the small levels composing  $\mathbf{L}$ . Such a tension arc follows the idea of building up the tension by increasing the difficulty of the level as the player evolves into the game, until reaching the climax (small levels 5 and 6 in  $\mathbf{L}$ ) and dropping on small levels 7 and 8.

For a given collection of annotated levels  $\Gamma$ , tension arc  $T$  and integer  $k$ , HCTA builds  $\mathbf{L}$  as shown in Algorithm 1. HCTA selects a level  $l$  in  $\Gamma$  for each difficulty value  $d$  in  $T$  and appends  $l$  to  $\mathbf{L}$ . The parameter  $k$  controls  $\mathbf{L}$ ’s quality by allowing HCTA to select only the levels amongst the  $k$  with highest scores for aesthetics and enjoyment. For small values of  $k$   $\mathbf{L}$  will be composed only by small levels from  $\Gamma$  with the highest human perceived aesthetics and enjoyment, but the levels produced by HCTA might be repetitive—larger values of  $k$  allow for less repetition but for a likely loss in content quality. In our experiments we use  $k = 50$ .

One can easily notice that our proposed approach can be implemented in ways different than the one we describe in this paper. For example, instead of using a tension arc with growing tension, one could try to use a constant difficult value throughout, or even random values. Moreover, instead of using NLG to generate the collection  $\Gamma$  one could use any other PCG system. Also, instead of using human

computation, one could also use different measures (e.g., the metrics introduced in Sorenson et al. [2011]; Horn et al. [2014]). Evaluating some of these variations with human subjects is an interesting direction of future work.

## 3.8 Empirical Evaluation

In this section we present our empirical evaluation of the HCTA system with human subjects. This section is organized as follows. In Section 3.8.1 we detail the methodology used in our experiments, in Section 3.8.3 we present the quantitative results which are discussed in Section 3.8.4. Finally, in Section 3.8.5 we present a few qualitative results.

### 3.8.1 Methodology

#### 3.8.1.1 Systems Tested

We evaluate four different systems: HCTA with the tension arc shown in Figure 3.3 (HCTA+P, where the P stands for “parabolic”, the shape of the tension arc), HCTA with a random tension arc (HCTA+R), the Occupancy-Regulated Extension generator (ORE) Mawhorter & Mateas [2010] which was the winner of the 2011 Mario AI Competition, and NLG. We wish we could evaluate more systems in our experiment (e.g., other participants in the Mario AI Competition and HCTA with different tension arcs), but that would substantially increase the time required to run the experiment and would also require a larger number of subjects. We chose to use ORE because it was the winner of the 2011 competition and the system code was available online.<sup>2</sup>

#### 3.8.1.2 Evaluated Metrics

The systems were evaluated according to the following criteria: enjoyment, aesthetics, difficulty, and Turing. The Turing criterion was meant to measure whether the participant thought that the level played was designed by a human or machine. In the beginning of the experiment we stated that the participants could be playing levels generated either by humans or by machines.

Each participant was asked to answer how much they agreed or disagreed, in a 7-likert scale, with following statements.

---

<sup>2</sup>see <http://www.marioai.org/LevelGeneration> for details.

1. This level is enjoyable to play.
2. This level has good aesthetics.
3. This level is difficult.
4. This level was developed by a machine.

A score of 1 for enjoyment and aesthetics mean that the participant strongly agrees that the level played is enjoyable and has excellent aesthetics; a score of 1 for the Turing criterion means that the participant strongly agrees that the level was designed by a machine; finally, a score of 1 for difficulty means that the participant strongly agrees that the level is difficult.

### 3.8.1.3 Participants

Our within-subject experiment had 34 participants: 30 males and 4 females with an average age of 23.73 and standard deviation of 4.31. The experiment was carried out online. Namely, we made our system available in the Internet and advertised our experiment in different mailing lists. Participation was anonymous and volunteered.

### 3.8.1.4 Experimental Design

In the beginning of the experiment the subjects filled a questionnaire informing their age, and their skills in the game of Infinite Mario (i.e., how much Mario they played before). Subjects were instructed about the controls of the game before playing a practice level. The practice level is important for participants to get acquainted with the keyboard control as most of them had played the Nintendo version of the game in a joystick. The practice level was generated by the NLG system. Only after playing the practice level that the participants evaluated the levels generated by the PCG systems. After playing each level the participants gave scores according to the criteria described above in a 7-likert scale. In addition to the scores, the participants had the option to enter comments justifying their scores, informing us of technical issues they might have had, or making general suggestions on the experiment. Since all participants played levels generated by the four tested systems, we used a balanced Latin square design to counteract ordering effects.

In order to have a fair comparison of the levels generated by different systems we had all systems generating levels of the same size:  $160 \times 15$ . We chose such size because we did not want the experiment to be too long. In total each participant

played 5 levels (1 practice level and 4 other levels for evaluation), and we could not afford creating larger levels as it could be tiring for the participants.

Moreover, we controlled the systems HCTA+P, HCTA+R, and NLG to generate levels with difficulty similar to the levels generated by ORE. This was done in the HCTA approaches by bounding the difficulty value used in the tension arcs, and in the NLG approach by bounding the value of  $d$ . We evaluate difficulty in this experiment only to make sure we were able to control such variable on the levels tested.

### 3.8.1.5 Data Cleaning

The data provided by the participants who were not able to play all levels in our experiment is not included in our results. We also removed the data of a single participant who had never played the game of Mario before. By examining the logs of the experiment we noticed that this participant was not able to get too far into the game and thus not able to properly evaluate the levels. The number of 34 participants was obtained after removing such data.

## 3.8.2 Hypotheses

We are interested in testing if the HTCA approach is able to generate levels with good aesthetics and that are enjoyable to play. We also want to test if the participants are tricked into thinking that the levels the HTCA approach generates are produced by human designers. Finally, we are also interested in testing the effect of different tension arcs in HCTA.

Specifically, we test the following hypotheses:

**H1** On average, the HCTA+P and HCTA+R approaches generate levels which are more enjoyable to play than the levels generated by the other approaches tested.

**H2** On average, the HCTA+P and HCTA+R approaches generate levels with better aesthetics than the levels generated by the other approaches tested.

**H3** On average, the HCTA+P approach is better at tricking players into thinking that the levels were produced by humans.

**H4** Different tension arcs can influence how much enjoyment the player has while playing levels generated by HCTA.

Tabela 3.1: Empirical evaluation of different PCG systems. Lower values of enjoyment and aesthetics indicate levels which are more enjoyable to play and have better aesthetics; larger values of Turing indicate levels which participants were more prone to believe that were generated by humans.

