

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CIÊNCIA DA COMPUTAÇÃO

Andrei Michel Schwingel

**"Classics Grandmaster" - Um Jogo Digital Sobre Jogos Lógicos de
Tabuleiro Inspirado no Programa de Extensão LoBoGames**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Ciência da Computação.

Orientador: Prof. Dr. Renato Perez Ribas

Porto Alegre
2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do Curso de Ciência da Computação: Prof. Carlos Arthur Lang Lisbôa

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Quero agradecer a todos que tornaram este trabalho possível, bem como toda minha trajetória para concluir o curso de Ciência da Computação.

Começo pelos meus familiares, por serem sempre os mais envolvidos, que ficaram mais felizes ou mais tristes com cada acontecimento ao longo desses anos. Em especial agradeço aos meus pais, Charles e Luiza, porque fizeram tudo que foi possível para que eu chegasse até aqui e por incontáveis vezes sei que estiveram orando para que essa conclusão fosse possível. Amo vocês, vocês são os maiores merecedores desta conclusão.

Outro agradecimento especial vai para minha esposa Cristiane, que desde os primeiros dias em Porto Alegre, antes até de sermos namorados, sempre me ajudou. Acompanhou de perto todos os anos de curso e parece estar até mais empolgada do que eu por esta conclusão. Te amo!

Agradeço também aos meus amigos e colegas que em algum momento do curso participaram de trabalhos, de estudos ou deram alguma orientação. Em especial agradeço à minha amiga Anita, que aceitou trabalhar nesse jogo, fazendo mais um excelente trabalho.

Agradeço aos meus professores do curso, por contribuírem com meu conhecimento que permitiu fazer este trabalho. Antecipadamente aos professores da banca de avaliação, por suas orientações. Em especial ao professor Renato, meu orientador, porque mesmo sem me conhecer nem ter dado uma aula para mim sequer, compartilhou suas grandes ideias e acreditou que eu poderia levar uma delas adiante através deste trabalho, possibilitando que eu realizasse o desejo de criar um jogo como trabalho de conclusão.

Este trabalho é um pouco de cada um de vocês.

Por fim agradeço a Deus, por ter realizado toda engenharia desses anos, me dado a possibilidade de conhecer todas essas pessoas, me dado força, orientação, e tudo mais que precisei para chegar aqui.

RESUMO

O *Classics Grandmaster* é um jogo digital que exercita o raciocínio lógico do jogador através da aprendizagem de vários jogos lógicos de tabuleiro conhecidos. Esses jogos são organizados em uma sequência de dificuldade crescente, como se fossem as fases do jogo. Esse roteiro corresponde à didática utilizada no Programa de Extensão LoBoGames, um projeto que acontece na Universidade Federal do Rio Grande do Sul. Além de aprender, o jogador também tem a possibilidade de jogar qualquer um desses jogos contra o computador ou um segundo jogador, como qualquer jogo lógico de tabuleiro em sua versão digital. O *Classics Grandmaster*, portanto, se diferencia dos demais jogos de tabuleiro presentes no mercado pela utilização desse método de aprendizagem. Este trabalho apresenta o estudo de tecnologias e os detalhes de implementação envolvidos na sua criação. Tanto o estudo quanto a implementação detalham apenas o que é mais relevante no contexto do curso de Ciência da Computação, a saber: decisões sobre plataforma e ambiente de desenvolvimento; a estrutura que define cada jogo de tabuleiro dentro do ambiente de desenvolvimento; o funcionamento geral dos jogos de tabuleiro implementados; como a composição dos jogos foi pensada para que seja fácil incluir novos jogos no futuro; e como foram implementados alguns requisitos da interface e experiência do usuário.

Palavras-chave: Jogos lógicos de tabuleiro. Desenvolvimento de jogos. Jogos digitais.

"Classics Grandmaster" - a Digital Game About Logical Board Games Inspired by the Extension Program LoBoGames

ABSTRACT

Classics Grandmaster is a digital game that exercises the player's logical reasoning by learning various known logical board games. These games are arranged in a sequence of increasing difficulty, like the stages of the game. This script corresponds to the didacticism used in Extension Program LoBoGames, a project that happens in the Federal University of Rio Grande do Sul. In addition to learning, the player can also play any of these games against the computer or a second player, as any logical board game in its digital version. *Classics Grandmaster*, therefore, differs from other board games on the market by the use of this learning method. This paper presents the study of technologies and implementation details involved in its creation. Both the study and the implementation detail only what is most relevant in the Computer Science course's context, namely: decisions about platform and development environment; the structure that defines each board game within the development environment; the overall functioning of the implemented board games; how the games' composition was designed to make it easy to include new games in the future; and how some interface and user experience requirements were implemented.

Keywords: Logical board games. Game development. Digital games.

LISTA DE FIGURAS

Figura 4.1 – Comunicação entre a peça e o tabuleiro.....	20
Figura 4.2 – Seleção da peça na jogada <i>mover</i>	24
Figura 4.3 – Seleção da casa na jogada <i>mover</i>	24
Figura 4.4 – Os oito alinhamentos do Tic-Tac-Toe	26
Figura 4.5 – Tela inicial	28
Figura 4.6 – Tela de configurações.....	29
Figura 4.7 – Tela de instruções.....	29
Figura 4.8 – Tela de registro de jogador	30
Figura 4.9 – Tela de escolha do módulo	31
Figura 4.10 – Tela de escolha do jogo	31
Figura 4.11 – Tela de jogo	32

LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
SDK	Software Development Kit
UFRGS	Universidade Federal do Rio Grande do Sul

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Metodologia do Lobogames	9
1.2 Proposta	10
2 ESTUDO DE TECNOLOGIAS E REQUISITOS	11
2.1 Plataforma de destino.....	11
2.2 Sistema operacional.....	12
2.3 Ambiente de desenvolvimento	13
2.4 Portabilidade	14
2.5 Apresentação visual	14
3 DESCRIÇÃO DO JOGO.....	16
4 IMPLEMENTAÇÃO	19
4.1 Estruturação dos jogos no motor gráfico	19
4.2 Programação genérica	20
4.2.1 Movimento das peças.....	22
4.2.2 Condição de vitória.....	24
4.2.3 Inteligência artificial	26
4.3 Apresentação visual	26
4.3.1 Arquitetura da interface	27
4.3.2 Experiência do usuário	32
4.3.3 Arte.....	35
4.3 Inclusão de um novo jogo.....	36
5 CONCLUSÕES E TRABALHOS FUTUROS	38
REFERÊNCIAS.....	40

1 INTRODUÇÃO

Os jogos de tabuleiro hoje são um passatempo bastante popular. Com todo seu tempo de existência foi possível estudá-los e perceber quão bons eles são em exercitar o raciocínio lógico (Huizinga, 1955; Bright, 1983; Friedrich e Preiss, 2006). Muitas escolas hoje utilizam jogos de tabuleiro para exercitar essa capacidade nas crianças¹.

A inserção de jogos de tabuleiro em escolas, especificamente os jogos lógicos, é justamente um dos objetivos do Programa de Extensão LoBoGames. O LoBoGames é um programa de extensão da UFRGS que trabalha para divulgar e despertar o interesse pelo aprendizado dos jogos lógicos de tabuleiro.

Para alcançar esse objetivo, o Programa promove muitas atividades. Entre elas: atuação em escolas, formação de professores, pesquisa acadêmica e construção de jogos. Estas são as atividades que fizeram este trabalho existir.

Os jogos construídos no Programa tem diferentes modalidades. Partindo da mais básica, o jogo de mesa, como tradicionalmente eles são conhecidos. Depois duas bem diferentes, que são as modalidades “gigante”, na qual o tabuleiro é realmente bastante grande, tendo até dois metros de largura, e a “vivo”, na qual as peças são os próprios participantes. A última é com os jogos no computador, no qual o jogador humano joga contra o computador, um programa que utiliza técnicas de Inteligência Artificial para definir suas jogadas. Nesse contexto de construção de jogos para computador é que este trabalho começou a se formar.

1.1 Metodologia do LoBoGames

O Programa utiliza uma metodologia própria para apresentar os jogos lógicos para quem está querendo aprender. Ela é fruto de pesquisa dos seus organizadores e não encontra nenhum método similar. Ela se resume em apresentar os jogos em uma sequência didática, começando do “jogo mais simples do mundo” e crescendo em regras e complexidade a cada novo jogo, conforme o participante compreende cada um.

¹ Disponível em http://www.inf.ufrgs.br/lobogames/wp-content/uploads/2015/07/pesquisa_artigo_2015_06.pdf (acesso em: 17/12/2015)

Os jogos são divididos em módulos, cada um contendo um conjunto de jogos de uma determinada mecânica. As mecânicas, assim como os jogos, também crescem em complexidade. Dessa forma, o participante não é apresentado diretamente a um jogo cheio de regras que ele pode ter muita dificuldade em compreender. Ele começa com algo muito simples e vai progredindo, crescendo em conhecimento a cada novo jogo. Assim, com o raciocínio construído aos poucos, ele chega nos jogos difíceis, como Xadrez, com muito mais segurança.

