

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

Artur Borges Corrêa  
Caio César Nascimento Silva  
Igor Richard Dias Silva

**Trabalho Prático 2**

Old School Games (Fishing Derby)

Contagem  
Julho de 2022

Artur Borges Corrêa  
Caio César Nascimento Silva  
Igor Richard Dias Silva

## **Trabalho Prático 2**

Relatório abordando o desenvolvimento de um clone do jogo Fishing Derby, do Atari 2600, recriado utilizando a biblioteca LibGDX em Java.

**Professor(a):** Alisson Rodrigo dos Santos

**Disciplina:** Linguagem e Técnicas de Programação II

**Turma:** INF-3

Contagem  
Maio de 2022

## Resumo

O relatório tem por objetivo realizar uma análise a respeito dos métodos utilizados antes e durante a criação de uma réplica do jogo Fishing Derby, lançado originalmente em 1980 para o console Atari 2600. Para isso, utilizou-se a biblioteca libGDX, que indicada pelo Orientador, e implementada utilizando o editor de código VSCode para a construção do mesmo. Com o estudo da documentação e de vídeos online, o grupo foi capaz de concluir a réplica do jogo, se utilizando de grande parte das funcionalidades da biblioteca mencionada. Durante o relatório, serão abordados os temas de contextualização dos jogos digitais e consoles, uma iniciação a libGDX, discussão sobre estrutura de dados, componentes do jogo original e suas devidas implementações na recriação, além de claro, demonstrar o jogo efetivamente.

**Palavras-chave:** libGDX; Programação Orientada a Objetos; Fishing Derby

## Sumário

<b>Trabalho Prático 2</b>	<b>1</b>
<b>Trabalho Prático 2</b>	<b>2</b>
<b>Resumo</b>	<b>3</b>
<b>Sumário</b>	<b>4</b>
<b>Lista de ilustrações</b>	<b>5</b>
<b>1 Introdução</b>	<b>6</b>
<b>2 Biblioteca LibGDX</b>	<b>7</b>
2.1 Definição	7
2.2 Estrutura	7
<b>3 Fishing Derby - Atari 2600</b>	<b>9</b>
3.1 Console	9
3.2 Fishing Derby	9
3.3 Características Analisadas	10
<b>4 A Recriação do Jogo</b>	<b>11</b>
4.1 Escolha de Componentes Gráficos	11
4.2 Configuração Inicial	11
4.3 Implementação	11
4.4 Lógica de Funcionamento	12
<b>5 Considerações Finais</b>	<b>15</b>
<b>6 Referências</b>	<b>16</b>

## Lista de ilustrações

Figura 1: Criador da LibGDX, Mario Zechner .....	7
Figura 2: O console Atari 2600 .....	9
Figura 3: O jogo Fishing Derby .....	10
Figura 4: Exemplo de Assets do pacote .....	11
Figura 5: Tela para a escolha do modo de jogo .....	13
Figura 6: Tela de jogo padrão .....	16

## 1 Introdução

Desde a década de 1970, um dos meios de entretenimento mais frequentes no nosso mundo são os videogames, servindo para causar diversão nos mais variados públicos, sejam eles jovens ou adultos, iniciantes ou veteranos do mundo do entretenimento digital. Os jogos digitais surgiram como uma forma fácil e prática para se aliviar o estresse diário, ou competir com um amigo por qualquer motivo que seja, eles ajudam as pessoas de diversas formas, e muitas vezes até como um certo controle emocional.

Em 1952, Alexander Shafto Douglas descreveu em sua tese de Phd de Cambridge um pensamento sobre interação humano-máquina, e anos mais tarde foi responsável pelo desenvolvimento do primeiro jogo gráfico de computador, o famoso “Tic-Tac-Toe”, ou Jogo da Velha, para os mais íntimos. Já em 1958, William Higinbotham criou o primeiro videogame de todos os tempos, intitulado “Tênis para Dois”, e em 1962 seu amigo Steve Russell criou “Space War”, o primeiro jogo destinado ao uso via computador.

Utilizando a ideia do “Tênis para Dois”, que servia mais como um conceito para ser apresentado em feiras tecnológicas Nollan Key Bushnell aplicou suas idéias e, em 1972, fundou uma nova empresa, a “Atari, Inc.”, juntamente de Ted Dabney.

E um dos famosos jogos criados para esta plataforma é o Fishing Derby, publicado pela Activision em 1980, um jogo de pesca idealizado por David Crane. Como objetivo deste projeto, o grupo irá recriar Fishing Derby utilizando a biblioteca libGDX, da linguagem Java. e ...

Faltou descrever os requisitos do trabalho alem da descrição do relatório.