	HCTA+P	HCTA+R	ORE	NLG
Enjoyment	2.45 +/- 1.87	2.91 +/- 2.04	3.54 +/- 1.93	2.82 +/- 1.98
Aesthetics	2.48 +/- 1.77	2.42 +/- 1.93	3.60 +/- 1.65	3.02 +/- 2.19
Turing	3.42 +/- 2.03	3.22 +/- 2.13	2.71 +/- 2.09	3.17 +/- 2.32
Difficulty	3.45 +/- 1.69	3.22 +/- 2.28	3.08 +/- 1.57	3.71 +/- 1.72

### 3.8.3 Quantitative Results

The mean results +/- the standard deviations of our experiment are shown in Table 3.1. Shapiro-Wilk tests show that our data is not normally distributed ( $p < .0001$  for all criteria). Therefore, we use the non-parametric Friedman test which shows a significant difference on enjoyment ( $\chi^2(3)=8.18$ ,  $p < .05$ ) and on aesthetics ( $\chi^2(3)=9.18$ ,  $p < .05$ ) across different systems; there was no statistical significance for Turing and difficulty. The small difference in the difficulty score and the lack of statistical significance are evidences that difficulty was indeed controlled in our experiment, allowing a fair comparison of the different approaches. We now turn to post-hoc tests (Wilcoxon signed-rank) of the systems with respect to enjoyment and aesthetics.

#### 3.8.3.1 Pairwise Comparison on Enjoyment

HCTA+P generates levels which are significantly more enjoyable to play than the levels HCTA+R generates ( $p < .05$ ) and the levels that ORE generates ( $p < .005$ ). The levels HCTA+R generates are significantly more enjoyable to play than the ones ORE generates ( $p < .05$ ). Finally, the levels NLG generates are significantly more enjoyable to play than the ones ORE generates ( $p < .05$ ).

#### 3.8.3.2 Pairwise Comparison on Aesthetics

HCTA+P generates levels with significantly better aesthetics than the levels ORE generates ( $p < .05$ ) and than the levels NLG generates ( $p < .05$ ). Also, HCTA+R generates levels with significantly better aesthetics than the levels ORE generates ( $p < .01$ ) and than the levels NLG generates ( $p < .05$ ).

### 3.8.4 Discussion

In this section we discuss the results presented in Table 3.1. Our discussion follows the order of the four hypotheses we are interested in testing in our experiment.

#### 3.8.4.1 Testing H1

In general we observe that the participants enjoyed playing the levels generated by all systems. For example, a score of two and three for enjoyment means that the participant agrees and somewhat agrees, respectively, that the level is enjoyable to play. HCTA+P was the system which generated levels which the participants enjoyed playing the most, with a score of 2.45. ORE was the system which generated levels which participants enjoyed playing the least. The average score for enjoyment for ORE was of 3.54, which is the closest to the score of 4 which means that the participant neither agrees nor disagrees with the statement that the level is enjoyable to play.

The levels HCTA+P generates are significantly more enjoyable to play than the levels generated by HCTA+R and ORE. The difference between HCTA+P and NLG was not significant in our experiment. These results partially support **H1** in the sense that both HCTA+P and HCTA+R generated levels which were more enjoyable to play than those generated by ORE, but no statistical difference was observed with respect to the levels NLG generates. We conjecture that the issue regarding the level size and enjoyment discussed in Section 3.6.2 played a role in the results.

#### 3.8.4.2 Testing H2

Similar to the results on enjoyment, in general the participants liked the aesthetics of the levels generated by all systems. HCTA+R was the system with best score (2.42) and HCTA+P was the second best (2.48). Both ORE and NLG had an average score above 3. Both HCTA systems generated levels with significantly better aesthetics than ORE and NLG. These results support **H2** in that the HCTA approaches are able to generate levels with better aesthetics.

#### 3.8.4.3 Testing H3

The system which performed best in tricking people into thinking that the level was developed by a human designer was HCTA+P, followed by HCTA+R, NLG, and

ORE. However, the average score of 3.42 for HCTA+P means that people somewhat agree that the level was developed by a machine. Our results do not support **H3**.

#### 3.8.4.4 Testing H4

Enjoyment was the only criterion in which there was a significant difference between the HCTA approaches. HCTA+P was slightly better than HCTA+R in the Turing criterion and it produced levels which were slightly easier than HCTA+R. However, such results were not statistically significant. The average score of aesthetics for both systems is nearly the same. We conjecture that the perceived aesthetics depends only on what the player sees on the screen at a given time. This is in contrast with enjoyment, where there could be a relation with the level's structure as discussed in Section 3.6.2.

It is interesting to observe that the tension arc shape can influence the enjoyment the players experience. While it is standard to have tension arcs depicting growing tension in storytelling Vogler [2007], it was not immediately clear to us that such a tension arc could also have a positive impact in IMB. This result fully supports **H4**. While the tension arc shown in Figure 3.3 performed better than the random tension arc, we do not know whether there are better tension arcs to be used to generate levels in platform games or even if the tension arc shape should be player-dependent. Investigating these issues are interesting directions of future work.

#### 3.8.5 Qualitative Results

Written evaluation was optional in our experiment and only a few participants entered comments about the levels played. Here we highlight a few of such comments.

A participant noticed the carry-on effect of playing multiple levels: *“this level seems to be easier than the first I played, maybe it is because I am getting used to the game”*. Although the ordering effect exists, as mentioned by the participant, an indication that our balanced Latin square design counteracted such effects is the lack of statistical significance in the difficulty scores of the levels generated by the systems.

Another participant wrote about a level generated by the ORE approach: *“The randomness of how the objects are placed on the screen make me believe this level was generated by a machine.”* The same participant wrote about the same level: *“The aesthetics isn't good because there are objects in places where they aren't needed.”* A level generated by ORE also received a compliment: *“If this level was developed*

*by a machine, then this is really good!*”. Another participant wrote about a level generated by HCTA+P: *“Despite the large number of enemies, I believe that the structure of the level was built by a human designer”*.

### 3.9 Conclusions

In this paper we introduced HCTA, a PCG system for the game of IMB which uses human computation to evaluate the content generated. HCTA uses an existing PCG system to generate a large number of small levels which are evaluated by human workers. Then, HCTA uses the mathematical model of tension arcs to combine a number of such small annotated levels into a larger level of the game. We also performed a systematic experiment with human subjects to evaluate levels generated by the proposed approach and also by different systems. Our results showed that (i) the levels generated by the HCTA approaches had better aesthetics than the levels generated by all other schemes tested on the average case; (ii) the levels generated by the HCTA approaches were more enjoyable to play than one of the other schemes tested on the average case—the other HCTA comparisons on enjoyment were not statistically significant.

