



MODELO DE DESENVOLVIMENTO DE JOGOS COM PROPÓSITO BASEADO
EM MECÂNICAS TRADICIONAIS DE JOGOS

Arthur Siqueira de Lima

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Geraldo Bonorino Xexéo

Rio de Janeiro

Agosto de 2013

MODELO DE DESENVOLVIMENTO DE JOGOS COM PROPÓSITO BASEADO
EM MECÂNICAS TRADICIONAIS DE JOGOS

Arthur Siqueira de Lima

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof.^o Geraldo Bonorino Xexéo, D.Sc.

Prof.^o Jano Moreira de Souza, Ph.D.

Prof.^a Esteban Walter Gonzalez Clua, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

AGOSTO DE 2013

Lima, Arthur Siqueira de

Modelo de desenvolvimento de jogos com propósito baseado em mecânicas tradicionais de jogos/ Arthur Siqueira de Lima. – Rio de Janeiro: UFRJ/COPPE, 2013.

XII, 93 p.: il.; 29,7 cm.

Orientador: Geraldo Bonorino Xexéo

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2013.

Referências Bibliográficas: p. 90-93.

1. Computação humana. 2. Jogos com Propósito. 3. Mecânicas de jogo. I. Xexéo, Geraldo Bonorino. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III Título.

À meus pais, meus modelos de pessoa.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Cleber e Laís, por tudo que fizeram por mim. Pela educação que me deram e formação que me proporcionaram. Pelo apoio e amor incondicional. Muitos dos valores que marcam a pessoa que sou hoje devo à eles.

Agradeço, também, toda minha família, destacando minha irmã Lílian, meu afilhado Matheus e tia Adriana, pessoas importantes que estiveram presentes comigo nesse período. Minha vó Maria e meu vô Francisco que sempre me perguntava sobre o andamento dos meus estudos.

Agradeço o professor Geraldo Xexéo pela orientação, por ter aceitado a proposta de pesquisa que apresentei a ele e pela a liberdade para realizá-la. Isso contribuiu para que o desenvolvimento da pesquisa fosse conduzido de uma forma tranquila.

Aos meus amigos que fizeram parte dessa história desde a graduação, Amanda, Bianca, Bernardo Araújo, Cláudio, Daniel, Gustavo, “Lond”, Marcelo Monteiro, Marcos, Perrotta. Sem esquecer daqueles que, como eu, encararam o mestrado, Allan “Big”, André “Dedé”, Bernardo “Meu Nobre”, Diego, Filipe, Roque, Marcelo “Super” e Marcelo “Shiryu” e aqueles que desistiram também, “Pedrinho”. Um agradecimento especial ao Filipe que me ajudou no insight que foi base da idéia da minha pesquisa.

Ao professor Guilherme Horta e meus amigos do grupo ESE: Arilo, Breno, Jobson, Karen “Japa”, Paulo Sérgio, Vitor Lopes, Vitor Farias, Victor Vidigal e Taísa que fizeram parte dessa trajetória, alguns durante a graduação e outros no mestrado.

A todos os que participaram dos experimentos e que colaboraram de uma forma ou de outra. À toda e qualquer pessoa não mencionada aqui, mas que fez parte de algum modo, valeu!

E a Deus, que tenho certeza, sempre esteve presente.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MODELO DE DESENVOLVIMENTO DE JOGOS COM PROPÓSITO BASEADO EM MECÂNICAS TRADICIONAIS DE JOGOS

Arthur Siqueira de Lima

Agosto/2013

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

Jogos com propósito surgem como sistemas de computação humana, baseados em entretenimento, para auxiliar na solução de tarefas de custo computacional muito alto. Em levantamento feito nesse trabalho percebe-se que os jogos com propósito existentes possuem as mecânicas de jogo muito próximas, praticamente análogas, as atividades para a solução do problema real. Posteriormente, é apresentado um procedimento de desenvolvimento de jogos com propósito que possibilita incorporar mecânicas de jogos tradicionais no *gameplay* dos jogos com propósito. Esse procedimento é composto por quatro processos que visam à transposição dos dados reais do problema para o ambiente do jogo, a utilização dos dados no jogo de forma a atingir a solução do problema a ser resolvido e a implementação de mecânicas tradicionais de jogos. Por fim, foi desenvolvido um jogo de classificação de dados seguindo o modelo proposto. Junto com a apresentação do jogo são mostrados os resultados obtidos através dele.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

GAMES WITH A PURPOSE DEVELOPMENT MODEL BASED ON GAME'S
TRADICIONAL MECHANICS

Arthur Siqueira de Lima

August/2013

Advisor: Geraldo Bonorino Xexéo

Department: Computer Science Engineering

Games With A Purpose (GWAPs) are platform for human computation focused on entertainment. We analysed GWAPs and noticed that gaming mechanics present on GWAP's gameplay are similar to real steps to solve a problem or a task. GWAPs using tradicional gaming mechanics can be more attractive to people and improve the computational power. We propose a model to help design GWAPs using tradicional gaming mechanics. This model has four processes. All processes help to use data to elaborate the game's stages and assist to incorporate traditional gaming mechanics on gameplay. We develop a data classifier GWAP using the steps present on the model proposed and , then, we compare our GWAP with classics data classifiers algorithms.

ÍNDICE

Capítulo 1 –Introdução	1
1.1 Motivação	2
1.2 Objetivo	2
1.3 Organização	3
Capítulo 2 –Computação Humana	5
2.1 – Sabedoria das Multidões	7
2.2 – Crowdsourcing	8
2.3 – Jogos com Propósito	10
Capítulo 3 –Jogo	19
3.1 – Gêneros de jogos	21
3.2 – Mecânicas que existem nos jogos	23
3.3 – Habilidades cognitivas e estratégias adotadas pelos jogadores	25
Capítulo 4 –Computação Humana para tarefa de classificação	29
4.1 – O que é classificar?	29
4.2 – Base dados disponíveis para tarefa	30
4.3 – Problemas e desafios encontrados em classificação	31
4.4 – Sistema humano de classificação	32
Capítulo 5 –Procedimentos de elaboração do jogo com propósito	36
5.1 - Procedimento de solução de problemas computacionais utilizando Jogos com Propósito	38
5.2 – Procedimento de projeto do jogo	45
5.3 – Procedimento de organização do jogo	57
5.4 – Procedimento de aplicação do jogo	64
5.5 – Conclusão	70
Capítulo 6 –Exemplo de jogo e experimento	72
6.1 – Chernoff Invaders: o jogo	72
6.2 – Experimento	81
6.3 – Conclusão	85

Capítulo 7 –Conclusões	87
7.1 Considerações acerca do trabalho	87
7.2 Contribuições	88
7.3 Limitações e trabalhos futuros	88
Referências Bibliográficas	90

LISTAGEM DE FIGURAS

Figura 1. Exemplo do ESP Game obtido durante partida do jogo por um <i>print screen</i>	12
Figura 2. Exemplo do jogo Google Image Labeler obtido no internet	13
Figura 3. Exemplo do jogo Peekaboom obtida no artigo (VON AHN <i>et al.</i> , 2006)	14
Figura 4. Exemplo do resultado obtido através do jogo Peekaboom. Imagem obtida do artigo (VON AHN <i>et al.</i> , 2006)	15
Figura 5. Exemplo de trecho classificado como ruim. Fonte (JOVIAN & AMPRIMO, 2011)	18
Figura 6. Exemplo de trecho classificado como neutro. Fonte (JOVIAN & AMPRIMO, 2011)	18
Figura 7. Classificação baseadas em habilidades cognitivas e estratégias utilizadas pelos jogadores quando jogam.	33
Figura 8. Visão geral do sistema humano de classificação de dados.	34
Figura 9. Visão detalhada (representação gráfica + jogo) do sistema humano de classificação de dados.	35
Figura 10. Visão detalhada (representação gráfica + jogo + compilador de resultados) do sistema humano de classificação de dados.	35
Figura 11. Procedimento para solução de problemas computacionais utilizando jogo com propósito.	39
Figura 12. Procedimento de projetar o jogo	46
Figura 13. Exemplo de representação gráfica utilizando rostos. (a) Exemplo de mapeamento do salário com o tamanho da boca. (b) Exemplo de inserção de significado na representação gráfica através do mapeamento do salário com o sorriso.	51
Figura 14. Passos do problema real relacionados com as mecânicas tradicionais inseridas no jogos.	53
Figura 15. Relação entre o passo real executado para solucionar o problema, a mecânica tradicional de jogo escolhida para substituí-lo e solução esperada para o problema.	54

Figura 16. Procedimento de organização do jogo.	58
Figura 17. Exemplo de representação gráfica para dados de diferentes padrões. Representação que será utilizada para ilustrar o treinamento do jogador.	60
Figura 18. Exemplo de uma fase de treinamento onde o jogador depara com representações gráficas (quadrados e triângulos) dos dados de treinamento.	60
Figura 19. Exemplo de <i>feedback</i> para induzir as decisões dos jogadores diante de elementos de representação gráfica de dados de treinamento de padrão d_1 .	61
Figura 20. Exemplo de <i>feedback</i> para induzir as decisões dos jogadores diante de elementos de representação gráfica de dados de treinamento de padrão d_2 .	61
Figura 21. Exemplo de representação gráfica utilizando dados reais. Destaque para perceber que as diferenças entre as representações dos padrões de dados não são tão nítidas como no exemplo utilizado para explicar a utilização de <i>feedbacks</i> para o treinamento do jogador.	62
Figura 22. Procedimento de aplicação do jogo.	66
Figura 23. Distribuição das classes da base Iris. <i>Iris setosa</i> representada pelo losango, <i>iris versicolor</i> representada pelo círculo e <i>iris virginica</i> representada pelo triângulo.	74
Figura 24. Exemplos de representações das faces de Chernoff dos diferentes programas de estatística. (a) MatLab; (b) R; (c) Mathematica; e (d) Stata. Não foram utilizados os mesmos dados para a representação dos rostos.	75
Figura 25. Telas da primeira versão do jogo com propósito desenvolvido para solucionar o problema de classificação de dados.	77
Figura 26. Telas da versão final do jogo com propósito desenvolvido para solucionar o problema de classificação de dados.	79
Figura 27. Tela inicial do jogo Chernoff Invaders. Configuração inicial da fase de treinamento.	80
Figura 28. Localização espacial dos dados classificados de maneira errada. Os dados classificados de maneira errada estão em destaque.	84

LISTAGEM DE TABELAS

Tabela 1. Classificação dos jogos com propósito em relação ao gênero de jogo	23
Tabela 2. Exemplos de mecânicas de jogo coletadas da análise realizada por DILLON (2010)	24
Tabela 3. Classificação dos jogos com propósito em relação as mecânicas de jogo	24
Tabela 4. Alguns exemplos de trechos de entrevista utilizados para identificar uma determinada habilidade cognitiva e estratégia utilizada pelo jogador	28
Tabela 5. Mapeamento dos atributos da base Iris e as características da face de Chernoff.	76
Tabela 6. Quantidade de classes presentes em cada fase.	79
Tabela 7. Dados classificados de maneira errada pelo jogo.	83
Tabela 8. Comparação entre o desempenho do jogo nas fases compostas por duas classes e por três classes.	84
Tabela 9. Comparação de desempenho fase por fase do jogo.	85
Tabela 10. Desempenho do algoritmo KNN alterando o número de vizinhos.	85
Tabela 11. Desempenho do algoritmo de Naive Bayes alterando o tamanho do conjunto de treinamento.	86
Tabela 12. Comparação entre os algoritmos KNN, <i>Naive Bayes</i> e o jogo com propósito Chernoff Invaders.	86

Capítulo 1 – Introdução

Existem tarefas que são extremamente custosas, até mesmo impossíveis, de serem realizadas por computadores. Contudo, essas mesmas tarefas são ridiculamente simples de serem realizadas pelo ser humano. Essas tarefas estão, geralmente, relacionadas à interpretação, percepções cognitivas, como por exemplo, o reconhecimento de objetos em imagens, entre outros.

Mesmo atingindo qualidades satisfatórias, uma tradução de um texto feita por algoritmos computacionais não consegue atingir os níveis das realizadas por seres humanos. As pessoas conseguem interpretar e perceber os recursos linguísticos utilizados pelo autor, como as figuras de linguagens, e acabam encontrando alternativas melhores para traduzir o texto.

Explorar as capacidades dos seres humanos com o objetivo de resolver problemas computacionais é o principal pilar da computação humana (VON AHN, 2005). Imaginando os seres humanos como um processador em um sistema distribuído, utilizam-se as respostas obtidas de diferentes pessoas para se chegar à solução final. As pessoas estão sujeitas a falhas, porém a resposta mais comum obtida de uma mesma tarefa realizada diversas vezes e por diferentes pessoas pode ser considerada a resposta correta.

Seguindo nessa linha de comparação entre ser humano e máquinas, o ser humano leva desvantagem no quesito repetição de uma mesma tarefa. Os computadores são capazes de realizar as mesmas tarefas repetidas vezes obtendo sempre os mesmos resultados. Já o ser humano, ao realizar as mesmas atividades, pode ficar entediado e isso acaba influenciando os resultados obtidos.

Contudo esse mesmo ser humano é capaz de passar horas e horas na frente de um computador, console ou dispositivo móvel jogando um jogo que o desafie. E a cada rodada ele sempre tenta buscar o melhor resultado para melhorar seu desempenho, ir mais longe em uma fase ou conseguir uma pontuação melhor que outro competidor.

Os jogos com propósito (VON AHN & DABBISH, 2008) tentam incorporar a resolução das tarefas computacionalmente impossíveis dentro da mecânica (*gameplay*) do

jogo. Assim, ao final da partida, o jogador acaba produzindo soluções computacionais como efeito colateral da partida disputada.

1.1 Motivação

As principais tarefas que os jogos com propósitos buscam dar suporte estão ligadas a percepções cognitivas. Os jogos ESP Game (VON AHN & DABBISH, 2004) e Google Image Labeler ajudam a rotular imagens. O Peekaboom (VON AHN *et al.*, 2006) auxilia na tarefa de identificar a posição de um objeto em uma imagem. O Phylo (KAWRYKOW *et al.*, 2012) ajuda no alinhamento de seqüências de DNA. O Foldit (COOPER *et al.*, 2010) colabora com determinação da estrutura de proteínas. O TypeAttack (JOVIAN & AMPRIMO, 2011) auxilia na digitalização de documentos.

Foi observado que nos jogos com propósito a mecânica de jogo está diretamente ligada a tarefa que se quer realizar com o auxílio do jogo. O ESP Game e o Google Image Labeler ajudam a rotular imagens e na mecânica de jogo o jogador deve informar rótulos para a imagem mostrada. O TypeAttack auxilia na digitalização de documentos e na mecânica do jogo o jogador deve digitar os trechos de texto que são mostrados. Isso se repete para todos os exemplos citados anteriormente.

As mecânicas de jogo presentes nos jogos com propósito são diferentes das mecânicas que compõe os jogos mais populares. Os jogos com propósito parecem uma forma mais divertida (atraente) de realizar as tarefas que são feitas normalmente. Dessa forma, os jogos com propósito não exploram a característica lúdica presente nos jogos, subutilizando o potencial que os jogos podem oferecer.

1.2 Objetivo

Pelo prisma de computação humana essas mecânicas de jogo atingem seus objetivos, que é obter resultados para as tarefas propostas. Contudo, as mecânicas desses jogos estão distantes das mecânicas dos jogos mais conhecidos. Nenhum desses jogos com propósito se parece com os jogos mais populares como, por exemplo, jogos de tiros ou de plataforma.

Os jogos com propósito possuem elementos comuns aos outros jogos que contribuem para aumentar o interesse como, por exemplo, o sistema de pontuação, *ranking*, interação com

outros jogadores. Contudo seus *gameplays* são pouco atrativos, limitando-se a solução do problema.

O número de jogadores e de partidas jogadas influencia diretamente na qualidade dos resultados. Quanto maior o número de respostas obtidas através dos jogadores, melhor é a resposta que se pode encontrar para a tarefa que o jogo se propõe a solucionar, já que diminui a influência do erro na obtenção do resultado final.

O objetivo desse trabalho é mostrar uma forma de incorporar as mecânicas dos jogos mais conhecidos nos jogos com propósitos, e assim, aumentar o número de jogadores interessados nesse tipo de jogo e, por consequência, melhorar a qualidade das respostas obtidas. O jogo mostrado como exemplo tem o intuito de ser abrangente o suficiente de modo que possa dar suporte a diferentes tarefas a partir da troca de dados de entrada que o alimentam. Além disso, foi criado de forma a permitir um experimento com menos efeitos externos aos dados usados, para facilitar a avaliação do resultado.

1.3 Organização

A dissertação é composta de 7 capítulos. O primeiro é o capítulo introdutório.

No Capítulo 2 – Computação Humana, são apresentados os conceitos fundamentais utilizados no desenvolvimento desse trabalho que estão relacionados com computação humana. Também são apresentados exemplos práticos dos conceitos atrelados à computação humana, bem como os exemplos de jogos com propósito.

No Capítulo 3 – Jogo, são apresentados conceitos importantes de jogos utilizados nesse trabalho. Conceitos esses como gênero de jogos, mecânicas de jogos e habilidades dos jogadores utilizadas durante uma partida.

No Capítulo 4 – Computação Humana para tarefa de classificação, são apresentados conceitos básicos de classificação de dados, problemas e desafios encontrados na classificação de dados. Por último, é apresentado um sistema humano de classificação de dados.

No Capítulo 5 – Procedimentos de elaboração do jogo com propósito, são apresentados os procedimentos para o desenvolvimento de jogos com propósito que facilitem

a inserção de mecânicas de jogos tradicionais. Esse procedimento é composto por quatro processos que visam a transposição dos dados reais para o ambiente do jogo com propósito e a utilização dos dados no jogo para atingir a solução real do problema.

No Capítulo 6 – Exemplo de jogo e experimento, é apresentado o processo de implementação de um jogo com propósito para auxiliar na classificação de dados seguindo os procedimentos proposto no capítulo 5. O jogo concebido é mostrado. Ainda são apresentados os experimentos realizados e os resultados obtidos.

Por fim, no Capítulo 7 – Conclusões, são apresentadas as conclusões observadas além das suas contribuições e perspectivas futuras.

Capítulo 2 – Computação Humana

Os computadores evoluíram e continuam evoluindo drasticamente, tornando-se cada vez mais potentes. Contudo, ainda não conseguem realizar de maneira satisfatória tarefas que os seres humanos conseguem com relativa facilidade. Como, por exemplo, tarefas de capacidades perceptivas, como identificar objetos em imagens.

Imaginando cada ser humano como um processador em um sistema distribuído, pode-se explorar as capacidades dos homens para resolver problemas computacionais em larga escala ou coletar dados para o treinamento de máquinas, com o intuito que as últimas possam realizar com sucesso tarefas humanas mais simples (VON AHN, 2005).

Computação humana é utilizar a poder de processamento do ser humano para resolver problemas que ainda não podem ser resolvidos por computadores. (VON AHN, 2005). É a capacidade de combinar o poder dos seres humanos e das máquinas com o objetivo de resolver problemas que não são possíveis de serem realizados sozinhos pelas partes (VON AHN, 2009).

Uma das formas que essa cooperação pode ser realizada é através de ambientes computacionais que permitam a colaboração entre os seres humanos e as máquinas em benefício da humanidade.

Diferente das máquinas, as pessoas precisam ser motivadas a colaborar coletivamente. As pessoas podem ser motivadas a participar desses processos de diferentes formas (QUINN & BEDERSON, 2011):

- i **remuneração:** a pessoa recebe um pagamento por uma determinada quantidade de tarefas realizadas;
- ii **altruísmo:** a pessoa acredita que o problema a ser resolvido é importante e tem vontade de ajudar;
- iii **divertimento:** através de atividades agradáveis como, por exemplo, jogos, as pessoas realizam as tarefas que vão acarretar na solução de algum problema;
- iv **reputação:** quando o problema está relacionado a alguma organização com prestígio e as pessoas participam para receber um reconhecimento público; e

- v **trabalho implícito:** é quando a tarefa que ajuda na resolução do problema já faz parte das atividades que as pessoas realizam.

A seguir será apresentado um exemplo de aplicação de computação humana baseada em trabalho implícito.

- **reCAPTCHA**

Uma forma de verificar se as requisições feitas na Internet são oriundas de um ser humano ou uma máquina é através de um simples teste. Esse teste é conhecido como CAPTCHAs (*Completely Automated Public Turing test to tell Computers and Humans Apart*) (VON AHN *et al.*, 2008) e consiste em mostrar uma imagem com caracteres distorcidos aos usuários e estes devem digitar os caracteres de forma correta provando que a requisição partiu de um ser humano.

Em um único dia as pessoas ao redor do mundo resolvem mais de 100 milhões de CAPTCHAs (VON AHN *et al.*, 2008). Cada pessoa gasta poucos segundos de esforço resolvendo um CAPTCHA, porém a humanidade chega a gastar milhares de horas por dia. Para que todo esse esforço não se tornar em vão, VON AHN (2008) propõe a utilização de CAPTCHAs para ajudar no processo de digitalização de textos impressos. Surge assim a idéia do reCAPTCHA.

O reCAPTCHA consiste em apresentar no lugar de caracteres distorcidos, as palavras que não foram reconhecidas por algoritmos de reconhecimento óptico de caracteres após a digitalização do documento (VON AHN *et al.*, 2008), permitindo que textos impressos antigos sejam transformados em textos digitais através do esforço das pessoas ao redor do mundo para solucionar os CAPTCHAs.

O reCAPTCHA é mais eficiente que os CAPTCHAs pois além de utilizar as mesmas distorções que o CAPTCHA utiliza as palavras selecionadas que já não foram identificadas por algoritmos de OCR e o próprio processo de digitalização insere ruído na palavra, dificultando os procedimentos para burlar o teste.

Com o intuito de evitar as possíveis respostas erradas dadas pelas pessoas, a mesma palavra é exibida para diferentes usuários. Ao final do processo é escolhida a resposta com o

maior número de “votos”, em outras palavras, é escolhida a resposta mais vezes apresentadas pelas pessoas.

O reCAPTCHA é um exemplo real que mostra a capacidade das pessoas para resolver problemas que ainda não podem ser realizados por computadores.

2.1 – Sabedoria das Multidões

Alguns autores defendem que grupos de pessoas, sob as circunstâncias corretas, são impressionantemente inteligentes, e frequentemente são mais inteligentes que a pessoa mais inteligente em seu interior (SUROWIECKI, 2006). Uma pessoa pode ter uma resposta melhor do que as outras do grupo. Contudo, a cada novo problema são indivíduos diferentes que se destacam. Assim, as melhores respostas são obtidas do grupo, e não de uma pessoa específica.

Segundo SUROWIECKI (2006) existem três condições necessárias para que o grupo seja sábio. Uma é a diversidade de opinião, a outra é que os indivíduos sejam independentes e, por último, que haja uma espécie de descentralização. As melhores decisões coletivas são fruto de discordância e contestação e não do consenso e acordo.

As pessoas que compõe o grupo devem possuir um conhecimento especializado e formado com um conhecimento local. A diversidade acrescenta diferentes perspectivas ao grupo e, assim, ampliam o conjunto de soluções possíveis. Além disso, facilita a exposição do que os indivíduos realmente pensam.

A independência significa a liberdade da influência dos outros. A decisão do indivíduo não deve ser influenciada pelas decisões dos outros integrantes do grupo. As pessoas devem tomar decisões baseadas no seu conhecimento e a decisão coletiva é uma combinação dessas decisões. Assim, evita-se que erros de avaliação pessoal não prejudiquem a avaliação coletiva do grupo.

A descentralização implica em colocar um grupo de pessoas independentes para trabalhar de forma individual, em vez de tentar dirigir seus esforços de cima para baixo. A descentralização favorece a especialização e a produtividade e acaba encorajando a independência. Quanto mais próximo de um problema mais provável que tenha uma solução para ele. O Linux é um exemplo de descentralização.

A sabedoria das multidões colabora com a computação humana. Tarefas exploradas na computação humana podem ser realizadas por somente um indivíduo, porém, ao buscar diferentes soluções dessas tarefas em um grande grupo de pessoas, a tendência é que se obtenha uma solução melhor.

2.2 – Crowdsourcing

Baseada na computação humana e na sabedoria das multidões o *crowdsourcing* aparece como uma nova maneira para encontrar soluções de problemas e elaborar produtos. É um novo modelo de negócio que explora a criação de soluções por redes distribuídas compostas por indivíduos que são convocados a apresentarem uma solução para o problema proposto (BRABHAM, 2008).

Para HOWE (2006) o *crowdsourcing* ganha força quando companhias e organizações percebem que tarefas realizadas por empregados podem ser realizadas, de uma forma mais barata, por grupos de pessoas conectados pela Internet.

A Internet desempenha um papel importante no *crowdsourcing*. Possibilita que grupos heterogêneos, compostos por especialista, “hobbyistas” e amadores, possam utilizar e compartilhar informações na rede e através dessas informações criar conteúdos e soluções de problemas. A Internet ainda possibilita a participação de pessoas em uma escala global e possui baixos custos.