Essa metodologia hoje, só é aplicada utilizando os jogos físicos. Não existe nenhuma aplicação dela com jogos digitais. Percebendo essa lacuna que surgiu a ideia de criar uma ferramenta computacional que realizasse essa tarefa, o que seria uma grande contribuição para o Programa. Sobre essa ideia que surgiu a proposta deste trabalho.

1.2 Proposta

O objetivo do *Classics Grandmaster* é ser um jogo digital que exercita o raciocínio lógico através do aprendizado e prática de jogos lógicos de tabuleiro. Ele apresenta ao jogador diversos jogos conhecidos, organizados em uma sequência especial, como se fossem as fases do jogo. O jogador precisa vencer uma partida contra o programa em um jogo para liberar o próximo e assim prosseguir no jogo. Esse fluxo corresponde à didática do LoBoGames e é a principal contribuição deste trabalho.

A organização da sequência de jogos é mais importante do que os jogos em si. Não é importante ter um determinado jogo de tabuleiro ou outro. O foco está em criar uma sequência que é modular, com um processo simples para o desenvolvedor incluir novos jogos ou modificar a ordem. Essa é a segunda contribuição importante deste trabalho, a capacidade de estender o jogo com facilidade, incluindo novos jogos.

Este trabalho apresenta primeiramente o estudo de tecnologias e requisitos, o qual definiu a plataforma para a qual esse jogo vai ser desenvolvido e algumas expectativas sobre sua apresentação visual. Depois são detalhados os aspectos de implementação: a engenharia em torno da montagem dos diversos jogos de tabuleiro, sua execução dentro do jogo e como foram atendidas as duas contribuições citadas.

2 ESTUDO DE TECNOLOGIAS E REQUISITOS

A criação de um jogo é uma tarefa multidisciplinar. Equipes de desenvolvimento normalmente contam com profissionais de diferentes áreas, cada um contribuindo com seu conhecimento em diferentes momentos. Iniciando com estudos de mercado, passando pela criação da arte conceitual, implementação, até o marketing do jogo concluído. Não é o objetivo deste trabalho detalhar esse processo.

O estudo que será apresentado foca em decisões tecnológicas indispensáveis para o desenvolvimento e o atendimento dos requisitos da proposta descrita anteriormente. Também foram consideradas no estudo as experiências anteriores de adaptação de jogos do LoBoGames, trabalhos anteriores similares a este e a experiência pessoal do autor.

2.1 Plataforma de destino

A primeira decisão indispensável é para qual plataforma destinar o jogo. Saber esse destino restringe as possibilidades de ferramentas de desenvolvimento, as possibilidades de interagir com o jogo e a possibilidade de elaboração visual - decisões importantes em passos seguintes.

Com o crescimento do mercado de jogos e o surgimento de novas plataformas que conseguem executar um jogo, existem hoje muitas opções. Alcançar mais de uma delas também é uma possibilidade, não são poucos os casos de jogos multiplataforma. Porém, para uma primeira versão do *Classics Grandmaster*, foi decidido escolher uma só, considerando a facilidade de portabilidade a partir dela.

Desde a proposta inicial de desenvolver este trabalho, já havia uma preferência por executar o jogo em smartphones e tablets. A maior mobilidade, menor necessidade de equipamentos extras e espaço em relação ao computador e especialmente aos videogames, fazem essas plataformas serem muito mais práticas. A experiência de jogo é mais próxima da versão física, ainda que distante, por utilizar as mãos diretamente na tela, do que nas outras possibilidades, em que há algo mais entre o jogo e o jogador (mouse ou joystick). Por se tratarem de jogos simples, não há nenhum aproveitamento da maior capacidade processamento das outras opções.

Além da preferência prévia, as pesquisas também apontam que optar pelos tablets e smartphones é um acerto. No Brasil, segundo a Pesquisa Game Brasil

2015², as plataformas de jogos eletrônicos mais utilizadas são: computador, videogame, tablet e smartphone. Dentre elas, a mais utilizada e preferida é o smartphone. No mundo, a porcentagem de pessoas que possui um smartphone já é maior que a das que possuem computador. Entre as pessoas que jogam jogos digitais, o celular também é a plataforma preferida. As crianças já têm também a mesma preferência³.

As projeções para os próximos anos indicam que esses números favoráveis aos smartphones vão aumentar. Somando tudo isso, conclui-se que desenvolver jogos para essa plataforma, aproveitando seu crescimento e popularização entre jogadores, é um bom plano.

Por fim, como o LoBoGames já tinha tido algumas experiências com jogos no computador. Para uma nova experiência, a opção por smartphones e tablets foi definida como a inicial.

2.2 Sistema operacional

A partir da decisão da plataforma, estudou-se a atual distribuição de sistemas operacionais no mercado de smartphones e tablets. Uma pesquisa⁴ recente mostrou que ele é amplamente dominado pelos que rodam Android. Se considerarmos junto o iOS, a presença dos demais concorrentes é insignificante e existe uma previsão de que isso não vai mudar nos próximos dois ou três anos⁵. Mesmo entre os dois, a presença do Android é tão maior que quase não há necessidade de avaliar essa questão: mais de 80% dos smartphones hoje rodam Android e mesmo em tablets⁶, aparelho que o iOS já foi totalmente dominante, a presença hoje é dominante do Android.

O que se avaliou em favor do iOS é sua menor quantidade de aparelhos com diferentes configurações de hardware no mercado. Isso ajuda no desenvolvimento,

² Disponível em <http://www.pesquisagamebrasil.com.br/> (acesso em: 30/11/2015)

³ Disponível em <http://www.gamevicio.com/i/noticias/221/221498-pc-nao-e-mais-a-plataforma-preferida-das-criancas/> (acesso em: 30/11/2015)

⁴ Disponível em <http://www.idc.com/prodserv/smartphone-os-market-share.jsp> (acesso em: 30/11/2015)

⁵ Disponível em <http://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems/> (acesso em: 16/12/2015)

⁶ Disponível em <http://www.statista.com/statistics/273840/global-market-share-of-tablet-operating-systems-since-2010/> (acesso em: 30/11/2015)

pois se tem uma quantidade menor de especificações, sendo mais previsível o comportamento do produto naquele sistema operacional. Outro fator interessante é que o público da loja de aplicativos para iOS costuma gastar mais dinheiro que o da loja do Android em compras de aplicativos⁷. Também há uma dificuldade maior de se publicar nessa loja, portanto há uma concorrência menor, porém de maior qualidade.

Por outro lado, desenvolver para iOS requer um computador com sistema operacional OS X, algo que não acontece com o Android. Os aparelhos para teste também são mais caros e menos pessoas os têm.

Considerando todos esses pontos, a decisão foi de desenvolver inicialmente para o sistema operacional Android. Atualmente existem dispositivos rodando diversas versões desse sistema. A versão mínima para rodar o *Classics Grandmaster* é a 4.0.3. Segundo os dados coletados pelo Google⁸, ela é suportada por aproximadamente 96% dos dispositivos com Android atualmente.

2.3 Ambiente de desenvolvimento

A oferta de ambientes de desenvolvimento de jogos é muito grande. Desde frameworks de código aberto, passando por diversos *SDK*, até motores de jogos. Cada uma das opções funciona com diferentes linguagens de programação e oferece diferentes variedades de ferramentas internas para facilitar e incrementar o desenvolvimento. Se forem combinados os diferentes ambientes com os possíveis editores de código, as combinações são tão diversas que a decisão final fica mesmo por conta da preferência de cada desenvolvedor.

Para a implementação deste trabalho, a escolha acabou sendo por experiência prévia e familiaridade do autor. A ferramenta escolhida foi o motor de jogos Unity3D⁹ (também chamado somente de Unity). Não que as vantagens e desvantagens não sejam importantes, mas para o desenvolvimento de um jogo em curto espaço de tempo e sem uma equipe completa, o mais importante é atingir o resultado que se quer com a ferramenta que se sabe usar. Se não há um motivo muito forte, algo que

⁷ Disponível em <http://www.mobileindustryreview.com/2014/12/android-revenue-less-than-you-think.html> (acesso em: 30/11/2015)

⁸ Disponível em <http://developer.android.com/about/dashboards/index.html> (acesso em 30/11/2015)

⁹ Disponível em <http://www.unity3d.com> (acesso em 16/12/2015)

impeça o desenvolvimento, o recomendado entre os desenvolvedores é que cada um use a ferramenta que domina.

A Unity3D é a ferramenta mais utilizada por desenvolvedores de jogos profissionais hoje¹⁰, e provavelmente também seja entre os iniciantes. Ela suporta tanto jogos 2D como 3D, sendo que para este trabalho foram utilizadas apenas as ferramentas 2D. Suporta três linguagens de programação, dentre as quais C#, que foi a utilizada. Também suporta a publicação para várias plataformas, um ponto positivo pensando na portabilidade do jogo. Durante o processo de desenvolvimento, por exemplo, rapidamente foi possível testar, a partir do projeto para Android, uma versão WebGL em navegadores que suportam essa tecnologia. Não é um motor gráfico de código aberto, e existe uma versão Pro, que é paga. Para o jogo desenvolvido, não se sentiu a necessidade de modificar códigos internos e a versão gratuita foi suficiente.