Our results suggest that the human-in-the-loop approach is feasible for the task of PCG in platform games and that such an approach can produce content of good quality. Finally, we believe that the HCTA approach is general to be applied to other Mario-like platform games such as Sonic the Hedgehog.

## Capítulo 4

# Considerações Finais

Este trabalho apresentou um gerador automático de fases para o jogo de plataforma 2D *Infinite Mario Bros* (IMB). Este gerador utiliza o conceito de computação humana para geração de níveis para o IMB. Como contribuições científicas, além de um novo tipo de gerador, foi possível observar que a avaliação humana pode também ser utilizada para auxiliar sistemas de geração de conteúdo de jogos de plataforma 2D.

O trabalho foi desenvolvido em duas partes. A primeira consistiu na criação e avaliação dos pequenos níveis. A criação dos pequenos níveis se deu através de sistemas existentes na literatura, enquanto a avaliação dos mesmos foi através da computação humana. A segunda parte consistiu em concatenar os pequenos níveis através do modelo matemático de arcos de tensão, usado para controlar o nível de dificuldade das fases geradas. Finalmente, realizou-se uma comparação detalhada de duas versões do método proposto com outros dois geradores de conteúdo para o mesmo jogo encontrados na literatura. Uma das versões do sistema proposto utilizou um arco de tensão aleatório, onde a dificuldade da fase variava de forma aleatória à medida que o jogador progredia no jogo. A segunda versão do sistema proposto utilizou um arco de tensão inspirado em conceitos de narrativa de estórias. O objetivo do arco de tensão inspirado em narrativa de estórias foi fazer com que a tensão do jogador aumentasse à medida que ele progredisse no jogo.

Os níveis gerados pelo sistema que utiliza computação humana e o arco de tensão inspirado em narrativa de estórias, obteve, no geral, resultados mais positivos que os demais sistemas avaliados. Com base nos resultados uma questão pode ser levantada: qual a melhor curva de dificuldade para os jogos? Esta é uma questão que pode ser respondida em trabalhos futuros.

# Referências Bibliográficas

- Bakkes, S.; Whiteson, S.; Li, G.; Visniuc, G. V.; Charitos, E.; Heijne, N. & Swellegrebel, A. (2014). Challenge balancing for personalised game spaces. In *Games Media Entertainment*, pp. 1--8. IEEE Press.
- Barros, L. M. & Musse, S. R. (2008). Towards consistency in interactive storytelling: Tension arcs and dead-ends. *Computers in Entertainment (CIE)*, 6(3):43.
- Canossa, A. & Smith, G. (2015). Towards a procedural evaluation technique: Metrics for level design. In *FDG*. ACM.
- Compton, K. & Mateas, M. (2006). Procedural level design for platform games. In *Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp. 109--111. AAAI.
- Dahlskog, S. & Togelius, J. (2013). Patterns as objectives for level generation. In *Proceedings of the Workshop on Design Patterns in Games at FDG*.
- Dahlskog, S. & Togelius, J. (2014a). A multi-level level generator. In *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, pp. 1--8.
- Dahlskog, S. & Togelius, J. (2014b). Procedural content generation using patterns as objectives. In *Proceedings of the European Conference Applications of Evolutionary Computation*, pp. 325--336.
- Dahlskog, S.; Togelius, J. & Nelson, M. J. (2014). Linear levels through n-grams. In *Proceedings of the International Academic MindTrek Conference*.
- Hendriks, M.; Meijer, S.; Van Der Velden, J. & Iosup, A. (2013). Procedural content generation for games: A survey. *ACM Transactions Multimedia Computing, Communications. Applications*, 9(1):1:1--1:22.
- Horn, B.; Dahlskog, S.; Shaker, N.; Smith, G. & Togelius, J. (2014). A comparative evaluation of level generators in the mario ai framework. In *FDG*. ACM.

- Karakovskiy, S. & Togelius, J. (2012). The mario ai benchmark and competitions. *Computational Intelligence and AI in Games, IEEE Transactions on*, 4(1):55–67.
- Kerssemakers, M.; Tuxen, J.; Togelius, J. & Yannakakis, G. N. (2012). A procedural procedural level generator generator. In *Conference of Comp. Intell. and Games*, pp. 335–341. IEEE.
- Liapis, A.; Yannakakis, G. N. & Togelius, J. (2013). Towards a generic method of evaluating game levels. In *Conference on Artificial Intelligence and Interactive Digital Entertainment*.
- Mariño, J. R. H.; Reis, W. M. P. & Lelis, L. H. S. (2015). An empirical evaluation of evaluation metrics of procedurally generated mario levels. In *Conference on Artificial Intelligence and Interactive Digital Entertainment*. AAAI Press.
- Mateas, M. & Stern, A. (2003). Façade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference*.
- Mawhorter, P. A. & Mateas, M. (2010). Procedural level generation using occupancy-regulated extension. In *Computational Intelligence and Games (CIG)*, pp. 351–358. IEEE.
- Nintendo (1985). Super mario bros.
- Nintendo (2015). Super mario maker. Disponível em <http://www.nintendo.com/games/detail/i7YeZxnWhEFX6ST-sLdUBGmLoZivzNP5>, acessado 11-Setembro-2015.
- Pedersen, C.; Togelius, J. & Yannakakis, G. N. (2009). Modeling player experience in Super Mario Bros. In *Computational Intelligence and Games (CIG)*, pp. 132–139. IEEE Press.
- Piselli, P.; Claypool, M. & Doyle, J. (2009). Relating cognitive models of computer games to user evaluations of entertainment. In Whitehead, J. & Young, R. M., editores, *FDG*, pp. 153–160. ACM.
- Quinn, A. J. & Bederson, B. B. (2011). Human computation: A survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1403–1412. ACM.
- Reis, W. M. P.; Lelis, L. H. S. & Gal, Y. (2015). Human computation for procedural content generation in platform games. In *Computational Intelligence and Games (CIG)*. IEEE.

