Relatório Técnico do Click Game

Cauã R. Brasil[1], Giliardo J. de Medeiros[2]

[1]Instituto Metróple Digital – Universidade Federal do Rio Grande do Norte (UFRN)
Caixa Postal 1524 – 59078-900 – Natal – RN – Brazil

caua.brasil.017@ufrn.edu.br, giliardo.junior.702@ufrn.edu.br

Abstract. This meta-article describes the final project of the Programming Language II discipline, chosen by the group formed by Cauã Brasil and Giliardo Junior, the Click Game. The Click Game was chosen based on the "osu!" developed by Dean Herbert..

Resumo. Este meta-artigo descreve o projeto final da disciplina de Linguagem de Programação II, escolhido pelo grupo formado por Cauã Brasil e Giliardo Junior, o Click Game. O Click Game foi escolhido com base no jogo "osu!" desenvolvido por Dean Herbert.

1. Introdução

Para esse projeto, foi escolhido um jogo de click utilizando o "osu!" como referência para a criação do Click Game. O Click Game é uma versão simplificada do original, com o foco apenas na velocidade de reação do jogador, onde não há música e ritmo, esse jogo foi criado para testar o quão rápido é a velocidade do player ao clicar em um botão que aparece em algum lugar aleatório da tela. A fim de mensurar e classificar as reações de cada usuário, é implementado um sistema de score, de maneira a cada botão clicado, aumentar a pontuação.

Com o intuito de diversificar a experiência, existem 3 dificuldades para jogar, fácil, normal e difícil, a cada dificuldade existe um diferente sistema de pontuação e tempo de duração do botão na tela. Por exemplo, a pontuação é dobrada e os botões aparecem e desaparecem rapidamente no modo difícil.

2. Metodologia de Desenvolvimento

2.1. Configurações e Estrutura Escolhidos

O projeto foi desenvolvido em duas IDEAs, eclipse e intellij, utilizando a versão 17 do java e do jfx, auxiliado pela ferramenta maven e estruturado pelo intellij para o uso do maven com o padrão de projeto DAO, acompanhado do diretório resource, que armazena os arquivos fxml.

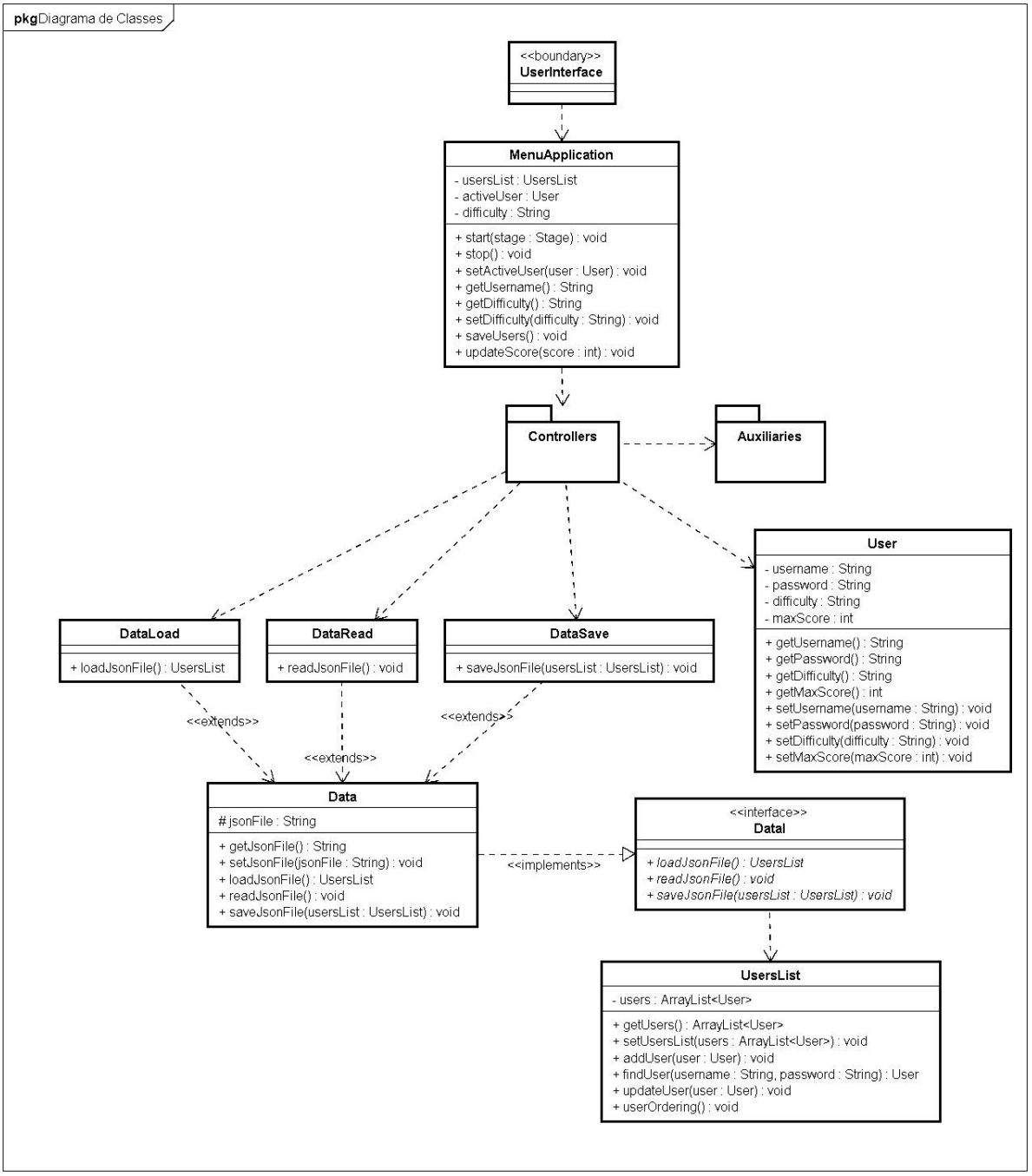
O padrão DAO foi escolhido devido a simples organização do projeto em 4 packages, model, dao, view e controller. facilitando a modularização e encapsulamento das classes e seus atributos. A ferramenta maven assiste o gerenciamento de dependências do código, de maneira a automatizar o processo de adicionar bibliotecas externas, como a jackson, que é uma manipuladora de arquivos json.

