

SnakeGame

1st Gustavo Lopes dos Santos
Sistemas da Informação
Unieuro
Brasília, Brazil
idmhannibalz@gmail.com

2nd Wesley Alcântara Alves
Sistemas da Informação
Unieuro
Brasília, Brazil
wesleyalvesskt15@gmail.com

3rd Rodrigo de Souza Ferreira
Sistemas da Informação
Unieuro
Brasília, Brazil
rodrigoandriago@hotmail.com

4th Aldo Henrique Dias Mendes
Sistemas da Informação
Unieuro
Brasília, Brazil
aldo003679@unieuro.com.br

Resumo — Esta pesquisa, por meio de amor aos jogos e ao aprendizado à programação, propõe a visibilidade de como foi construído um simples jogo em programação orientada a objetos na linguagem Java. Um projeto completo, que nasceu na primeira reunião de todas as partes presentes no projeto, em que foi decidida a ideia e como seria construído o jogo, juntamente com o desenvolvimento do código fonte e finalizando com este artigo.

Palavras-Chave — Jogo, snake game, POO, java

I. INTRODUÇÃO

Snake (também conhecido como "jogo da cobrinha") é um jogo famoso que teve diversas versões, inicialmente, a primeira versão do jogo lançada foi a arcade Blockade em 1976. Já no ano 1997 a empresa Nokia introduziu o jogo nos seus celulares, popularizando o jogo nos anos 2000, desde então diversas variações do jogo foram criadas tanto em videogames, quanto em computadores, celulares e smartphones.

O Objetivo principal do projeto SnakeGame é recriar a aplicação pela qual o usuário possa jogar com novos e diferentes recursos do clássico jogo Snake. O projeto foi escrito na linguagem Orientada a Objetos Java com o auxílio de bibliotecas importadas para utilizar recursos de interface gráfica (AWT e SWING). A fim de entender melhor a metodologia de implementação do código, foi utilizado no projeto SnakeGame, o estudo de (GONÇALVES 2018)[1], que pode proporcionar melhor entendimento sobre os conceitos citados anteriormente. Quanto a interface gráfica que nos permitirá interagir com o programa SnakeGame foi usado como referência o estudo de (CARMO 2017)[2].

II. DIAGRAMA DE CLASSES

A primeira implementação do projeto, deu início com o Diagrama de Classes, o mesmo foi utilizado para definir o corpo do projeto. E com o decorrer das reuniões, o Diagrama de Classes lapidou-se e tomou a forma do final do projeto Fig. 1.

A. Classe Cobra.

A Classe Cobra define o tamanho das variáveis cabeça e corpo, o tamanho total da cobra e os encapsulamentos.

B. Classe Fruta.

A Classe Fruta é uma superclasse possuindo um método para definir o tamanho das variáveis, encapsulamentos e possui a classe filha Fruta Amarela.

C. Classe Player.

A Classe Player define o nick e a pontuação, e também altera, remove e visualiza nicks criados.

D. Classe Interface Jogo.

A Classe Interface estende a classe Painel e utiliza as extensões das bibliotecas Java.awt e javax.swing para dar forma à interface do jogo. A mesma contém objetos das classes Cobra, Fruta e Player, além das variáveis que definem a resolução da tela e métodos que definem a parte principal do código. A atribuição de movimentos no teclado, é implementada pelas classes interface's KeyListener e ActionListener.

1) *InterfaceJogo*.: É um construtor da classe InterfaceJogo, que define a resolução do jogo com altura e largura da tela dividida pelo tamanho das unidades, ou seja, os locais da tela que a cobra nasce, por onde ela pode andar e onde as maçãs nascerão é padronizado em quadrados, dividindo a tela em 10 unidades de largura e 10 unidades de altura, totalizando 100 unidades de tela. O método InterfaceJogo será chamado na classe PainelJogo como um novo objeto.