- Shaker, N. & Abou-Zleikha, M. (2014). Alone we can do so little, together we can do so much: A combinatorial approach for generating game content. In *Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp. 167--173. AAAI Press.
- Shaker, N.; Nicolau, M.; Yannakakis, G. N.; Togelius, J. & O'Neill, M. (2012a). Evolving levels for super mario bros using grammatical evolution. In *Computational Intelligence and Games (CIG)*, pp. 304--311. IEEE.
- Shaker, N.; Togelius, J. & Nelson, M. J. (2015). *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer.
- Shaker, N.; Yannakakis, G. & Togelius, J. (2013). Crowdsourcing the aesthetics of platform games. *IEEE Transactions on Computational Intelligence and AI in Games*, 5(3):276--290.
- Shaker, N.; Yannakakis, G. N. & Togelius, J. (2010). Towards automatic personalized content generation for platform games. In *Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp. 63--68. AAAI Press.
- Shaker, N.; Yannakakis, G. N. & Togelius, J. (2011). Feature analysis for modeling game content quality. In *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, pp. 126--133. IEEE.
- Shaker, N.; Yannakakis, G. N. & Togelius, J. (2012b). Digging deeper into platform game level design: Session size and sequential features. In *EvoApplications*, volume 7248 of *Lecture Notes in Computer Science*. Springer.
- Shaker, N.; Yannakakis, G. N.; Togelius, J.; Nicolau, M. & O'Neill, M. (2012c). Evolving personalized content for Super Mario Bros using grammatical evolution. In *Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp. 75--80. AAAI Press.
- Smith, A. M.; Lewis, C.; Hullet, K. & Sullivan, A. (2011a). An inclusive view of player modeling. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pp. 301--303, New York, NY, USA. ACM.
- Smith, G.; Cha, M. & Whitehead, J. (2008). A framework for analysis of 2d platformer levels. In *ACM SIGGRAPH Symposium on Video Games*, pp. 75--80. ACM.

- Smith, G.; Treanor, M.; Whitehead, J.; Mateas, M.; Treanor, M.; March, J. & Cha, M. (2011b). Launchpad: A rhythm-based level generation for 2d platformers. *IEEE Transactions on Computing Intelligence and AI in Games*, 3(1):1–16.
- Smith, G. & Whitehead, J. (2010). Analyzing the expressive range of a level generator. In *Proceedings of the Workshop on Procedural Content Generation in Games*, pp. 1–7. ACM.
- Smith, G.; Whitehead, J. & Mateas, M. (2010). Tanagra: a mixed-initiative level design tool. In *FDG*, pp. 209–216. ACM.
- Sorenson, N.; Pasquier, P. & DiPaola, S. (2011). A generic approach to challenge modeling for the procedural creation of video game levels. *IEEE Transactions on Computing Intelligence and AI in Games*, 3(3):229–244.
- Togelius, J.; De Nardi, R. & Lucas, S. M. (2006). Making racing fun through player modeling and track evolution. In *Proceedings of the SAB Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games*.
- Togelius, J.; Karakovskiy, S.; Koutnik, J. & Schmidhuber, J. (2009). Super mario evolution. In *Computational Intelligence and Games (CIG)*, pp. 156–161. IEEE.
- Togelius, J.; Shaker, N.; Karakovskiy, S. & Yannakakis, G. N. (2013). The mario AI championship 2009–2012. *AI Magazine*, 3(34):89–92.
- Togelius, J.; Shaker, N. & Nelson, M. J. (2014). Introduction. In Shaker, N.; Togelius, J. & Nelson, M. J., editores, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer.
- Togelius, J.; Yannakakis, G. N.; Stanley, K. O. & Browne, C. (2011). Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):172–186.
- Tsay, J.-J.; Chen, C.-C. & Hsu, J.-J. (2011). Evolving intelligent mario controller by reinforcement learning. In *Technologies and Applications of Artificial Intelligence (TAAI), 2011 International Conference on*, pp. 266–272. IEEE.
- Vogler, C. (2007). *The Writer's Journey: Mythic Structure For Writers*. Michael Wiese Productions, United States.
- Von Ahn, L. (2009). Human computation. In *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*, pp. 418–419. IEEE.

- Von Ahn, L. & Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 319--326. ACM.
- Von Ahn, L.; Maurer, B.; McMillen, C.; Abraham, D. & Blum, M. (2008). re-captcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465--1468.
- Yahoo (2015). Japonês completa nível mais difícil do game super mario e a sua reação é absurda. Disponível em <https://br.noticias.yahoo.com/blogs/super-incrível/japonês-completa-nível-mais-difícil-do-game-super-mario-e-a-sua-reação-é-absurda-150250022.html>, acessado 28-Setembro-2015.
- Yerkes, R. M. & Dodson, J. D. (1908). The relation of strength of stimulus to rapidity of habit formation. *Journal of Comparative Neurology and Psychology*, 18:459--482.
- Youtube (2015). Man collapses into tears of joy after competing one of the hardest super mario levels. Disponível em <http://www.nintendo.com/games/detail/i7YeZxnWhEFX6ST-sLdUBGmLoZivzNP5>, acessado 28-Setembro-2015.

# APÊNDICE A

## Os sistemas utilizados para coleta de dados

Neste capítulo são descritos os sistemas utilizados para coleta de dados durante este trabalho.

A proposta apresentada neste trabalho, denominada **HCTA** (sigla do inglês *Human-Computation Tension Arc-Based*), foi um gerador de níveis. O caso de estudo utilizado foi a geração de níveis para o *Infinite Mario Bros*. Como descrito no Capítulo 3, o trabalho foi realizado em duas partes que podem ser representadas pela Figura A.1.

Na primeira parte, avaliou-se o conteúdo dos pequenos níveis gerados. Enquanto na segunda parte, o objetivo foi a avaliação dos níveis completos e a comparação com geradores da literatura. Nas seções a seguir, são exibidas capturas de telas dos sistemas utilizados para coleta dos dados.

### A.1 Avaliação dos pequenos níveis

Para a primeira etapa da coleta de dados, foi elaborado um sistema que possuía a finalidade de coletar informações de jogadores voluntários em relação a pequenas fases geradas por um gerador já existente.

A Figura A.2 é a primeira tela do sistema. A segunda tela, exibida na Figura A.3 mostra as teclas a serem utilizadas para jogar. A terceira tela, mostrada na Figura A.4 destaca os três quesitos que deverão ser avaliados. As telas seguintes, mostradas nas Figuras A.5, A.6, A.7, A.8, A.9, A.10, A.11 são exemplos de avaliações dos quesitos de diversão, visual e dificuldade de alguns pequenos níveis. A ideia principal dessas telas é tentar explicar os quesitos de avaliação. Após as