Como destaca BRABHAM (2008), *crowdsourcing* é diferente das iniciativas de código aberto. Nos sistemas código aberto é permitido acesso ao código fonte para que haja um desenvolvimento colaborativo com distribuição do produto final. Já no caso do *crowdsourcing*, como existe pagamento da melhor solução encontrada, a organização é dona da solução do problema que ela propôs.

O processo de *crowdsourcing* é composto por quatro etapas (VUKOVIC, 2009). (i) Primeiramente a organização descreve o problema e define suas características, período que vai receber soluções, parâmetros utilizados entre outros. (ii) Em seguida, o problema é divulgado. Em outras palavras, é realizada uma chamada pública para os interessados em colaborar. (iii) Durante o período em que as soluções estão sendo desenvolvidas, a

organização oferece suporte e uma maneira de integração entre os participantes. (iv) Por último, as soluções são avaliadas pela organização e a solução vencedora é escolhida.

- **Amazon Mechanical Turk**

Mechanical Turk é um serviço web de *crowdsourcing* oferecido pela Amazon. Ele funciona como um *pool* de tarefas que ficam acessíveis para serem resolvidas por seres humanos. Funcionando como uma espécie de inteligência humana disponível para resolver problemas, contribuir com pesquisas, testar usabilidades de interfaces entre outros.

Na plataforma online os *requesters* criam tarefas (chamadas HITs, *Human Intelligent Tasks*) e recrutam *workers* para realizar as tarefas em troca de uma remuneração, chamada *reward*. Ao criar uma tarefa o *requester* define a remuneração, o tempo estimado para a realização da tarefa e ainda pode definir quais tipos de *workers* estão aptos a realizar aquela tarefa.

Como a motivação para a participação nas atividades é financeira, pode-se questionar a qualidade dos dados obtidos. Contudo, os dados obtidos através do Mechanical Turk são pelo menos tão confiáveis que os dados obtidos por métodos tradicionais (BUHRMESTER *et al.*, 2011). Ainda segundo BUHRMESTER *et al.* (2011), uma remuneração justa não afeta a qualidade dos dados.

O valor da remuneração e o tempo previsto de duração da tarefa afetam a quantidade de participantes da tarefa e o tempo para conseguir um determinado número de participantes (BUHRMESTER *et al.*, 2011). Em outras palavras, quanto mais simples for a tarefa e mais alta a remuneração, mais participantes respondem em um menor espaço de tempo.

O ambiente permite que o *requester* bonifique *workers* com bom desempenho e rejeite pagamento para *workers* com respostas de má qualidade. Isso contribui para manter a qualidade dos dados. Por outro lado, os *workers* podem marcar *requesters* que não derem uma justificativa satisfatória para suspender o pagamento. Evitando assim a exploração dos *workers*.

Grupos de usuários obtidos através do Mechanical Turk tem uma representatividade melhor em relação a realidade do que grupos de teste selecionados em um ambiente

específico. Analisando somente os *workers* americanos, eles são indiscutivelmente mais próximos da população americana do que grupos recrutados dentro das universidades americanas (PAOLACCI *et al.*, 2010). Os *workers*, de uma forma geral, são mais demograficamente distintos do que as amostras padrões de usuários obtidos através da Internet e significativamente mais diversificados do que amostras conseguidas em uma universidade americana (BUHRMESTER *et al.*, 2011).

O Mechanical Turk permite que tarefas computacionalmente complicadas possam ser realizadas por pessoas de uma forma rápida e com baixo custo sem afetar a qualidade dos dados obtidos.

2.3 – Jogos com Propósito

A existência de tarefas que são triviais para os seres humanos, porém extremamente custosas para os computadores. O fato de que as pessoas, especialmente os jovens, estão passando muitas horas no computador e jogando. O aumento da população com acesso a Internet. São os três fatos observados por VON AHN & DABBISH (2008) que levaram ao seguinte questionamento: “E se todo esse tempo e energia fossem canalizados para resolver problemas computacionais e treinamento de algoritmos de inteligência artificial?”¹.(VON AHN & DABBISH, 2008) Surge assim o conceito de Jogos com Propósito, do inglês *games with a purpose* (GWAP).

Jogos com Propósito são jogos em que os jogadores produzem algum resultado computacional como efeito colateral de uma partida do jogo (VON AHN & DABBISH, 2008). As pessoas não jogam com o intuito de colaborar com a solução de algum problema computacional, mas sim para se divertir e entreter com o jogo (VON AHN & DABBISH, 2008).

Jogos com Propósito são uma plataforma para computação humana que não focam na remuneração ou no trabalho implícito para motivar a participação do ser humano. O principal estímulo nesse caso é o divertimento.

¹ Tradução do autor para o seguinte trecho: “What if time and energy were also channeled toward solving computational problems and training AI algorithms?” (VON AHN & DABBISH, 2008)

Todo jogo com propósito deve estar relacionado a algum problema computacional. Os jogos com propósito devem ser concebidos de forma a fazer com que os jogadores realizem os passos necessários para resolver um problema computacional e, se possível, garantir que as respostas obtidas através do jogo sejam corretas, mesmo que o jogador não tenha a intenção de ser correto (VON AHN & DABBISH, 2008).

Baseados nas estruturas dos jogos, VON AHN & DABBISH (2008) definem três tipos de jogos com propósito:

- i ***output-agreement***: jogadores recebem uma mesma entrada e devem produzir a mesma resposta;
- ii ***inversion-problem***: um jogador recebe uma entrada e o outro tem que produzir como resposta a entrada recebida pelo primeiro jogador; e
- iii ***input-agreement***: cada jogador recebe uma entrada e eles devem determinar se essas entradas são iguais ou diferentes.

A seguir são apresentados exemplos de jogos com propósito.

- **ESP Game**

Bases de imagens rotuladas podem ser úteis para melhorar o desempenho de máquinas de busca de imagens, para realizar o treinamento de algoritmos de máquina de aprendizado, auxiliar na acessibilidade entre outros. Porém rotular as imagens de um modo manual é tedioso e custoso.

O ESP Game (VON AHN & DABBISH, 2004) surge como um jogo que enquanto o jogador se entretém jogando ele ajuda a determinar os objetos presentes na imagem informando rótulos para as mesmas. O Jogo explora as capacidades perceptivas das pessoas, que são superiores as das máquinas, e a vontade de se entreter através do jogo.

O jogo fica disponível na Internet² e é jogado por duas pessoas de forma cooperativa. Os pares de jogadores são formados de maneira aleatória e a comunicação entre os jogadores da dupla não é permitida.

Na perspectiva do jogador, o objetivo do jogo é adivinhar o rótulo que o parceiro está digitando para a imagem. Quando os jogadores fornecem a mesma palavra para a imagem eles recebem uma pontuação e aparece uma nova imagem para ser rotulada. Esse procedimento se repete até que o período de tempo acabe. Assim sendo, a palavra em que os dois jogadores concordaram é considerada um bom rótulo para a imagem mostrada.

Quando uma mesma palavra é utilizada por diversas duplas de jogadores ela passa a ser definida como um rótulo para a imagem. Com o intuito de aumentar o número de rótulos e obter rótulos mais específicos para as imagens, é utilizado o conceito de palavras tabu. São palavras obtidas através do próprio jogo e por já rotularem a imagem não são mais permitidas para as novas rodadas. Assim obriga o jogador a informar novos rótulos para aquela imagem.

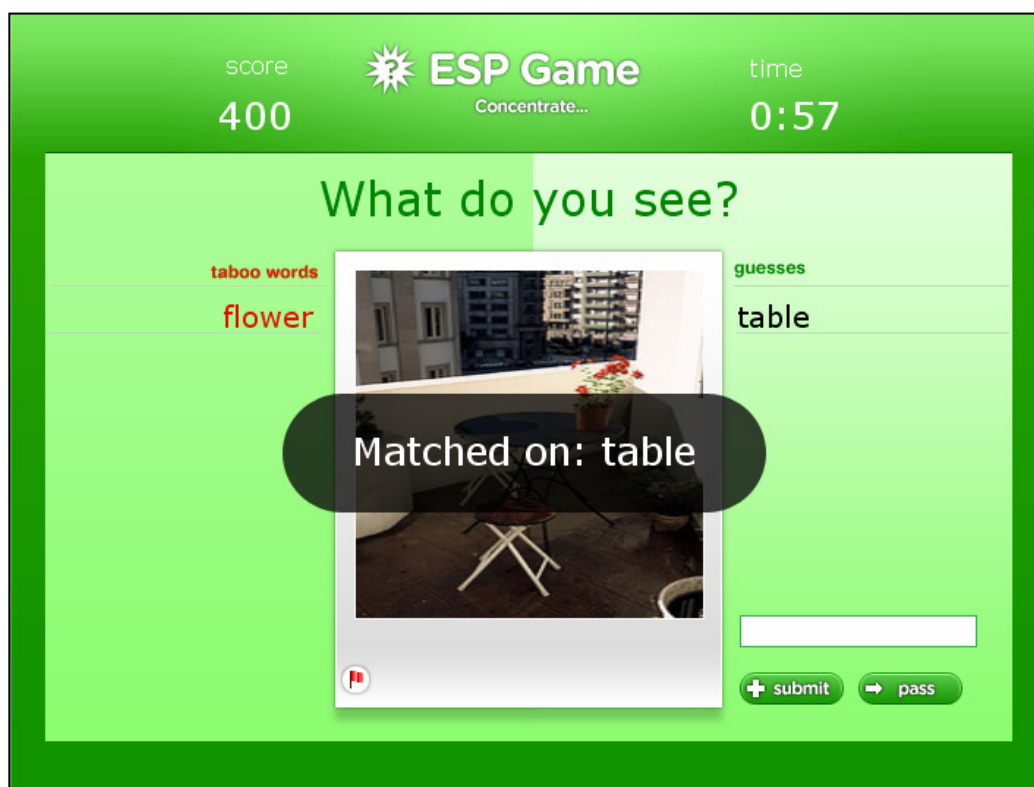


Figura 1. Exemplo do ESP Game obtido durante partida do jogo por um *print screen*

² <http://www.gwap.com/gwap/gamesPreview/espgame/>

No exemplo da Figura 1 os jogadores concordaram com a palavra *table* para a imagem mostrada. A palavra *flower* não pode ser utilizada, pois já é considerada um rótulo válido para a imagem. Sendo impedido de utilizar determinadas palavras, o jogador passa a buscar novas palavras para a imagem aumentando o número de rótulos possíveis para as imagens.

- **Google Image Labeler**

Em 2006 o ESP Game foi licenciado ao Google para ser utilizado a fim de melhorar o desempenho da sua popular máquina de busca de imagens (VON AHN, 2009). Em 2011 o jogo foi descontinuado junto com uma serie de serviços do Google (EUSTACE, 2011).

O jogo do Google apresenta a mesma mecânica de jogo que o ESP Game. A principal diferença é que as imagens utilizadas nas partidas são oriundas da própria base de imagens do Google.

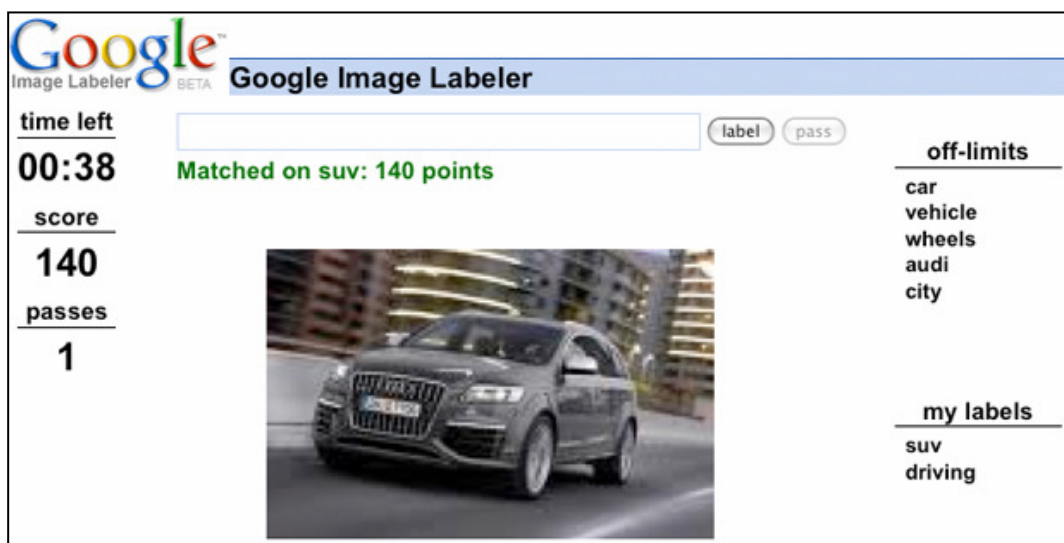


Figura 2. Exemplo do jogo Google Image Labeler obtido no internet

- **Peekaboom**

Seres humanos lidam com imagens todos os dias sem muitas dificuldades, contudo computadores ainda tem problemas com tarefas visuais simples como localizar objetos em imagens (VON AHN *et al.*, 2006). O Peekaboom visa explorar essa facilidade do ser humano para localizar objetos nas imagens.

O Peekaboom (VON AHN *et al.*, 2006) é jogado por duplas de jogadores formadas de maneira aleatória. A cada rodada um jogador faz o papel de *peek* e o outro de *boom*. O *peek* começa com uma tela em branco e o *boom* começa com uma imagem e uma palavra relacionada a ela (obtida através do jogo ESP Game).

O papel do *boom* é revelar partes da imagem para o *peek* que deve adivinhar a palavra associada à imagem. Para melhorar o desempenho o *boom* deve mostrar apenas partes da imagem que são relacionadas à palavra associada. O *boom* ainda pode dizer se os palpites do *peek* estão quente (perto da resposta correta) ou frio (longe da resposta correta).

Na Figura 3 o *boom* possui a palavra *cow* (vaca, em inglês) associada a sua imagem (lado direito da Figura 3). Para ter o melhor desempenho o *boom* deve tornar visível ao *peek* somente a área da imagem que possui a vaca.

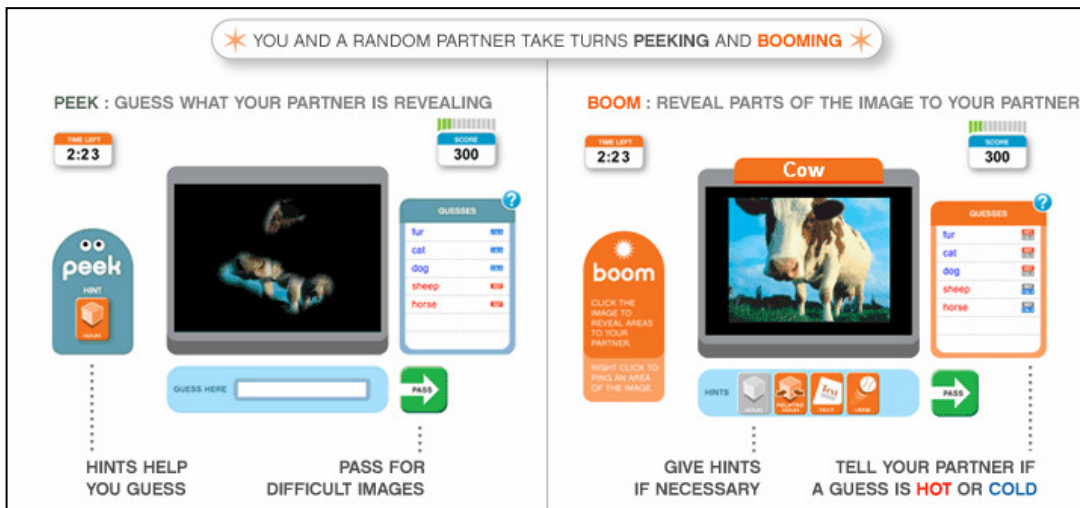


Figura 3. Exemplo do jogo Peekaboom obtida no artigo (VON AHN *et al.*, 2006)

A Figura 4 mostra o resultado obtido através do jogo. Um conjunto de imagens com metadados que informam a região da imagem que está presente no objeto. No exemplo pode-se perceber a localização dos olhos (retângulo vermelho) e do nariz (retângulo azul) em cada imagem.

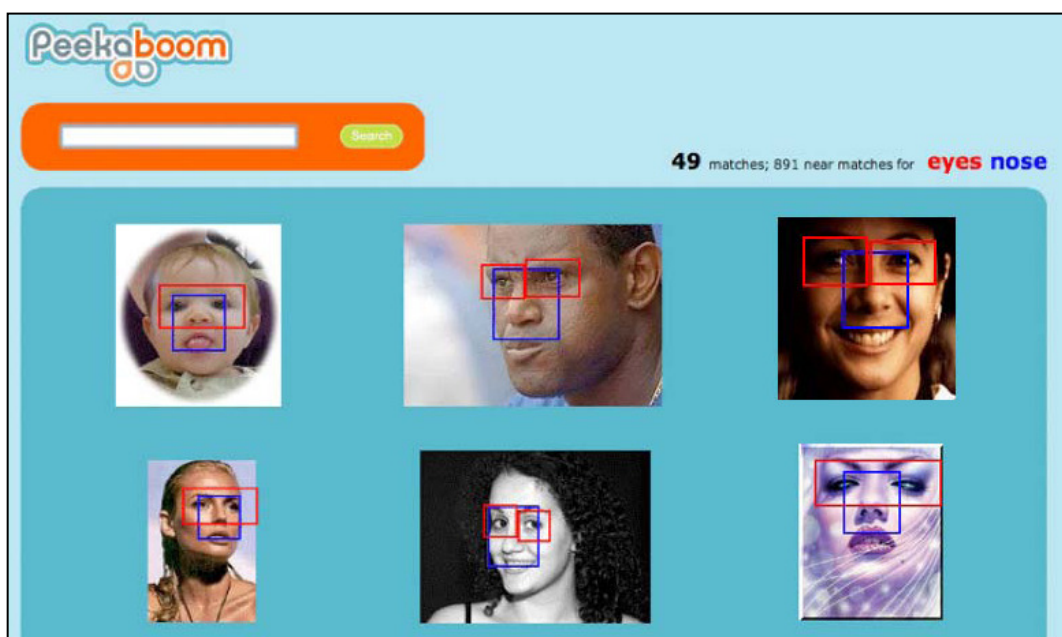


Figura 4. Exemplo do resultado obtido através do jogo Peekaboombot. Imagem obtida do artigo (VON AHN *et al.*, 2006)

- **Foldit**

Determinar a estrutura de uma proteína é um problema computacional complexo. A estrutura das proteínas é determinada pelas seqüências de aminoácidos. As metodologias conhecidas para determinar a estrutura das proteínas atende a uma parcela pequena das proteínas conhecidas.

Foldit³ (COOPER *et al.*, 2010) é um jogo multiplayer online que possibilita que jogadores não cientistas resolvam problemas complexos de previsão de estruturas das proteínas. Os jogadores interagem com as estruturas protéicas através de ferramentas de manipulação baseadas em versões simples dos algoritmos de previsão de estrutura de proteína já conhecidas.

Formações incorretas de proteínas são apresentadas aos jogadores que devem rearranjar as proteínas de forma em que eles acham que conseguirão mais pontos. Os

³ <http://fold.it/portal/>

jogadores tem a possibilidade de realizar diferentes manipulações como, por exemplo, rotação.

Para tornar o jogo acessível para um número maior de jogadores existe uma série de fases introdutórias que buscam passar algumas noções de biologia molecular. As fases iniciais também ensinam o uso das ferramentas de manipulação, as diferentes visualizações e algumas estratégias.

Em 2011 um grupo de jogadores do Foldit conseguiu desvendar a estrutura de uma proteína importante para a reprodução de vírus como o HIV (KHATIB *et al.*, 2011), possibilitando assim avanços nos estudos de vacinas e remédios.

- **Phylo**

O alinhamento de sequências de DNA, RNA ou proteínas permite que cientistas infiram origens evolucionárias das espécies, ilustrem mutações e consigam traçar a origem de determinadas doenças genéticas.

Para gerar alinhamentos precisos são necessários algoritmos computacionalmente custosos. Esses algoritmos são NP-difícil e necessitam de um número de heurísticas que tornam os resultados imprecisos.

Pensando em aproveitar as facilidades dos homens para reconhecer padrões visuais foi criado o Phylo⁴ (KAWRYKOW *et al.*, 2012), um jogo que visa solucionar problemas de alinhamento de múltiplas sequências de DNA. O jogo transcreve o problema científico em um jogo casual para torná-lo acessível a jogadores não cientistas.

As sequências de DNA são representadas por sequências de blocos de quatro cores diferentes, representando os quatro nucleotídeos do código genético (adenina, guanina, citosina e timina). As sequências de blocos são apresentadas em até oito linhas. Os blocos podem ser movidos horizontalmente. O objetivo do jogo é organizar os blocos de uma maneira que maximize o número de blocos da mesma cor em uma coluna e minimize o número de espaços entre blocos.

⁴ <http://phylo.cs.mcgill.ca/eng/play.html>

A base de sequências do jogo é obtida através de subsequências que apresentam sinais de desalinhamento. Essas subsequências formam as sequências de blocos que são apresentadas ao jogador. A solução dada pelo jogador é inserida à sequência que deu origem a subsequência utilizada e é avaliada. Se o resultado obtido for melhor que a original ele é mantido, caso contrário é descartado.

No melhor caso o resultado obtido pelo Phylo foi 70% melhor que o obtido pelo Multiz (programa de computador que faz alinhamento de sequências de DNA). E na média os resultados obtidos pelo Phylo levaram metade do tempo que no Multiz.

- **TypeAttack**

Digitalização é o processo de converter informação “analógica” em informação digital. Documentos antigos possuem mais ruídos devido ao tempo, a conservação e a qualidade de impressão da época. Isso ocasiona baixa precisão dos algoritmos de reconhecimento ótico de caracteres, do inglês *optical character recognition* (OCR). Necessitando de um passo adicional no processo de digitalização para correção.

O TypeAttack (JOVIAN & AMPRIMO, 2011) é um jogo para o facebook que surge como uma alternativa para o tradicional processo de correção de OCR. Explorando as habilidades dos seres humanos e sua vontade de diversão. O jogo foca na digitalização de documentos antigos que não puderam ser transcritos de maneira satisfatória pelos programas de OCR.

No TypeAttack os jogadores competem para digitar os textos presentes nas imagens digitalizadas de trechos dos documentos o mais rápido possível e da maneira mais correta. Em uma partida, o jogador compete com outros quatro jogadores que já digitaram o trecho do documento. É possível acompanhar a velocidade que outros jogadores digitaram o trecho e ver o quão rápido ou devagar deve-se digitar pra ganhar.

As imagens de textos foram obtidas através de artigos do jornal *The Straits Times* do ano de 1938. Os artigos foram divididos em trechos de cinco linhas para se adequar melhor ao jogo. Esses trechos foram pré-processados e classificados em bom, neutro e ruim. O jogo utiliza apenas os trechos classificados como neutro. Os bons foram descartados por possuir

um resultado satisfatório com os algoritmos de OCR e os ruins, porque causam impacto negativo e confusão no jogo, como exemplo, um trecho com números e pontuações excessivos.

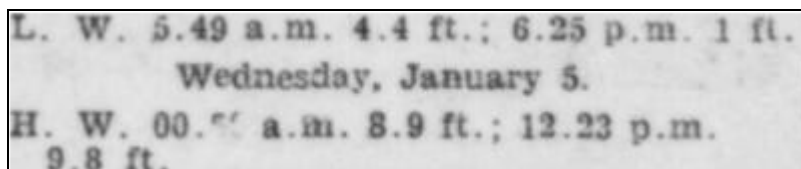


Figura 5. Exemplo de trecho classificado como ruim. Fonte (JOVIAN & AMPRIMO, 2011)

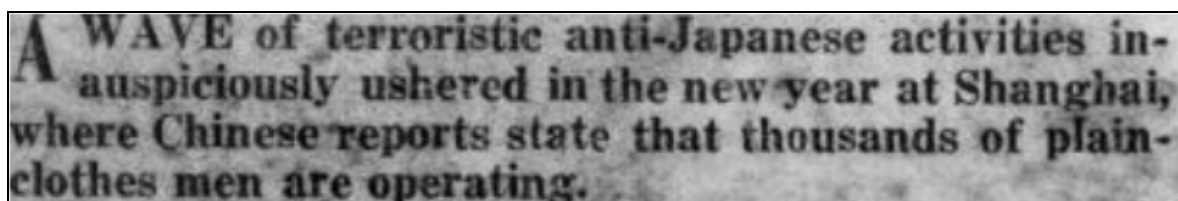


Figura 6. Exemplo de trecho classificado como neutro. Fonte (JOVIAN & AMPRIMO, 2011)

O sistema valida a resposta de cada jogador antes de armazená-la. Ele compara a resposta do jogador com o resultado do pré-processamento e com as respostas de outros jogadores. Quando um trecho é transcrito o número suficiente de vezes ele é retirado do jogo e substituído por outro trecho.

Os resultados obtidos pelo TypeAttack foram equivalentes aos de profissionais. Os trechos utilizados no jogo tinham, em média, 83.1% de precisão. E após as partidas esses trechos tiveram 99% de precisão em suas transcrições. Esses dados foram coletados de 505 trechos transcritos por 289 jogadores (JOVIAN & AMPRIMO, 2011).

