

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO DE INFORMÁTICA
TURMA: INFORMÁTICA - 3

Iago Gonçalves Moysés
Pedro Rangel Ferreira de Aguiar
Samuel Henrique da Cunha Machado

Trabalho Prático 2

Old School Games

Contagem
Julho de 2022

Iago Gonçalves Moysés
Pedro Rangel Ferreira de Aguiar
Samuel Henrique da Cunha Machado

Trabalho Prático 2

Trabalho formalizado da disciplina de Linguagens e Técnicas de Programação II, que possui como objetivo ampliar o conhecimento da linguagem de programação Java por meio de técnicas de programação, juntamente com o uso da biblioteca LibGdx. A proposta é explorar aspectos avançados da biblioteca no desenvolvimento de aplicações multimídias.

Professor(a): Alisson Rodrigo dos Santos

Disciplina: Linguagens e Técnicas de Programação II

Turma: INF-3

Contagem
Julho de 2022

RESUMO

O devido trabalho tem por objetivo o desenvolvimento de um jogo, que visa melhorar a capacidade de uso da P.O.O.(Programação Orientada a Objeto), com uma análise de métodos de desenvolvimento acerca de um projeto realizado na linguagem de programação Java. Os discentes foram delegados a recriar um dos jogos de Atari famosos da década de 80. Para tal feito, a biblioteca multiplataforma para desenvolvimento de jogos LibGdx foi utilizada. Possuindo diversas classes com funções já definidas, como *sounds*, *sprites* e *textures*, o desenvolvimento de um game nesse escopo é acelerado. O jogo desenvolvido pelo grupo redator do presente relatório foi o “*Frogger*”.

Palavras-chave: Atari. Frogger. Programação em Java.

LISTA DE ILUSTRAÇÕES

Figura 1 - Anúncio oficial da Sega do Frogger	5
Figura 2 - Logo LibGdx.....	6
Figura 3 - Jogo Frogger em execução	8

SUMÁRIO

1 INTRODUÇÃO	5
2 DESENVOLVIMENTO.....	6
2.1 - MODELO DE APLICAÇÃO	6
2.1.1 - Biblioteca LibGdx.....	6
2.2 - CLASSES E ELEMENTOS	7
2.2.1 - Classe FroggerGame	7
2.2.2 - Classe DesktopLauncher	7
2.2.3 - Classe Abstrata MoveI	7
2.2.4 - Classe Frogg.....	7
2.2.5 - Classe Veículos	7
2.2.6 - Classe InitScreen.....	8
2.2.7 - Texturas, textos e sons	8
3 CONSIDERAÇÕES FINAIS.....	9
REFERÊNCIAS.....	10

1 INTRODUÇÃO

Frogger é um jogo de Video Game criado em 1981 em uma parceria entre as então jovens empresas de informática SEGA e Konami.



Figura 1 - Anúncio oficial da Sega do Frogger

https://bojoga.com.br/files/2018/06/pub_frogger_sega-233x300.jpg

Inicialmente fabricado para máquinas de arcade, logo foi fabricado em grande escala para o console mais popular da época, o Atari 2600, tornando-se também um de seus jogos mais populares. A mecânica do jogo era simples, onde o jogador controla um sapo que deve atravessar, com tempo limitado, uma estrada movimentada e um rio cheio de obstáculos até chegar ao seu ninho. A cada cinco travessias bem sucedidas, o jogo progredia em novas fases de dificuldade crescente.

Diversas empresas independentes desenvolveram versões do Frogger, principalmente para PC.

Partindo desse pressuposto, o grupo em questão desenvolveu sua própria versão do Frogger, através da biblioteca Java LibGdx e utilizando seus conhecimentos gerais de Programação Orientada a Objetos.

2 DESENVOLVIMENTO

Inicialmente, foram realizadas pesquisas acerca das regras do Frogger, além da procura de uma inspiração para norteamento da codificação.

2.1 - MODELO DE APLICAÇÃO

O funcionamento do jogo não é de difícil entendimento, e este será pormenorizado no decorrer desta seção.

Todo o código principal fica no corpo da classe *FroggerGame* que por meio do mecanismo de herança, estende a classe “*ApplicationAdapter*”. Nesta classe, todo o encapsulamento principal é realizado, declarando-se as texturas, os sons, objetos, variáveis e funções. A criação desses elementos na tela e mecanismos do jogo se deu por meio da utilização da biblioteca LibGdx.

2.1.1 - Biblioteca LibGdx

A biblioteca LibGdx é uma estrutura de aplicação com enfoque no desenvolvimento de jogos Mobile e Desktop de código aberto, por meio de linguagens como Java. Foi lançada pela primeira vez, versão 1.0, no segundo semestre de 2014 com cerca de 250 mil downloads mensais através do Maven Central, um repositório fornecido pela comunidade Maven. A principal característica desta biblioteca é o suporte a multiplataforma, sendo possível ser executada em diversos sistemas operacionais. Ela possui diversas extensões como Box2d com enfoque centrado em física, Controllers para a configuração do jogo com um controle de videogame, entre outros.



Figura 2 - Logo LibGdx

https://iteracao.info/processed_images/6e8dc7f3fc0e739300.png

2.2 - CLASSES E ELEMENTOS

Conforme dito anteriormente, diversas classes e funções foram utilizadas, e elas serão detalhadas nos sub-tópicos:

2.2.1 - Classe *FroggerGame*

Classe que engloba o jogo completo.

2.2.2 - Classe *DesktopLauncher*

Classe que possui o Main e será possível nela rodar o jogo.

2.2.3 - Classe Abstrata *Movel*

Esta é a classe que possui os dados e funções referentes a qualquer objeto que se movimenta no jogo. É uma classe abstrata para que todas as instâncias dessa classe sejam obrigadas a implementar os métodos abstratos.