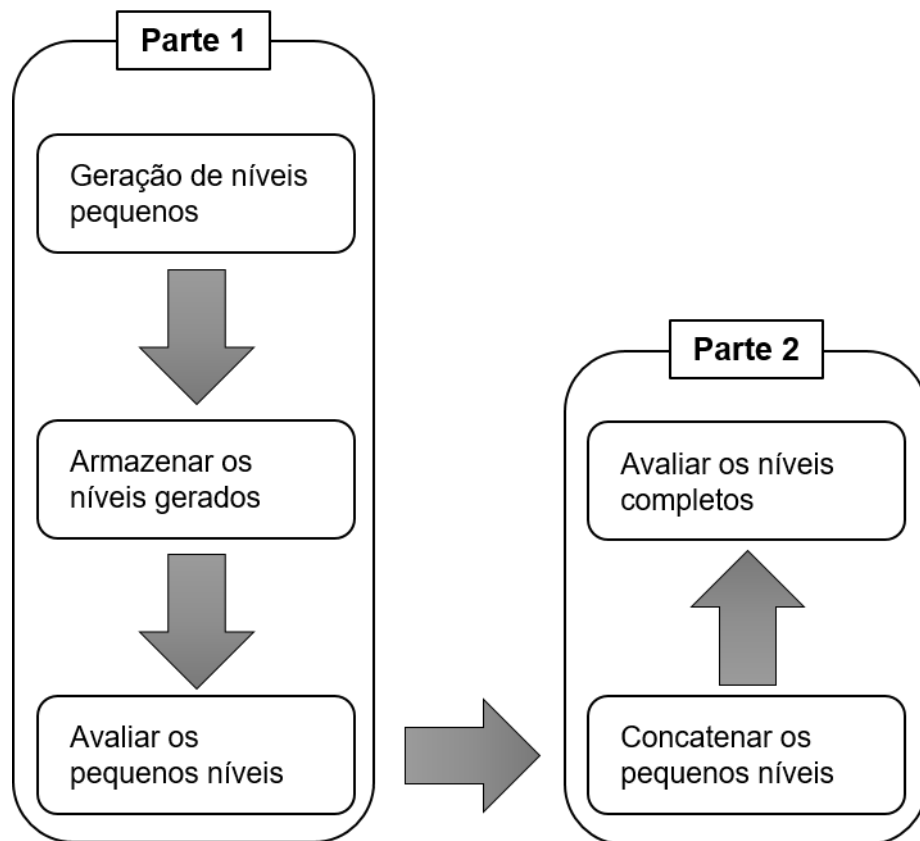


Figura A.1: Fluxograma HCTA

instruções é mostrada uma tela que explica o *ranking* utilizado neste sistema - veja Figura A.12. Este *ranking* é utilizado com intuito de despertar a competição nos jogadores, a fim de incentivá-los a jogar. Após as instruções é aberta uma tela de pré-jogo, onde são mostrados os quesitos de avaliação e existe a opção de reler as instruções ou começar a jogar; esta tela pode ser vista na Figura A.13. Caso o jogador saia do jogo e volte a jogar novamente, ele não terá de passar, obrigatoriamente, por todas as telas de instruções. Ao clicar em "jogar", o jogo irá iniciar, conforme a Figura A.14. Ao terminar uma fase é aberta a tela de avaliação onde também é exibido o *ranking* - veja Figura A.15.

## A.2 Avaliação dos níveis completos

Finalizada a avaliação das fases pequenas, a segunda etapa do experimento foi de geração de fases completas e comparação das mesmas com fases geradas por outros geradores encontrados na literatura.

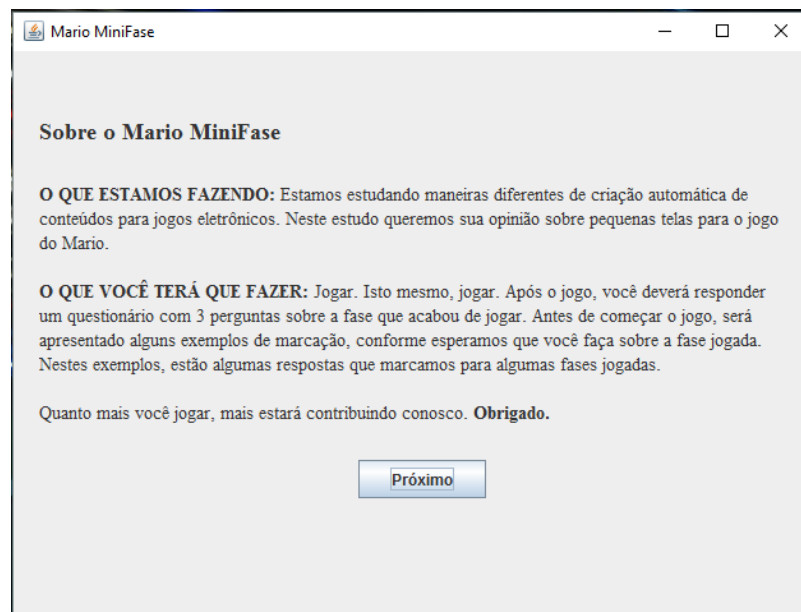


Figura A.2: Tela sobre o sistema.

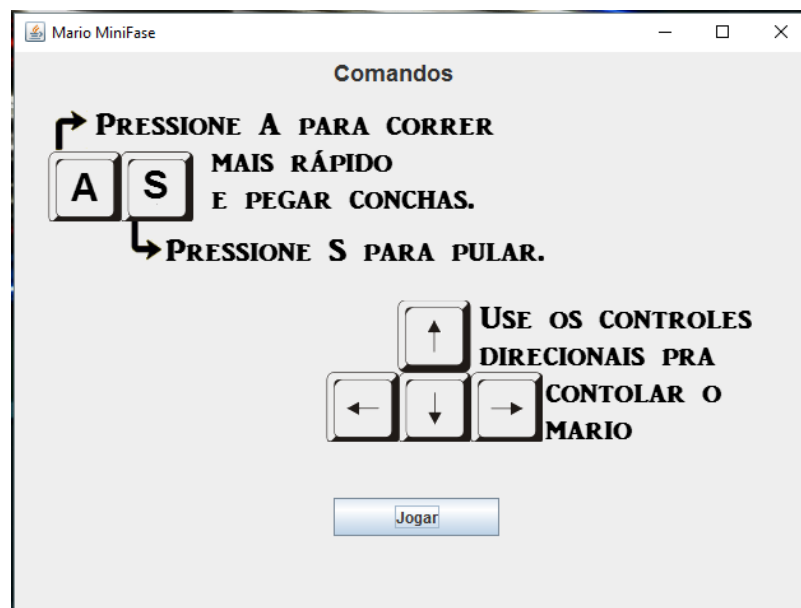


Figura A.3: Tela de comandos.