Enquanto o reCAPTCHA (VON AHN *et al.*, 2008) é focado em corrigir palavras, o foco do TypeAttack são trechos de texto. Além disso, a motivação de participação é diferente, no reCAPTCHA se dá pelo trabalho implícito e no TypeAttack pela diversão.

Capítulo 3 – Jogos

Quando se fala de jogo não existe um consenso na academia, existem muitas definições para esse conceito. Isso se deve, em parte, por ser estudado por especialistas de diferentes áreas acadêmicas o que torna a visão sobre jogos bastante variada. Jogos são estudados por psicólogos, antropólogos, artistas, por profissionais de computação e da própria área de jogos.

Não é incomum um autor estudar diferentes definições encontradas para formalizar e apresentar o seu entendimento sobre jogo. Da mesma forma, nesse capítulo, são apresentadas definições sobre jogos de outros autores e é finalizado com a visão que se adequa melhor ao que é entendido por jogo nesse trabalho.

Em SCHELL (2008) o autor começa a discussão sobre o que é jogo com uma definição bem simples e que não diz muita coisa: “Jogo é alguma coisa que se joga”. A definição se torna mais imprecisa se pensarmos ela em inglês, “*A game is something you play*”, devido as diferentes interpretações que o verbo *to play* pode assumir na língua inglesa.

È possível definir jogo a partir do verbo jogar. Contudo, o contrário também é possível, definir o verbo jogar a partir de jogo. Por isso é melhor buscar outras maneiras de definir jogo evitando utilizar o par jogo x jogar. Uma forma comum é elaborar uma lista de condições para que se possa classificar algo como um jogo. Seguindo esse raciocínio, para ROGERS (2010) um jogo é uma atividade que:

- (i) Precisa de pelo menos um jogador;
- (ii) Possui regras; e
- (iii) Possui uma condição de vitória.

Seguindo a mesma linha de ROGERS (2010), a partir do estudo de diferentes definições para jogo encontradas na academia e na indústria, SCHELL (2008) elabora uma lista de características presentes nos jogos. Essa lista é mais detalhada e mais abrangente que a lista proposta anteriormente. Para SCHELL (2008) os jogos:

- (i) são jogados voluntariamente;

- (ii) possuem objetivos;
- (iii) possuem conflitos;
- (iv) possuem regras;
- (v) permitem que haja vitória ou derrota;
- (vi) são interativos;
- (vii) possuem desafios;
- (viii) constroem seus próprios valores internos;
- (ix) são envolventes para os jogadores; e
- (x) são sistemas formais fechados.

Em um segundo momento SCHELL (2008) muda a perspectiva de suas análises. Passar a analisar como as pessoas se relacionam com os jogos. E chega a conclusão que as pessoas jogam pois gostam de resolver problemas. Encontrar a solução para um tipo de problema é de algum modo prazeroso para as pessoas.

De certa forma os objetivos do jogo são propostos ao jogador na forma de solução de problemas, como por exemplo, descobrir um modo de completar um desafio ou uma fase, até mesmo, fazer mais pontos que um oponente. As características identificadas nos jogos também estão presentes na resolução de problemas (SCHELL, 2008).

Por isso, “Um jogo é uma atividade de **solução de problemas**, abordada por uma perspectiva lúdica.”^{5 6} (SCHELL, 2008)

A definição de jogo proposta por SCHELL (2008), que explora o conceito de solução de problemas, se enquadra perfeitamente com o que é entendido por jogos com propósito. Já que nos jogos com propósito pretende-se solucionar problemas computacionais reais através dos jogos.

⁵ Tradução do autor para o trecho “A game is a problem-solving activity, approached with a playful attitude” (SCHELL, 2008) pág 37

⁶ Grifo nosso.

Esses problemas reais são transcritos de uma forma que possam ser incorporados ao *gameplay* de um jogo, tornando lúdica a abordagem feita sobre o problema e, que ao final, irá produzir um resultado válido para o problema.

3.1 – Gêneros de jogos

Existem jogos de tabuleiro, jogos de carta, jogos de azar, jogos de bar, jogos eletrônicos, jogos de computador, videogames e diversos outros tipos de jogos. Em todos eles estão presentes os mesmos princípios: o jogador, as regras e a(s) condição(ões) de vitória(s). E apesar das diferenças presentes entre os tipos de jogos é possível identificar eles de uma forma única como jogo.

Diferente de quando se fala de jogo, existe um consenso sobre videogames. Na literatura os autores condicionam o videogame à presença de um dispositivo de vídeo:

“Videogame é um jogo que usa uma tela de vídeo digital de algum tipo, de alguma forma.”⁷ (BRATHWAITE & SCHREIBER, 2009)

“Videogame é um jogo que é jogado em uma tela de vídeo.”⁸ (ROGERS, 2010)

Apesar de não mencionar a presença de dispositivos de entrada, estes fazem parte da experiência desse tipo de jogo. É através deles que o jogador interage e informa suas decisões durante uma partida. As consequências de suas decisões são mostradas para o jogador através da tela de vídeo.

Os jogos possuem gêneros, da mesma forma que acontece na literatura e no cinema. O gênero do jogo auxilia a identificar o tipo de *gameplay* presente. Nada impede que um jogo combine diferentes tipos de *gameplay*, esses são chamados de híbridos. Entre os gêneros mais comuns estão presentes:

⁷ Tradução do autor para o trecho “A video game is a game that uses a digital video screen of some kind, in some way” (BRATHWAITE & SCHREIBER, 2009) pág 5

⁸ Tradução do autor para o trecho “ vídeo game is a game that is played on a vídeo screen” (ROGERS, 2010) pág

- (i) **Aventura:** possui narrativas fortes, foco na solução de desafios e coleção de itens. No início esses jogos eram baseados em textos.
- (ii) **Ação:** exige uma coordenação elevada entre olhos e mãos. O jogador deve responder em tempo real ao que acontece no jogo.
- (iii) **Estratégia:** exige raciocínio e planejamento nas jogadas. O jogador deve gerenciar recursos limitados para atingir os objetivos.
- (iv) **Simulação:** tenta emular condições reais de máquinas e experiências nos jogos.
- (v) **Tiro:** baseado em destruir os inimigos através de tiros.
- (vi) **Esporte:** baseado em esportes e competições atléticas. Reproduzem as regras e estratégias das competições esportivas.
- (vii) **Quebra-cabeça:** baseia-se em lógica e reconhecimento de padrões para criar desafios intelectuais. Existem por si só, diferente dos desafios que aparecem nos jogos de aventura.

Os gêneros possuem subgêneros para tornar mais específicos ou evidenciar uma determinada característica do jogo. O gênero tiro, por exemplo, possui subgênero que enfatizam a perspectiva da câmera presente no jogo:

- (i) **Primeira pessoa:** a câmera é posicionada de maneira a simular a visão do personagem do jogador. A visão, como um todo, é mais limitada, porém permite uma imersão maior;
- (ii) **Terceira pessoa:** a câmera é posicionada atrás do jogador. Permite visualizar o personagem do jogador e o seu entorno; e
- (iii) **Shoot 'em up:** a câmera permite uma visão de cima ou de lado do jogo. O jogador atira em uma grande quantidade de inimigos enquanto desvia dos perigos.

Os jogos com propósito não se enquadram perfeitamente nessa classificação. Ao tentar classificá-los de acordo com os principais gêneros de jogos percebe-se que alguns deles não se enquadram em nenhum gênero.

O Foldit e o Phylo podem até ser considerados quebra-cabeças, pois se baseiam em reconhecimento de padrões para encontrar a solução do problema. Já o ESP Game, o Google

Image Labeler, o Peekaboom e o TypeAttack não se enquadram rigorosamente em nenhum dos gêneros. .

Tabela 1. Classificação dos jogos com propósito em relação ao gênero de jogo

Jogo com Propósito	Gênero de Jogo
ESP Game	X
Google Image Labeler	X
Peekaboom	X
Foldit	quebra-cabeça
Phylo	quebra-cabeça
TypeAttack	X

A dificuldade de classificar os jogos com propósito nos gêneros mais conhecidos torna evidente as diferenças que eles possuem em relação aos jogos mais populares. Para incorporar a solução de problemas reais dentro do método de jogo acabam-se criando novos estilos de jogos. Porém, para os jogadores em geral, encontrar uma forma de jogo que já é familiar a eles pode tornar os jogos com propósito mais atrativos.

3.2 – Mecânicas que existem nos jogos

Os jogadores interagem com os jogos como uma forma de avisar ao jogo as decisões tomadas por eles. Essas decisões são informadas ao jogo através de ações que o *gameplay* permite o jogador realizar. Em um jogo de tiro o jogador pode atirar, mirar, recarregar, entre outros. As ações básicas que podem ser tomadas pelo jogador são chamadas de mecânicas.

As mecânicas de jogo definem o que o jogador pode fazer enquanto joga, são as regras do jogo. As mecânicas que prevalecem definem o gênero do jogo. O jogo no qual as mecânicas permitem que o jogador se mova, mire e atire é classificado como um jogo de tiro. As mecânicas de jogo definem o jogo em si.

Para exemplificar, a seguir é mostrada uma lista com algumas mecânicas retiradas de (DILLON, 2010) onde o autor analisa diferentes jogos clássicos como *Space Invaders*, *Pacman*, entre outros.

Tabela 2. Exemplos de mecânicas de jogo coletadas da análise realizada por DILLON (2010)

mover	parar	virar	pular	correr
mirar	atirar	recarregar	esconder	matar
comprar	vender	dirigir	acelerar	voar
bater	socar	chutar	defender	quebrar

Mecânicas podem ser interpretadas como se fossem os verbos do jogo. É como o jogador explicaria um jogo. “Você vai **andando** pela fase, **coletando** moedinhas e **pulando** em cima dos inimigos”. É um exemplo de como se definiria os populares jogos de plataforma da série *Super Mario World*.

Fazendo algo semelhante ao que foi feito por DILLON (2010), realizamos um levantamento das mecânicas presentes nos jogos com propósito. No quadro a seguir são apresentadas as principais mecânicas encontradas nos jogos com propósito estudados.

Tabela 3. Classificação dos jogos com propósito em relação as mecânicas de jogo

Jogo com Propósito	Mecânicas de Jogo
ESP Game	digitar
Google Image Labeler	digitar
Peekaboom	digitar, revelar
Foldit	mover, rodar, ligar combinar
Phylo	mover, combinar
TypeAttack	digitar

Pode-se perceber que as mecânicas encontradas são diferentes das presentes nos jogos clássicos levantadas por DILLON (2010). Fazendo uma exceção a mecânica **mover**. Esta se encontra presente nos jogos Foldit e Phylo.

Este levantamento deixa claro que existem diferenças significativas nas mecânicas dos jogos tradicionais e dos jogos com propósito. Isso se dá, em parte, ao fato de que o *gameplay* pensado para o jogo com propósito se basear fortemente no método utilizado para resolver o problema real. Logo as ações permitidas pelo jogo são semelhantes às possíveis de se realizar no problema real.

No caso do ESP Game e do Google Image Labeler o jogo auxilia na atribuição de rótulos as imagens. A forma que essa atividade é realizada na vida real é a mesma que a proposta pelo jogo. Uma imagem é mostrada e as pessoas devem informar rótulos para a imagem. A mecânica presente na vida real e no jogo são as mesmas, escrever e/ou digitar um rótulo para a imagem.

Utilizar mecânicas que estejam presentes em jogos tradicionais pode tornar os jogos mais interessantes, pois os jogadores estariam mais familiarizados com a forma de jogar desses jogos. Como consequência poderia atrair mais jogadores, e assim, aumentaria o número de soluções possíveis para o problema.

3.3 – Habilidades cognitivas e estratégias adotadas pelos jogadores

Diferente de outras formas de entretenimento, como livros e músicas, o consumo dos jogos é relativamente imprevisível. A cadeia de eventos pensados para a experiência a ser oferecida pelo jogo nem sempre é praticada pelo jogador da mesma forma que foi pensada pelos criadores.

Jogos são muito mais do que simples atividades que exigem coordenação entre os olhos e as mãos. Os jogadores reagem de forma diferente aos acontecimentos do jogo e cada um escolhe uma estratégia para resolver o desafio proposto. Os jogos são experiências interativas, narrativas e, até mesmo, de aprendizagem.

Os jogos incorporam princípios de aprendizagem. Os desafios iniciais são apresentados de forma mais simples para que o jogador aprenda a interagir com eles. Com o

conhecimento adquirido o jogador se torna capaz de resolver os desafios mais complexos ou experimentar novas abordagens para obter a solução.

Por serem atividades complexas que exigem aprendizado e interação, os jogos requerem habilidades cognitivas e estratégias por parte dos jogadores. Trabalhos como HENDERSON (2005) e HENDERSON *et al.* (2010) estudam o comportamento dos jogadores durante uma partida com a finalidade de identificar as habilidades cognitivas e as estratégias mais utilizadas pelos jogadores para obter o melhor desempenho.

Nos trabalhos desenvolvidos por HENDERSON (2005) e HENDERSON *et al.* (2010), foram identificadas as seguintes habilidades cognitivas e estratégias utilizadas pelos jogadores:

- (i) **Afeição (*Affect*):** representa os sentimentos despertados pelo jogo. Incluindo empatia pelos personagens;
- (ii) **Análise (*Analyzing*):** decomposição do problema (desafio) em pequenas partes;
- (iii) **Antecipação (*Anticipating*):** imaginar as possibilidades de ocorrer um evento, ou da importância de um conteúdo;
- (iv) **Aplicação (*Applying*):** considerar o uso de uma ideia ou tática em um contexto diferente;
- (v) **Categorização (*Categorizing*):** organizar itens, ideias habilidades em diferentes grupos;
- (vi) **Comparação (*Comparing*):** identificar diferenças e similaridades entre conceitos, ideias, modelos, situações, etc;
- (vii) **Confirmação (*Confirming*):** validar que as ideias presentes nos jogos dão suporte as práticas e as táticas;
- (viii) **Diagnosticar (*Diagnosing*):** identificar os pontos positivos e negativos de uma ideia, estratégias ou ponto de vista;
- (ix) **Avaliação (*Evaluating*):** perceber o valor do jogo em si, atividades, estratégias, gráficos, etc;

- (x) **Criar (*Generating*):** formular as próprias questões, ideias, problemas. Indo além do que está no jogo;
- (xi) **Imaginar (*Imaging*):** criar uma imagem mental de uma ideia para obter uma compreensão mais completa;
- (xii) **Justificar (*Justifying*):** explicar, fornecer razões para as suas ações e pensamentos;
- (xiii) **Associação (*Linking*):** vincular duas ou mais ideias , experiências, tarefas, etc;
- (xiv) **Metacognição (*Metacognizing*):** avaliar, refletir sobre a certeza de entendimento. Pensar novas maneiras de solucionar uma falta de entendimento;
- (xv) **Recordar (*Recalling*):** retomar uma ideia, opinião, fato já armazenado na memória;
- (xvi) **Reflexão (*Reflecting*):** pensar cuidadosamente sobre uma consideração já tomada;
- (xvii) **Selecionar (*Selecting*):** identificar a ideia chave ou o que é relevante para uma avaliação; e
- (xviii) **Traduzir (*Translating*):** usar as próprias palavras para interpretar, explicar o que está acontecendo na história ou com um personagem.

Essas habilidades cognitivas estratégias foram levantadas pelos pesquisadores analisando entrevistas realizadas com os jogadores. Para melhor entendimento das habilidades cognitivas e estratégias, a Tabela 4 a seguir mostra alguns exemplos de trechos das entrevistas que serviram de base para identificar a habilidade cognitiva utilizada pelo jogador.

Tabela 4. Alguns exemplos de trechos de entrevista utilizados para identificar uma determinada habilidade cognitiva e estratégia utilizada pelo jogador

Habilidade Cognitiva	Trecho da Entrevista
Afeição	“Eu estava realmente com medo.” ⁹
Comparação	“Eles sabem disso e eu não.” ¹⁰
Diagnosticar	“Huh, não funcionou, mas eu ainda consegui alguma coisa” ¹¹
Avaliação	“É difícil. É muito difícil pra eu entender corretamente.” ¹²
Metacognição	“Eu estava pensando que eu sei que preciso descobrir o que os botões fazem” ¹³
Recordar	“No início ele disse: ‘Você deve perguntar a outras pessoas’.” ¹⁴

⁹ Tradução do autor para o trecho "I was really scared." (HENDERSON *et al.*, 2010)

¹⁰ Tradução do autor para o trecho “They know it and I don't.” (HENDERSON *et al.*, 2010)

¹¹ Tradução do autor para o trecho “Huh, that didn’t work but I still got something.” (HENDERSON *et al.*, 2010)

¹² Tradução do autor para o trecho “It’s hard. It’s too hard for me to understand properly.” (HENDERSON *et al.*, 2010)

¹³ Tradução do autor para o trecho "I was thinking that I know I need to find out what the buttons do." (HENDERSON, 2005)

¹⁴ Tradução do autor para o trecho “At the start he said, ‘You should ask other people’.” (HENDERSON *et al.*, 2010)

Capítulo 4 – Computação Humana para tarefa de classificação

O avanço tecnológico tornou relativamente fácil o armazenamento de dados em quantidades sem precedentes na história da civilização. Permitiu também que organizações mantivessem informações sobre suas atividades cotidianas. As organizações podem tomar facilmente decisões estratégicas em condições de certeza limitada, se conseguirem transformar essa quantidade de dados em conhecimento útil.

Para extrair conhecimento útil é preciso explorar as grandes quantidades de informações, descobrir conhecimento em bancos de dados, do inglês, *knowledge discovery in database* (KDD). Analisar e avaliar os dados para reconhecer padrões e estabelecer relacionamentos entre as informações presentes na base de dados.

Uma das possibilidades de avaliação de dados é através da classificação dos dados em grupos de uma forma que eles possam revelar algum tipo de padrão. Esses padrões, depois de analisados, podem revelar algum conhecimento útil presente na base de dados.

4.1 – O que é classificar?

Classificação é o processo para encontrar um modelo ou fórmula que distinga as classes dos dados. O propósito desse modelo é ser capaz de determinar a classe dos dados que não possuem classificação (HAN & KAMBER, 2006). O modelo obtido é baseado na análise de dados que a classificação já é conhecida. Esse conjunto de dados é conhecido como dados de treinamento.

O modelo de classificação pode ser representado de diversas formas: como um conjunto de regras, como árvores de decisão, como fórmulas matemáticas ou como redes neurais. As árvores de decisão podem ser facilmente transformadas em um conjunto de regras. Existem outros métodos para construir modelos de classificação, como o vizinho mais próximo (*k-nearest neighbor*, K-NN), o método bayesiano e a máquina de vetores de suporte.

O processo de classificação de dados é composto de dois passos. O primeiro passo consiste em analisar os dados de treinamento para obter uma fórmula geral de classificação. O

segundo passo é utilizar o classificador obtido no primeiro passo para determinar as classes dos dados sem rótulos.

O primeiro passo é chamado de treinamento. É o momento no qual um algoritmo de classificação analisa os rótulos e os dados do conjunto de treinamento e infere uma fórmula de classificação $y = f(x)$ que seja capaz de definir a que classe y um determinado dado x pertence. Sendo esse dado x um vetor n -dimensional.

Esse primeiro passo também é conhecido com aprendizado supervisionado (HAN & KAMBER, 2006). Já que o aprendizado do classificador é “supervisionado” pelas classificações já existentes dos dados de treinamento. O algoritmo pode comparar suas respostas com o rótulo esperado e ajustar o modelo de forma a maximizar os acertos. Para questão de comparação, o processo de clusterização de dados é um aprendizado não-supervisionado, pois não utiliza dados de treinamento rotulados.

O segundo passo é utilizar o modelo obtido para realizar a classificação dos dados não rotulados. Contudo, é preciso estimar a precisão do classificador. Para calcular a precisão do modelo é utilizado um conjunto de teste, que é composto por dados diferentes dos dados do conjunto de treinamento. A precisão do modelo de classificação é determinada pela porcentagem de classificações corretas obtidas no conjunto de teste (HAN & KAMBER, 2006).

A classificação de dados é caracterizada por rótulos com valores discretos (classes), onde a ordem entre os valores não tem significado. Contudo, quando as saídas do modelo são valores contínuos é caracterizada uma predição. Estimar temperatura é um exemplo de predição.

4.2 – Base dados disponíveis para tarefa

Classificação pode ser aplicada em inúmeras situações, como diagnóstico de doenças, análise de crédito, detecção de fraudes, campanhas de marketing direto, entre outras. No caso do marketing direto, baseado em uma amostra de clientes, pode-se criar um modelo que permita o direcionamento de ações de marketing para os clientes com provável aceitação de um produto ou serviço.

A classificação é uma ferramenta útil em diversas áreas e seus benefícios, limitações e desafios são bastantes conhecidos. Por isso, ela é estudada em diferentes áreas acadêmicas como, por exemplo, estatística, inteligência artificial e bancos de dados. E para facilitar os estudos nessa área, existe um grande número de bases de dados disponíveis para esse fim.

4.3 – Problemas e desafios encontrados em classificação

Um dos problemas de classificação é a distribuição desbalanceada de padrões entre as classes. O problema de classes desbalanceadas corresponde a domínios nos quais uma classe é representada por um grande número de exemplos, enquanto que a outra é representada por poucos exemplos (BATISTA, 2003).

Esse problema é importante quando se trata da análise de fraudes, por exemplo. O grupo de transações legítimas possui um grande número de exemplos, praticamente a totalidade de casos, enquanto que o conjunto de dados de exemplos de transações fraudulentas é bem pequeno.

Os sistemas falham em induzir um modelo que seja capaz de prever a classe minoritária com precisão na presença de dados com classes desbalanceadas, pois assumem que as classes estão relativamente balanceadas (BATISTA, 2003). Em geral, a tendência é produzir modelos de classificação que favorecem as classes com maior probabilidade de ocorrência (CASTRO & BRAGA, 2011). O classificador acaba tendo uma boa precisão para dados da classe dominante e um desempenho ruim nas classes minoritárias.

Outro problema é de conseguir um conjunto de treinamento adequado para obter um modelo classificador. Em alguns casos não é possível garantir que os dados da amostragem utilizada para treinamento representem adequadamente o sistema como um todo (SILVA, 2008).

Um exemplo que ilustra bem esse problema é o de classificação de textos da Internet. Nesse caso é complexo e extremamente custoso obter um conjunto de treinamento suficientemente abrangente que represente de forma adequada o conjunto completo. O espaço de estudo é demasiadamente grande para ser possível gerar uma amostra estatisticamente

representativa (SILVA, 2008). Uma amostragem limitada do conjunto de treinamento não fornece informação suficiente para a indução de um modelo classificatório com boa precisão.

Alguns conjuntos de treinamento tem sua representatividade comprometida, pois o processo para obtenção de dados classificados para compor o conjunto de treinamento é extremamente custoso ou complexo. É possível aumentar a representatividade do conjunto de treinamento através de classificação manual, utilizando pessoas para realizar a classificação dos dados até que se tenha uma quantidade ideal. Contudo esse processo demanda tempo e recursos.

4.4 – Sistema humano de classificação

Na seção 3.3 foram apresentadas habilidades cognitivas e estratégias utilizadas pelos jogadores enquanto jogam. A classificação não aparece explicitamente como uma dessas estratégias. Porém, é possível identificar a presença da classificação combinando algumas habilidades. Para demonstrar isso é mostrado o processo de classificação baseado nas habilidades cognitivas e estratégias encontradas nos trabalhos de HENDERSON (2005) e HENDERSON *et al.* (2010).

Para realizar uma classificação qualquer, uma pessoa **(i)** identifica diferenças e similaridades entre conceitos, itens, modelos..., essa ação equivale às habilidades de comparar e diagnosticar; em seguida, **(ii)** a pessoa organiza os conceitos, itens e modelos em diferentes grupos, o que equivale à habilidade de categorização; depois, **(iii)** identifica as características relevantes para a avaliação, que equivale à habilidade selecionar; e por fim, **(iv)** vincula esse novo conceito ou item analisado ao grupo de semelhantes, o que é equivalente à habilidade de associação.

Esse processo de classificação quando aplicado no jogo ainda é seguido pela habilidade de antecipação. Na qual, depois de associar um novo conceito ou item com os já vistos anteriormente no jogo, o jogador imagina o que pode vir a ocorrer ou a importância desse conceito ou item.

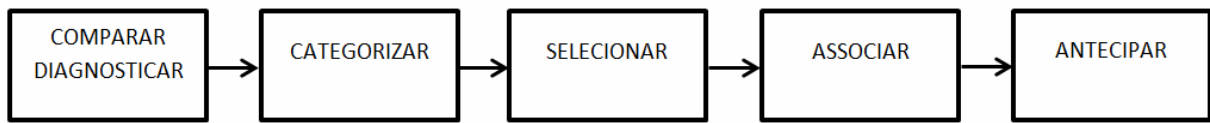


Figura 7. Classificação baseadas em habilidades cognitivas e estratégias utilizadas pelos jogadores quando jogam.