2.2.4 - Classe *Froga*

Como o nome diz, esta é a classe responsável pela implementação do sapo, contendo tudo necessário para sua criação, como o número de vidas, a função de movimentação e as verificações de perda ou ganho da partida. Como é um objeto que se move, esta classe estende a classe *Movel*. O sapo é controlado pelo jogador pelas setas do teclado, ou pelas teclas WASD, e deve se mover até o outro lado da rodovia sem colidir com nenhum dos veículos (carro, caminhão e caminhão de bombeiro). O jogador possui 40 segundos e 3 vidas para terminar a partida. Porém, ao chegar na segunda fase, o número de vidas é restrito a apenas 1, o que aumenta a dificuldade.

2.2.5 - Classe *Veículos*

Responsável pela criação dos veículos. Assim como a classe *Sapo*, como os veículos se movem, essa classe é filha da classe *Movel*.

2.2.6 - Classe *InitScreen*

Contém os atributos necessários para a construção da rodovia que aparece de imediato na tela do game.

2.2.7 - Texturas, textos e sons

Todas as imagens utilizadas foram sprites com fundo transparente retirados da internet, fonte também dos efeitos sonoros do game. Os textos foram utilizados com a finalidade de mostrar as vidas restantes, tempo e a fase em que o jogador se apresenta.

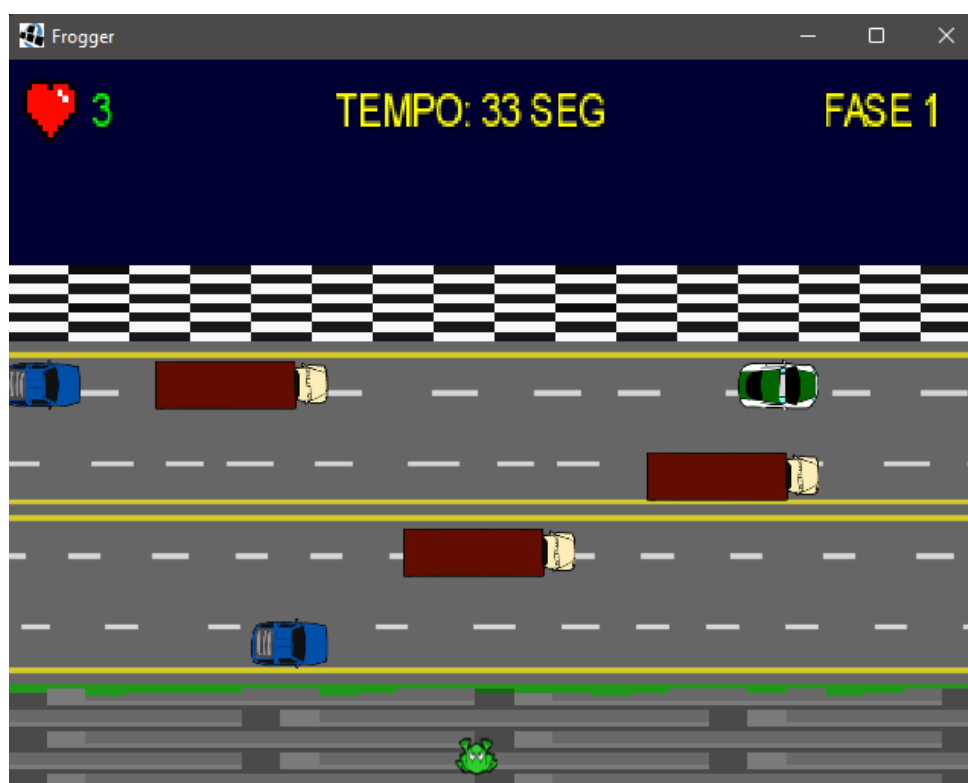


Figura 3 - Jogo Frogger em execução

Autoria própria

3 CONSIDERAÇÕES FINAIS

O grupo concluiu com êxito a tarefa delegada, recriando a própria versão do clássico jogo de Fliperamas e Atari dos anos 80, Frogger. Conforme dito anteriormente neste relatório, a biblioteca Java LibGdx foi utilizada para o desenvolvimento do game. Embora os recursos da biblioteca não sejam difíceis de implementar, a falta de documentação existente foi um empecilho que retardou bastante a codificação do jogo. Ao final, percebeu-se a vantagem da utilização da ideia de polimorfismo, uma vez que várias classes compartilhadas foram instanciadas de maneiras diferentes.

REFERÊNCIAS

FROGGER. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2021. Disponível em: <<https://pt.wikipedia.org/w/index.php?title=Frogger&oldid=62134894>>. Acesso em: 02 jul. 2022.

FROGGER Sprite Sheet PNGITEM. [S.I] [2019]. Disponível em: <https://www.pngitem.com/middle/TRxbJwm_game-graphics-png-sprite-frogger-sprite-sheet-transparent/> Acesso em: 27 jun. 2022.

GPJECC. Como inserir texto no jogo - LibGDX. [S.I] [2016]. Disponível em: <<http://gpjecc.blogspot.com/2016/10/como-inserir-texto-no-jogo.html>> Acesso em: 27 jun. 2022.

GULARTE, Daniel. Frogger(Konami, 1981). BoJoga, 2018. Disponível em: <<https://bojoga.com.br/artigos/retroplay/arcade-pinball/frogger-konami-1981/>>. Acesso em: 28 jun. 2022.

LIBGDX. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2022. Disponível em: <<https://pt.wikipedia.org/w/index.php?title=LibGDX&oldid=63081896>>. Acesso em: 04 jul. 2022.