A Figura A.16 é a primeira tela do sistema, onde são coletadas algumas informações básicas dos jogadores. Essas informações não são suficientes para identificar o participante. A segunda tela, exibida na Figura A.17 mostra as teclas a serem utilizadas para jogar. A terceira tela, mostrada na Figura A.18, é um nível utilizado para que os jogadores possam treinar os controles. Este nível não é utilizado na avaliação. Na terceira tela há uma mensagem em destacando que as próximas fases serão avaliadas, conforme pode-se observar na Figura A.19. Clicando em jogar é

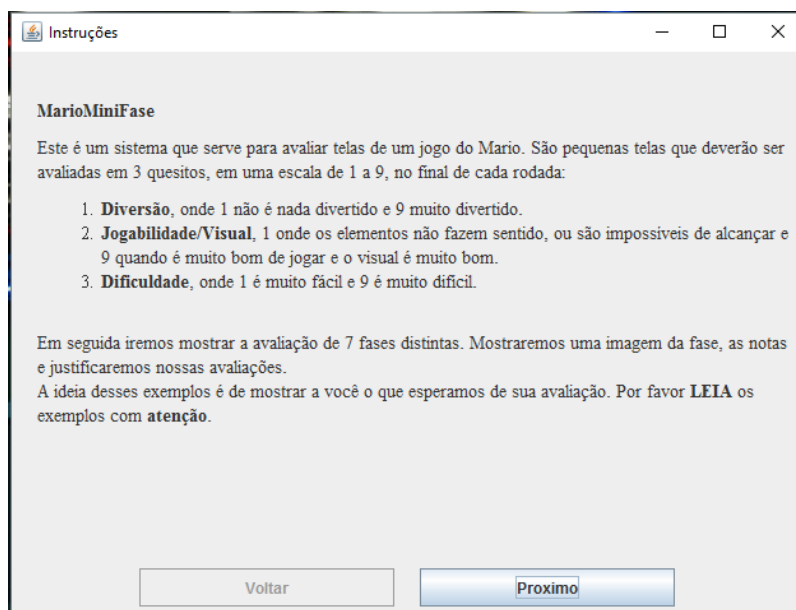


Figura A.4: Tela de instruções nº 1.



Figura A.5: Tela de instruções nº 2.

aberta a tela do jogo, onde o nível jogado foi gerado por algum gerador de conteúdo avaliado neste trabalho - veja Figura A.20. Após terminar de jogar o nível é aberta a tela onde o jogador irá avaliar o nível jogado - veja Figura A.21. A tela do jogo e a tela de avaliação são exibidas quatro vezes, uma para cada gerador avaliado. Após as quatro avaliações é aberta uma tela, como a da Figura A.22, onde o jogador pode escrever observações sobre o experimento.



Figura A.6: Tela de instruções nº 3.



Figura A.7: Tela de instruções nº 4.



Figura A.8: Tela de instruções nº 5.

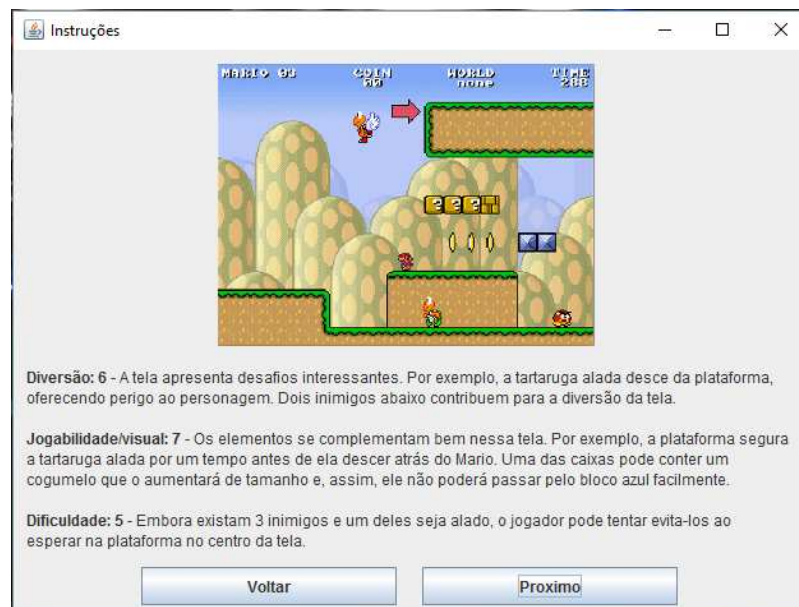


Figura A.9: Tela de instruções nº 6.



Figura A.10: Tela de instruções nº 7.

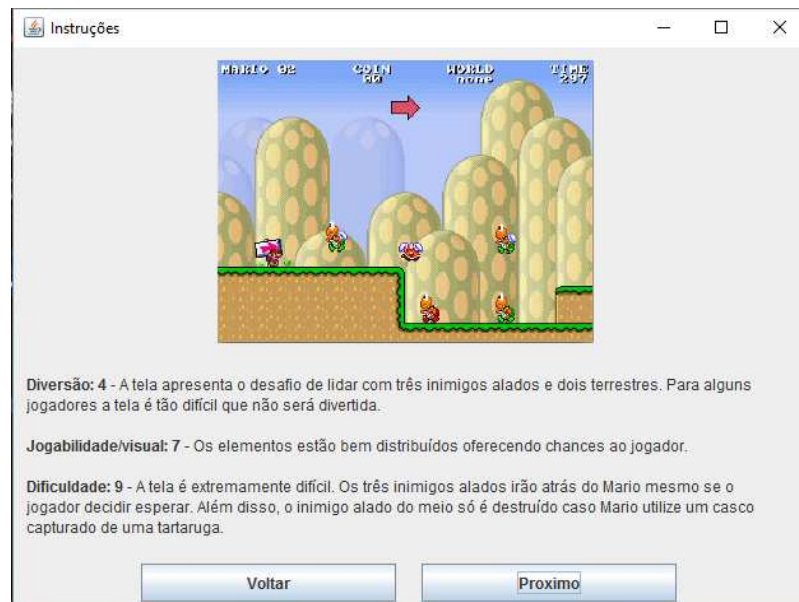


Figura A.11: Tela de instruções nº 8.