Nesse processo de classificação descrito baseados nas habilidades cognitivas e estratégias identificadas nos trabalhos de HENDERSON (2005) e HENDERSON *et al.* (2010) fica clara a presença dos dois passos do processo de classificação. O primeiro passo, o de treinamento, corresponde aos passos (i), (ii) e (iii). Já o passo (iv) equivale à classificação dos dados em si. E o passo de antecipação seria a aplicação do conhecimento útil adquirido.

Classificação está presente nos jogos mais populares e auxilia na tomada de decisões. Em um jogo hipotético, o jogador se depara com o inimigo. Baseado nas características do inimigo e nos inimigos que ele já enfrentou durante o jogo, ele classifica esse inimigo como um determinado tipo de inimigo. Assim, o jogador está apto a escolher o ataque mais eficiente contra inimigos desse grupo de inimigos.

Já no jogo de computador *Counter Strike*¹⁵ os jogadores são divididos em dois grupos, o de policiais e o de terroristas. Os jogadores do grupo de policiais devem matar os jogadores do grupo de terroristas e vice-versa. Diante de um personagem, o jogador deve definir se ele é ou não do mesmo grupo que o seu baseado nas características (roupas, principalmente) do personagem.

Na série de jogos *Assassin's Creed*¹⁶, existe um modo de jogo *multiplayer* no qual, em um mesmo ambiente, estão personagens controlados por jogadores e pela máquina. O jogador, para somar pontos, deve matar somente os personagens controlados por outros jogadores. Ele deve observar o comportamento dos personagens para decidir se ele é ou não controlado por outro jogador.

Como as informações passadas pelos jogadores são majoritariamente visuais, a classificação nos jogos depende de características identificadas nos elementos do jogo. Com

¹⁵ <http://www.counter-strike.net>

¹⁶ <http://assassinscreed.ubi.com>

isso, o jogador deve prestar atenção no que está sendo mostrado pra ele, desde características marcantes como roupas e cores, até características mais sutis, como o comportamento e ações. De uma forma mais clara ou não, a classificação é uma estratégia utilizada pelos jogadores.

Jogos com propósito estão sendo utilizados pra auxiliar na resolução de problemas computacionais. O ato de classificar já está presente nos jogos. Com isso, explorando todo o potencial dos jogos com propósito, pode-se propor um sistema humano que dê suporte ao processo de classificação de dados.

A ideia desse sistema é ser um jogo com propósito focado em classificação de dados. Enquanto o jogador está jogando uma partida, sem perceber, estaria classificando os dados que são apresentados a ele. Pode-se imaginar o jogo como uma “caixa preta” que classifica dados. Assim, a entrada do jogo seria a base de dados que se deseja classificar e a saída, seria uma base de dados classificada.

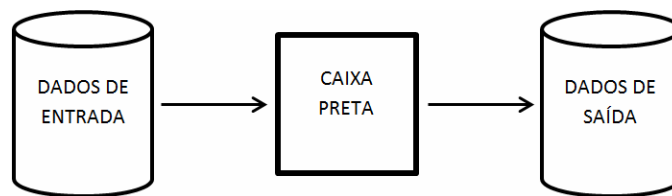


Figura 8. Visão geral do sistema humano de classificação de dados.

A base de dados de entrada deve conter todos os conjuntos de dados utilizados num processo tradicional de classificação: o conjunto de treinamento, o conjunto de teste e o conjunto de dados a serem classificados.

Um dos desafios percebidos, num jogo onde o foco é a classificação de dados, é apresentar os dados a serem classificados ao jogador. Além desses dados serem mostrados de forma amigável, eles devem ser incorporados ao jogo. Por isso, o sistema deve ter um passo, na elaboração do jogo, que seja responsável por representar graficamente os dados.

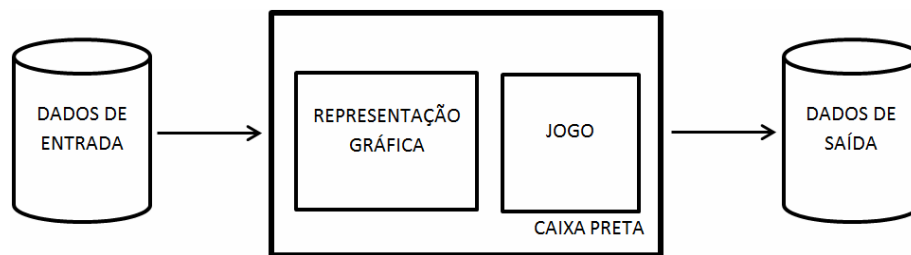


Figura 9. Visão detalhada (representação gráfica + jogo) do sistema humano de classificação de dados.

Ao final de cada partida jogada obtêm-se os resultados dos jogadores, junto com os dados classificados por ele. Como uma pessoa é passível de erros, é importante combinar as classificações de todos os jogadores com o intuito de diminuir o impacto dos erros, caso ocorram. Assim, percebe-se a necessidade de um último passo para compilar as respostas dadas pelos jogadores durante as partidas e informará uma classificação final.

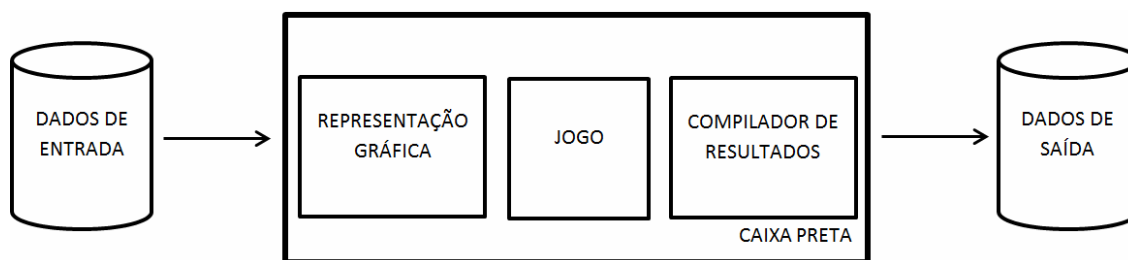


Figura 10. Visão detalhada (representação gráfica + jogo + compilador de resultados) do sistema humano de classificação de dados.

É preciso entender o problema de classificação de dados para conseguir elaborar um jogo com propósito que possa ser classificado de acordo com os gêneros de jogos mais populares. Além disso, o jogo deve ser organizado de forma que garanta que as respostas obtidas através dele sejam confiáveis. No capítulo a seguir isso será explorado mais detalhadamente.

Capítulo 5 – Procedimentos de elaboração do jogo com propósito

Entre os dez jogos mais vendidos de 2012 estão jogos de tiro (4 de 10), esportes (3 de 10), aventura (2 de 10) e dança (1 de 10). Isso mostra que os jogos mais consumidos possuem um gênero bem definido. Ao contrário do que acontece com os jogos com propósito, que não se enquadram em grandes gêneros de jogos e acabam formando um gênero a parte.

As principais mecânicas de jogo encontradas nos jogos com propósito são diferentes das mecânicas dos jogos mais populares. Os jogos convencionais possuem as mecânicas características dos gêneros de jogos a que pertencem. Já as mecânicas encontradas nos jogos com propósito costumam reproduzir a solução real do problema abordado através do jogo.

Como visto na seção 3.2, a única mecânica comum entre os jogos com propósito analisados e os jogos estudados por DILLON (2010) é a mecânica de mover. Contudo, a mecânica de mover é uma das mais básicas dos jogos, é a forma mais clara de permitir a interação do jogador com o jogo.

Pelas mecânicas identificadas nos jogos com propósito, fica nítido que a solução do problema real foi inserida na mecânica de jogo de uma forma muito direta. O que acaba diminuindo o poder da solução lúdica que pode ser explorada através dos jogos. Assim, deixa-se de explorar características importantes dos jogos. Jogar um jogo com propósito passa a ser a realização de passos de um algoritmo de uma forma camuflada, de uma maneira que se torne interessante para as pessoas.

Os jogos possuem elementos como, por exemplo, pontuação e ranking, que buscam aumentar o interesse e inserir novos aspectos e desafios aos jogadores. A pontuação no jogo estimula o jogador a conseguir um desempenho melhor toda vez que ele joga uma nova partida. O ranking serve para comparar o desempenho com os outros jogadores e estimular a competitividade entre eles. Existem outros elementos como esses presentes nos jogos. Da mesma forma que os jogos tradicionais, os jogos com propósito conseguem incorporar esses elementos.

Os jogos com propósito se focam, principalmente, nesses elementos extras para manter o interesse do jogador no jogo por mais tempo. Todos os jogos com propósito estudados possuem, por exemplo, alguma característica que estimule a interação e/ou competição entre os jogadores.

Nos jogos com propósito é interessante que se tenha um grande número de pessoas jogando durante bastante tempo um determinado jogo. Isso aumenta o número de soluções para o problema que o jogo com propósito pode oferecer. Contudo, ao se focar somente nos elementos extras para atingir esse objetivo, acaba-se subutilizando todo o poder lúdico, de imersão e interação que os jogos podem proporcionar. Todas essas características contribuem para que os jogadores se interessem mais pelo jogo e, conseqüentemente, passem mais tempo jogando.

A seguir são apresentados procedimentos para o projeto e a organização de jogos com propósito. Esses procedimentos visam facilitar a forma de se explorar todas essas outras características dos jogos, distanciar os jogos com propósito da possível solução do problema e aproximá-lo dos jogos mais tradicionais.

Todos os procedimentos foram organizados em quatro modelos. O primeiro modelo aborda, como um todo, o processo de solução de problemas computacionais reais utilizando jogos com propósito baseados em mecânicas tradicionais de jogos. Para facilitar o entendimento, atividades presentes nesse processo foram aprofundadas em modelos independentes. Como é o caso das atividades Projetar Jogo e Aplicar Jogo. O último modelo é referente a atividade Organizar Jogo presente no modelo Projetar Jogo.

Esses procedimentos possuem duas características principais. A primeira delas é voltada para a transposição dos dados e informações do problema real a ser abordado para o ambiente do jogo. A outra é a utilização dos dados no jogo de uma forma que se possa atingir a solução do problema a ser resolvido.

No decorrer desse capítulo todos os procedimentos serão aprofundados.

5.1 - Procedimento de solução de problemas computacionais utilizando Jogos com Propósito

Uma característica dos jogos com propósito destacada nesse trabalho é que a mecânica de jogo adotada por eles refletem diretamente a solução do problema abordado. Com isso, se torna difícil classificar os jogos com propósito dentro dos gêneros de jogos mais tradicionais.

Esse procedimento mostra como deve ocorrer a solução de problemas computacionais utilizando jogos com propósito. Inicialmente, deve-se projetar o jogo que será utilizado. Em seguida é importante definir o público alvo do jogo, bem como toda a estrutura necessária que permita que os jogadores utilizem o jogo. O próximo passo é permitir que os jogadores joguem o jogo. Em seguida, os resultados obtidos são analisados para se obter a solução do problema. Por último, todo o processo é analisado para identificar pontos positivos e negativos para que as execuções futuras melhorem.

A Figura 11 mostra o procedimento de solução de problemas computacionais utilizando jogos com propósito. Todas as atividades serão exploradas detalhadamente no decorrer dessa seção.

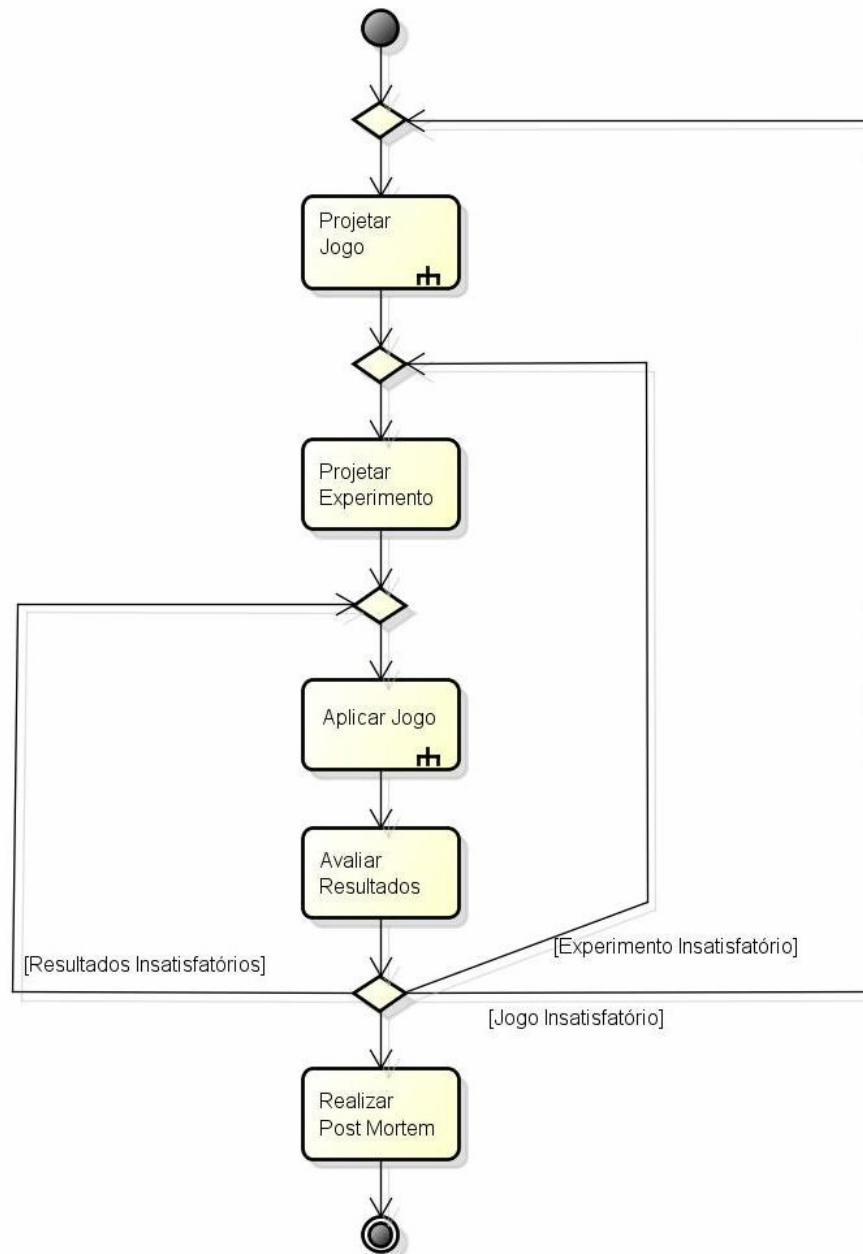


Figura 11. Procedimento para solução de problemas computacionais utilizando jogo com propósito.

- **Projetar Jogo**

O jogo com propósito é a ferramenta que será utilizada para encontrar a solução do problema. Por isso, o primeiro passo do procedimento é desenvolver o jogo com propósito que vai ser utilizado no processo. Essa ferramenta deve ser adequada ao problema que se

deseja solucionar. Por isso, além de desenvolver o jogo com propósito, deve-se realizar um estudo aprofundado do problema.

Existem elementos que caracterizam os jogos. Por isso, a solução lúdica encontrada como ideal para solucionar o problema deve ser organizada de uma maneira que se forme um jogo. Por exemplo, os jogos iniciam com desafios mais simples e no decorrer da partida vão se tornando mais complexos. O ideal é que o jogo com propósito também siga esse comportamento.

Além de todos os processos de desenvolvimento e organização do jogo, são realizados testes em todas as etapas para garantir que o jogo tenha o comportamento esperado e atinja o principal objetivo que é indicar uma solução correta para o problema abordado.

Sob a perspectiva desse trabalho, essa atividade é a mais importante do processo, pois além do desenvolvimento do jogo com propósito propriamente dito, é nesse momento que as mecânicas de jogos tradicionais serão inseridas no *gameplay* do jogo. Pela sua importância e complexidade, essa atividade é tratada em um modelo próprio que será aprofundado na próxima seção.

- **Projetar Experimento**

Nesse momento o jogo está pronto e apto para ser jogado. Mas antes de disponibilizar o jogo publicamente é importante definir alguns pontos para não ser surpreendido durante o período que o jogo está sendo jogado pelas pessoas.

Primeiramente, deve-se definir como o jogo será disponibilizado para o público. As respostas dadas pelos jogadores serão utilizadas pelo jogo com propósito para definir a solução final do problema. Essas respostas deverão ser armazenadas para análises futuras. Uma vez que isso foi definido, deve-se preparar toda a infra estrutura para receber as respostas dos jogadores. Deve-se garantir que todo o resultado produzido pelos jogadores sejam armazenados, não é interessante que o jogador jogue uma partida, produza resultados e esses sejam perdidos. No caso de um jogo que tenha um servidor que irá receber as respostas do jogadores, esse servidor deve estar sempre em serviço e ter a capacidade de suportar o número de jogadores.

O ideal é que o jogo seja acessível para todas as pessoas. Porém, um problema mais complexo, a interface ou o *gameplay* utilizados podem exigir um conhecimento prévio ou um dispositivo específico. Para esses casos em que o jogo possui restrições, deve-se definir precisamente o público alvo para evitar os casos que não atendem os requisitos produzam influências negativas no resultado final.

Nos casos que exista a necessidade de um dispositivo específico, deve-se preocupar se os jogadores possuem esses dispositivos ou se precisam de algum auxílio para maneja-los. Nos casos que o acesso a esses dispositivos é complicado é preciso disponibilizar acesso a eles e ter alguém para dar suporte aos jogadores. Isso deve ser providenciado para não causar interferências na obtenção dos resultados.

Com o público alvo definido é necessário convocar as pessoas para utilizar os jogos. Deve-se definir a melhor maneira de realizar esse convite e certificar que o convite chegue aos interessados de fato. Um convite atraente desperta o interesse dos jogadores em participar do jogo e colaborar com a solução do problema.

Por último, deve-se estabelecer o período que o jogo estará disponível para ser jogado. Esse período deve ser suficiente para que se consiga determinar com uma precisão definida a solução do problema. O período de tempo varia de acordo com a eficiência do jogo. VON AHN & DABBISH (2008) definem o rendimento de um jogo com propósito como sendo o número de soluções obtidas por homem hora. Durante os testes é possível definir o rendimento do jogo e assim estabelecer com mais precisão o tempo ideal que o jogo ficará disponível para ser jogado.

Ao final desse processo foram tratados fatos como, por exemplo, infra estrutura, público alvo e tempo de duração, para diminuir o risco de que problemas aconteçam durante a aplicação do jogo e coleta de respostas.

- **Aplicar jogo**

Essa atividade representa o momento em que o jogo está disponível para ser jogado pelos jogadores e, conseqüentemente, coletar as respostas deles. Como foi mencionado na

atividade Projetar Jogo, que será melhor detalhada na próxima seção, o jogo precisa ser organizado. É neste momento que entendemos melhor a importância dessa organização.

A estrutura de organização seguida procura ensinar e mostrar para o jogador o comportamento que se espera dele enquanto ele joga o jogo. Para que as decisões tomadas pelos jogadores estejam dentro do que foi planejado no jogo. Ainda dentro dessa estrutura, enquanto o jogo está colhendo as respostas do jogador, ele está realizando um teste de desempenho com o jogador.

O jogador é testado para ajudar a definir a qualidade das respostas obtidas desse jogador. O teste possibilita que se descarte as respostas de jogadores com um desempenho ruim. Assim, melhora a qualidade da solução que será encontrada para o problema.

As respostas obtidas de jogadores com desempenho acima do mínimo exigido são separadas das respostas dos jogadores com desempenho abaixo. A solução final é obtida através da análise das respostas dos melhores jogadores. As respostas dos jogadores de desempenho baixo serão utilizadas apenas para o cálculo de estatísticas do jogo.

Essa atividade possui um modelo específico para ela que ajuda a explicar melhor toda a estrutura de ensino, treino e teste do jogador. Na seção 5.4 – Procedimento de aplicação do jogo, esse processo será estudado mais detalhadamente, abordando, inclusive, a validação de desempenho do jogador.

- **Avaliar Resultados**

Os jogos com propósito são ferramentas de computação humana. Logo, uma de suas características é a participação das multidões. Nesse momento espera-se que o jogo tenha sido jogado por diversas pessoas. Por isso, deve-se agrupar e compilar todos os resultados dos jogadores que participaram.

As respostas dos jogadores que tiveram o desempenho abaixo do desejado já foram separadas. Sobraram, teoricamente, as melhores respostas que puderam ser coletadas. Para chegar à solução final do problema deve-se levar em consideração todas as respostas dadas por todos os jogadores selecionados.

Como as respostas são dadas por seres humanos, elas podem causar alguma desconfiança sobre a solução final. Para aumentar a confiabilidade na solução obtida através do jogo com propósito deve-se informar a precisão do sistema. Por isso, são contabilizadas somente as respostas dos jogadores que obtiveram o desempenho melhor que o estipulado.

É preciso analisar se é possível chegar a uma solução utilizando as respostas obtidas através do jogo. Verificar se as respostas obtidas pelo jogo são suficientes para determinar uma solução final com um determinado grau de certeza. Essa verificação se torna importante para evitar que ações tomadas de forma aleatória pelo o jogador influenciem na solução final.

Uma forma de determinar a certeza da resposta é utilizar a distribuição normal das respostas obtidas dos jogadores e a distribuição normal aleatória das possíveis respostas do problema. Deve-se comparar as duas distribuições para chegar ao grau de certeza da resposta obtida pelo sistema. Quanto mais distante a distribuição normal das respostas obtidas dos jogadores estiver da distribuição normal aleatória, mais certeza se tem da resposta obtida. Quanto mais similares forem as duas distribuições menos certeza se tem da resposta obtida pelo sistema.

Quando o grau de certeza for baixo pode-se executar mais vezes o jogo a fim de aumentar a certeza da resposta. O grau de certeza pode servir, também, como um indicador de parada para a obtenção de resposta para um determinado item. Esse item então, seria substituído por um outro.

É importante deixar claro que o grau de certeza não está relacionado à correção da resposta. Ele informa que as respostas obtidas dos jogadores são suficientes para se chegar a uma solução, não necessariamente a uma solução correta. Em outras palavras, o jogo pode indicar, com um determinado grau de certeza, uma resposta errada como solução do problema.

Estando com as melhores respostas dos jogadores e tendo certeza que essas respostas levarão a um resultado, está na hora de determinar a solução final do problema. Todas as respostas selecionadas devem ser combinadas de maneira a levar a uma solução. Existem combinações que são determinadas pela maioria (também conhecida como votação), pela média ponderada, pela soma, pelo produto, entre outras. Cabe decidir pela melhor forma que se aplique as respostas obtidas e leve a solução final.

Foram vistos os tratamentos de dados referentes ao problema a ser resolvido pelo jogo com propósito. Também é interessante realizar estudos sobre o desempenho do jogo em si. Para isso deve-se obter dados a respeito de diversas características do jogo. Essas informações servem para análise do jogo propriamente dito e auxilia a determinar melhorias e ajustes que podem ser realizados.

Medidas comuns para a realização desse estudo são, por exemplo, a média, a mediana, a moda, os valores mínimo e máximo, porcentagem, variância entre outros. O tempo médio que o jogador permanece jogando, os valores mínimo e máximo de respostas certas dos jogadores e a resposta mais comum (moda) dada pelos jogadores, são exemplos de dados estatísticos a respeito do jogo que podem ser interessantes de serem obtidos.

Essas informações servem de material para análise sobre o jogo em si. Elas colaboram positivamente nas discussões e nas decisões que ocorrem na próxima atividade do procedimento.

- **Realizar *Post Mortem***

O *post mortem* é uma atividade comum ao final de grandes produções e projetos. É o momento em que as pessoas envolvidas se reúnem para avaliar o projeto, identificar o que aconteceu de errado e o que pode ser realizado de uma forma diferente. Essa atividade é importante para evitar que problemas se repitam.

Nesse procedimento não será diferente. As informações obtidas na reunião de *post mortem* devem ser utilizadas para implementar alterações com o objetivo de melhorar o processo nas próximas execuções.

É interessante a participação de todos os envolvidos no processo. Cada integrante tem uma visão particular dos fatos que ocorreram durante o processo. Podendo contribuir com alguma observação que passou despercebida pelos outros.

Ao realizar esses procedimentos espera-se encontrar uma solução para o problema abordado. Como pode se notar, existem atividades que são mais complexas e que para melhor entendimento foram abordadas em modelos específicos. A seguir, será abordado de forma mais completa o processo de Projetar Jogo.

5.2 – Procedimento de projeto do jogo

O importante na fase da elaboração do jogo com propósito é entender o problema e os dados do problema com o objetivo de conseguir incluir mecânicas de jogos mais tradicionais, como atirar, pular, entre outras, de uma maneira natural no *gameplay* pretendido para o jogo.

Dando uma visão geral do procedimento de desenvolvimento do jogo, primeiro, deve-se entender bem o problema a ser resolvido. Entender o problema, passa por entender as informações disponíveis do problema e os dados que serão utilizados. Em seguida, deve-se escolher o modo de incluir os dados no jogo e definir a forma que os dados vão assumir dentro do jogo, para, assim, estabelecer a melhor forma de transformar os dados em elementos do jogo. Feito isso, pensar no *gameplay* que utilize esses elementos no jogo seguindo as mecânicas de jogos mais conhecidas. Por fim, deve-se organizar a forma que os dados serão apresentados para o jogador. Não menos importante, testes devem ser realizados ao longo do processo para garantir o funcionamento do que foi definido.