2.4 Portabilidade

A possibilidade de levar esse jogo para outras plataformas é bem importante no objetivo de encontrar espaço no mercado de jogos e tornar o jogo acessível para muitos usuários. Nesse sentido, a utilização da Unity torna-se muito útil, já que ela suporta a publicação multiplataforma.

Cada nova plataforma exige um pouco de adaptação. Muda o público, a maneira de monetizar, a interação por parte do usuário - só para citar pontos relevantes a este jogo. Mesmo assim, por sua semelhança, a portabilidade para iOS é simples. Para navegadores que suportam WebGL, é ainda mais simples, e deve ser um primeiro trabalho futuro. Outras opções de plataforma não parecem vantajosas no momento, então não receberam atenção neste estudo.

2.5 Apresentação visual

Com experiências anteriores do Programa LoBoGames, dois pontos foram os principais aprendizados em relação ao visual do jogo: 1) a interface deve ser comercial, isto é, deve ser bem acabada, bem definida, atrativa para o jogador; 2) o jogo pode ser simples, não há necessidade de gráficos elaborados, basta algo que

¹⁰ Disponível em <http://unity3d.com/public-relations> (acesso em 30/11/2015)

deixe claro para o jogador cada momento do jogo, como o movimento das peças, por exemplo: o ideal é que exista movimento na tela, não que a peça desapareça de um lugar e apareça em outro.

Com esses aprendizados e considerando que a plataforma de destino são smartphones e tablets, a primeira preocupação visual foi em ter um jogo leve. Os gráficos 2D foram a opção nesse caso. Além de ser mais leve que o 3D, é o mais indicado para projetos com poucas pessoas envolvidas, como é esse.

Essa opção, porém, traz um pouco mais de necessidade de cuidado artístico, para que todos jogadores, com diferentes tamanhos de tela, tenham a mesma experiência. A interface deve se adaptar para que não fique nenhuma informação fora da tela, problema comum com dispositivos que possuem telas com diferentes aspectos. Cada aparelho deve receber gráficos adequados à sua resolução: com mais resolução para os que tem mais resolução, para que as imagens não fiquem *pixeladas*; com menos resolução os que tem menos resolução, para não utilizar mais memória que o necessário para carregar imagens.

No estudo da interface também foi importante considerar que o jogo deve ser acessível para pessoas de idades bastante diferentes, de crianças a idosos. Também para pessoas com diferentes experiências em jogos de tabuleiro, já que ele se propõe a ensinar a jogar.

Foi preciso estudar a utilização de informações que dão a ideia de aprendizado e que auxiliam nesse processo. Desde os menus antes de uma partida até os botões disponíveis durante ela, o funcionamento foi pensado não para um único jogo, mas considerando que vários jogos são apresentados e com o objetivo de criar um ambiente de aprendizado e progressão.

Para a representação dos tabuleiros e peças, a principal preocupação foi que eles fossem facilmente identificados. No LoBoGames, na modalidade de jogos de mesa, esses dois elementos são bastante simples. O tabuleiro é impresso em uma folha e as peças são tampinhas de garrafa. Para uma representação digital, o objetivo foi manter essa simplicidade.

3 DESCRIÇÃO DO JOGO

O *Classics Grandmaster* é, primeiramente, um jogo digital que exercita o raciocínio lógico do jogador. Para isso, ele utiliza diversos jogos lógicos de tabuleiro conhecidos, organizados em módulos e em uma sequência crescente de dificuldade, como se fossem as fases do jogo.

O Programa LoBoGames é a grande influência e inspiração desse jogo. As duas contribuições que ele traz são do interesse do Projeto. A primeira é que o fluxo de jogo corresponde à didática do Programa.

Quando um jogador inicia o *Classics Grandmaster* pela primeira vez, ele encontra apenas um módulo de jogos disponível, e dentro dele um único jogo disponível. Esse jogo é o mais simples de todos que poderão ser jogados. Isso é o que acontece também quando uma pessoa inicia no método do LoBoGames, ela sempre começa do jogo mais simples. Uma vez que a pessoa compreende esse jogo, ela passa para o próximo. Nesta “versão digital” também, uma vez que o jogador supera o computador em um jogo, ele libera o próximo daquele módulo.

Esse fluxo se repete para todos os jogos daquele módulo. Quando o jogador vencer em todos, o módulo seguinte é liberado e ele segue o mesmo roteiro. Hoje, no LoBoGames, são seis módulos de jogos no total. Para a versão do *Classics Grandmaster* apresentada neste trabalho, foi planejado apenas um módulo com jogos. O código interno, porém, deve considerar esse sistema de módulos.

A segunda contribuição é a possibilidade de facilmente ampliar o jogo, incluindo mais módulos e jogos. Todos os jogos de tabuleiro que compõe os módulos são jogos já existentes. O importante não é a presença de um ou outro jogo especificamente, mas a sequência em que eles estão organizados e são apresentados para o jogador. Portanto, é importante ter a capacidade de facilmente modificar essa sequência, seja incluindo mais jogos, ou alterando ordem em que eles aparecem. Para isso ser algo simples, é preciso que o método de inclusão de jogos seja simples. A engenharia na estruturação dos jogos na Unity3D deve ser feita pensando nisso.

O *Classic Grandmaster* pode ser visto como um jogo no qual o jogador passa por diferentes “mundos” e em cada um deles passa por várias fases que crescem em dificuldade. Os módulos seriam os “mundos” e cada fase um jogo de tabuleiro diferente que acrescenta mais regras e/ou dificuldade em relação ao anterior. Cada módulo representa uma mecânica específica que os jogos dentro dele têm em comum.

Em segundo lugar, o *Classics Grandmaster* também serve para praticar esses jogos. Uma vez que um jogo é liberado, o jogador pode voltar a ele a qualquer momento. Assim é possível ter acesso a muitos jogos num mesmo “lugar”, sem precisar ter cada um individualmente. A liberação de um jogo também permite que se jogue contra um segundo jogador humano. Assim, todos jogos devem oferecer a opção de serem jogados por dois jogadores humanos.

Por último, o *Classics Grandmaster* também serve para ensinar outras pessoas a jogar os que ele apresenta. Se o LoBoGames quiser utilizá-lo nas suas atividades, isso é possível. Alguém com conhecimento pode controlar o jogo, utilizando seus tabuleiros digitais no lugar dos físicos. Pessoas que desconhecem o LoBoGames também podem usar o jogo dessa maneira. Toda a sequência de jogos já está definida para facilitar o aprendizado e, assim, também o ensino.

Por essas três características, o *Classics Grandmaster* é um jogo diferente dos demais na categoria de jogos de tabuleiro para smartphones e tablets. Com sua maneira de apresentar os jogos, explorando o aprendizado e não somente a prática, pretende ser inovador no mercado de jogos digitais e conseguir um espaço entre a grande quantidade de opções.

Se aproveitando de não existir um jogo semelhante e sabendo que o método do LoBoGames é eficaz presencialmente, criar um jogo com esse método aplicado digitalmente tem grandes chances de encontrar espaço no mercado e fazer surgir novos jogadores de tabuleiro. Os jogos utilizados podem ser facilmente encontrados em diferentes plataformas: portais de jogos para navegadores na Internet, sites sobre inteligência artificial em jogos, lojas de aplicativos para smartphones e tablets. Porém, eles foram desenvolvidos pensando apenas na prática das pessoas que já os conhecem e jogam. Não estão agrupados em nenhum tipo de ordem e em geral são encontrados individualmente, ficando totalmente por conta do jogador encontrá-los e aprendê-los.

O fato de ser digital ainda traz algumas possibilidades bastante úteis em relação ao modo presencial. A capacidade de alcance é muito maior que a de um grupo que se encontra presencialmente, rapidamente se alcança o mundo todo com um jogo disponível *online*. Com tabuleiros e peças digitais, todos jogos estão no mesmo “lugar” e é fácil e rápido trocar de jogo. É possível jogar sozinho, enfrentando o programa; personalizar tabuleiros e peças para jogadores de diferentes idades;

utilizar elementos visuais para fazer indicações de jogadas, destacar peças, registrar jogadas.

Mesmo não sendo um jogo de tabuleiro digital comum, é importante que o jogo acompanhe a concorrência na apresentação visual. O visual gráfico deve ter aparência comercial, ser bem desenvolvido. Pessoas de todas as idades já participaram do LoBoGames e tiveram sucesso. Por isso é importante que a arte do jogo represente os tabuleiros e peças com clareza e simplicidade para todas elas. Por experiência com outras ferramentas computacionais no Programa LoBoGames, concluiu-se que a falta de cuidado com o visual impedia o jogo de ser atraente para os participantes e eles ficavam sem vontade de jogar.