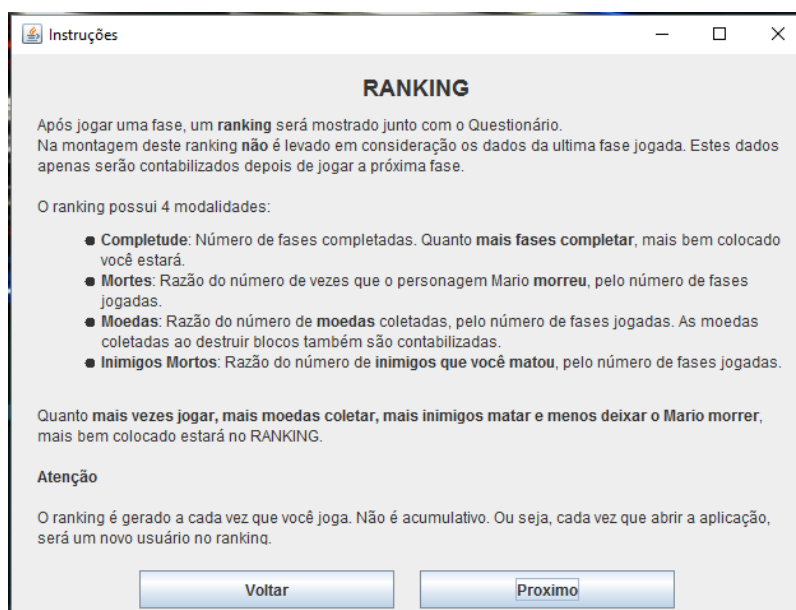


Figura A.12: Tela com informações sobre o ranking.

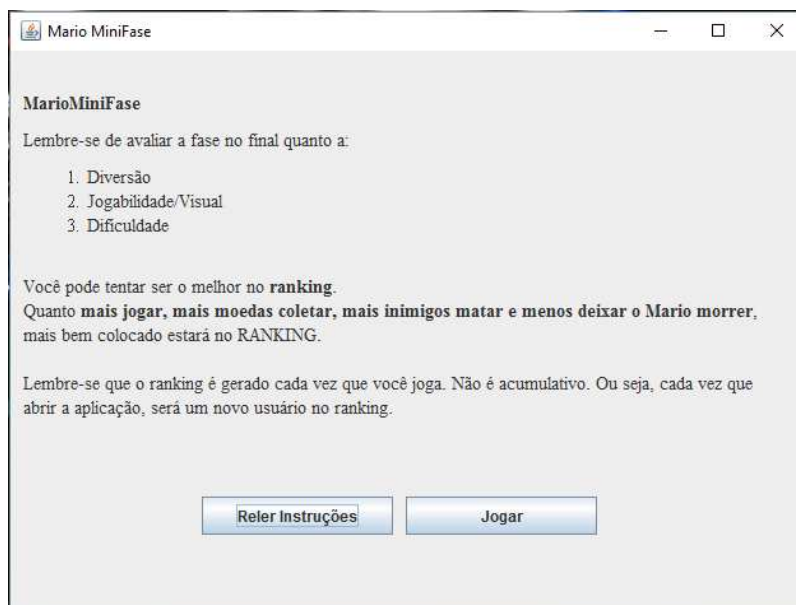


Figura A.13: Tela do pré-jogo.



Figura A.14: Tela do jogo.

Mario MiniFase

De 1 a 9, avalie a tela quanto as questões abaixo:

Diversão.....  1  2  3  4  5  6  7  8  9

Jogabilidade/Visual.....  1  2  3  4  5  6  7  8  9

Dificuldade.....  1  2  3  4  5  6  7  8  9

Sugestões/Opiniões (opcional):

### Rankings

Até o momento, 147 pessoas já jogaram.

**Completo:** Você é o 139º colocado. Já jogou 7 fases e o 1º jogou 34 fases.

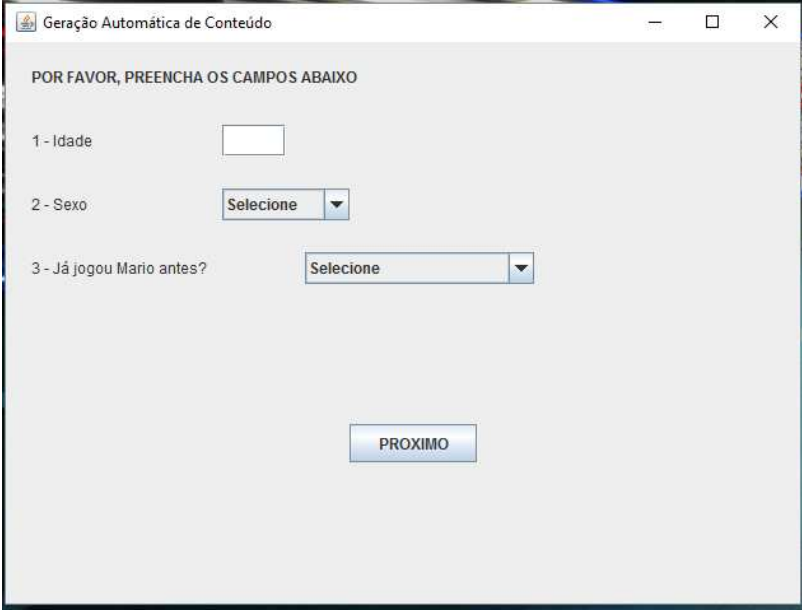
**Mortes:** Você é o 145º colocado. Você morreu 12 vezes. O 1º morreu 7 vezes em 25 fases.

**Moedas:** Você é o 119º colocado. Você coletou 29 moedas. O 1º coletou 100 em 12.

**Inimigos Mortos:** Você é o 97º colocado. Você matou 14 inimigos. O 1º colocado matou 43 em 11 fases.

[Jogar proxima tela](#)

Figura A.15: Tela de Avaliação.



Geração Automática de Conteúdo

POR FAVOR, PREENCHA OS CAMPOS ABAIXO

1 - Idade

2 - Sexo  ▼

3 - Já jogou Mario antes?  ▼

Figura A.16: Tela de solicitação de algumas informações sobre o jogador.

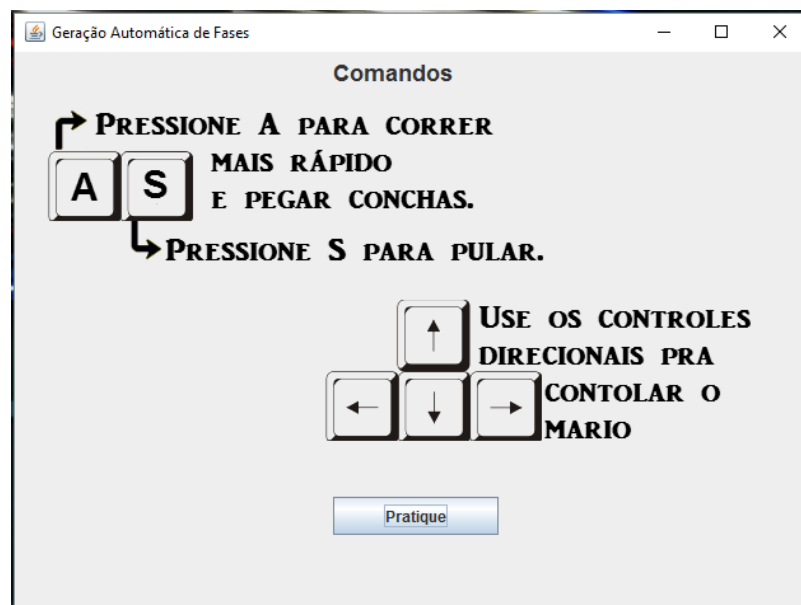


Figura A.17: Tela exibindo as teclas a serem utilizadas para jogar.