A Figura 12 mostra uma visão geral do procedimento de projeto do jogo. No desenvolvimento desse capítulo será explorado detalhadamente cada passo desse processo.

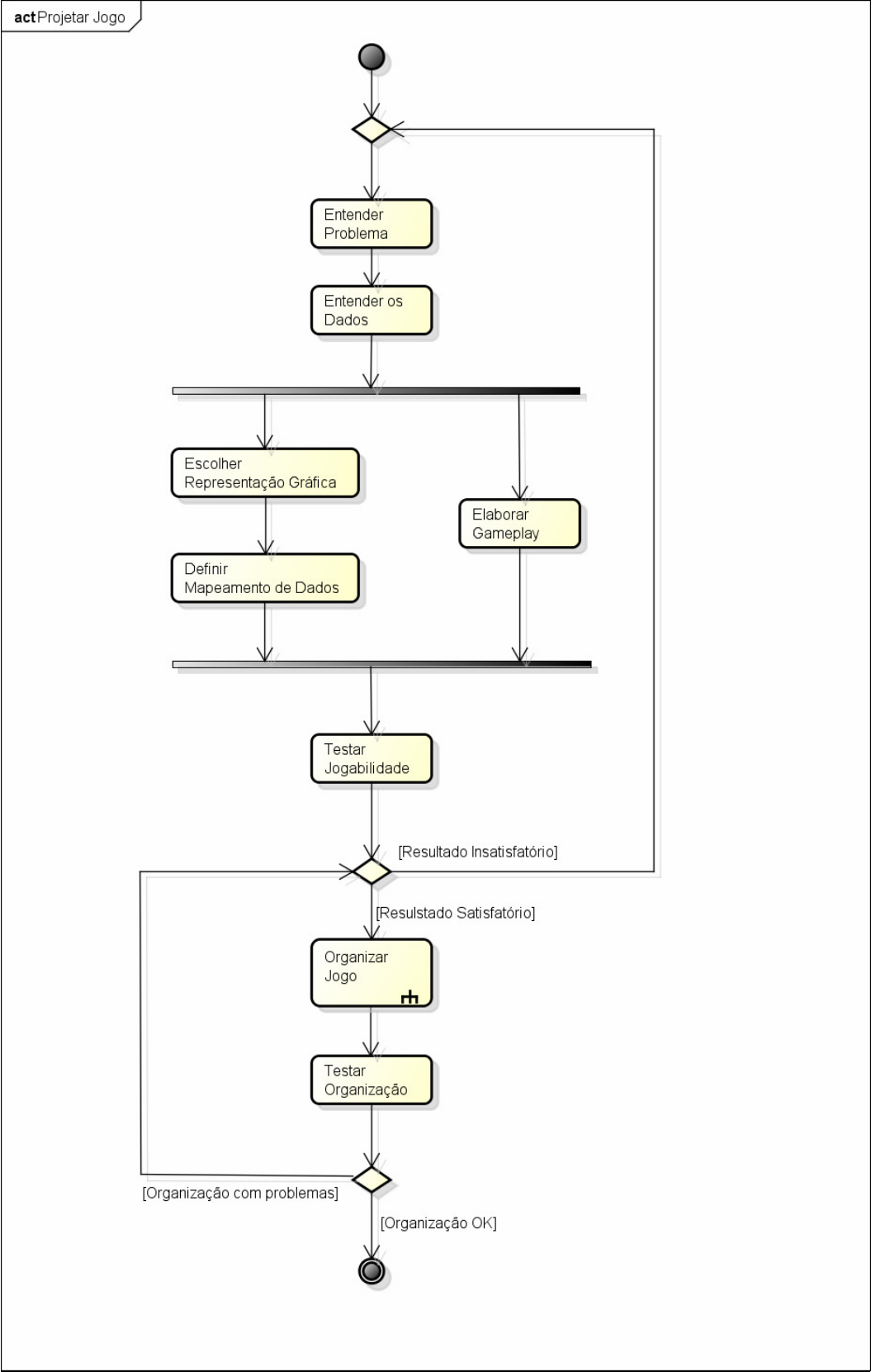


Figura 12. Procedimento de projetar o jogo

- **Entender o problema**

O jogo com propósito tem que permitir que uma determinada entrada se torne uma saída após a interação realizada pelo jogador. Para se justificar a utilização do jogo, a saída pretendida não pode ser capaz de ser obtida por máquinas. (VON AHN, 2005) Nos casos em que os resultados também podem ser encontrados através de algoritmos, a utilização dos jogos com propósito se justifica caso esse método seja mais rápido ou mais preciso.

Sendo assim, não são todos os problemas que são passíveis de serem resolvidos utilizando jogos com propósito. Além disso, nem todos os problemas que podem ser resolvidos através de jogos com propósito se encaixam nesse modelo proposto. Por isso, deve-se estudar e entender o problema a ser resolvido.

Para auxiliar no entendimento do problema é preciso ter de forma clara dois pontos: **(i)** a questão que pretende ser respondida; e **(ii)** a solução que se espera encontrar. Encontrar a solução do problema é o principal propósito do jogo. Por isso é importante que ela esteja bem definida desde o primeiro momento.

O problema tem informações que o caracterizam. Os dados do problema fazem parte dessas informações. No próximo passo, o principal foco é o entendimento desses dados. Porém, nesse momento, é importante saber se os dados disponíveis são suficientes para se obter a solução pretendida. E, se for necessário, se é possível obter mais informações que auxiliem na modelagem do problema e na obtenção dos resultados esperados.

É preciso identificar os passos para encontrar a solução do problema. Esses passos são importantes para definir se é possível ou não utilizar jogos com propósito para se obter uma solução para o problema. Esses passos também serão utilizados no momento de definição de *gameplay* do jogo. Eles auxiliarão na identificação de mecânicas a serem utilizadas. Uma boa fonte para identificar esse passo-a-passo é o procedimento que já vem sendo utilizado para solucionar o problema.

Ao final dessa atividade é importante ter definidos as seguintes informações:

1. O problema ser solucionado;
2. A solução esperada;
3. A completude das informações que caracterizam o problema; e

4. O passo-a-passo para obter a solução desejada.

De posse de todas essas informações em relação ao problema, já é possível definir se os jogos com propósito são capazes de auxiliar na obtenção dos resultados esperados. Estando apto a continuar ao próximo passo do procedimento de projeto do jogo.

- **Entender os dados**

Definir a melhor maneira de utilizar os dados no jogo. Saber qual informação é mais relevante e, assim, escolher a melhor forma de inseri-la no jogo. Escolher quais são os dados menos importantes, podendo subutilizá-los para melhorar a experiência sem comprometer os resultados a serem obtidos através do jogo. Todas essas decisões são tomadas com maior certeza quando se compreende o conjunto de dados como um todo.

Antes de utilizar os dados nos jogo é preciso estudá-los. Utilizar análises clássicas de outras áreas como estatística, por exemplo, é uma opção válida. O estudo dos dados visa identificar se existe alguma relação entre as variáveis, alguma tendência de comportamento, algum agrupamento especial, alguma estrutura subjacente ou dados muito diferentes dos restantes.

O principal foco é entender a estrutura do conjunto de dados para auxiliar na inserção deles no jogo. Porém, os dados possuem significados também. Se for possível estender a compreensão ao nível do significado dos dados, a utilização dos dados pode ser explorada de uma forma mais abrangente. Por exemplo, no caso de dados relacionados a temperatura, é interessante explorar o conhecimento popular que aceita que cores mais vibrantes, como o vermelho, estão ligadas a altas temperaturas e cores , como o azul, estão relacionadas a baixas temperaturas.

Mais adiante os dados se tornarão elementos do jogo. Para facilitar a inclusão dessa informação no jogo é interessante que os dados do problema sejam quantitativos. Por isso, nesse momento, também é importante identificar se é possível converter os dados qualitativos em dados quantitativos. Dados quantitativos apresentam menos dificuldades para serem incorporados no jogo.

Conhecer as principais características e as disposições dos dados permite definir a melhor maneira de utilizá-los no jogo. Todas as informações relacionadas aos dados obtidas nesse passo servirão de base para as decisões tomadas futuramente. Decisões essas de como representar os dados e a melhor maneira de realizar esse mapeamento na representação. Esses próximos passos serão detalhados a seguir.

- **Escolher a representação gráfica**

Representar graficamente o conjunto de dados pode facilitar a análise das características, tendências e padrões de comportamento dos dados. Possibilitando a identificação de informações relevantes.

Dados n -dimensionais, com n pequeno, podem ser representados das maneiras mais tradicionais como gráficos lineares, de barras e de pizza sem maiores problemas. Contudo para dados n -dimensionais com n grande, devem-se utilizar representações gráficas específicas para dados multidimensionais.

Dentre as diferentes possibilidades de representação gráfica de dados multidimensionais existem aquelas que procuram representar os dados em formas já conhecidas e familiares aos seres humanos. Existem representações baseadas no formato das mãos (KABULOV & TASHPULATOVA, 2010), de castelos (KLEINER & HARTIGAN, 1981), de árvores (KLEINER & HARTIGAN, 1981), de rostos (CHERNOFF, 1973), dentre outros.

Esses tipos de representação dos dados combinam com a ideia lúdica presente nos jogos com propósito. Permitem a inserção dos dados de uma forma mais suave no ambiente do jogo. Porém, inserção dos dados do problema no jogo não precisa se restringir as formas da representação gráfica. Pode-se explorar outras características visuais como cores, por exemplo.

As percepções cognitivas dos seres humanos também são influenciadas pelo som. A representação sonora também pode ser utilizada para incluir dados nos jogos. Contudo, a preferência é utilizar o som como um canal de *feedback* para o jogador. É melhor empregado quando mostra ao jogador se ele está agindo do modo esperado ou não.

A forma da representação gráfica escolhida influencia na temática do jogo. Os dados se transformam em elementos dos jogos e passam a fazer parte do ambiente de jogo, de modo que os jogadores podem e devem interagir com eles. Características do jogo devem ser concebidas para que os elementos oriundos dos dados possam ser inseridos sem perder a coerência do mundo do jogo.

Foram citadas formas de representação gráficas encontradas na literatura, porém é possível desenvolver uma nova representação que se enquadre melhor nas características do jogo. Cada nova representação gráfica criada para ser utilizada num determinado jogo, é mais uma possibilidade de representação que pode ser utilizada em um outro jogo.

A representação gráfica e o *gameplay* possuem uma relação forte. A representação dos dados influencia o *gameplay* e vice-versa. Portanto, é preciso reavaliar as decisões tomadas na representação dos dados quando ocorrer alguma mudança significativa no *gameplay*. O caso contrário também deve ser reavaliado, quando ocorre uma mudança na representação gráfica o *gameplay* deve ser analisado.

As modificações ocorridas tanto no *gameplay* quanto na representação gráfica também devem ser observadas sob a perspectiva do mapeamento dos dados. Esse relacionamento de influência entre as atividades fica evidenciado no modelo através do paralelismo entre as atividades de Escolher Representação Gráfica, Definir Mapeamento de dados e Elaborar *Gameplay*.

- **Definir o mapeamento dos dados**

A representação gráfica é transformar uma informação de um modo mais complexo para uma forma mais atraente e fácil de ser entendida. É o mapeamento de uma dimensão dos dados em uma característica da representação gráfica. Por exemplo, é desenhar o tamanho da boca, numa representação gráfica que forme um rosto, de acordo com o salário de uma tabela de funcionários, como mostra a Figura 13 (a).

Mas será que o tamanho da boca é a característica mais adequada para representar o salário dos funcionários? A resposta para essa pergunta é subjetiva. Já que as informações

visuais serão interpretadas por pessoas, e as pessoas possuem percepções distintas umas das outras.

Os estudos realizados na atividade Entender os dados e na atividade para compreender as características da representação escolhida servem de base para decidir o mapeamento a ser feito. É interessante criar mais de um mapeamento para ser utilizado e realizar testes com os diferentes mapeamentos e grupos de pessoas. O importante nesse momento é verificar se todos os dados relevantes estão sendo representados e se podem ser percebidos.

Para agregar mais informação é interessante utilizar uma representação implícita para representar os dados. Continuando com o exemplo dos funcionários e a representação através do rosto. Teoricamente, o funcionário que ganha mais está mais satisfeito que o funcionário que ganha menos. O rosto poderia refletir isso do modo que o sorriso correspondente ao funcionário que ganha mais seja mais “feliz” que o sorriso do funcionário que ganha menos. Como pode ser visto na Figura 13 (b).

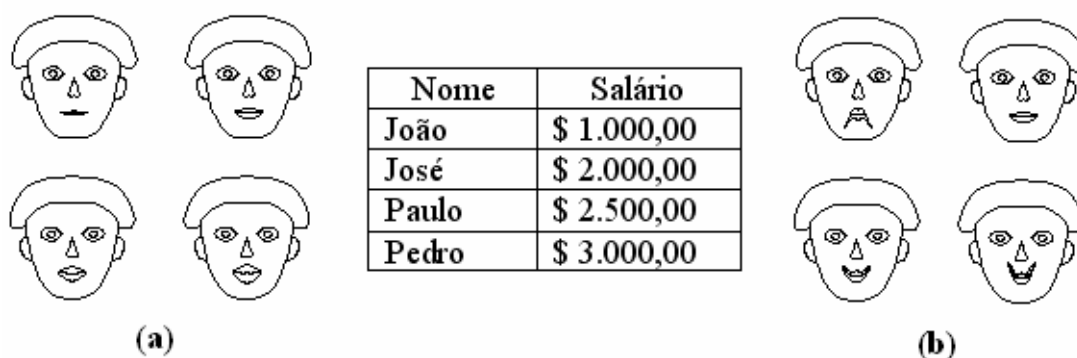


Figura 13. Exemplo de representação gráfica utilizando rostos. (a) Exemplo de mapeamento do salário com o tamanho da boca. (b) Exemplo de inserção de significado na representação gráfica através do mapeamento do salário com o sorriso.

Esse ponto está relacionado com o levantamento de significado dos dados abordado anteriormente na atividade de Entender os dados. Contudo, deve-se ter um cuidado extra com esse mapeamento para evitar interpretações equivocadas que tirem o foco do jogo. Por exemplo, ao representar dados referente à violência utilizando cores, evitar que um dado que esteja relacionado a uma taxa de maior violência esteja ligada a cores que induzam determinadas raças. Esse tipo de representação gera uma discussão com foco em questões sociais e o jogo fica em segundo plano.

Por causa de interdependência entre as atividades Escolher Representação Gráfica, Definir Mapeamento de Dados e Elaborar *Gameplay* deve-se rever as decisões tomadas em relação a representação gráfica e o *gameplay* toda vez que o mapeamento de dados utilizados sofra uma alteração.

- **Elaborar o *gameplay***

Segundo VON AHN (2005), pedir diretamente a solução computacional não é útil, torna o jogo muito algoritmo e produz pouco entretenimento (diversão). É preciso maquiagem a pergunta de uma forma que torne o jogo agradável. Esse é o principal desafio nessa etapa de elaboração do *gameplay*.

Para isso procura-se utilizar as mecânicas de jogos já conhecidos com a intenção de aumentar a empatia pelo jogo e torná-lo mais divertido. Com isso, esse passo se torna o mais importante do procedimento proposto pelo trabalho.

Para definir o *gameplay* do jogo serão utilizados conceitos já definidos nos passos anteriores. Oriunda da atividade de Entender o Problema, será utilizado o levantamento do passo-a-passo necessário para se obter a resposta e a solução desejada. Será utilizada a representação gráfica que foi definida na atividade de escolha da representação gráfica. E, por fim, as mecânicas de jogos tradicionais também serão utilizadas.

A representação gráfica está diretamente ligada ao *gameplay* do jogo. Com isso, é possível que uma definição mais concreta sobre que tipo de representação que será utilizada só ocorra nesse momento.

No primeiro momento deve-se encontrar mecânicas de jogos que equivalham aos passos necessários para se obter a solução do problema. Pode-se combinar mais de uma mecânica para realizar um mesmo passo ou, até mesmo, mais de um passo pode ser representado por uma única mecânica, como pode ser visto na Figura 14.

Não existe uma relação fixa entre os passos reais e as mecânicas tradicionais inseridas no jogo. Para tornar o *gameplay* mais fluido pode-se inserir mecânicas de jogo que não possuam um passo correspondente para a solução do problema real. Também podem existir

passos reais que não precisam de uma mecânica presente no jogo como, por exemplo, analisar alguma coisa.

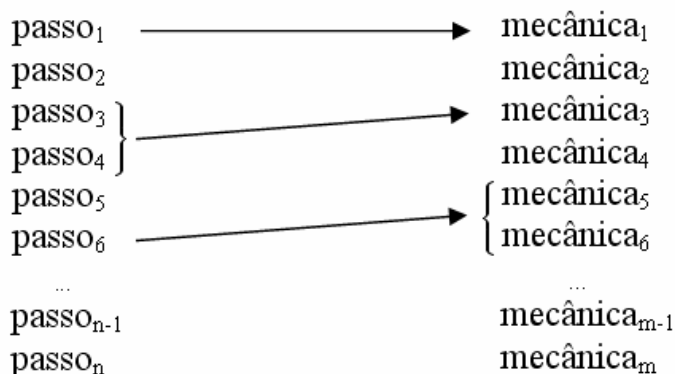


Figura 14. Passos do problema real relacionados com as mecânicas tradicionais inseridas no jogos.

Com esse tratamento evita-se que o passo utilizado na solução real do problema entre de forma direta no jogo. A ação para solucionar o problema deixa de ser uma mecânica de jogo e passa a ser equivalente a mecânicas já existentes.

Após definir as mecânicas que serão utilizadas no jogo, deve-se verificar se os elementos do jogo, que são formados pela representação gráfica, podem sofrer as interações das mecânicas escolhidas.

A mecânica do jogo é a ação que o jogador pode realizar enquanto está jogando. Nesse momento deve-se testar se a representação gráfica definida anteriormente aceita a ação pretendida. Se a mecânica escolhida for pular, por exemplo, verificar se faz sentido pular naquele elemento.

Verificar também se através da representação gráfica o jogador tem todas as informações necessárias para definir qual a melhor ação a ser realizada. E, se a decisão tomada pelo o jogador é uma forma de se obter a resposta real do problema.

Acaba-se formando um relacionamento entre o passo real para se obter uma solução real e a mecânica tradicional de jogo utilizada. Quando o jogador realiza as ações definidas pela mecânica de jogo ele chega à solução pretendida inicialmente ao executar o passo real do problema, como mostra a Figura 15.

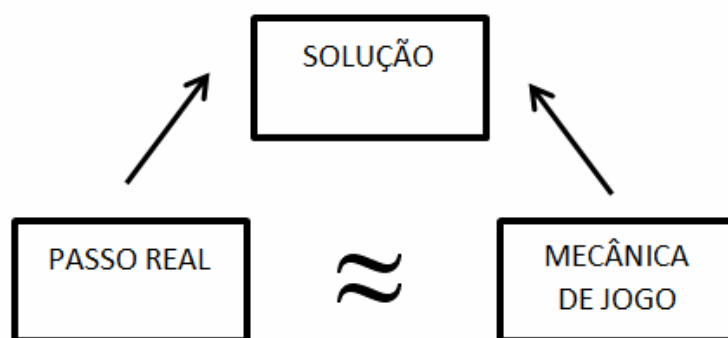


Figura 15. Relação entre o passo real executado para solucionar o problema, a mecânica tradicional de jogo escolhida para substituí-lo e solução esperada para o problema.

No passo de escolher a representação gráfica foi mencionado que o *gameplay* poderia interferir na representação gráfica, é nesse momento que isso pode acontecer. Se a mecânica definida não possibilitar interação com o elemento definido pela representação gráfica escolhida anteriormente pode-se optar por uma nova representação ou, ainda, tentar escolher uma nova mecânica para representar a ação definida e manter a representação gráfica.

Ao fim dessa atividade, já deve-se ter a forma lúdica de solução do problema real para ser utilizada no jogo com propósito definido. Na seção 5.3 – Procedimento de organização do jogo, será abordada a organização que o jogo deve possuir para atender as interações com o jogador e possuir uma estrutura de jogo. Ensinar as mecânicas ao jogador, mostrar ao jogador o que se espera que ele faça, testar as respostas dadas pelo jogador, serão abordadas adiante. Contudo, no decorrer dessa seção será dada continuidade ao estudo do procedimento de projeto do jogo.

- **Testar jogabilidade**

A solução lúdica que será utilizada para encontrar a solução do problema foi definida nos passos anteriores. Nesse momento será realizado um teste de jogabilidade para verificar a eficiência e o entendimento implícito no *gameplay* utilizado. Além de testar o jogo com propósito através do contexto de uma ferramenta de solução de problema, ele também deve ser testado como jogo em si, verificando se as mecânicas estão coerentes com os elementos do jogo.

Os testes realizados devem analisar, principalmente, se as mecânicas tradicionais de jogos utilizadas:

1. são eficientes para encontrar a solução do problema; e
2. fazem sentido dentro do ambiente criado para o jogo.

É interessante utilizar um pequeno grupo para esse teste e que seja possível, ao final da participação dos jogadores testadores, realizar uma entrevista ou um bate papo para identificar as impressões deles, evitando o uso de questionários como única forma para obter as opiniões dos jogadores. Ter uma pessoa acompanhado a sessão de jogo do testador também é uma forma a mais para se perceber as dificuldades e comportamentos dos jogadores.

Não existe uma característica especial para escolher as pessoas que irão compor esse grupo de testadores. Porém, pensando na utilização das mecânicas tradicionais de jogo para a solução do problema, é interessante que alguns membros tenham conhecimento sobre o problema que está se buscando solução, visando, assim, o melhor proveito das sessões de testes.

Realizar os testes com foco na jogabilidade nesse momento do processo permite o aprimoramento das decisões tomadas antes de estar com o jogo completamente finalizado. Se impressões dos jogadores e os resultados dos testes forem insatisfatórios deve-se voltar as atividades de elaboração de *gameplay*, definição de representação gráfica e mapeamento de dados ou, até mesmo, de entendimento do problema e dos dados, dependendo da melhor forma para se tratar as demandas encontradas.

Na atividade seguinte a principal preocupação é em relação a organização da solução do problema para transformá-la em um jogo. As demandas identificadas antes da conclusão da elaboração do jogo são tratadas com menos esforço e causam menos impacto no processo como um todo.

- **Organizar o jogo**

No processo de organização do jogo a solução lúdica encontrada é organizada na forma de um jogo. É nesse momento que a ferramenta jogo com propósito passa a ser um jogo em si.

Para dar uma visão geral do que ocorre, é nesse momento que a estrutura do jogo é pensada. São elaboradas as fases de treinamento de mecânicas de jogos, as fases tutoriais, as fases de coleta de respostas dos jogadores, bem como testes que são realizados com os jogadores durante a partida.

É nesse momento também que são definidos quais tipos de dados que serão utilizados em cada uma das fases que compõe o jogo. Por exemplo, dados de treinamento para as fases de tutorial. Essa atividade será estudada mais detalhadamente na seção seguinte.

- **Testar Organização**

Ao final do processo de projeto de jogo espera que se tenha um jogo com propósito que sirva para solucionar um problema de forma lúdica utilizando as mecânicas que não se pareçam com os passos da solução real do problema. Deve-se, nesse momento, realizar testes para verificar se o jogo criado se comporta da forma esperada e se as respostas obtidas através dele levam a uma solução para o problema.

O grupo de testadores dessa vez pode ser um pouco maior que o utilizado para testar a jogabilidade, e não necessariamente deve possuir pessoas que conheçam o problema, contudo a presença desse tipo de pessoa deve agregar qualidade ao resultado do teste. É interessante utilizar algumas pessoas que participaram da primeira sessão de teste pois já estão familiarizadas com o jogo. Pessoas que estão tendo o primeiro contato com o jogo nesse momento também agregam fatores positivos aos resultados da atividade.

O foco do teste é o jogo como um todo. No teste anterior foi verificado que ele funciona como ferramenta. Agora é verificado se ele funciona como jogo, se o jogador sozinho consegue entender e agir da forma que o jogo espera que ele haja.

Qualquer problema ou demanda identificada deve-se rever a estrutura estabelecida para o jogo. Verificar se o período de treinamento é suficiente, se as informações disponibilizadas são necessárias para o entendimento do jogador, entre outros. Os resultados dos testes sendo satisfatórios o jogo está finalizado e pronto para ser disponibilizado.

Nesse momento também se pode obter métricas iniciais sobre o jogo que irão determinar decisões que serão tomadas na atividade de Projetar o Experimento. Por exemplo,

o número de respostas obtidas por homem hora para definir o tempo que o jogo ficará disponível para ser jogado. Observar o comportamento dos jogadores para definir algum cuidado extra para quando for realizar a coleta de dados quando o jogo estiver disponível para o público em geral.

Ao final desse procedimento temos o jogo concluído e apto para ser disponibilizado para a aquisição das repostas dos jogadores como a forma lúdica de solucionar o problema. Antes de focar nas atividades restantes será realizado o estudo mais aprofundado do procedimento de organização do jogo. A seção seguinte busca explorar mais detalhadamente essas atividades.