2) *telaInicial*.: A tela do jogo é construída com a altura e largura da tela e dividida pelo tamanho das unidades, ou seja, os locais da tela em que a cobra nasce, por onde ela pode andar e onde as maçãs nascerão é padronizado em quadrados, dividindo a tela em 10 unidades de largura e 10 unidades de altura, totalizando 100 unidades de tela. O método InterfaceJogo será chamado na classe PainelJogo como um novo objeto.

3) *draw*.: Desenha na tela início "Bem-vindo ao jogo da Cobra" e se início = false e rodando = true pinta as frutas e pinta a cabeça e corpo da cobra.

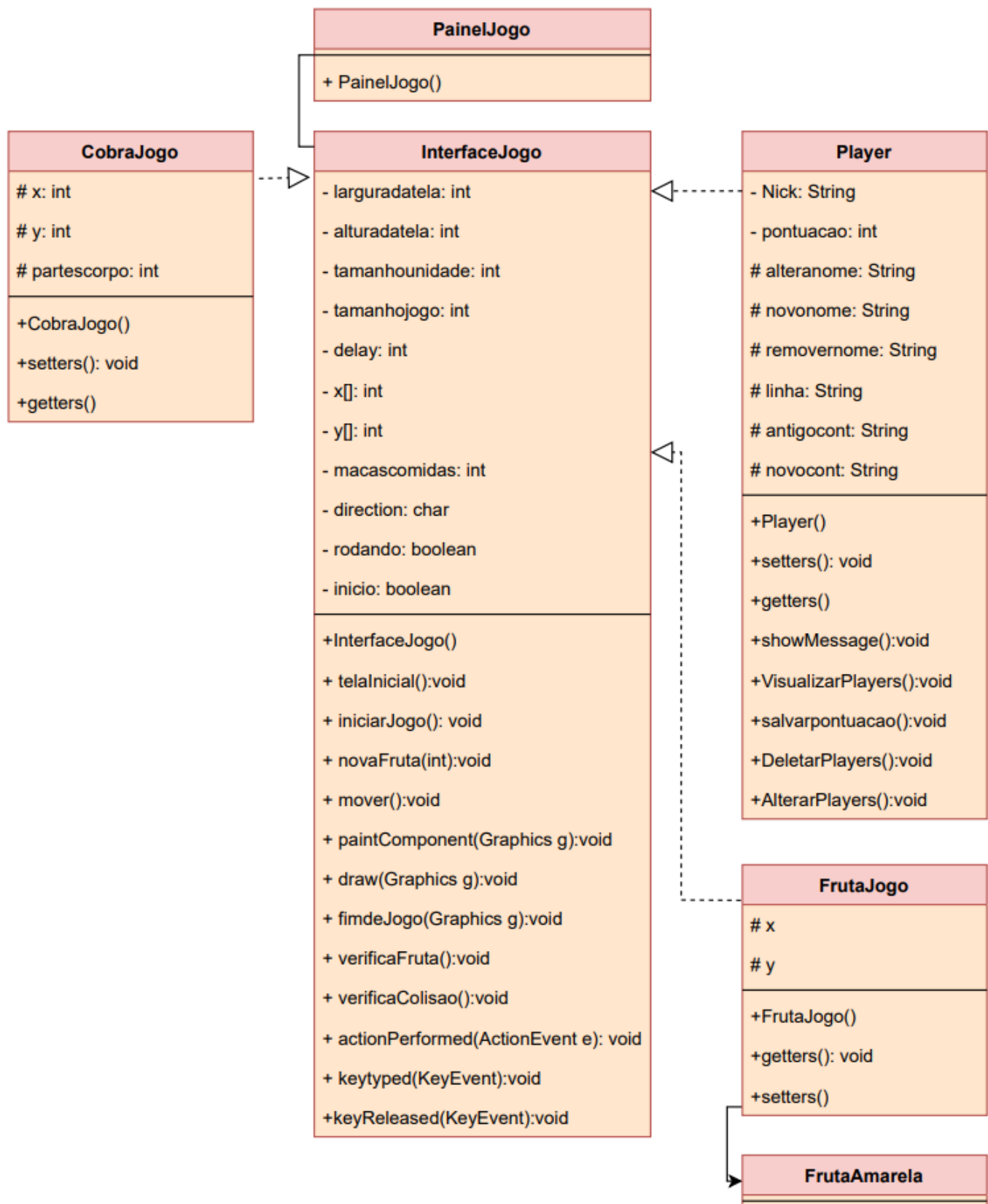


Fig. 1. Diagrama de Classes

4) *paintComponent*.: Usa o método draw para pintar e dar vida ao projeto.

5) *novaFruta*.: É chamada pelo método IniciarJogo e adiciona uma fruta de cada na tela.

6) *verificaFruta*.: Caso a Fruta seja coletada chama o método novaFruta novamente e adiciona mais uma na tela.

7) *actionPerformed*.: Quando um botão for apertado chama os métodos para fazer alterações no código.

8) *keyPressed*.: Caso o usuário tecele durante o jogo rodando com W, A, S, D, ↑, ←, ↓, → *elasmoveparaadireçãodesejada*.

9) *moverCobra*.: Faz com que cada parte do corpo da cobra siga a cabeça, sempre -1 do seu x ou y. Já a direção, em que ela vai se movimentar pela cabeça, utiliza um switch-case, usando os métodos usados no keyPressed, assim, a cobra segue na direção Up (y-1), Down(y+1), Left (x-1) ou Right(x+1).

10) *iniciarJogo*.: O método iniciar jogo remove os botões, aciona os métodos de adicionar fruta e rodando a variável booleana sai de false para true, assim, a cobra também é adicionada a tela.