4 IMPLEMENTAÇÃO

Para este trabalho, a parte importante da implementação foi atender aos requisitos e contribuições. Não foi o objetivo a implementação total do jogo, como já comentado inicialmente, mas implementar o fluxo de jogo e o modelo que permite adicionar facilmente outros jogos. Como teste mínimo para esse funcionamento, foram escolhidos alguns jogos do Programa LoBoGames para serem incluídos na versão que este trabalho apresenta. Os jogos incluídos são todos do “Módulo 1: jogos de bloqueio e alinhamento”, e são: Pong Hau K'i, Madelinette, Mu Torere, Picaria, Tapatan, Three Men's Morris, Altan Xaracaaj, Shisima, Tic-Tac-Toe, Tsoro Yematatu, a variante com posicionamento do Three Men's Morris e o Five Men's Morris. As regras e tabuleiros desses jogos podem ser encontrados no site do Programa LoBoGames¹¹.

4.1 Estruturação dos jogos no motor gráfico

O primeiro passo para entender como o *Classics Grandmaster* foi implementado para alcançar os objetivos propostos, é entender de um modo geral a estruturação que os jogos têm dentro da Unity3D. É preciso saber que a Unity estrutura todos elementos do jogo através da lógica de orientação a componentes. Tudo que for usado dentro do jogo será usado em forma de componente: desde os diversos programas escritos pelo programador (os *scripts*), até qualquer gráfico criado por um artista.

A estrutura interna mais básica na qual se pode adicionar componentes é um objeto chamado *GameObject*. Um *GameObject* vazio é aquele que não tem nenhum componente adicionado, tendo apenas um registro de sua posição no mundo 3D, chamado de *Transform*. Todo elemento do jogo é inicialmente um *GameObject* vazio, e conforme recebe componentes vai se tornando aquilo que o desenvolvedor quer.

Uma peça de qualquer um dos jogos implementados, por exemplo, é composta por dois componentes principais: o *Image*, um componente que exhibe uma imagem 2D na tela, no caso o desenho da peça; e um componente definido pelo programador, um *script*, que resolve a lógica contida na peça.

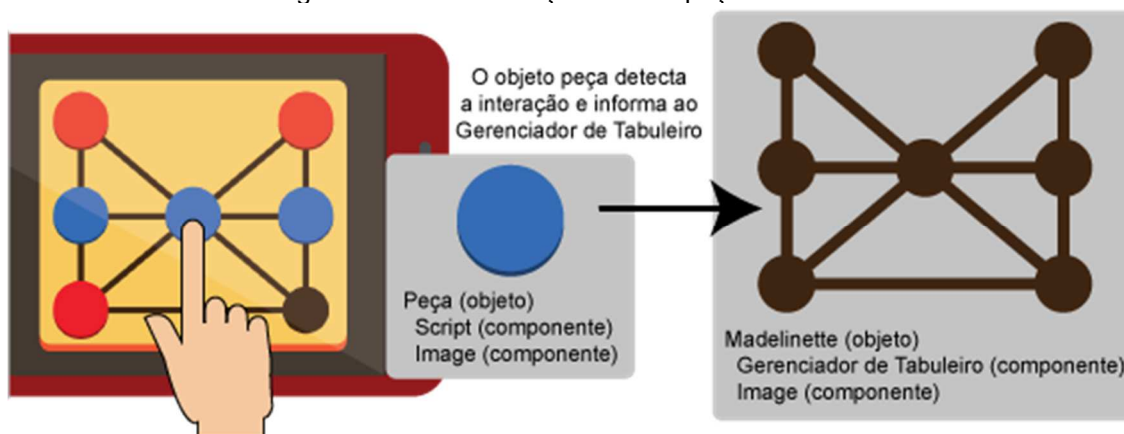
¹¹ <http://www.inf.ufrgs.br/lobogames/#jogos> (acesso em: 30/11/2015)

É possível adicionar um *GameObject* a outro criando uma hierarquia. Todos os jogos implementados foram criados tendo como objeto-pai o desenho de seus tabuleiros. Cada tabuleiro é um objeto com vários componentes e diversos objetos-filho, também formados por diversos componentes, e organizados em dois grupos: um com as casas do tabuleiro e outro com as peças. Quando o jogador decide qual jogo ele quer jogar, o tabuleiro correspondente é carregado para a cena de jogo e pode ser exibido na tela. A cena de jogo é o ambiente virtual no qual estão todos os elementos do jogo já carregados.

Existem diferentes componentes em diferentes objetos que controlam e gerenciam o funcionamento do jogo como um todo. O principal componente que será citado neste trabalho é o *Gerenciador de Tabuleiro*. Ele é um *script* presente em cada jogo implementado. Ele recebe comunicações das casas e peças e informa ao controlador geral do jogo sobre o estado da partida.

Por exemplo, quando o jogador toca na tela, querendo que uma peça se movimente, ele está interagindo com um objeto *Peça* pertencente ao *Gerenciador de Tabuleiro*. Esse objeto tem, além do componente que exibe a peça na tela, um componente que detecta a interação do jogador e um *script* que processa a interação e a informa ao *Gerenciador de Tabuleiro*, para que ele decida o que fazer. A figura abaixo representa essa comunicação:

Figura 4.1 – Comunicação entre a peça e o tabuleiro



Fonte: o autor

4.2 Programação genérica

Um dos principais desafios relacionados à programação foi atender ao requisito de implementar os jogos de tabuleiro de uma maneira genérica. Como o objetivo é ter

um processo simples de inclusão de jogos, não se poderia gastar tempo pensando em como implementar cada jogo individualmente.

Normalmente, quando se planeja a programação de um jogo, já se têm definidas as suas regras. Na implementação deste, porém, foi necessário pensar numa maneira de programar o funcionamento dos jogos que viesse a suportar diferentes regras, incluindo a possibilidade de surgirem novas no futuro, com a inclusão de novos jogos.

A solução foi criar *scripts* que implementam comportamentos comuns a vários jogos, evitando programar jogos específicos. Se um *script* estivesse atendendo ao comportamento de um único jogo, ele já não poderia ser usado em outro, e isso é o que se queria evitar. O objetivo foi ter a programação feita de tal forma que cada novo jogo incluído gerasse o menor trabalho possível em programação específica para que ele funcionasse. Qualquer pessoa que quisesse incluir um jogo no futuro deveria poder fazê-lo, bastando seguir o processo utilizado.

Para atingir esse objetivo, foi feita uma análise dos jogos atualmente utilizados pelo LoBoGames. Foram analisadas suas regras e mecânicas, procurando por padrões que pudessem se tornar códigos reutilizáveis em mais de um jogo.

Foi possível dividir o funcionamento dos jogos em diferentes partes, tendo um componente diferente responsável por cada uma. Se dois jogos diferentes tivessem o mesmo funcionamento em uma dessas partes, o componente responsável por elas poderia ser o mesmo, atingido o objetivo de reutilizar códigos em mais de um jogo - ou, sendo mais específico, reutilizar componentes. A utilização da Unity acabou sendo um grande benefício nessa abordagem. Como ela funciona internamente no padrão de orientação a componentes, combinou perfeitamente com esse objetivo.

As partes em que os jogos foram divididos são justamente os padrões de regras e mecânicas que foram procurados nos jogos. Os padrões encontrados foram: movimento das peças, condição de vitória e inteligência artificial. Todos os jogos de tabuleiro analisados têm essas três partes bem definidas: uma ou mais regras para movimentar as peças, uma condição para se vencer uma partida e, pela possibilidade de jogar contra o programa, um código com alguma técnica de inteligência artificial para gerar as jogadas do jogador artificial.

Para cada uma delas foi criada uma classe base. Para cada novo jogo de tabuleiro incluído, um *script* que deriva dessa classe base e implementa essa parte do

jogo deve ser incluído no seu *Gerenciador de Tabuleiro*. Jogos que apresentam o mesmo comportamento em uma dessas partes, utilizam o mesmo código.

4.2.1 Movimento das peças

O primeiro padrão encontrado é que em todos os jogos os jogadores interagem de alguma maneira com as peças disponíveis. Cada jogo tem seu tabuleiro e uma quantidade de peças, e todos eles definem uma ou mais regras sobre como se pode movimentar as peças no tabuleiro. Assim, para que um jogo funcione corretamente, seu *Gerenciador de Tabuleiro* precisa ter um componente responsável pela lógica de movimentação das peças.

Para isso, foi criado um *script* base do qual todas as regras de movimentação de peças que venham a ser criadas devem herdar - é a aplicação do conceito de herança de orientação a objetos. No *Gerenciador de Tabuleiro* existe uma variável do tipo dessa classe base que espera receber o componente que gerencia a movimentação das peças. Essa mesma lógica é utilizada nos outros dois padrões.

A partir da análise dos jogos do *Módulo 1*, foram detectados três padrões de movimentação de peças. Apesar de serem doze jogos, todos eles são jogados com apenas três tipos de interação com as peças. Para cada um deles, foi criado um *script* que define sua lógica, sendo possível utilizar apenas três componentes para definir a movimentação das peças de doze jogos.

Esse é a lógica por trás da programação ser genérica: o foco está em programar as regras, não os jogos. Aproveitando a repetição das mecânicas, não foram programadas doze regras de movimentação, apenas três. Com esses três componentes, associando cada um ao jogo que o utiliza, foram definidos doze jogos.