5.3 – Procedimento de organização do jogo

Para obter as respostas dos jogadores, preocupa-se com a forma que os dados serão utilizados para auxiliar o jogador a atingir a solução do problema. No primeiro momento o foco é na interação que será realizada com o jogador. Posteriormente, para obter a solução para o problema, deve-se realizar testes com o jogador e decidir se as respostas dele serão utilizadas para encontrar a solução final do problema.

Em linhas gerais esse procedimento ocorre de forma análoga ao que ocorre no processo de classificação. O jogador deve ser treinado para realizar a tarefa que vai levar a solução do problema proposto e, posteriormente, deve ser testado para decidir se suas respostas serão levadas em consideração ou não para determinar a solução do problema.

A Figura 16 mostra o esquema do processo de organização do jogo como um todo. A seguir serão abordados individualmente os processos.

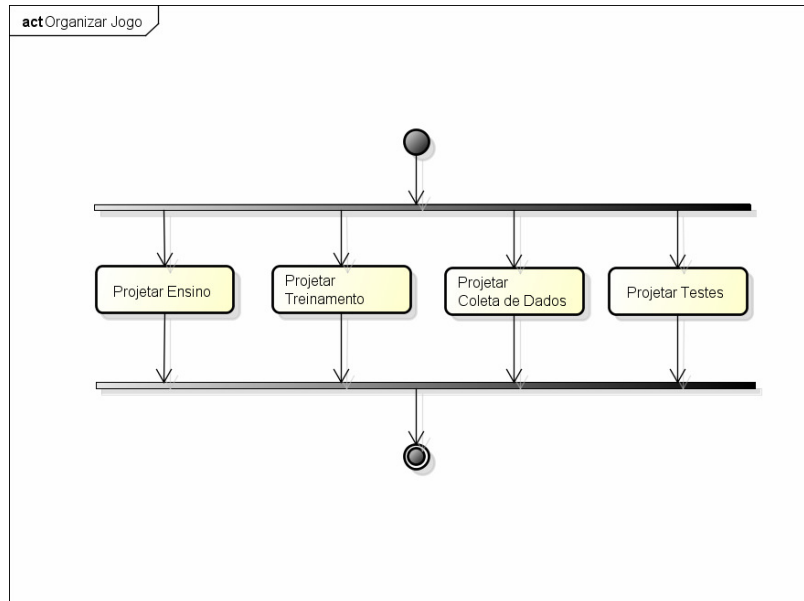


Figura 16. Procedimento de organização do jogo.

- **Projetar ensino**

Imaginando um primeiro contato do jogador com o jogo deve-se ter em mente ensinar ao jogador como o jogo funciona. Ensinar as mecânicas de jogo e seus respectivos comandos. Mostrar onde o jogador pode encontrar informações como pontuação, tempo de jogo, vida, entre outras. Para tornar o jogo familiar ao jogador. O melhor lugar para isso é no tutorial do jogo.

O Tutorial permite que o jogador entre em contato com o ambiente e as mecânicas do jogo sem que seja penalizado caso falhe. Tornando menos frustrante o processo de aprendizado. É interessante dar a possibilidade de o jogador pular essa etapa, ao retornar ao jogo ele já sabe como funciona e não precisa passar pelo tutorial novamente. Uma maneira de atender isso é, no menu do jogo, dar acesso direto ao tutorial e ao jogo em si. Dessa forma, ainda existe a possibilidade de o jogador retornar ao modo tutorial caso queira aprimorar alguma habilidade do jogo.

Não se pode assumir que o jogador vai realmente jogar o modo tutorial (BATES, 2004). Por isso, é importante que durante o jogo apareçam dicas para o jogador. Essas dicas podem aparecer através de texto simples, ou obrigando a interação com outros elementos do jogo, como outros personagens, por exemplo.

O objetivo ao final do processo é que o jogador saiba quais são os comandos que ele deve acionar para realizar a ação desejada no jogo. O jogador deve conhecer quais são as possibilidades que ele possui diante de um desafio do jogo, assim pode escolher a melhor maneira de interagir com o jogo e solucionar os desafios.

- **Projetar treinamento**

Depois de conhecer as ações que pode tomar durante o jogo, o jogador está apto a decidir quais são as melhores alternativas para resolver os desafios encontrados enquanto joga. Diante de um obstáculo, o jogador já possui todas as informações para superá-lo e seguir em frente.

Contudo, nos jogos com propósito, espera-se que as ações escolhidas pelo jogador colaborem com a obtenção da solução do problema que se deseja resolver. Com isso, deve-se treinar o jogador para que as ações escolhidas levem a melhor resposta possível. Durante as fases de treinamento mostra-se ao jogador o que se espera dele, essa comunicação é feita através de *feedbacks*.

Para qualquer ação que o jogador tome ele espera que aconteça uma reação no jogo. Se o jogador faz alguma coisa, o jogo faz alguma coisa em resposta (BATES, 2004). Os *feedbacks* são essas respostas dadas ao jogador. Eles podem ser visuais, sonoros, até mesmo, táteis (o controle pode vibrar, por exemplo). Além disso, os *feedbacks* podem ser positivos e negativos, informando se o jogador tomou uma atitude correta ou não.

Segundo SCHELL (2008), “A informação retornada ao jogador pelo jogo afeta drasticamente o que o jogador vai fazer a seguir (...) esse *feedback* exerce uma forte influência no quanto o jogador entendeu e aproveitou do jogo”¹⁷. Assim sendo, os *feedbacks* são elementos que podem ser utilizados para mostrar ao jogador se ele está ou não agindo da forma esperada, e como ele deve agir futuramente.

¹⁷ Tradução do autor para o trecho: “The information that is returned to the player by the game dramatically affects what the palyer will do next. [...] this feed back can exert a powerful influence on how much the palyer understands and enjoys the game.” (SCHELL, 2008)

As fases iniciais são ideais para treinar o jogador, pois são os primeiros contatos dele com o jogo, ele está se familiarizando com o jogo e ele ainda não conhece a fundo o seu comportamento. Durante os treinos são utilizados como entrada os dados que já se conhece a resposta esperada. Funcionam como um conjunto de treinamento. Sendo assim, para uma entrada e_1 que é apresentada ao jogador são gerados *feedbacks* positivos caso as decisões tomadas por ele levem a saída esperada e *feedbacks* negativos caso as ações do jogador não levem a saída correta.

Utilizando um exemplo simples para ilustrar uma fase de treinamento, imagine que no conjunto de treinamento os dados de padrão d_1 gerem representação gráfica na forma de quadrados e que dados de padrão d_2 gerem representação gráfica na forma de triângulos, como indica a Figura 17.

$$d_1 \rightarrow f(d_1) = \square \cdot$$

$$d_2 \rightarrow f(d_2) = \triangle$$

Figura 17. Exemplo de representação gráfica para dados de diferentes padrões.
Representação que será utilizada para ilustrar o treinamento do jogador.

Suponha agora que foi escolhido utilizar mecânicas comuns aos jogos de plataforma. No decorrer da fase de treinamento o jogador vai se deparar com quadrados e triângulos, as representações gráficas dos dados de treinamento, como mostra a Figura 18. Diante desses elementos ele deverá optar pelas ações permitidas para interagir com eles.

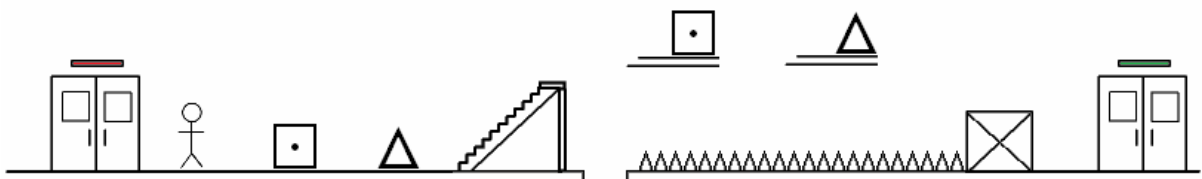


Figura 18. Exemplo de uma fase de treinamento onde o jogador depara com representações gráficas (quadrados e triângulos) dos dados de treinamento.

Para determinar a que padrões pertencem novos dados, deve-se condicionar o jogador a realizar uma determinada ação diante dos dados de padrão d_1 e outra diante de dados de padrão d_2 . Para induzir o comportamento do jogador serão utilizados os *feedbacks*. Suponha

que foi escolhido que o jogador pule em cima do quadrado. Se ele pular em cima ganha ponto e emite um efeito sonoro positivo, caso opte por outra ação, perde ponto e emite um efeito sonoro negativo. Figura 19.

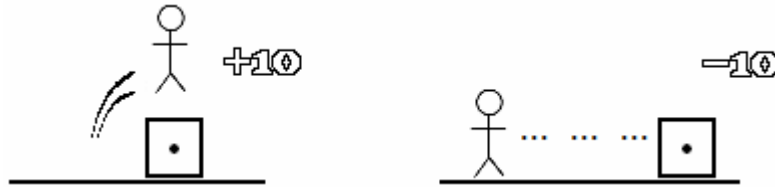


Figura 19. Exemplo de *feedback* para induzir as decisões dos jogadores diante de elementos de representação gráfica de dados de treinamento de padrão d_1 .

Foi escolhido, por exemplo, que o comportamento diante dos triângulos (representação gráfica dos dados de padrão d_2) deve ser atirar. Logo, se o jogador pular em cima do triângulo ele perde ponto e emite um sinal sonoro negativo, caso atire ele é recompensado positivamente. Figura 20.

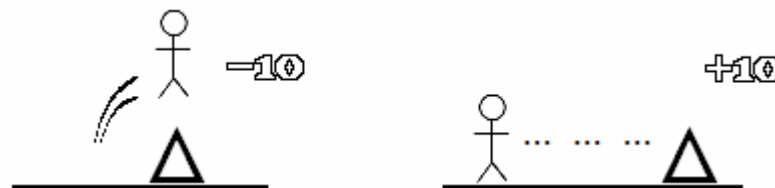


Figura 20. Exemplo de *feedback* para induzir as decisões dos jogadores diante de elementos de representação gráfica de dados de treinamento de padrão d_2 .

No exemplo utilizado as diferenças das representações gráficas são bem marcantes. Em problemas reais as representações gráficas podem se confundir mais entre os padrões de dados, tornando mais importantes ainda os cuidados para se treinar o jogador. A escolha dos dados que serão utilizados como treinamento e os *feedbacks* que serão apresentados para o jogador devem ser claros para não influenciar negativamente o resultado final.

Uma representação gráfica com dados reais seria como na Figura 21, onde foi utilizada a representação gráfica de estrela do programa de estatística R¹⁸ para os dados de criminalidade de algumas cidades americanas. Nota-se que as diferenças entre as representações gráficas não são tão nítidas. Os *feedbacks* servirão como mais uma informação para induzir as ações dos jogadores.

¹⁸ <http://www.r-project.org/>

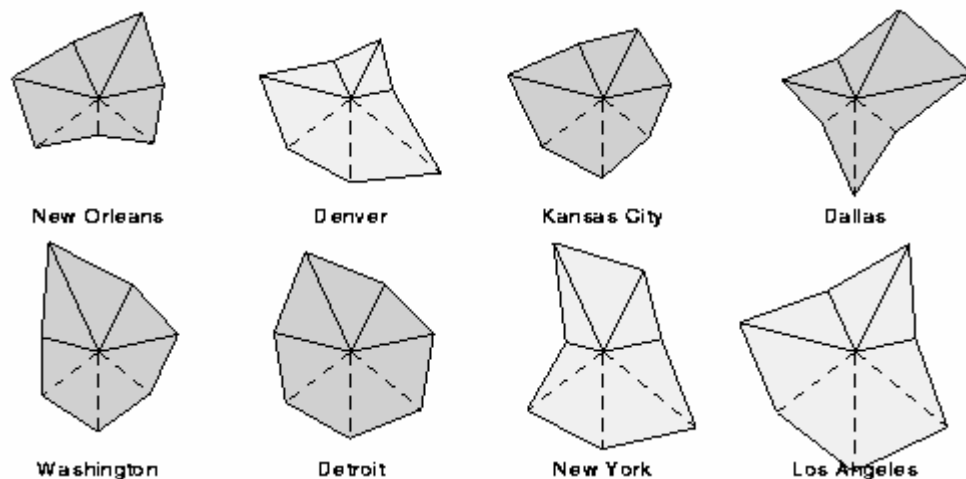


Figura 21. Exemplo de representação gráfica utilizando dados reais. Destaque para perceber que as diferenças entre as representações dos padrões de dados não são tão nítidas como no exemplo utilizado para explicar a utilização de *feedbacks* para o treinamento do jogador.

Quando chega ao final do processo de treinamento, o jogador já é capaz de identificar as melhores ações que deve tomar diante de um desafio para que obtenha o melhor desempenho, conseqüentemente, baseado nas ações do jogador, nas fases futuras será possível dizer a que padrão uma nova entrada e_x pertence.

- **Projetar coleta de dados**

O jogador já conhece as mecânicas disponíveis e já sabe como se comportar diante dos desafios que o jogo propõe pra ele. Então o jogador já está preparado para colaborar para encontrar a solução do problema que o jogo busca resolver. É nesse momento de coleta de dados que são obtidas as respostas do jogador.

Quanto mais tempo o jogador passar jogando mais ele colabora com a solução do problema, ele fornece mais respostas que serão utilizadas para definir a solução final. Tendo isso em mente, o jogo deve ser capaz de manter o jogador o maior tempo possível jogando e, se possível, fazer com que o jogador retorne ao jogo futuramente.

Uma forma de conseguir isso é balancear os desafios que são apresentados durante a partida. Os desafios não devem ser fáceis nem difíceis demais a ponto de fazerem o jogador abandonar o jogo, seja por ser muito fácil ou muito difícil. Os desafios que serão apresentados

ao jogador devem se tornar mais complexos gradualmente. Assim o jogador é desafiado de acordo com sua capacidade.

Existem outros elementos que colaboram para aumentar o tempo em que o jogador passa jogando. Pontuação e classificação com o desempenho (*ranking*) estimulam o jogador a jogar outras vezes para melhorar seu próprio desempenho ou superar o desempenho de amigos. O sistema de recompensas também aumenta o tempo dedicado ao jogo. Por exemplo, ao atingir uma determinada pontuação o jogador recebe um item especial ou ao jogar um determinado número de partidas o jogador recebe uma medalha (*badge*). Esses elementos também devem ser explorados para aumentar o número de respostas obtidas.

Todos esses fatores combinados com a utilização de mecânicas de jogos tradicionais colaboram para aumentar a aceitação e a vida útil do jogo. Conseqüentemente, aumentam o número de respostas obtidas dos jogadores, aumenta a relação respostas obtidas por homem hora. Assim, o jogo possui um rendimento maior, tornando possível o encontro de uma solução adequada em um menor tempo.

Outro fator que vai afetar o número de respostas obtidas é a quantidade de pessoas que estão jogando o jogo. A forma que o jogo será disponibilizado e a necessidade de algum dispositivo específico afetam diretamente o número de pessoas aptas a jogar o jogo. O jogo estando acessível a qualquer momento e não possuir um pré-requisito para jogá-lo atinge um público maior. Isso deve ser levando em consideração para aumentar a eficiência do jogo com propósito.

- **Projetar testes**

O ser humano é passível de erro. Diferentemente das máquinas, diante de uma mesma entrada nem sempre um ser humano vai produzir a mesma saída. Ele é influenciado por elementos externos que podem gerar resultados diferentes para uma mesma entrada. O jogador pode optar por decisões diferentes daquelas que foram induzidas a ele.

Por causa disso, é interessante verificar o desempenho do jogador. Para isso deve-se testar o jogador para decidir se as respostas dadas por ele são coerentes e assim, utilizá-las para obter a solução final do problema.

Durante a fase de treinamento foram utilizados dados para os quais a resposta já era conhecida. Para testar o jogador também serão usados como entrada dados para os quais já se conhece a saída. Porém, é interessante que os dados de treinamento sejam diferentes dos dados de teste, para evitar qualquer influência na avaliação do desempenho do jogador.

A fase de treinamento também poderia ser utilizada para determinar o desempenho do jogador. Porém, o jogador depara-se com ela em um momento inicial no qual ainda está se familiarizando com o jogo, aprendendo como ele funciona. Ele ainda não conhece completamente o jogo. Por isso, o desempenho do jogador durante a fase de treinamento não deve ser utilizado para avaliar a qualidade das suas respostas.

Os dados que a saída é desconhecida são utilizados para compor os desafios que irão aparecer no decorrer das demais fases que constituem o jogo. No meio dessas fases que devem ser inseridos desafios com os dados de teste, dados esses que já se conhece a saída. Conhecendo a saída sabe-se a ação que o jogador deve escolher e assim, é possível determinar se as respostas dadas por aquele jogador são boas ou não.

Os elementos formados a partir dos dados de teste servirão como elementos de controle, e serão eles que irão determinar o desempenho do jogador. As respostas dos jogadores com desempenho acima do mínimo esperado serão utilizadas para determinar a solução final do problema. As repostas dos jogadores com desempenho abaixo do mínimo esperado serão utilizadas somente para cálculos estatísticos.

5.4 – Procedimento de aplicação do jogo

Esse processo representa o momento em que o jogo está disponível e os jogadores estão jogando o jogo e, conseqüentemente, solucionando o problema computacional. Através das partidas jogadas são obtidas respostas dos jogadores que serão compiladas para obter a solução final.

Nesse momento o jogo esta sendo jogado e informações estão sendo obtidas dos jogadores. O procedimento que será abordado nessa seção pode ser dividido em dois blocos principais:

1. obter as respostas do jogador e,

2. obter as respostas para solução do problema.

O primeiro bloco compreende o momento em que o jogador está realizando as ações dele para superar os desafios propostos. O segundo bloco foca em selecionar as respostas que serão utilizadas para obtenção da solução do problema baseado no desempenho do jogador durante os testes que são submetidos a ele.

O jogo deve desempenhar alguns papéis, e esses papéis estão caracterizados nas atividades desse diagrama: Ensinar as mecânicas de jogo, treinar o jogador para realizar ações que levem a solução do problema e testar o jogador. Esses são alguns exemplos de papéis que o jogo com propósito deve assumir durante uma partida.

Para cada um destes papéis foi projetado um comportamento para o jogo no processo de organização do jogo visto na seção 5.3 – Procedimento de organização do jogo. Durante essa seção será feita a relação entre os comportamentos projetados e o papel do jogo. No final, será abordado o que deve ser feito após testar o jogador.

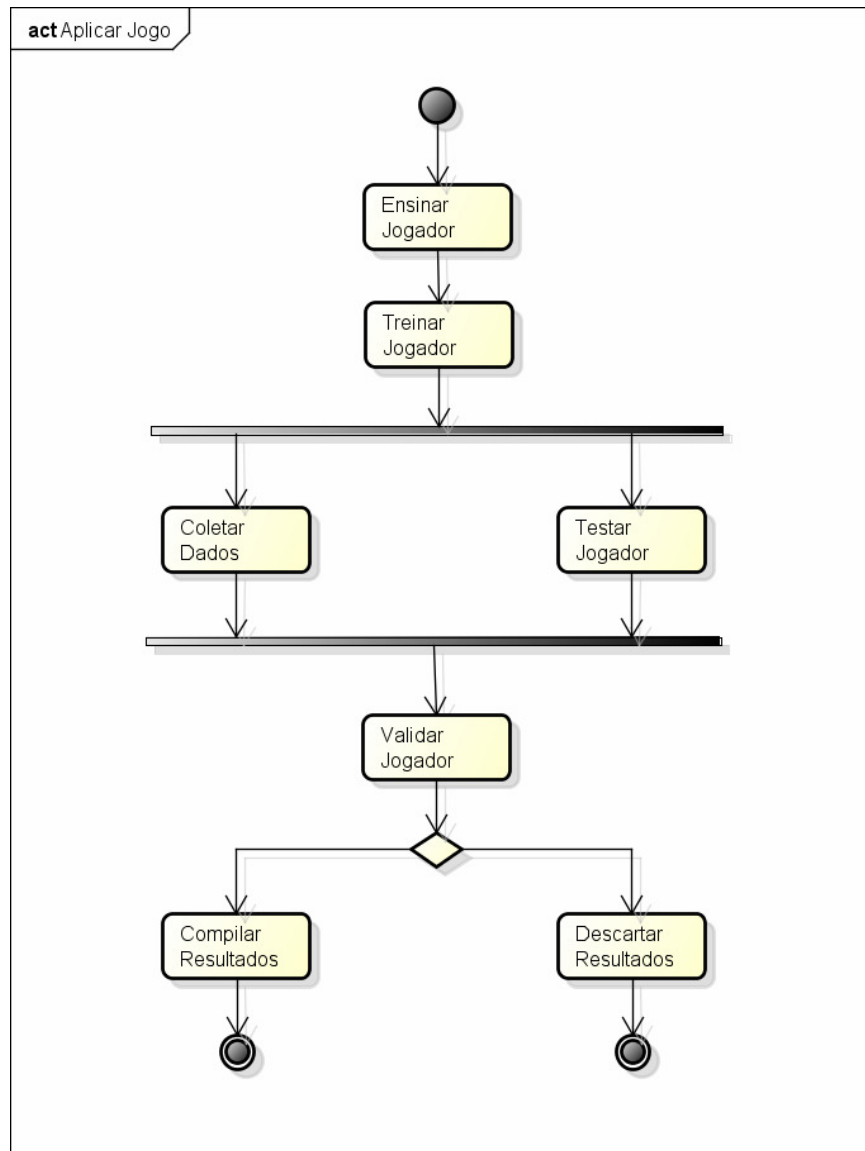


Figura 22. Procedimento de aplicação do jogo.

- **Ensinar o jogador**

Imaginando o primeiro contato do jogador com o jogo, é nesse momento que o jogo assume o papel de ensinar o jogador as mecânicas de jogo, ensinar qual é o comando que deve ser feito para se realizar uma determinada mecânica do jogo e mostrar para que serve determinada mecânica.

Isso acontece, comumente nos jogos durante o tutorial. Porém, não se pode garantir que o jogador irá realizar o tutorial. Para evitar esses casos os momentos iniciais do jogo

também ensinam as mecânicas e comandos do jogo. Nesses casos, os momentos iniciais se comportam como um tutorial, contudo de uma forma camuflada.

Ao final do momento de aprendizagem o jogador já conhece as ações que ele pode realizar no ambiente do jogo.

- **Treinar o jogador**

O objetivo final do jogo com propósito é encontrar a solução para um problema. Para obter essa solução espera-se um determinado comportamento do jogador. É nesse momento que o jogador é treinado para se comportar da maneira esperada.

Os desafios apresentados aos jogadores foram criados utilizando dados que já se conhece a resposta. Dessa maneira, o jogador vai aprendendo o que se espera que ele realize diante de um certo tipo de desafio. Cria-se assim um padrão de comportamento que será importante para se obter a resposta correta do jogador.

Como o jogador já conhece todas as mecânicas que ele pode realizar, ao se deparar com um tipo de desafio ele vai tomar um determinado conjunto de ações. Diante de outro tipo de desafio, outro conjunto de ação será necessário para superá-lo. Essas ações distintas acabam criando um padrão de comportamento que irá refletir o padrão de respostas esperadas que o jogador forneça.

As repostas obtidas durante essa fase não serão utilizadas para determinar a solução final do problema, tendo em vista que os dados utilizados para a elaboração dos desafios já possuem a resposta conhecida.

Ao final dessa atividade espera-se que o jogador já esteja apto para encarar os desafios oriundos dos dados sem respostas de forma correta.

- **Coletar os dados**

Esse momento também pode ser entendido como coleta de respostas. Já que é durante essa fase que o jogador está jogando o jogo e dando informações para encontrar a solução do problema.

Toda a assistência que o jogo poderia dar ao jogador foi fornecida nos momentos anteriores. Assume-se agora que o jogador já é capaz de realizar tudo o que se espera dele de uma maneira correta. As repostas dadas pelo jogador serão utilizadas futuramente para a obtenção da solução final do problema.

É durante esse período que o jogo está sendo jogado propriamente dito e que o jogador está resolvendo um problema como um efeito colateral do jogar. O jogador estará se entretendo com o jogo, se divertindo. O tempo que cada jogador passa diante do jogo vai variar dependendo da aceitação do jogo pelo jogador e pela maneira que ele é influenciado pelo elementos que foram inseridos no jogo.

Ao chegar ao final dessa atividade o jogo tem um grupo de repostas obtidas do jogador que serão utilizadas para determinar a solução final do problema.

- **Testar o jogador**

Como o ser humano é passível de cometer erros e não se pode garantir que a compreensão das mecânicas e dos conceitos presentes no jogo são iguais para todos os jogadores que estão participando. Por isso, o jogador deve ser testado enquanto está jogando.

Durante a fase de coleta de dados o jogo contempla o jogador com desafios oriundos de dados que já se conhece a resposta final. Esse tipo de desafio aparece de forma esporádica no meio de desafios que a resposta não é conhecida. O jogador não é capaz de determinar o tipo de desafio que ele está resolvendo.

O objetivo desses testes é garantir uma qualidade mínima das repostas que são obtidas dos jogadores e que serão utilizadas para encontrar a solução final do problema.

- **Validar o jogador**

Ao final da partida o jogador possui dois grupos de resultados oriundos dos desafios compostos por:

1. dados de saída desconhecida; e
2. dados de teste.