11) *verificaColisao*.: Caso a cabeça cobra colida com as bordas da tela ou com o próprio corpo o player é levado a tela fim de jogo.

12) *fimDeJogo*.: A tela fim de jogo exibe o player atual e sua pontuação, além dos botões de Alterar Nome, Remover Nome e Visualizar a pontuação de todos os player anteriores.

E. Classe Painel Jogo.

A Classe painel é uma classe estendida da biblioteca JFrame, a classe painel contempla o frame do jogo e um novo objeto InterfaceJogo da classe InterfaceJogo, assim, cria a Interface, ou seja, a borda que delimita a tela, o nome do jogo, as opções de minimizar, aumentar tela ou fechar o jogo, pela biblioteca JFrame. Já o restante da implementação é chamada pelo novo objeto InterfaceJogo, trazendo toda a complexidade dos métodos anteriores.

III. O JOGO

O objetivo do jogo é controlar uma cobra e se movimentar pela tela, consumindo frutas para que seu corpo cresça ou diminua, aumentando a dificuldade do jogo e a pontuação do jogador. A chamada "cobrinha" não pode colidir de forma alguma com seu próprio corpo ou com as "paredes" que formam o campo de jogo. O Jogador controla a direção da cabeça da serpente (para cima, para baixo, esquerda e direita) e seu corpo segue. O jogo está dividido em partes, a tela inicial, a tela em jogo e a tela de derrota.

A. Tela Inicial.

A tela inicial tem o proposito de receber o Nick que o usuário digita, para no final do jogo salvar a pontuação do jogador juntamente ao nome demonstrado na Fig. 2.

B. Iniciar Jogo.

O jogador pode iniciar o jogo clicando no botão Iniciar demonstrado na Fig. 3.

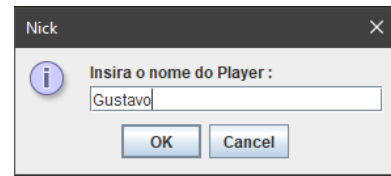


Fig. 2. Tela Inicial.