As três regras de movimentação de peças implementadas foram:

- a. movimentação sobre as linhas do tabuleiro. Essa regra se aplica aos jogos em que o jogador só pode mover sua peça da casa que ela está ocupando para uma casa vizinha que está vazia. A casa vizinha é qualquer uma que esteja “ligada” a ela no tabuleiro, por isso o nome de movimentação “sobre as linhas do tabuleiro”. Os jogos Pong Hau K’i e Shisima são exemplos desse tipo de movimentação.
- b. movimentação livre. Assim como a regra anterior, a peça só pode ser movimentada da casa que está ocupando para uma casa vazia. Porém, nessa

regra, a casa destino não precisa ser vizinha da casa de origem. É possível “pular” pelo tabuleiro, indo para qualquer casa vazia. No jogo Tsoro Yematatu é possível mover peças dessa maneira na segunda fase do jogo.

- c. movimento de posicionamento. Essa é a regra utilizada quando um jogador posiciona uma peça sua, que está fora do tabuleiro, em alguma casa livre dentro do tabuleiro. No jogo Tic-Tac-Toe, essa é a única movimentação utilizada. Todas as peças iniciam o jogo fora do tabuleiro e, na sua vez de jogar, cada jogador posiciona uma peça no tabuleiro.

É importante notar a natureza modular dessa solução. É possível, por exemplo, criar um novo jogo apenas trocando a regra de movimento de um jogo já existente. No jogo Shisima, se trocarmos a regra de movimentação *sobre as linhas* pela de movimentação *livre*, teremos um novo jogo. Essa é uma possibilidade não explorada neste trabalho, mas o exemplo serve para demonstrar a capacidade da solução.

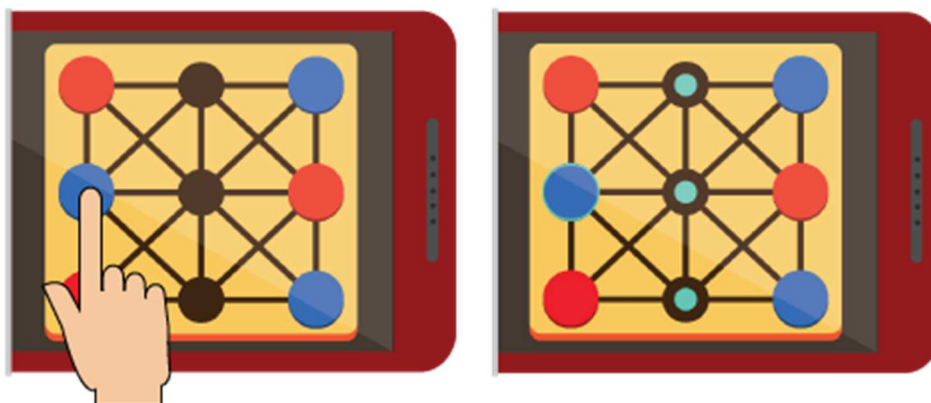
Um exemplo que aconteceu neste trabalho, foi que após ter sido codificada a regra de movimentação das peças para o jogo Pong Hau K'i, o primeiro a ser incluído, todos os jogos incluídos depois, e que tinham a mesma mecânica de movimentação, aproveitaram o mesmo componente. Não foi preciso programar um novo *script*.

Durante uma partida de qualquer um dos jogos, com a regra de movimentação devidamente incluída no *Gerenciador de Tabuleiro*, a comunicação entre os componentes para executar uma jogada segue o seguinte fluxo:

1. o *Gerenciador de Tabuleiro* avisa qual jogador está na vez de jogar e qual tipo de jogada ele precisa fazer: *mover* uma peça no tabuleiro ou *posicionar* uma peça de fora do tabuleiro. Essa informação é dada na tela de jogo através de uma mensagem de texto;
2. se o tipo de jogada é *mover*, o jogador toca na peça que deseja mover; se é *posicionar*, o jogador toca na casa vazia na qual deseja posicionar a peça;
3. a peça ou a casa selecionada vai comunicar o *Gerenciador de Tabuleiro* que ela foi tocada. O *Gerenciador de Tabuleiro* é o responsável por movimentar e posicionar as peças no tabuleiro;
4. se a jogada é *mover* e só existe uma casa vazia vizinha àquela em que a peça está, a peça tocada é movimentada até ela. Se houver mais de uma casa vizinha vazia, o jogador ainda precisa definir qual a casa destino do movimento. Se a jogada é *posicionar*, uma peça de fora do tabuleiro é posicionada na casa vazia que o jogador tocou.

As imagens abaixo demonstram uma jogada do tipo *mover*. Na primeira imagem o jogador toca na peça que quer movimentar. Como há mais de uma possibilidade de movimento, as casas vazias, para as quais se pode fazer a jogada, são destacadas.

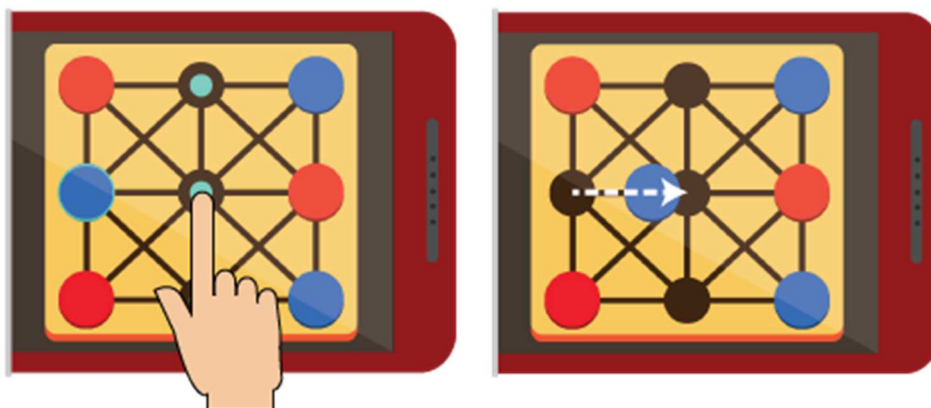
Figura 4.2 – Seleção da peça na jogada *mover*



Fonte: o autor

A seguir, o jogador toca na casa para a qual deseja movimentar a peça. A peça, então, é movimentada para lá.

Figura 4.3 – Seleção da casa na jogada *mover*



Fonte: o autor

4.2.2 Condição de vitória

Outro padrão encontrado é a condição de vitória. Nos jogos do *Módulo 1*, que são de bloqueio e alinhamento, as condições de vitória são exatamente as do título do módulo:

- a. vitória por bloqueio. Quando algum jogador não tem nenhum movimento possível para realizar com as suas peças, ele perde. É a condição de vitória dos três primeiros jogos do módulo.
- b. vitória por alinhamento. Quando um jogador consegue alinhar um certo número de peças, definido nas regras, no tabuleiro, na sua vez de jogar, ele vence a partida.

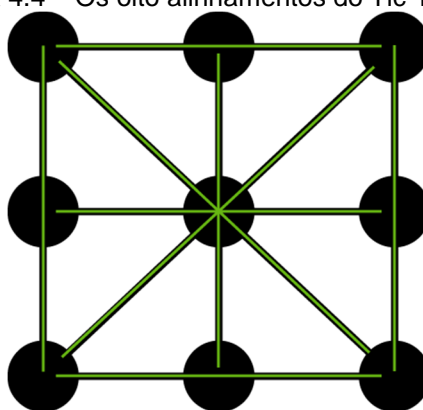
A comunicação do *Gerenciador de Tabuleiro* com o componente que verifica a condição de vitória ocorre ao final de cada jogada, a fim de verificar se alguém venceu a partida antes de passar a vez para o outro jogador. Portanto, o componente dessa parte entra em ação logo após o componente de movimentação das peças.

Para realizar a detecção das condições de vitória, são utilizadas as estruturas de dados criadas para as peças e casas do tabuleiro. Para detectar a vitória por bloqueio, as peças do jogador que fará a próxima jogada são verificadas. Se uma delas está livre para se movimentar, o jogador ainda não foi derrotado, ele tem uma possibilidade de jogada e o jogo continua.

A detecção por alinhamento tem uma estrutura própria que armazena os diferentes alinhamentos presente no tabuleiro. Quando um jogo a ser incluído utiliza essa condição de vitória, o desenvolvedor precisa configurar a regra na *Unity*. É preciso informar quantas peças precisam ser alinhadas para se obter a vitória e preencher a estrutura que armazena os alinhamentos. A ideia básica dessa estrutura é armazenar quais as combinações de casas que formam os alinhamentos do tabuleiro. Ao final de cada jogada, esses alinhamentos são verificados para conferir se em algum deles ocorreu um alinhamento do tamanho necessário para um jogador vencer a partida.

Para o jogo Tic-Tac-Toe, por exemplo, a configuração é feita indicando os oito alinhamentos do tabuleiro e informando que são necessárias três peças de um mesmo jogador em qualquer um dos alinhamentos para vencer. A imagem abaixo destaca, em verde, os oito alinhamentos:

Figura 4.4 – Os oito alinhamentos do Tic-Tac-Toe



Fonte: o autor

4.2.3 Inteligência artificial

Todos jogos também possuem uma inteligência artificial. Não é o foco desse trabalho elaborará-la - e por isso as jogadas do computador são bem fracas - mas também existe um espaço reservado no *Gerenciador de Tabuleiro* para este componente que gerencia a jogada do computador.