O resultado obtido utilizando dados de saída desconhecida será utilizado para determinar a solução do problema. Já o resultado do grupo de dados de teste será utilizado para determinar a qualidade das respostas do jogador.

A qualidade da solução final do problema estará ligada ao desempenho do jogador. Garantindo o uso de respostas de jogadores com bom desempenho diminui a presença de erros que podem comprometer a qualidade da solução final.

No final da partida o jogador possui uma reputação que é calculada de acordo com o seu desempenho diante dos desafios de teste. A qualidade das respostas que se espera obter através do jogo com propósito irá determinar o nível de aceitação das respostas dos jogadores.

- **Descartar as respostas**

A partir da definição do nível de desempenho desejado é possível determinar se as repostas do jogador serão utilizadas para se obter a solução final do problema. As respostas dos jogadores com o desempenho abaixo do nível esperado são descartadas e não são utilizadas para determinar a solução final.

As respostas descartadas não são perdidas. Elas são armazenadas em locais diferentes. Apesar de não serem consideradas para se encontrar a solução final do problema, elas são importantes para cálculos estatísticos do jogo com propósito. Número de partidas, razão de respostas obtidas por homem hora, razão de respostas úteis por respostas totais, entre outros.

Se o desempenho do jogador estiver acima do nível definido para o jogo com propósito, as respostas do jogador serão selecionadas para serem utilizadas no próximo passo, compilar os resultados. Caso o desempenho do jogador esteja aquém do esperado suas resposta são armazenadas em outro local.

- **Compilar os resultados**

Já foram descartadas as respostas dos jogadores que tiveram o desempenho abaixo do desejado. Sobraram, teoricamente, as melhores respostas que puderam ser coletadas. Nesse momento serão utilizadas somente as respostas classificadas como aceitáveis.

O ideal é que o jogo com propósito tenha como saída do sistema a resposta final do jogador. Porém, dependendo de como o *gameplay* foi planejado ou da natureza do problema que deve ser solucionado, pode haver a necessidade de realizar um último processamento das respostas dos jogadores. Para então combinar todas as respostas e chegar a uma solução final.

Esse processamento tem objetivo de agrupar e organizar as respostas de forma que facilitem a análise final que vai determinar a solução final do problema. Para a análise final deve-se ter somente as informações necessárias para se obter a solução do problema.

Os processamentos necessários já foram identificados na elaboração da solução lúdica do problema. O *gameplay* utilizado irá implicar em alguns processamentos, por exemplo, na atividade de Elaborar *Gameplay* do processo de Projetar Jogo visto na seção 5.2 – Procedimento de projeto do jogo, a definição das mecânicas tradicionais de jogo que são utilizadas foram definidas baseadas em passos da solução real do problema. Um passo real podendo ser representado por somente uma mecânica de jogo ou por mais de uma. Nos casos em que o processo real foi mapeado em mais de uma mecânica poderá ocorrer um processamento para combinar as duas ações tomadas pelo jogador.

No final dessa atividade, os dados estão organizados de uma forma simples que irá facilitar e possibilitar que se encontre a solução final do problema de uma maneira mais direta.

5.5 – Conclusão

Ao final desses procedimentos espera-se que o jogo com propósito possua mais características de jogos tradicionais. Que o *gameplay* presente esteja distante dos passos realizados para encontrar a solução real do problema. Escondendo os procedimentos para encontrar a solução do problema dentro de mecânicas de jogos conhecidas pelos jogadores. E, assim, possuir uma forma verdadeiramente lúdica para solucionar o problema.

Existem partes do procedimento que funcionam independentemente, como é o caso da representação gráfica. Sendo assim, toda representação gráfica utilizada para o desenvolvimento de jogos com propósito pode ser armazenada em um repositório de representações gráficas. Assim, ao desenvolver um novo jogo com propósito, procurar-se-ia

nesse repositório uma representação que pudesse ser utilizada antes de conceber uma nova representação gráfica.

Outra facilidade a ser destacada, é que uma mesma solução pode ser utilizada para mais de um problema. Sendo necessário somente realizar a troca dos dados de entrada. Substituindo os dados de entrada em uso por dados que caracterizem esse novo problema, desde que a solução utilizada para um problema também sirva para obter a solução do novo problema.

Foi desenvolvido um jogo com propósito e foram realizados alguns experimentos para testar os procedimentos propostos. No capítulo a seguir será explorado mais a fundo o processo de criação do jogo com propósito através do jogo desenvolvido. Os experimentos utilizando os jogos com propósito também serão apresentados e analisados.

Capítulo 6 – Exemplo de jogo e experimento

Afim de testar o procedimento proposto para o desenvolvimento de jogos com propósito utilizando mecânicas tradicionais de jogos, foi desenvolvido um jogo com propósito que auxilia na classificação de dados. O jogo foi desenvolvido baseado nos processos definidos no capítulo anterior. Com o jogo finalizado, foi realizado um experimento para verificar a eficiência do jogo criado.

O processo de desenvolvimento será abordado durante esse capítulo. Todas as decisões tomadas e as dificuldades que apareceram durante a elaboração do jogo serão discutidas no decorrer desse capítulo. No fim, serão discutidos os experimentos realizados e os resultados obtidos.

6.1 – Chernoff Invaders: o jogo

Foi escolhido desenvolver um jogo com propósito que auxilie na classificação de dados. Para iniciar o processo de elaboração do jogo com propósito deve-se entender o problema. O estudo realizado para o entendimento do problema de classificação foi realizado e aparece de uma forma mais detalhada no Capítulo 4 – desse trabalho.

Como pode ser observado no decorrer do capítulo 4, foram identificados os principais passos que uma pessoa realiza durante uma classificação, vide Figura 7. De uma forma simplificada, esse passo a passo é composto por:

1. Comparar;
2. Categorizar;
3. Selecionar;
4. Associar; e
5. Antecipar.

Também durante esse estudo inicial do problema, foi pensada a estrutura base que o sistema de classificação humano deveria ter. Esse sistema deverá conter os seguintes elementos:

1. Dados de entrada;
2. Representação gráfica;
3. Jogo;
4. Compilador de resultados; e
5. Dados de saída.

Com a realização do estudo sobre o problema, a próxima atividade é realizar um estudo sobre o conjunto de dados que será utilizado. O problema de classificação de dados é amplamente estudado. Por isso existe uma grande oferta de bases de dados para testar novos sistemas de classificação. Para o jogo desenvolvido foi escolhido utilizar a base de dados Iris (FRANK & ASUNCION, 2010).

Iris é um tipo de flor muito comum em regiões de clima temperado. Possuem aparência semelhante aos lírios, que são mais conhecidos no Brasil. A base de dados é composta por 150 instâncias divididas em 3 classes: *iris setosa*, *iris versicolor* e *iris virginica*. Cada classe possui 50 instâncias. Os dados possuem 4 atributos, o comprimento e a largura da sépala e da pétala, dados em centímetros.

Uma classe é linearmente separada das outras duas, como pode ser visto na Figura 23. A base de dados forma dois *clusters*. Um cluster é composto pela *iris setosa*, o outro pela *iris versicolor* e *iris virginica*. Essa característica insere um desafio a mais para a base e, conseqüentemente, para o jogo.

Após o entendimento dos dados que serão utilizados, foi escolhida a representação gráfica que será utilizada. Existe a possibilidade de utilizar uma representação gráfica já existente ou desenvolver uma que se enquadre de maneira coerente ao ambiente do jogo. Durante os estudos foi encontrada uma representação gráfica através de rostos (CHERNOFF, 1973). Essa representação foi escolhida por se adequar de maneira satisfatória na ideia de jogo que estava sendo desenvolvida.

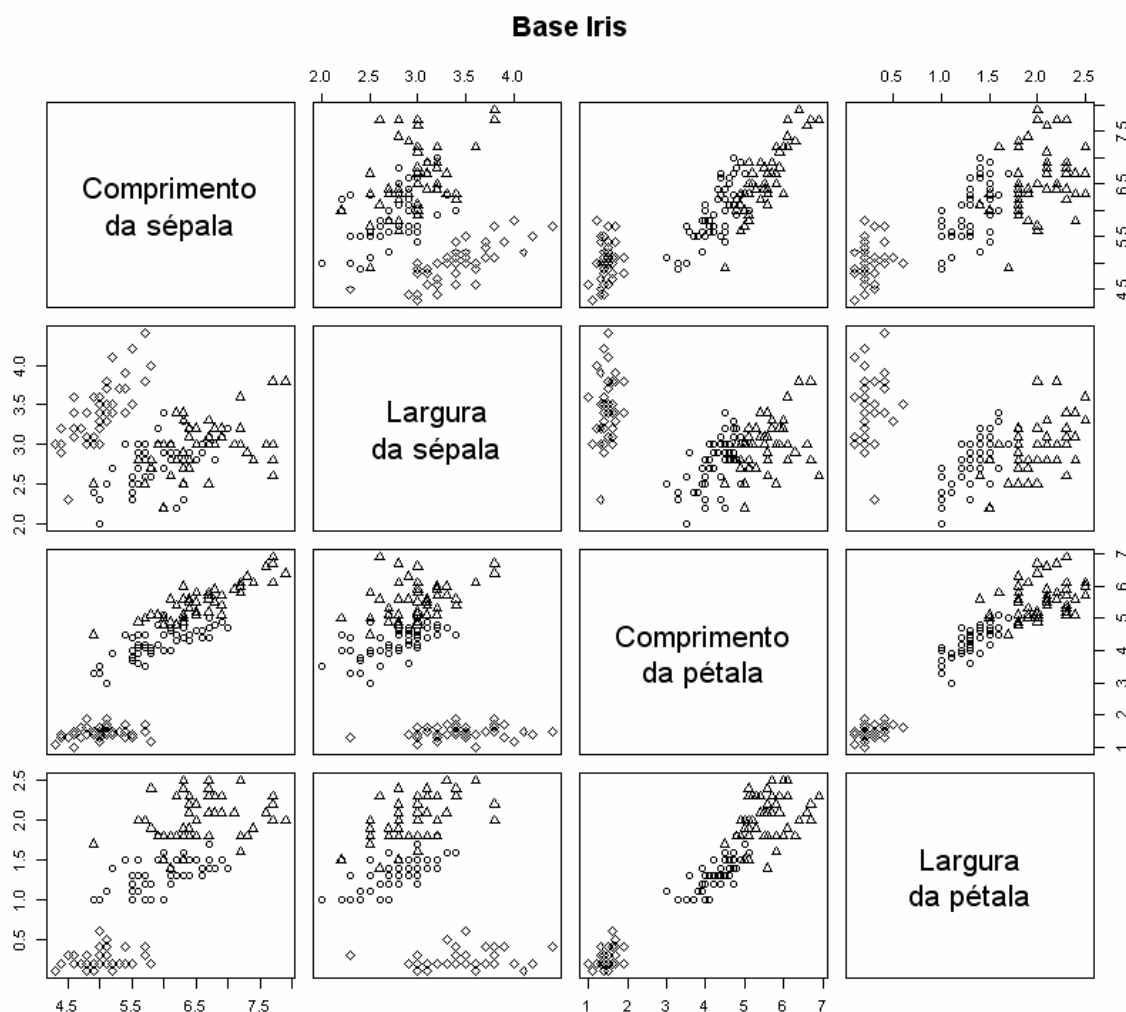


Figura 23. Distribuição das classes da base Iris. *Iris setosa* representada pelo losango, *iris versicolor* representada pelo círculo e *iris virginica* representada pelo triângulo.

O método de representação gráfica proposto por CHERNOFF (1973) “consiste em representar um ponto de um espaço n -dimensional através do desenho de um rosto no qual as características são representadas pela posição do ponto”. O tamanho do nariz, o formato do rosto, a distância entre os olhos, entre outros, são exemplos de características do rosto que são modificados de acordo com os valores observados.

O modelo proposto por CHERNOFF (1973) trabalha com até 18 variáveis. Porém, como o próprio autor defende, é relativamente fácil aumentar o número de variáveis acrescentando novas características na representação do rosto. Foram encontrados diferentes algoritmos que geram faces de Chernoff em programas de estatística. Cada algoritmo dos

programas com um número máximo de variáveis: o MatLab¹⁹ suporta até 17 variáveis, o R²⁰ consegue representar 14 variáveis, o Mathematica²¹ trabalha com 10 variáveis, já o Stata²² representa 18 variáveis se o rosto for simétrico, e é o único que dá suporte a criação de rostos assimétricos, podendo chegar a 36 variáveis nesses casos.

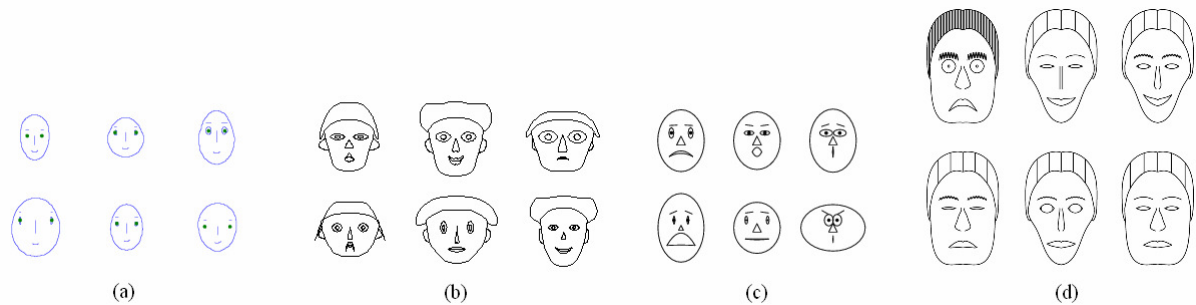


Figura 24. Exemplos de representações das faces de Chernoff dos diferentes programas de estatística. (a) MatLab; (b) R; (c) Mathematica; e (d) Stata. Não foram utilizados os mesmos dados para a representação dos rostos.

Segundo CHERNOFF (1973), a principal vantagem do método está no fato do ser humano estar acostumado a analisar e reagir a rostos. Isso o torna capaz de ignorar características irrelevantes e se focar nas mais importantes possíveis. Devido a experiências anteriores, pequenas diferenças são percebidas e grandes são ignoradas se não forem importantes. Porém, a interpretação dos rostos ainda é subjetiva, dependendo de quem está observando.

Com a representação gráfica definida começou-se a pensar na maneira de como os dados seriam inseridos na representação. Na análise dos dados não foi identificado nenhuma informação que pudesse ser utilizada para enriquecer o mapeamento. Em outras palavras, que possuísse algum significado incorporado ao dado.

¹⁹ <http://www.mathworks.com/>

²⁰ <http://www.r-project.org/>

²¹ <http://www.wolfram.com/mathematica/>

²² <http://www.stata.com/>

Sendo assim, optou-se por um mapeamento direto das colunas dos dados da base Iris com as características das faces de Chernoff. A tabela a seguir mostra o mapeamento final que foi utilizado para inserir os dados no jogo.

Tabela 5. Mapeamento dos atributos da base Iris e as características da face de Chernoff.

Atributo	Característica
Comprimento da sépala	Estrutura do rosto
Largura da sépala	Sorriso
Comprimento da pétala	Abertura dos olhos
Largura da pétala	Estilo do cabelo

Em seguida deve-se definir o *gameplay* para o jogo que vai auxiliar na classificação de dados. O momento em que a solução lúdica para o problema escolhido é desenvolvida.

A primeira definição tomada foi em relação ao gênero que o jogo com propósito se encaixaria. Foi escolhido o gênero de tiro, mais especificamente o gênero de tiro com visão aérea. Também conhecido como *shoot'em up*. Gênero de jogo muito popular com representantes bastante conhecidos como, por exemplo, Galaga, River Raid, Raiden, Sonic Wings e , o mais popular, o Space Invaders.

Com o gênero de jogo definido, conseqüentemente, já existia um grupo de mecânicas tradicionais de jogo disponíveis para ser utilizadas com o objetivo de que o jogo desenvolvido permanecesse dentro do gênero escolhido. Seriam mecânicas como mover, mirar e atirar. Dessa maneira, chegou-se a versão de como se comportaria a solução lúdica para o problema de classificação de dados.

O jogador irá utilizar o tiro para realizar a classificação. Sendo assim, a ação de classificar foi mapeada para a mecânica de tiro. Porém, surge um problema com essa definição. Ao classificar alguma coisa, a pessoa diz se algo possui ou não o mesmo padrão do restante dos outros elementos da classe. São duas ações, **(i)** pertence a classe e **(ii)** não pertence a classe, e apenas uma mecânica, atirar.

Para resolver esse problema foram inseridos dois significados à mecânica na forma de dois tiros. Um tiro representaria a ação de pertence a classe e o outro de não pertence a classe. Sendo assim, foi criado um tiro que salva e um tiro que mata. Ao escolher o tiro que salva, o jogador mostra que o elemento escolhido pertence a classe. Ao escolher o tiro que mata, está indicando que o elemento pertence a outra classe.

Com essas definições o jogo se comportaria da seguinte maneira. O jogador controla um rosto que pertence a uma determinada classe. Durante a fase rostos de diversas classes se movimentam em direção ao jogador. Enquanto isso, o jogador decide se mata ou salva o rosto de acordo com o rosto que ele controla.

O rosto que o jogador controla serve como exemplo de uma determinada classe. Baseado nesse exemplo, o jogador decide se o rosto que ele está mirando é ou não da mesma classe que a dele. Se o rosto for da mesma classe que a dele o jogador opta pelo tiro que salva. Se o rosto for de outra classe, o jogador escolhe o tiro que mata.

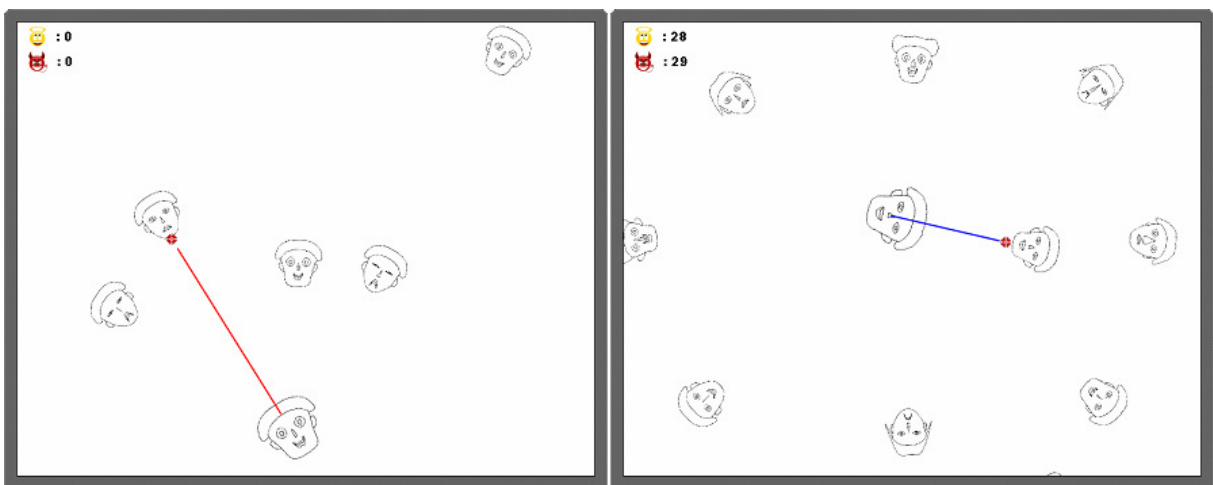


Figura 25. Telas da primeira versão do jogo com propósito desenvolvido para solucionar o problema de classificação de dados.

A Figura 25 serve de exemplo de como ficou essa primeira versão desenvolvida para o jogo. Com o *gameplay* que foi definido, o jogador estaria classificando um dado como sendo da mesma classe do jogador ou de outra classe de uma forma implícita.

Nesse momento foi realizado um teste com o intuito de verificar a qualidade da solução lúdica encontrada. Esse teste foi realizado com um pequeno grupo. Explicou-se a mecânica do jogo para o jogador e foi pedido para ele jogar. Acompanhou-se os jogadores

durante as partidas para tirar eventuais dúvidas e, principalmente, observar o comportamento deles durante a execução. Nem antes, nem durante a partida se comentou sobre classificação de dados. Ao final do período de jogo foi realizada uma conversa com os jogadores e, nesse momento, mencionou-se o trabalho implícito presente no jogo.

Durante essa conversa com os jogadores testadores, percebeu-se dois fatores principais:

1. o jogo parecia confuso, com muitos rostos vindo de todas as direções; e
2. o jogo possuía poucos exemplos de padrões das classes envolvidas.

Por causa do resultado desse teste, retornou-se a elaboração para aprimorar o jogo. O jogo deveria ser reformulado para tornar a experiência mais agradável e melhorar a qualidade das respostas.

A segunda versão do jogo continuou com a mesma mecânica de atirar para salvar e matar os rostos. Contudo, a disposição das fases foi alterada visando atender as demandas encontradas durante o teste.

A demanda que dizia que o jogo estava confuso, com os rostos vindos de todas as direções foi resolvida estabelecendo um bloco de rostos para cada fase. As fases se iniciariam com esse bloco já estabelecido e terminariam ao matar/salvar todos os rostos presentes nesse bloco. Esse bloco se movimentaria lateralmente ou verticalmente, dependendo da fase, de forma uniforme. Outro ponto que também foi alterado é que o jogador agora só pode se movimentar horizontalmente na parte inferior da tela.

A demanda que identificou poucos padrões para comparação foi resolvido inserindo dois rostos inimigos nos cantos superiores da tela. Esses dois inimigos sempre são de classes diferentes da do jogador. Com o objetivo de incluir um desafio maior, esses inimigos disparam tiros para matar o jogador. O bloco de rostos também influencia positivamente nessa demanda, já que tem vários exemplos de rostos de mais de uma classe lado a lado, facilitando a percepção de padrões entre as classes.

A Figura 26 mostra como ficou a versão final do jogo.

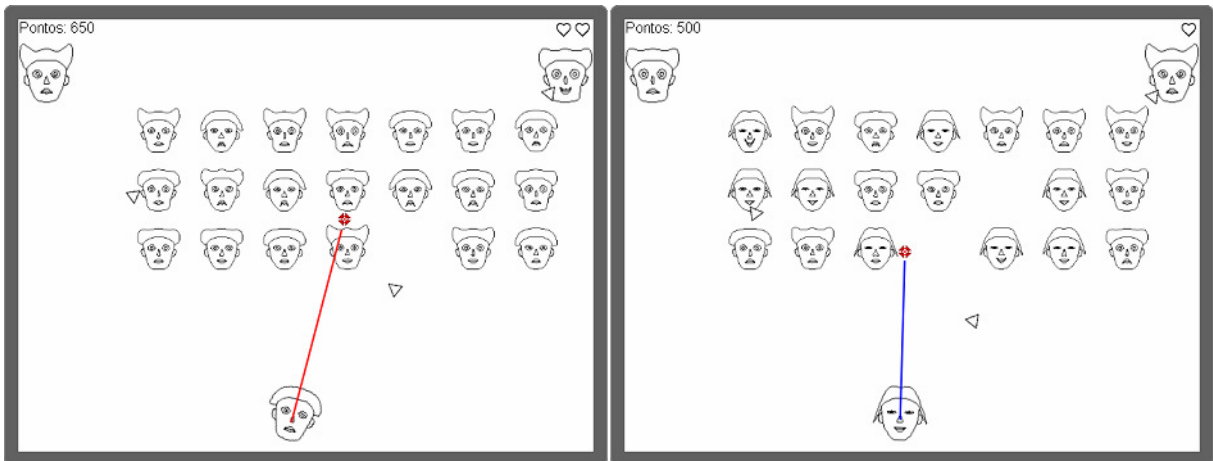


Figura 26. Telas da versão final do jogo com propósito desenvolvido para solucionar o problema de classificação de dados.

O jogo foi estruturado de uma forma que possui um tutorial que ensina as mecânicas de jogo. A primeira fase foi montada como uma fase de treinamento. Da segunda fase em diante são as fases que o jogador irá classificar os dados propriamente dito. O jogo é composto por onze fases. Dessas onze, uma é de treinamento e dez para classificação de dados. Das dez fases de jogo, seis são compostas por duas classes de cada vez e quatro são compostas pelas três classes presentes na base utilizada.

Tabela 6. Quantidade de classes presentes em cada fase.

Fase	Classes	Fase	Classes
1	2	7	3
2	2	8	3
3	2	9	2
4	2	10	2
5	3	11	2
6	3		

O jogador possui três vidas para completar o jogo. Ao ser atingido pelos tiros dos rostos atiradores o jogador perde uma vida. Ao ficar sem vidas, perde a partida. No decorrer das fases o nível de dificuldade vai alterando para manter o desafio aos jogadores. O

movimento dos rostos se tornam mais rápidos e a quantidade de tiros disparados pelos atiradores também aumenta.

O jogo ainda possui um sistema de ranking para que o jogador possa comparar seu desempenho com o dos outros jogadores. Servindo de estímulo para que o jogador realize novas partidas.