O desenvolvimento do projeto é focado em 2 telas principais da aplicação. A tela de Welcome, possuindo a responsabilidade de cadastrar e logar os usuários, seguida pela tela de Main Menu, responsável por disponibilizar as funções mais importantes do jogo.

Para isso, o padrão DAO é necessário para separar os controladores dessas telas, a aplicação do projeto, as classes auxiliadoras e o gerenciador de dados. Ao manter essas duas telas como foco, a produção foi dividida em 4 etapas, a primeira etapa focada em gerenciar os players, a segunda etapa focada em configurar o ambiente do main menu e suas funções, a terceira etapa responsável pelo jogo em si, e a quarta etapa, responsável por personalizar as telas.

2.2. Informações das Classes

A fim de facilitar a compreensão e visualização da estrutura das classes, foram criados 2 diagramas de classes, o diagrama do package controller, contendo as classes usadas como controllers; e o diagrama que contém as outras classes do projeto.

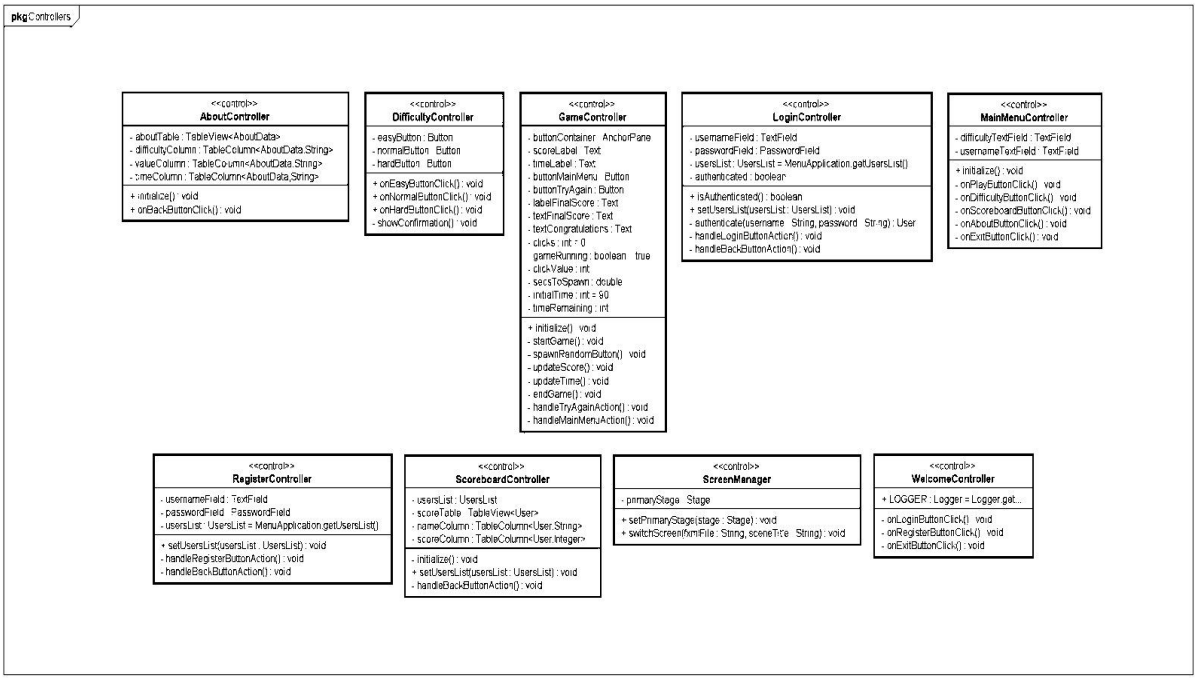


Como disposto na imagem, o programa tem início na classe MenuApplication, essa classe possui o executável do projeto, possuindo 3 atributos privados estáticos, difficulty: responsável por armazenar a dificuldade do jogo selecionada, activeUser: responsável por guardar o usuário atual e o usersList: responsável por estocar os usuários salvos. Além disso, ela possui a função que lança a página inicial da aplicação, e a partir dela as outras telas são chamadas.

Com o início da aplicação, o código segue para as classes do tipo controller, que utilizam das classes que herdam a Data. A classe Data possui a finalidade de lidar com o arquivo json, arquivo esse que provém a lista de usuários e suas informações. A DataRead tem a função de ler o json e verificar se não há anomalias nele, caso o arquivo não exista, ele o cria. A DataLoad é responsável por carregar os usuários salvos no json e os passar para o usersList. A DataSave possui a responsabilidade de salvar as alterações dos usuários no json no fim do programa.