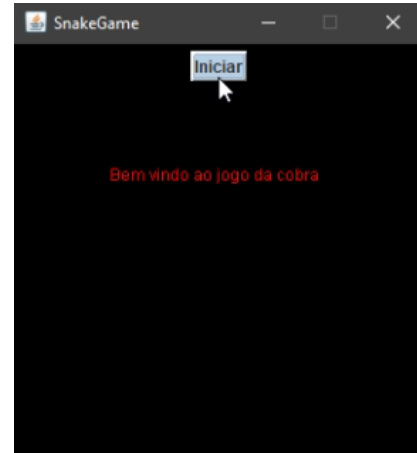


Fig. 3. Iniciar.

C. Jogo em Execução.

A tela do jogo em execução é onde o jogo ocorre exibido na Fig. 4. A Cobra começa com 4 unidades de corpo, sendo uma cabeça e 3 partes de corpo. O jogo possui colisão, nas paredes da tela, na fruta e na própria cobra, logo, a colisão da cobra e das paredes influenciam no fim de jogo, já as frutas interferem no tamanho da cobra. A cobra possui cabeça e corpo, ambas variáveis são compostas pelo tamanho da unidade e se comportam de maneiras semelhantes, porém, a cabeça é única e contém a cor verde claro, já o corpo, um verde escuro. A fruta amarela quando colidida pela cabeça da cobra, perde uma unidade de corpo como mostra na Fig 5. Já a fruta vermelha quando colidida pela cabeça da cobra ganha uma unidade de corpo como mostra na Fig 6. soma-se mais um ponto no score do jogador, já a amarela irá diminuir o delay da velocidade da cobra por um certo tempo. De acordo com cada colisão da cabeça da cobra com as maçãs, a pontuação do jogador cresce.

D. Tela Final.

Quando o jogador colidir com a cabeça da cobra no corpo da cobra ou contra as paredes do mapa exibido na Fig. 7, o mesmo será levado a tela de fim de jogo Fig. 8.

A tela final permite visualizar a pontuação obtida pelo jogador durante a ultima partida do SnakeGame. A tela final também implementa 3 botões: Visualizar Players que mostra o Placar Geral e a pontuação de todos os ultimos players e nicks, Remover Nome remove o player e pontuação, e Alterar Nome que altera o nome do player desejado. Fig. 8.

Mostrar o Placar Geral Fig. 9.

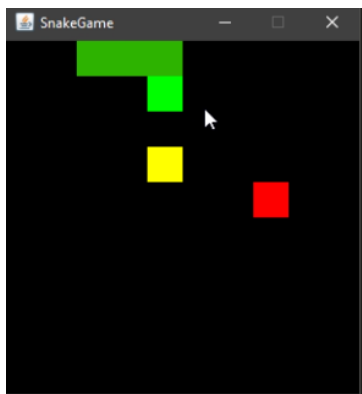


Fig. 4. Tela em Jogo.

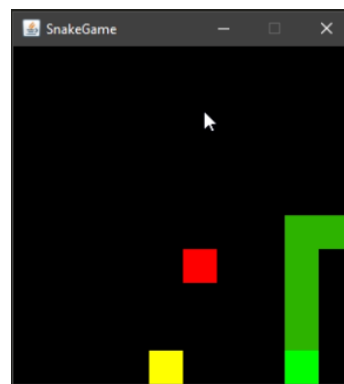


Fig. 7. Colisão Cobra Parede

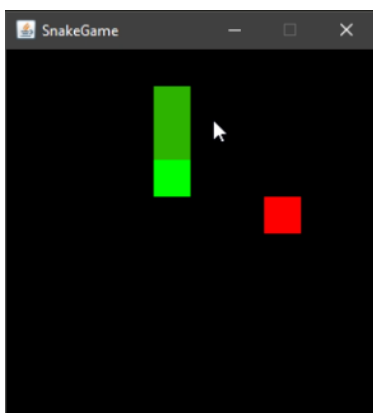


Fig. 5. Tela em Jogo.