O jogo, batizado de Chernoff Invaders, foi desenvolvido com o framework Construct 2²³ utilizando HTML 5. O jogo possui uma estrutura cliente-servidor. O lado do servidor foi desenvolvido em PHP com a base de dados MySQL²⁴ e o lado cliente, que é o próprio jogo, que foi desenvolvido em HTML 5. Dessa maneira o jogo pode ser jogado em qualquer navegador moderno.



Figura 27. Tela inicial do jogo Chernoff Invaders. Configuração inicial da fase de treinamento.

Na seção seguinte será abordado o experimento realizado com o jogo. Também serão comentados os resultados obtidos através do jogo inclusive, será apresentado o resultado de algoritmos matemáticos para classificação com a finalidade de comparar o desempenho do jogo.

²³ <https://www.scirra.com/construct2>

²⁴ <http://www.mysql.com/>

6.2 – Experimento

Foram realizados dois experimentos para avaliação do jogo com propósito. O primeiro, como foi comentado anteriormente, foi realizado durante o período de desenvolvimento para testar a solução lúdica encontrada para o problema. Já o segundo, foi realizado com um grupo de jogadores e de uma forma controlada. A seguir o segundo experimento será detalhado.

O segundo experimento contou com a participação de 11 (onze) alunos de graduação da turma de Projeto de Jogos do curso de bacharelado em Ciências da Computação da Universidade Federal do Rio de Janeiro. Foi realizado de uma forma controlada no laboratório. Inicialmente, foram explicadas as mecânicas de jogo e, em seguida, os alunos ficaram jogando. Ao final do experimento, foram explicados os conceitos presentes no jogo como, por exemplo, computação humana, jogos com propósito e classificação.

Durante o experimento os jogadores jogaram 111 partidas e passaram por 965 fases. Totalizando 19.299 classificações. Em média, cada jogador jogou 10,09 partidas, passou por 87,72 fases e realizou 1.754,45 classificações. O experimento durou um total de 92 minutos. Contando o tempo destinado ao jogo e as explicações iniciais e finais.

A duração média de cada partida jogada foi de 188,4 segundos, sendo que a partida mais longa demorou 453,3 segundos e a mais curta 27,6 segundos. O jogador da partida mais demorada perdeu na fase 10 e o jogador da partida mais curta jogou até a segunda fase. Já as fases tiveram uma duração média de 19,1 segundos. Sendo a fase com maior duração demorou 210,4 segundos e a mais curta 2,2 segundos.

Cada partida teve, em média, 173,86 classificações. Como a duração média das partidas foi de 188,4 segundos é possível determinar o rendimento do jogo desenvolvido. O número médio de classificações realizadas por partidas dividido pelo tempo médio de duração das partidas informa que o rendimento do jogo foi de 0,92 classificações por segundo.

Todos os rostos não foram classificados o mesmo número de vezes. Isso ocorre devido ao desempenho dos jogadores nas partidas. Por causa disso, rostos que aparecem nas fases iniciais recebem mais classificações que os rostos que aparecem nas outras fases. Os seis rostos mais classificados obtiveram, respectivamente, 200, 191, 190, 190, 189 e 189

classificações. Os seis rostos menos classificados obtiveram, respectivamente, 70, 71, 73, 74, 74 e 75 respostas. Em média cada rosto foi classificado 128,66 vezes.

Os dados apresentados até esse momento são focados na análise do jogo com propósito em si. Essas informações não colaboram para encontrar a solução para o problema, neste caso, classificação de dados. Porém, permitem um entendimento mais amplo do contexto em que a solução foi encontrada. A seguir serão apresentados os resultados finais referentes à classificação dos dados.

Serão apresentados resultados gerais obtidos pelo jogo. Em seguida, serão apresentados resultados específicos para cada fase e para os grupos compostos por 2 classes e 3 classes. Essa divisão foi realizada para enriquecer a análise sobre o desempenho do jogo. Ao final, serão apresentados resultados de classificações realizadas por algoritmos tradicionais de classificação de dados.

O jogo obteve 19.299 respostas dos jogadores. Desse total, 17.784 foram respostas corretas e 1.515 respostas erradas. Esses números levam em consideração todas as respostas dadas pelos jogadores, inclusive durante a fase de treinamento. Excluindo a fase de treinamento da análise, 16.905 respostas foram obtidas. Sendo 15.467 corretas e 1.438 erradas. No primeiro caso chegando a 92,14% de acerto, já no segundo caso, 91,49% de acertos.

Entende-se por resposta correta quando o jogador opta pela ação que é esperada que ele tome. Por exemplo, se o comportamento esperado é o jogador salvar um determinado rosto e o jogador escolhe o tiro que salva é tido como uma resposta correta. Caso o jogador opte por usar um tiro que mata em um rosto que era para ser salvo, é contabilizado como uma resposta errada.

Levando em consideração somente as respostas certas e erradas dadas pelos jogadores, sem contar a fase de treinamento, o jogo conseguiu classificar corretamente 126 instâncias de um total de 129, obtendo um desempenho de 97,67% de acertos.

Contudo, devido à organização do jogo, não é possível definir as classes dos 129 dados restantes. Isso ocorre porque o jogo possui fases com duas classes e com três classes. Nas fases compostas por duas classes, quando o jogador opta por utilizar o tiro que mata em um rosto, automaticamente, ele está dizendo que o rosto pertence à outra classe. O mesmo não

acontece nas fases com três classes. Quando o jogador opta pelo tiro que mata não é possível afirmar qual é a classe do rosto morto. Pode-se concluir, apenas, que o rosto não é da mesma classe do jogador.

Removendo os dados em que não é possível afirmar a classe que ele pertence, restam 89 instâncias classificáveis. Desse total, 86 dados foram classificados de forma correta. Sinalizando um desempenho de 96,63% de acerto.

Nos dois cenários, utilizando todas as instâncias e utilizando apenas as instâncias classificáveis, ocorreram três classificações erradas. A classe que foi determinada para o dado não é realmente a que o dado pertence. A tabela a seguir mostra os dados que foram classificados de forma errada.

Tabela 7. Dados classificados de maneira errada pelo jogo.

classe	comprimento da sépala	largura da sépala	comprimento da pétala	largura da pétala
<i>versicolor</i>	5,9	3,2	4,8	1,8
<i>virginica</i>	6,0	2,2	5,0	1,5
<i>virginica</i>	6,1	2,6	5,6	1,4

Aprofundando a análise sobre os dados classificados de maneira errada, percebe-se que eles pertencem justamente as duas classes que formam um único *cluster* (configuração em *clusters* da base pode ser vista na Figura 23). Além disso, esses dados estão presentes na fronteira entre as classes *versicolor* e *virginica*, o que mostra que esses dados estão bastante próximos no espaço. Logo, não causa estranheza eles serem classificados em classes que não pertencem. Na Figura 28 é possível identificar onde os dados estão presentes no espaço.

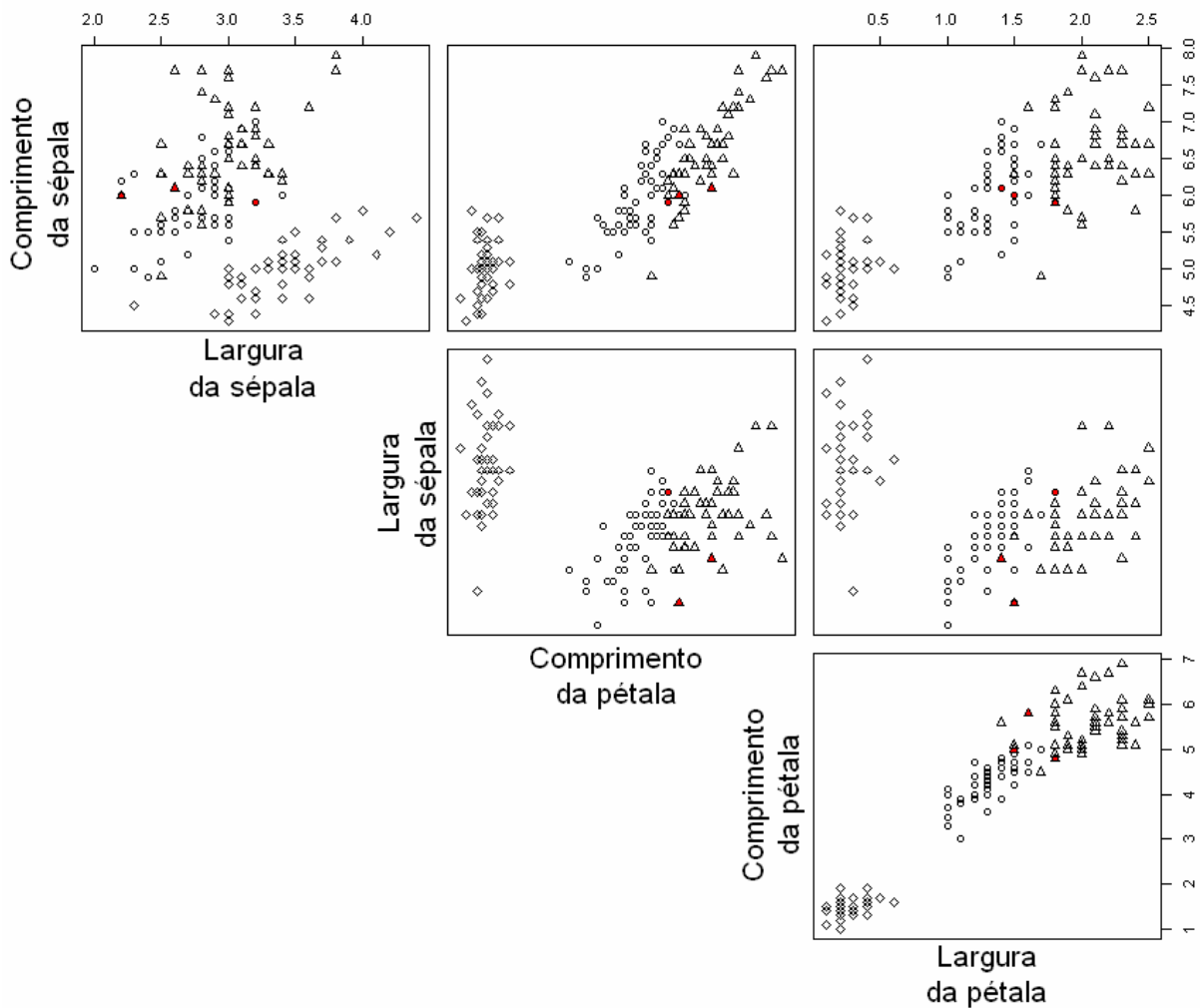


Figura 28. Localização espacial dos dados classificados de maneira errada. Os dados classificados de maneira errada estão em destaque.

Devido as diferentes configurações das fases com duas e três classes mostrou-se interessante estudar de forma individual o desempenho em cada configuração. O quadro abaixo mostra que, apesar de próximos, o desempenho nas fases com duas classes foi superior ao de três classes. O que pode indicar que uma classificação binária pode ser mais proveitosa para esse tipo de solução.

Tabela 8. Comparação entre o desempenho do jogo nas fases compostas por duas classes e por três classes.

Classes	desempenho
2	98,48%
3	94,14%

Uma última análise realizada foi em relação ao desempenho em cada fase. O quadro a seguir mostra o desempenho fase por fase. O interessante dessa análise é a constatação de que fase por fase existem mais erros do que no jogo como um todo. Os dados que foram classificados de forma errada pelo jogo aparecem nas fases 4, 5 e 8. Já os rostos que foram classificados de forma errada nas fases 10 e 11 tinham sido classificados de forma correta em fases anteriores. O que leva a conclusão que a dificuldade das fases pode influenciar na qualidade das respostas.

Tabela 9. Comparação de desempenho fase por fase do jogo.

Fase	2	3	4	5	6	7	8	9	10	11
Acertos	21	21	20	20	21	21	20	21	20	19
Erros	0	0	1	1	0	0	1	0	1	2

6.3 – Conclusão

Foram realizadas classificações da base Iris utilizando algoritmos de classificações já conhecidos. Foram escolhidos os algoritmos KNN (*k-nearest neighbor*) e o *Naive Bayes*. Os desempenhos dos algoritmos servirão como uma base de comparação para o desempenho do jogo com propósito.

Para executar o algoritmo KNN foi utilizado um conjunto de treinamento com o mesmo tamanho do utilizado pelo jogo. O algoritmo foi executado mais de uma vez mantendo fixo o tamanho do conjunto de treinamento e variando apenas o número de vizinhos utilizados. A tabela a seguir mostra o desempenho do algoritmo KNN variando o número de vizinhos de 1 até 9.

Tabela 10. Desempenho do algoritmo KNN alterando o número de vizinhos.

Vizinhos	1	2	3	4	5	6	7	8	9
Desempenho(%)	94,57	91,47	96,90	93,02	93,02	89,92	92,25	89,15	89,15

O algoritmo de *Naive Bayes* também foi executado mais de uma vez. Porém nesse caso o valor que foi alterado em cada execução foi o tamanho do conjunto de treinamento. Na tabela a seguir é possível notar os desempenhos do algoritmo de *Naive Bayes* para cada configuração diferente.

Tabela 11. Desempenho do algoritmo de *Naive Bayes* alterando o tamanho do conjunto de treinamento.

Conjunto treinamento	21	26	30	41	50	60	69	80	90
Desempenho (%)	87,60	89,52	89,17	88,99	92,00	96,67	96,30	95,71	93,33

O jogo com propósito Chernoff Invaders foi capaz de classificar corretamente 96,63% dos dados. Um desempenho muito próximo aos melhores resultados obtidos pelos algoritmos matemáticos. O KNN classificou corretamente 96,90% dos dados com a configuração de três vizinhos e o *Naive Bayes* conseguiu classificar corretamente 96,67% dos dados utilizando um conjunto de treinamento composto por 60 instâncias.

Tabela 12. Comparação entre os algoritmos KNN, *Naive Bayes* e o jogo com propósito Chernoff Invaders.

Método	KNN	<i>Naive Bayes</i>	Chernoff Invaders
Desempenho	96,90%	96,67%	96,63%

O desempenho do jogo como um classificador foi bastante próximo dos melhores desempenhos dos algoritmos. Contudo, o tamanho do conjunto de treinamento do utilizado pelo algoritmo de *Naive Bayes* foi, praticamente, três vezes superior ao tamanho do conjunto de treinamento utilizado pelo jogo (21 para o jogo e 60 para o algoritmo). O que mostra que para cenários em que o conjunto de treinamento para classificação for limitado, a utilização do jogo com propósito pode ser uma alternativa mais eficiente.

Capítulo 7 – Conclusões

Foram realizados estudos sobre computação humana com foco em jogos com propósito e mecânicas de jogos para elaboração de um modelo de desenvolvimento de jogos com propósito que possibilite o uso de mecânicas tradicionais de jogos. Em seguida foi realizado um estudo sobre classificação de dados para o desenvolvimento de um jogo com propósito que auxilie na classificação de dados utilizando mecânicas de jogos de tiro. Agora, serão abordadas as considerações finais referentes a este trabalho, suas contribuições e perspectivas futuras.

7.1 Considerações acerca do trabalho

O grande objetivo do trabalho é propor um modelo de desenvolvimento de jogos com propósito utilizando mecânicas de jogos tradicionais. Para a elaboração desse modelo foi estudado o processo de aplicação dos jogos com propósito como um todo. A abordagem não se limitou somente ao desenvolvimento do jogo em si, porém abordou o processo de uma forma global, passando pela obtenção de respostas dos jogadores e análise para encontrar a solução final do problema.

Por causa disso, o resultado final do trabalho foi um processo bastante completo do procedimento da solução de problemas computacionais utilizando jogos com propósito. Ao todo, foram elaborados quatro modelos para representar esse processo.

O primeiro modelo apresenta uma visão do processo como um todo. Ele inicia com o projeto do jogo, passando pela aplicação e análise de resultados. A parte de aplicação do jogo também foi contemplada em um modelo próprio. Já a parte principal do trabalho foi representada em dois modelos: Os procedimentos de projetar o jogo e de estruturar o jogo.

O modelo proposto permitiu o desenvolvimento de um jogo com propósito para classificação de dados utilizando mecânicas de jogos de tiro. O jogo desenvolvido, além de camuflar os passos para solucionar o problema real, obteve um desempenho bastante semelhante aos algoritmos de classificação.

O jogo com propósito desenvolvido obteve vantagem em relação aos algoritmos matemáticos no quesito do tamanho do conjunto de treinamento. Obteve um resultado semelhante com um conjunto de treinamento três vezes menor. O jogo se mostrou melhor aproveitado quando foram realizadas classificações binárias.

7.2 Contribuições

A principal contribuição do modelo proposto pelo trabalho foi a abordagem referente a inserção de mecânicas tradicionais de jogos no *gameplay* dos jogos com propósito. O estudo do problema real acarreta na identificação dos passos executados para a obtenção da solução do problema. Com esses passos identificados é possível representá-los dentro do ambiente do jogo utilizando mecânicas tradicionais de jogo. Assim, ao utilizar as mecânicas, o jogador será capaz de encontrar a solução do problema de uma maneira lúdica e distante da forma real.

Além do processo para a inserção de mecânicas de jogos tradicionais nos jogos com propósito, os modelos desenvolvidos permitiram a visualização do processo de solução de problemas computacionais utilizando jogos com propósito como um todo. Apesar do foco do trabalho ser o uso das mecânicas de jogos, o modelo desenvolvido contemplou todos os momentos envolvidos.

Tendo a visão completa do processo foi possível identificar pontos de reusabilidade, desde pequenos pontos como a representação gráfica até a solução do problema como um todo. Uma representação gráfica utilizada para uma determinada solução pode ser aplicada em outra. A própria solução lúdica pode ser utilizada em outro problema alterando-se apenas os dados de entrada.

7.3 Limitações e trabalhos futuros

Os procedimentos apresentados nesse trabalho possuem uma restrição em relação ao tipo de problemas que podem ser aplicados. Os problemas computacionais ficam restritos a problemas que possuam suas informações estruturadas. As entradas do modelo proposto ficam limitadas a bases de dados.

Devido a essa característica da solução proposta, pode-se pensar em um trabalho futuro para a evolução do modelo. Essa nova versão da proposta aumentaria o leque de problemas a serem atendidos. Tornando ele uma solução menos específica possível.

O modelo desenvolvido pode evoluir para um ambiente de desenvolvimento que auxilie na elaboração de jogos com propósito. Esse ambiente pode formalizar os passos necessários para o projeto do jogo. Pode, também, auxiliar centralizando soluções prévias para serem reutilizadas.

Esse ambiente possuiria vários repositórios onde poderiam ser encontradas diferentes representações gráficas, soluções definidas para os possíveis modelos de dados, entre outras facilidades, visando tornar o processo de desenvolvimento mais rápido e mais eficiente.

Referências Bibliográficas

- VON AHN, L., 2005. *Human Computation*. Tese de Doutorado: Carnegie Mellon University, Pittsburgh, Pennsylvania, United States.
- VON AHN, L., 2009, "Human computation". In: *Proceedings of the 46th Annual Design Automation Conference on ZZZ - DAC '09*. pp. 418.
- VON AHN, L., DABBISH, L., 2004. "Labeling images with a computer game". In: *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*. New York, New York, USA: ACM Press. 2004. pp. 319–326.
- VON AHN, L., DABBISH, L., 2008, "Designing games with a purpose". In: *Communications of the ACM*. v. 51, n. 8 (Aug.), pp. 57 – 67.
- VON AHN, L., LIU, R., BLUM, M., 2006. "Peekaboom: A Game for Locating Objects in Images". In: *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*. New York, New York, USA: ACM Press. 2006. pp. 55 – 64.
- VON AHN, L., MAURER, B., MCMILLEN, C., ABRAHAM, D., BLUM, M., 2008, "reCAPTCHA: human-based character recognition via Web security measures.". In: *Science (New York, N.Y.)*. v. 321, n. 5895 (Sep.), pp. 1465–8.
- BATES, B., 2004, *Game Design*. 2ª ed. Boston, Course Technology PTR.
- BATISTA, G.E. DE A.P.A., 2003. *Pré-processamento de Dados em Aprendizado de Máquina Supervisionado*. Tese de Doutorado: Universidade de São Paulo, São Paulo, São Paulo, Brasil.
- BRABHAM, D.C., 2008, "Crowdsourcing as a Model for Problem Solving: An Introduction and Cases". In: *Convergence: The International Journal of Research into New Media Technologies*. v. 14, n. 1 (Feb.), pp. 75–90.
- BRATHWAITE, B., SCHREIBER, I., 2009, *Challenges for Game Designers*. 1ª ed., Boston, Course Technology PTR.

- BUHRMESTER, M., KWANG, T., GOSLING, S.D., 2011, "Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data?". In: *Perspectives on Psychological Science*. v. 6, n. 1 (Feb.), pp. 3–5.
- CASTRO, C.L. DE, BRAGA, A.P., 2011, "Aprendizado supervisionado com conjuntos de dados desbalanceados". In: *Revista Controle & Automação*. v. 22, n. 5.
- CHERNOFF, H., 1973, "The use of faces to represent points in k-dimensional space graphically". In: *Journal of the American Statistical Association*. v. 68, n. 342, pp. 361–368.
- COOPER, S., KHATIB, F., TREUILLE, A., BARBERO, J., LEE, J., BEENEN, M., LEACER-FAY, A., BAKER, D., POPOVIC, Z., 2010, "Predicting protein structures with a multiplayer online game.". In: *Nature*. v. 466, n. 7307 (Aug.), pp. 756–760.
- DILLON, R., 2010, *On the Way to Fun: An Emotion Based Approach to Successful Game Design*. 1ª ed., Natick, A K Peters/CRC Press.
- EUSTACE, A., *A fall spring-clean* 2011. Disponível em: <<http://googleblog.blogspot.com.br/2011/09/fall-spring-clean.html>>. Acessado em: 9 Junho 2012.
- FRANK, A., ASUNCION, A., 2010. *UCI Machine Learning Repository*. <<http://archive.ics.uci.edu/ml>> Irvine,CA: University of California, School of Information and Computer Science.
- HAN, J., KAMBER, M., 2006, *Data Mining: Concepts and Techniques*. Second edition. S.I., Burlington, Morgan Kaufmann Publishers.
- HENDERSON, L., 2005. "Video games: A significant cognitive artifact of contemporary youth culture". In: *DiGRA 2005 Conference: Changing Views – Worlds in Play*. S.I.: s.n. 2005.
- HENDERSON, L., HENDERSON, M., GRANT, S., HUANG, H., 2010, "What are users thinking in a virtual world lesson? Using stimulated recall interviews to report student cognition, and its triggers". In: *Journal of Virtual Worlds* v. 3.
- HOWE, J., 2006, "The rise of crowdsourcing". In: *Wired magazine*. n. 14.

- JOVIAN, L.T., AMPRIMO, O., 2011. "OCR Correction via Human Computational Game". In: *2011 44th Hawaii International Conference on System Sciences*. S.l.: IEEE. Janeiro 2011. pp. 1–10.
- KABULOV, B.T., TASHPULATOVA, N.B., 2010, "Using hands for data visualization". In: *2010 4th International Conference on Application of Information and Communication Technologies*. pp. 1–2.
- KAWRYKOW, A., ROUMANIS, G., KAM, A., KWAK, D., LEUNG, C., WU, C., ZAROOUR, E., SARMENTA, L., BLANCHETTE, M., WALDISPUHL, J., 2012, "Phylo: a citizen science approach for improving multiple sequence alignment.". In: *PloS one*. v. 7, n. 3 (Jan.), pp. e31362.
- KHATIB, F., DIMAIO, F., COOPER, S., KAZMIERCZYK, M., GILSKI, M., KRZYWDA, S., ZABRANSKA, H., PICOVA, I., THOMPSON, J., POPOVIC, Z., JASKOLSKI, M., BAKER, D., 2011, "Crystal structure of a monomeric retroviral protease solved by protein folding game players.". In: *Nature structural & molecular biology*. v. 18, n. 10 (Oct.), pp. 1175–7.
- KLEINER, B., HARTIGAN, J.A., 1981, "Representing Points in Many Dimensions Representing by Trees and Castles". In: *Journal of the American Statistical Association*. v. 76, n. 374, pp. 260–269.
- PAOLACCI, G., CHANDLER, J., STERN, L.N., 2010, "Running experiments on Amazon Mechanical Turk". In: *Judgment and Decision Making*. v. 5, n. 5, pp. 411–419.
- QUINN, A.J., BEDERSON, B.B., 2011. "Human computation: a survey and taxonomy of a growing field". In: *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. New York, New York, USA: ACM Press. 2011. pp. 1403 – 1412.
- ROGERS, S., 2010, *Level Up! The Guide to Great Video Game Design*. 1^a ed. , New Jersey, John Wiley & Sons.
- SCHELL, J., 2008, *The Art of Game Design: A Book of Lenses*. , Burlington, Morgan Kaufmann Publishers.

SILVA, M.M., 2008. *Uma Abordagem Evolucionária Para o Aprendizado Semi-Supervisionado em Máquinas de Vetores de Suporte*. Tese de Doutorado. Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brasil.

SUROWIECKI, J., 2006, *A sabedoria das multidões*. 1ª edição, Rio de Janeiro, Editora RECORD.

VUKOVIC, M., 2009, "Crowdsourcing for Enterprises". In: *2009 Congress on Services - I*. pp. 686–692.