Uma diferença importante em relação aos padrões anteriores, é que para este foi criado um *script* para cada jogo. Embora seja possível criar um código genérico e utilizá-lo em jogos com a mesma maneira de jogar, é mais interessante que cada jogo tenha o seu próprio código para a inteligência artificial, para que seja possível especializá-la naquele jogo.

Na versão do *Classics Grandmaster* para este trabalho, alguns jogos que tem a mesma lógica de jogar utilizam o mesmo *script* para o gerenciamento da IA. Por exemplo, todos os jogos da parte “Movimento e Alinhamento” - Picaria, Tapatan, Three Men’s Morris, Altan Xaraacaj e Shisima - utilizam o mesmo código. Como são jogos com a mesma condição de vitória, mudando apenas o tabuleiro, isso foi possível. Porém, para que o computador ofereça um desafio maior ao jogador, é importante ter a opção de criar um código específico para cada jogo, que vai levar em conta as melhores posições de cada tabuleiro, mesmo que a maneira de jogar seja a mesma.

4.3 Apresentação visual

Em qualquer jogo digital existe uma grande necessidade de planejar bem a apresentação visual. Com a grande oferta de jogos que temos hoje, um jogo mal

apresentado facilmente é ignorado. Em grandes empresas existem pessoas que trabalham especificamente com cada parte envolvida nesse processo.

Jogos são feitos para o jogador jogar, ele precisa se sentir bem e entender o que está acontecendo. É indesejado que ele tenha dificuldade de realizar alguma ação, que tenha resultados inesperados ou que seja prejudicado em sua experiência. Se isso acontecer, ele vai acabar procurando outro jogo.

Num curso de Ciência da Computação não são adquiridos os conhecimentos necessários para resolver todos desafios da apresentação visual de um jogo. Por isso, para a versão final do *Classisc Grandmaster*, uma pessoa com o conhecimento adequado será responsável por isso. As imagens apresentadas a seguir representam um primeiro trabalho com a apresentação visual do jogo, embora ainda não seja a versão final.

Três partes foram alvo de uma maior preocupação: a experiência do usuário, a arquitetura da interface e a arte, que é a produção artística dos elementos visuais. As duas primeiras cuidam da experiência do jogador enquanto joga e a última se preocupa com como o jogo será visto, de qual público ele deve chamar atenção e como representar bem o seu estilo através da arte.

4.3.1 Arquitetura da interface

A arquitetura da interface é a tarefa de estudar e definir o posicionamento dos diferentes elementos (imagens, botões, textos, etc.) em cada tela de jogo. As telas de jogo que serão apresentadas, que são as que o jogo tem até agora, são suficientes para demonstrar o fluxo do jogo. Não significa, porém, que são todas que o jogo terá até a sua conclusão.

A tela do menu é a tela inicial do jogo. Ela é bem simples, semelhante a qualquer jogo digital: tem um espaço reservado para o logotipo do jogo e um menu com as opções.

Figura 4.5 – Tela inicial



Fonte: o autor

Através da opção “Configurações” na tela de menu, o jogador abre a tela de configurações. Nela estão as opções do jogo que podem ser modificadas. O jogador pode mudar seu perfil, que é o seu nome e a cor das suas peças. Conforme estiver definido nesta tela, será exibido na tela de jogo durante as partidas.

Os elementos para alterar o tema e o volume da música e efeitos sonoros estão presentes apenas para guardar espaço, pois ainda não há sons e diferentes temas no jogo. Com diferentes temas é possível criar temas específicos para jogadores de uma certa idade ou com algum tipo de necessidade visual especial. Por exemplo, crianças poderiam jogar o jogo com uma arte pensada para elas.

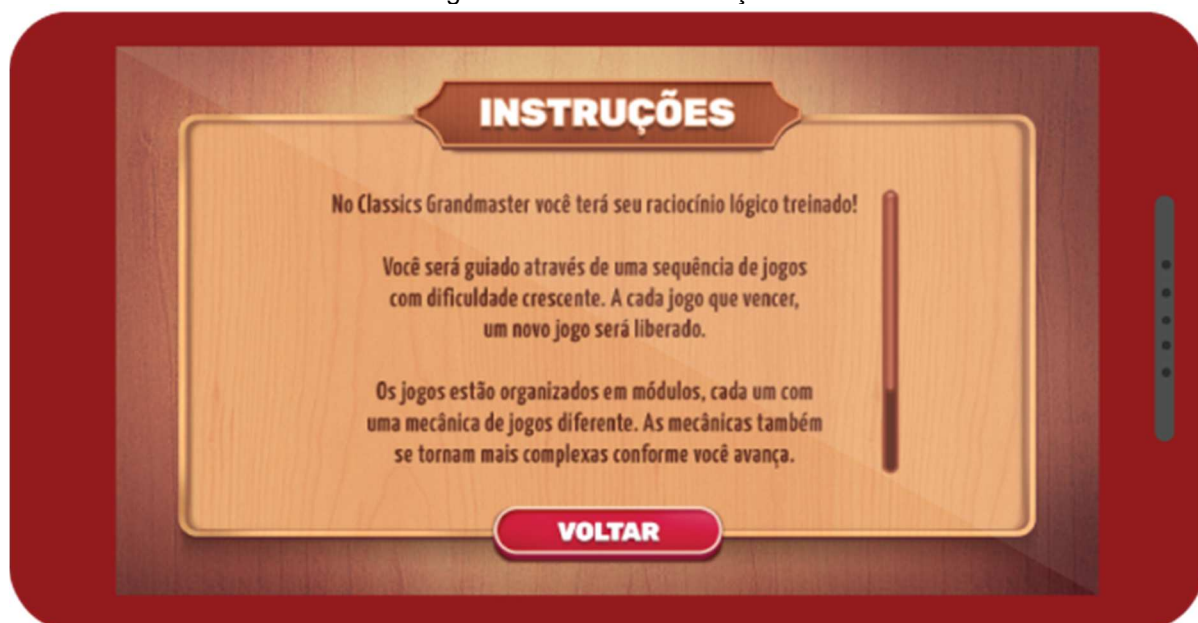
Figura 4.6 – Tela de configurações



Fonte: o autor

A opção “Instruções” da tela inicial leva o jogador à tela de instruções, a qual exibe as instruções básicas de funcionamento do jogo.

Figura 4.7 – Tela de instruções



Fonte: o autor

Quando o jogador entra pela primeira vez no *Classics Grandmaster*, ele ainda não tem um perfil salvo internamente. Por isso, quando a opção “Um jogador” for acessada pela primeira vez, o jogador é levado para a tela de registro.

Figura 4.8 – Tela de registro de jogador



Fonte: o autor

Nessa tela o jogador escolhe um nome para si, digitando no campo apropriado, e escolhe a cor das suas peças, tocando na cor que deseja. Conforme ele definir aqui, aparecerá na tela de jogo. Ao tocar no botão “Confirmar”, essas informações são salvas internamente no dispositivo que estiver sendo usado. Nas próximas vezes que o jogador entrar no jogo, não passará por essa tela de novo. Caso ele queira alterar essas informações, pode fazer isso na tela de configurações.

A opção do menu inicial “Dois jogadores”, por outro lado, sempre leva para a tela de registro de jogador. Como o segundo jogador pode ser sempre uma pessoa diferente e o progresso no jogo é referente somente ao jogador principal, não há necessidade de salvar suas informações. A cada vez que se quiser jogar uma partida de dois jogadores, o segundo define suas informações

Após confirmadas as informações na tela de registro, a próxima tela sempre é a de escolha do módulo. Nesta tela estão visíveis todos os módulos que compõe o jogo. Eles tem um logotipo, que está no meio da tela, um nome, que está acima do logotipo, e a porcentagem que indica o quanto aquele módulo está completo, abaixo do logotipo.

Figura 4.9 – Tela de escolha do módulo



Fonte: o autor

Conforme já comentado, o único módulo disponível é o *Módulo 1*. Os demais módulos são apenas representativos do que se deseja para o futuro. Quando o jogador toca no módulo que deseja acessar, ele acessa a tela de escolha de jogo.