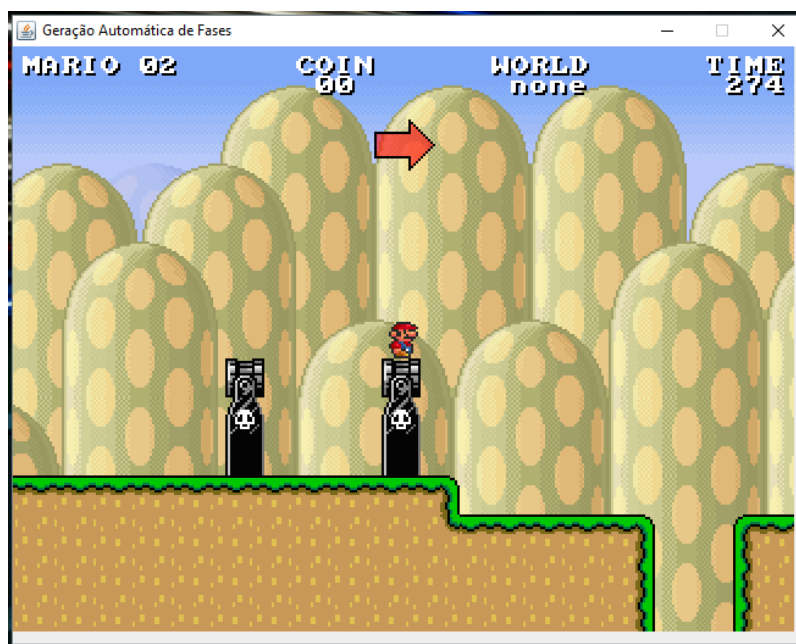


Figura A.18: Pequeno nível para que os jogadores possam treinar os controles.

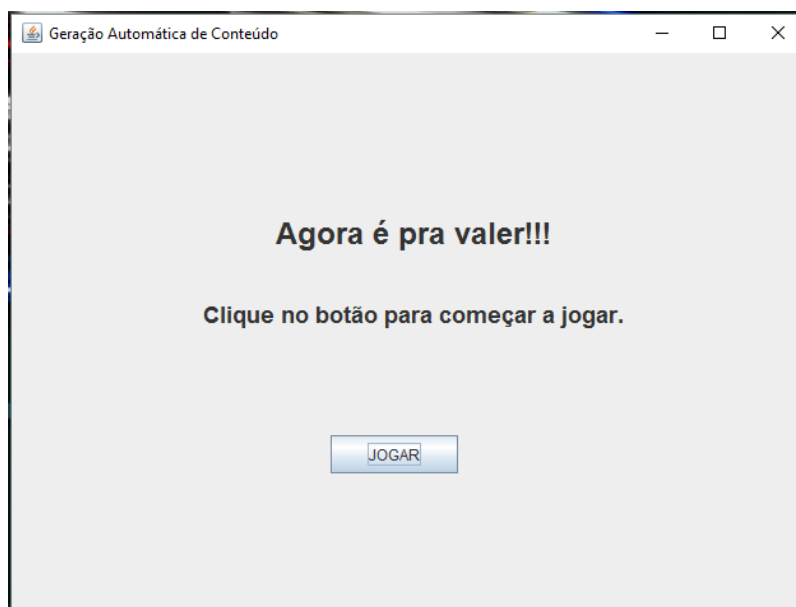


Figura A.19: Tela pré-jogo.



Figura A.20: Tela com o nível gerado por um dos geradores avaliados neste trabalho.

A screenshot of a survey form titled "Geração Automática de Conteúdo". It asks for ratings from 1 to 7 for four statements, with optional justification boxes and a "Continuar Jogando" button.

Dê uma nota de 1 a 7 para cada afirmação, onde:  
(1) Concordo totalmente - (2) Concordo - (3) Concordo fracamente - (4) Não concordo nem discordo -  
(5) Discordo fracamente - (6) Discordo - (7) Discordo totalmente

Gostei de jogar esta fase      Concordo totalmente  1  2  3  4  5  6  7      Discordo totalmente  
Justifique (opcional)

Esta fase é difícil      Concordo totalmente  1  2  3  4  5  6  7      Discordo totalmente  
Justifique (opcional)

Esta fase tem um bom visual (estética)      Concordo totalmente  1  2  3  4  5  6  7      Discordo totalmente  
Justifique (opcional)

Esta fase foi desenvolvida por um programa de computador      Concordo totalmente  1  2  3  4  5  6  7      Discordo totalmente  
Justifique (opcional)

Figura A.21: Tela para avaliar o nível jogado.

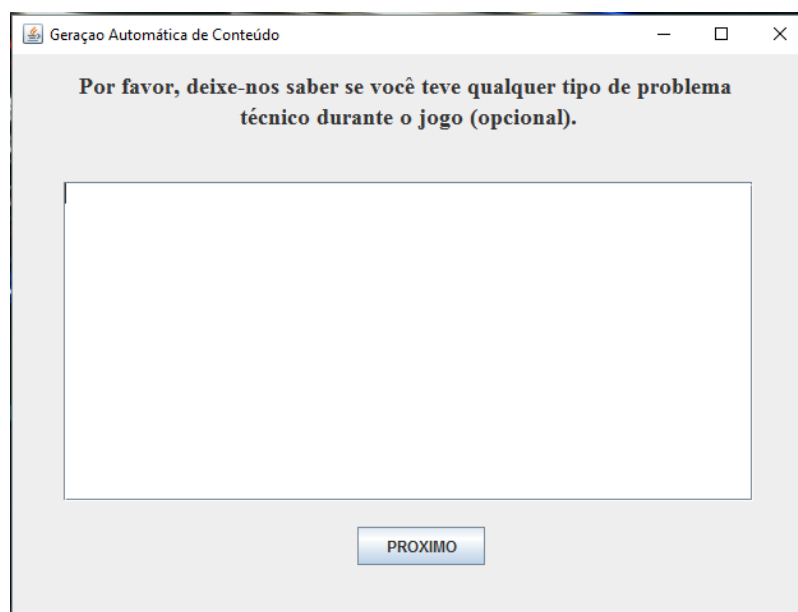


Figura A.22: Tela final que permite ao jogador fazer comentários sobre o experimento.