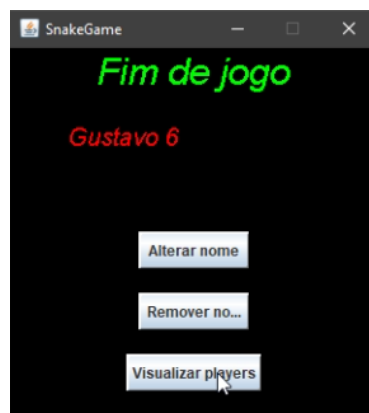


Fig. 8. FimDeJogo



Fig. 6. Tela em Jogo.

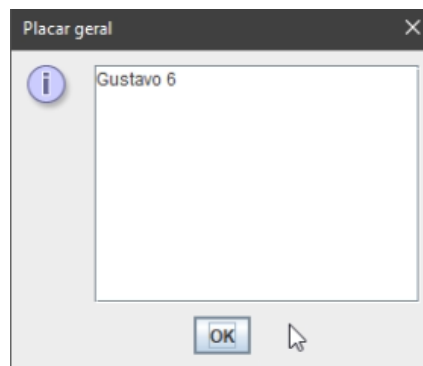


Fig. 9. Placar Geral

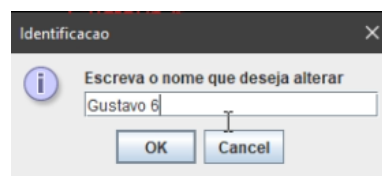


Fig. 10. Alterar Jogador

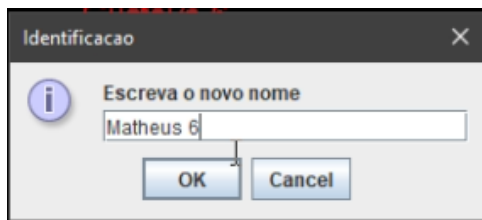


Fig. 11. Renomeção do Jogador

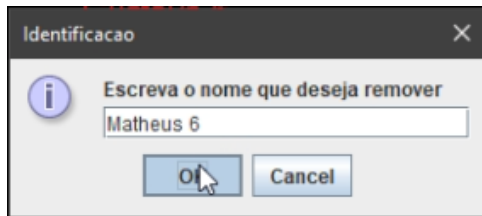


Fig. 12. Remover Jogador

ACKNOWLEDGMENT

A experiência de criar o nosso primeiro jogo em uma nova linguagem de programação com novos conceitos foi completamente diferente aos projetos passados. Desde o início do projeto com a construção do diagrama de classes, a busca e estudo das bibliotecas para dar vida ao projeto e a POO (Programação Orientada a Objetos) tornou a experiência ainda melhor.

REFERÊNCIAS

- [1] GONÇALVES, M. V. Programação Orientada a Objetos. Revista Ada Lovelace, [S. l.], v. 2, p. 106–110, 2018. Acesso em: 26 out. 2021.
- [2] CARMO, A. N. Bibliotecas Swing e JavaFx para softwares Java em ambiente desktop uma análise de usabilidade e código-fonte. 2017. Acesso em: 26 out. 2021.
- [3] SINGLA, Nishant; NARULA, Mandeep Singh. FPGA Implementation of Snake Game Using Verilog HDL. 2018. Acesso em: 27 out. 2021.
- [4] AGAMBEN, Giorgio. O aberto: o homem e o animal. Rio de Janeiro: Civilização Brasileira. 2013. Acesso em: 28 out. 2021.
- [5] BATES, Bob. Game Design: Second Edition. Boston: PTR. 2004. Acesso em: 28 out. 2021.
- [6] CAUQUELIN, Anne. No Ângulo dos Mundos Possíveis. São Paulo: Martins Fontes. 2011. Acesso em: 30 out. 2021.
- [7] DEWEY, John. Arte como experiência. São Paulo: Martins Fontes. 2010. Acesso em: 30 out. 2021.
- [8] HUIZINGA, John. Homo Ludens. São Paulo: Perspectiva. 2010. Acesso em: 31 out. 2021.