Figura 4.10 – Tela de escolha do jogo

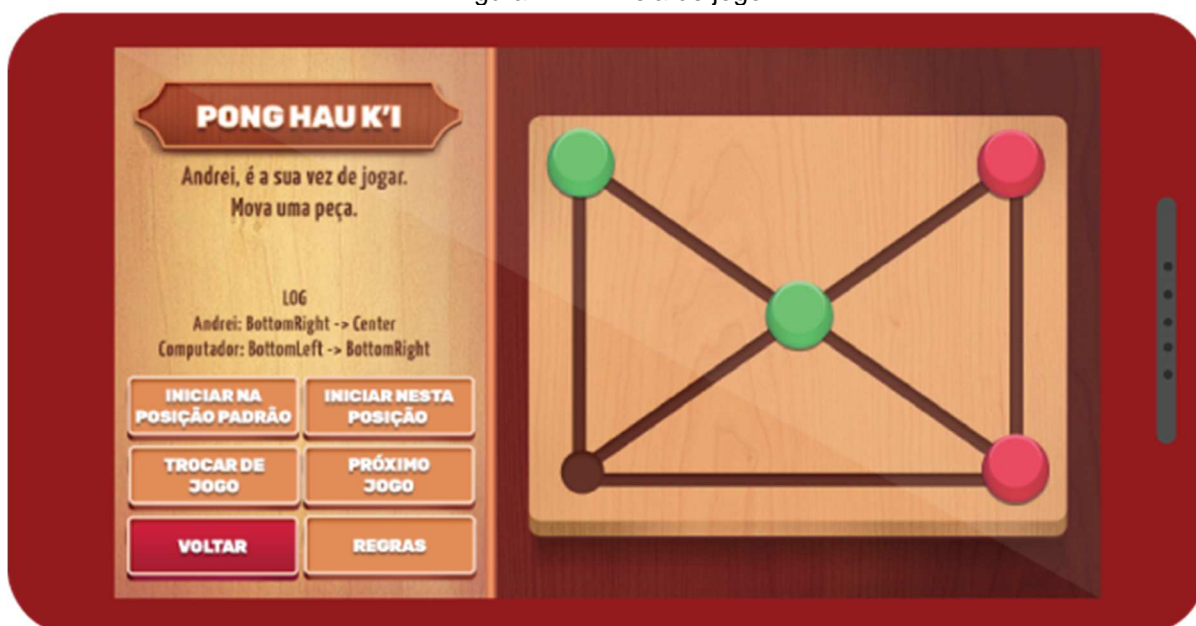


Fonte: o autor

No lado esquerdo da tela estão todos os jogos do módulo, divididos conforme as partes que o compõe. No lado direito está uma visualização maior do tabuleiro do jogo selecionado. Acima dele está o logotipo do módulo e abaixo o nome do jogo e

um botão para confirmar a escolha. Ao tocar nesse botão o jogo escolhido é carregado e aparece já na tela de jogo.

Figura 4.11 – Tela de jogo



Fonte: o autor

Na tela de jogo é que acontecem as partidas. Na direita está o tabuleiro do jogo escolhido. Ali acontece toda interação com as peças. Na esquerda estão as informações sobre a partida. No alto está o nome do jogo. Abaixo dele as mensagens sobre o andamento do jogo, que dão instruções como quem está na vez de jogar ou quem ganhou a partida. Abaixo dessa mensagem está uma caixa com o registro das jogadas realizadas na partida. Por último estão os botões de gerenciamento.

O botão “Regras” faz surgir uma aba com as regras do jogo que está sendo jogado. O botão “Configurar uma posição inicial” possibilita ao jogador montar uma configuração de peças para iniciar uma partida a partir dela. O botão “Iniciar na posição padrão” monta a configuração padrão daquele jogo e inicia a partida. “Voltar ao menu” e “Trocar de jogo” fazem aquilo que o nome diz. “Próximo jogo” passa para o próximo jogo do módulo, seguindo a ordem do módulo. Esse botão só fica disponível se o próximo jogo está liberado.

4.3.2 Experiência do usuário

Antes mesmo de iniciar a implementação deste trabalho, diversos objetivos foram listados pensando em proporcionar ao jogador uma boa experiência. Como os

jogos lógicos podem ser jogados por crianças até idosos, é preciso que numa versão digital todos entendam o que está acontecendo e consigam interagir, mesmo sem ter contato anterior com tablets ou smartphones. Também é importante lembrar que um jogador pode estar experimentando os jogos pela primeira vez e ainda sozinho. Nesse sentido o jogo precisa ajudá-lo a saber o que fazer em cada momento. Abaixo estão listados esses objetivos junto com suas soluções:

- a. “o jogador precisa saber o que fazer e quando fazer”. Para isso, foi reservado, conforme mostrado na tela de jogo na seção anterior, um espaço na interface da partida que exibe uma mensagem que indica ao jogador a sua vez de jogar, utilizando o nome que ele definiu para si. Essa mensagem é específica para cada tipo de jogada que precisa ser feita. Quando o jogador precisa movimentar uma peça, a mensagem é diferente daquela quando ele precisa posicionar uma peça que está fora do tabuleiro para dentro dele.
- b. “as possibilidades de jogada precisam ser claras”. Cada jogador precisa saber o que ele pode fazer com cada peça. Para atender isso, quando um jogador escolhe uma peça que tem mais de uma possibilidades de movimento, a peça selecionada e as possíveis casas para as quais ele pode movê-la são destacadas no tabuleiro. Quando a peça só tem um movimento possível, esse movimento é realizado sem a necessidade de selecionar a casa.
- c. “o jogador precisa entender por que o jogo acabou”. Quando um jogador vence o jogo, as peças que compõe a jogada vitoriosa são destacadas. Num jogo de alinhamento, por exemplo, as peças alinhadas que deram a vitória recebem destaque. Também há uma mensagem no final do jogo indicando qual jogador venceu ou se houve empate.
- d. “o jogador precisa conseguir acompanhar o que está acontecendo em cada momento do jogo”. Para isso, quando uma peça vai de uma casa para outra, ela se movimenta da casa de origem até a casa destino, não simplesmente some de uma casa e aparece na outra. Todas jogadas, sejam elas *mover* ou *posicionar*, também são registradas num local específico da tela de jogo, um registrador de jogadas, para que o jogador possa rever o histórico da partida. Esse histórico indica qual movimento foi feito, diferenciando cada um com um símbolo e a(s) casa(s) envolvida(s). Para uma jogada *mover*, a mensagem segue o padrão “*casa de origem* → *casa destino*”, sendo que a seta para a direita representa esse tipo de jogada. Para uma jogada *posicionar*, o padrão é

“↓ *nome da casa*”, sendo que a seta para baixo representa esse tipo de jogada. Esses símbolos foram escolhidos como uma maneira de representar visualmente o tipo de jogada. A seta para a direita dá o entendimento de ir de um lugar para outro. A seta para baixo dá o entendimento de estar colocando algo num lugar.

- e. “as peças e casas do tabuleiro precisam ser facilmente identificadas e selecionadas.” Todos os jogos do *Classics Grandmaster* são jogados com peças do mesmo formato. Não há, como no Xadrez, peças com formatos diferentes. Aproveitando isso, as peças foram desenhadas em formatos bem simples, com tamanho grande para facilitar sua seleção e cores bem diferentes para diferenciar os jogadores.

Alguns detalhes, que não estavam nessa lista de objetivos, mas enriqueceram a experiência do usuário, foram incluídos inspirados em mecânicas comuns em jogos digitais. Um deles é o indicador de progresso do módulo, citado na tela de escolha do módulo. Há uma porcentagem abaixo do logotipo de cada módulo indicando o quão completo ele está, isto é, a porcentagem de jogos dele que já foram liberados. Outro é a criação de logotipos para cada módulo, Isso cria uma identificação para cada módulo, reforçando a ideia de cada módulo ser um “mundo” do jogo.

Na tela de escolha de jogos, os jogos que o jogador já venceu recebem um destaque diferente, indicando que estão disponíveis para serem escolhidos. Isso dá ao jogador a sensação de estar passando de fase cada vez que vence uma partida contra o programa. É uma construção comum em jogos digitais que serve muito bem para o propósito de aprendizado, pois o aprendiz entende que está vencendo etapas do seu aprendizado.

Outros detalhes surgiram para auxiliar na experiência de quem está aprendendo com o jogo. Na interface da tela de jogo, foram incluídos dois botões que não precisariam existir se só existisse um único jogo. O botão “Regras” não seria necessário, ele poderia estar nos menus pré-jogo ou até não estar presente, caso fosse algum jogo lógico bem conhecido. Da mesma forma o botão “Próximo jogo”. Ele demonstra para o jogador a ideia da sequencialidade, de que existe uma ordem. Por causa disso, esse botão não é habilitado até que o jogador vença uma partida contra o programa naquele jogo, demonstrando que entendeu como o jogo funciona.

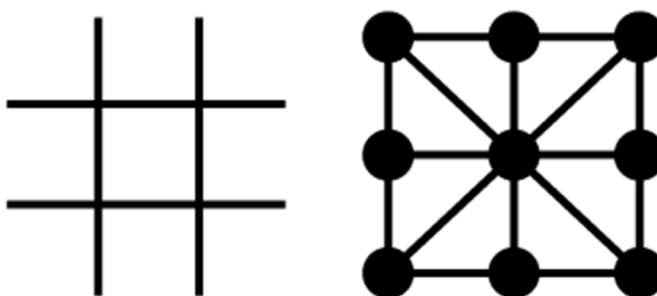
4.3.3 Arte

Algumas decisões sobre como representar visualmente os jogos foram tomadas e serão descritas aqui. Porém, maiores detalhes sobre a produção artística dos elementos do jogo fogem totalmente do objetivo deste trabalho. Uma pessoa com habilidades nessa área fará toda arte do jogo no futuro.