A User tem como atributos as características de um usuário, como nome, senha, dificuldade e pontuação máxima. Seus métodos são seus geters e seters, assim formando uma classe com o único objetivo de armazenar as características de cada usuário.

A UsersList possui as funções de armazenar os usuários que foram salvos no json, encontrar o usuário informado no login e adicionar usuários novos cadastrados no register, além de verificações de senha.



É observável que a maioria das classes que estão no package são responsáveis por fazer o controle de seus respectivos fxml's usados no projeto. Onde a WelcomeController, RegisterController, LoginController estão conectadas a primeira janela, a janela de welcome. Enquanto as outras classes estão ligadas a janela do main menu.

3. Funcionalidades da Aplicação

A tela de welcome possui três botões, o de login, register e exit. O login abre uma tela de login com espaço para digitar o usuário e senha, se ambos estiverem corretos, a tela do main menu é aberta, o botão de register abre uma tela de registro, fazendo verificações padrões de registro de usuário, e o exit que mostra uma alerta de confirmação de saída.

A tela do main menu tem 5 funções, a play que inicia o jogo, a scoreboard que mostra um ranking com os 20 jogadores com maior pontuação, a about que oferece informações sobre o jogo, a difficulty que fornece opções de escolha de dificuldade e por fim a exit, que mostra um alerta de confirmação de saída.

Ao pressionar o play na main menu, o jogo é iniciado com um tempo fixo de 1 minuto de duração da partida, ao fim do tempo é mostrado a pontuação obtida e é ofertado a opção de voltar para o main menu ou tentar novamente.

4. Conclusão

Dessa forma, o projeto possui 19 classes e 8 telas, com padrão de projeto DAO, utilizando de ferramentas como maven e jackson, usando o jfx para a produção da interface. Além disso, os requisitos para executar o projeto são o java e jfx 17 e o Maven.

Para a execução do mesmo, é necessário verificar se o maven está instalado através do comando “mvn -v”, compilar por meio do “mvn compile” e executar com “mvn clean jfx:run”.

5. Referências

https://maven.apache.org/

https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html

https://openjfx.io/