Quanto à representação visual do tabuleiro, a decisão foi por representar conforme eles aparecem no material do Programa LoBoGames. As casas do tabuleiro são pequenos círculos e, para indicar os alinhamentos ou vizinhanças das casas, são desenhadas linhas ligando esses círculos. Com esse tipo de desenho, é possível representar todos os jogos com um mesmo estilo. Assim os jogadores associam mais facilmente as possibilidades de movimento, alinhamento e ocupação, se preocupando mais com as jogadas do que com o entendimento do tabuleiro.

Para alguns jogos, o desenho fica igual à versão física, como o Pong Hau K'i. Para outros, fica um pouco diferente. O Tic-Tac-Toe, por exemplo, que é bem conhecido por seu tabuleiro com a forma do símbolo de número (#). A versão tradicional é mais limpa, como se pode ver na imagem abaixo à esquerda. Porém com a representação utilizada, a da direita, os alinhamentos possível ficam mais visíveis.

Figura 4.12 – Representações do tabuleiro Tic-Tac-Toe



Fonte: o autor

Nos jogos do LoBoGames não há nenhum em que as peças tem formatos especiais, como no Xadrez. Essa decisão poupa trabalho artístico e facilita a identificação de quem joga. Ainda torna possível a ideia já comentada de criar diferentes temas visuais para as peças. Já que em todos jogos as peças utilizam o mesmo elemento gráfico, é possível criar variações e deixar o jogador escolher uma delas, associando ao seu perfil. Repetindo o exemplo, pode existir uma opção na qual as peças tem uma aparência infantil, mais atrativa para crianças.

4.3 Inclusão de um novo jogo

Como já comentado, um dos objetivos desse trabalho é criar um processo simples para a inclusão de novos jogos. Isso é atendido pela estrutura criada com o *Gerenciador de Tabuleiro*. Alguns passos do processo completo, porém, são um pouco minucioso pela configuração da estrutura de gerenciamento da sequência de jogos e das telas do jogo. Caso se queira incluir um novo jogo em algum módulo, é preciso seguir os passos a seguir, descritos em termos gerais.

O primeiro passo é criar o desenho do tabuleiro. Alguns jogos compartilham do mesmo desenho do tabuleiro, Tapatan e Tic-Tac-Toe, por exemplo. Dentro da Unity, esses dois jogos são construídos utilizando a mesma imagem, não há nenhum problema nisso. Se o tabuleiro do novo jogo não estiver representado pelos desenhos já existentes, um novo desenho é necessário.

Para criar esse desenho, o artista utiliza o mesmo processo usado na criação dos anteriores: seguindo o estilo artístico do jogo, cria o desenho no seu programa de edição de imagens preferido. A única restrição é a necessidade de respeitar o tamanho limite em *pixels* para que o tabuleiro caiba dentro da tela de jogo.

O segundo passo é programar os novos componentes. Caso o novo jogo tenha regras diferentes das já definidas pelos componentes atuais, seja para movimentação ou para a condição de vitória, é necessário programar essa nova regra. O componente da IA, como já comentado, precisa ser criado sempre.

O terceiro passo é estruturar o tabuleiro na Unity. Como comentado na seção sobre a estruturação dos jogos, começa-se criando um *GameObject* vazio e dando ao ele o nome do jogo. Adiciona-se a ele os componentes comuns a todos os jogos: o *Image* e o *Gerenciador de Tabuleiro*. Depois, se adiciona, como objetos-filho, um *GameObject* com todas as casas e outro com todas as peças do jogo. As peças e casas também com seus componentes básicos: um *Image* e o *script* que gerencia peças ou casas.

Feito isso, esses componentes precisam ser configurados conforme o novo jogo. O desenho do tabuleiro é inserido no componente *Image*, e os componentes que implementam as três partes básicas do jogo (movimentação das peças, condição de vitória e IA) nos seus locais reservados no *Gerenciador de Tabuleiro*. As peças são configuradas indicando a qual jogador pertencem e qual sua posição inicial. As casas são configuradas indicando suas vizinhas.

O quarto passo é salvar a estrutura criada e adicionar um campo para o novo jogo no componente que guarda a sequência de jogos. Para salvar a estrutura, cria-se um *Prefab* desse jogo na pasta do módulo ao qual ele pertence. Um *Prefab* é uma estrutura interna da Unity. Na prática, é um objeto fechado com toda a estrutura e configuração criada. Assim, o tabuleiro e toda sua estrutura não fica carregado ocupando memória, só é carregado na hora que o jogador decide jogar aquele jogo. É partir do nome do *Prefab* e da pasta em que ele é salvo que o *Gerenciador de Jogo* pode encontrá-lo para carregá-lo. Essa informação precisa ser adicionada no componente que guarda a sequência de jogos, para que seja carregado na sequência correta.

Por último, é preciso criar o ícone para o jogo dentro da tela de escolha do jogo. Estando lá é que ele fica selecionável para o jogador.

5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho descreveu o processo de estudo e implementação do *Classics Grandmaster*, um jogo digital sobre jogos lógicos de tabuleiro. Foram apresentadas as decisões de projeto, escolhas tecnológicas e detalhes da implementação. O resultado final desse processo é um jogo para smartphones e tablets com Android que atende aos objetivos deste trabalho.

O jogo desenvolvido apresenta os jogos de forma simples e compreensiva, um fluxo de jogo conforme o método do LoBoGames, serve como ferramenta de ensino e está construído de forma que a inclusão de novos jogos seja um processo simples.

Ao término deste trabalho, o jogo encontra-se conforme o que foi descrito. Alguns trabalhos futuros, porém, já são previstos para que ele se torne mais completo antes de ser inserido na Play Store, a loja de aplicativos para Android, na qual buscará alcançar o objetivo maior de ter um espaço no mercado de jogos de tabuleiro digitais. Os planos são:

- a. ampliar a quantidade de jogos. Seguindo as regras que foram detalhadas, a inclusão de novos segue um fluxo simples, portanto é um objetivo imediato. Aproveitando a inspiração do Programa LoBoGames, a ideia é incluir pelo menos tantos jogos e módulos quanto existem no Programa.
- b. completar o jogo como produto. Para alcançar o padrão dos jogos presentes nas lojas de aplicativos, a atual versão do *Classics Grandmaster* precisa de incrementos nas áreas não abordadas neste trabalho. A parte visual, como já comentado, precisa de um trabalho profissional. Também a parte de sons e música. Depois encontrar um modelo de negócios adequado.
- c. utilizar uma técnica de IA. O desenvolvimento da IA não foi o foco deste trabalho, mas é sabido que para jogadores mais experientes o desafio de uma boa IA é essencial. Essa implementação pode vir junto com um sistema de níveis de dificuldade, no qual o nível mais fácil tem uma técnica de IA mais fraca e no mais forte uma mais elaborada.
- d. alcançar outras plataformas. A portabilidade para navegadores é praticamente direta da versão final deste trabalho, restando a necessidade de encontrar um bom modelo de negócio para essa plataforma. Versões para dispositivos móveis com outros sistemas operacionais, como iOS e Windows Phone,

também exigem uma pequena quantidade de trabalho a partir do que já está feito.

- e. estabelecer uma parceria com o Programa LoBoGames. Como o jogo foi criado inspirado no Programa, a ideia é que o público dele o teste para que seja refinado até o seu lançamento oficial.

REFERÊNCIAS

MENDES, THIAGO GODOLPHIM. **GAMES E EDUCAÇÃO: Diretrizes de Projeto para Jogos Digitais Voltados à Aprendizagem**. 2012. 134 f. Dissertação (Mestrado em Design) – Faculdade de Arquitetura, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

RAUTER, ARTHUR CARVALHO. **Plataforma Computacional de Apoio ao Projeto de Extensão jogos Lógicos de Tabuleiro**. 2015. 60 f. Monografia (Bacharel em Engenharia da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014.

HUIZINGA, JOHAN. **Homo ludens: a study of the play-element in culture**. Boston: Beacon Press. 1955.

FRIEDRICH G.; PREISS, G. Educar com a cabeça. **Viver Mente e Cérebro**, edição especial, número 8, fev. 2006.

BRIGHT, GEORGE W.; HARVEY, JOHN G.; WHEELER, MARGARIETE M. Use of a Game to Instruct on Logical Reasoning. **School Science and Mathematics**, volume 83, edição 5, p. 396-405, jun. 1983.

GIORDANI, LILIANE FERRARI; RIBAS, RENATO PEREZ. Movimentos no Currículo, Formação de professores e outras aprendizagens. Disponível em: <<http://www.inf.ufrgs.br/lobogames/#pesquisa>> Acesso em: 17 dez. 2015.

GRIDER, MARTIN. When Board Games Go Mobile. **IGDA Perspectives Newsletter**. Disponível em: <<http://newsletter.igda.org/2013/11/30/when-board-games-go-mobile>> Acesso em: 30 nov. 2015.

LOBOGAMES. **Lobogames, Jogos Lógicos de Tabuleiro**. Disponível em: <<http://www.inf.ufrgs.br/lobogames/>>. Acesso em: 30 nov. 2015.

UNITY MANUAL. **Unity Technologies**. Disponível em: <<http://www.unity3d.com>>. Acesso em 16 dez 2015.