## 2 Biblioteca LibGDX

### 2.1 Definição

A LibGDX é uma estrutura de aplicação e biblioteca de desenvolvimento de jogos livre, gratuita e de código aberto escrita em Java com componentes em C e C++. Criada por Mario Zechner em 2009, com o objetivo de ser um meio facilitador para, inicialmente, a criação de jogos em Android.



**Figura 1: Criador da LibGDX, Mario Zechner**

<https://i1.sndcdn.com/artworks-000209835570-udqz5m-t500x500.jpg>

Ela é distribuída utilizando a licença Apache 2.0, que permite sua utilização, modificação e redistribuição de forma livre e sem necessidade de pagamento de direitos autorais. É compatível com Windows, Linux, Mac OS X, Android, BlackBerry OS, iOS, Java Applet e JavaScript/WebGL.

### 2.2 Estrutura

Sua estrutura básica pode ser obtida utilizando o arquivo “gdx-setup.jar”, disponível para download em seu síte oficial, e é composta por uma estruturação de pastas, contendo arquivos como:

- DesktopLauncher.java: Responsável por inicializar a janela principal do jogo, contendo suas configurações básicas.
- MyGdxGame.java: Arquivo da classe principal do jogo, normalmente é renomeado para o nome do projeto, e é onde ficam as funções principais de funcionamento.

Dentro do arquivo da classe principal, temos algumas funções básicas que tem grande importância no funcionamento da biblioteca:

- Create: Responsável por inicializar os elementos da aplicação;
- Render: Chamada em cada atualização de quadros do jogo, fazendo verificações e chamadas de funções recorrentemente.
- Dispose: Chamado quando a aplicação é destruída, e deve finalizar os elementos que foram inicializados na função 'create'.



Não deixe espaços vazios no relatório.



### 3 Fishing Derby - Atari 2600

#### 3.1 Console

O Atari 2600 foi um console lançado em setembro de 1977, pela Atari, Inc. Ele vendeu mais de 30 milhões de cópias, e foi um dos grandes responsáveis pela explosão dos jogos digitais desde então.

Mesmo sendo descontinuado em 1992, o Atari 2600 trouxe muitos jogos e conceitos inovadores para a época, e um dos motivos disso era o seu alto nível de processamento e memória para os padrões que a indústria estava acostumada. Tinha incríveis 128 bytes de RAM, e contava com uma CPU 8-bits de 1.19 MHz, o que permitiu o desenvolvimento de diversos jogos durante o seu período de vendas, incluindo o foco deste relatório, Fishing Derby.



**Figura 2: O console Atari 2600**

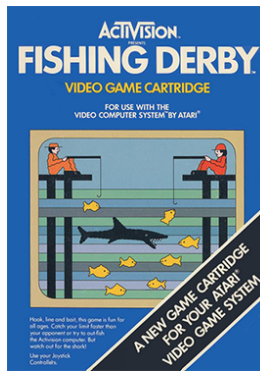
<https://upload.wikimedia.org/wikipedia/commons/0/02/Atari-2600-Wood-4Sw-Set.png>

#### 3.2 Fishing Derby

Escrito por David Crane, desenvolvido e publicado pela Activision para o Atari 2600 em 1980, Fishing Derby era um jogo com um conceito simples, mas que foi revolucionário.

O jogo consistia na competição de 2 pescadores, onde se deveria obter 99 libras em peixes para se vencer e podia ser jogado por dois jogadores no mesmo console, ou por um único jogador, que enfrentaria um pescador controlado pela máquina.

Foi um dos primeiros jogos digitais desenvolvidos pela Activision, e foi extremamente bem recebido pela crítica, sendo classificado como “imaginativo, colorido e divertido”, provendo as crianças um ânimo melhor do que o show de TV da manhã de sábado, e provendo aos adultos um desafiante jogo para testar a sua habilidade” pela *Arcade Alley*, sendo recomendado sempre como uma boa opção para se jogar em família.



**Figura 3: O jogo Fishing Derby**

[https://en.wikipedia.org/wiki/Fishing\\_Derby#/media/File:Fishing\\_Derby\\_coverart.png](https://en.wikipedia.org/wiki/Fishing_Derby#/media/File:Fishing_Derby_coverart.png)

### 3.3 Características Analisadas

Para a realização do clone de Fishing Derby, o grupo analisou e separou algumas características que são essenciais para esse estilo de jogabilidade, entre eles estão:

- Existência de 2 pescadores, capazes de movimentar sua linha e pescar um peixe;
- Diferença de peso, em libras, de peixes de diferentes alturas. Os dois primeiros tinham 2 libras, os 2 do centro possuíam 4 libras, e os peixes mais ao fundo, pesavam 6 libras;
- A subida automática da linha de pesca ao pescar efetivamente um peixe;
- A existência de um tubarão, que poderia comer os peixes antes dos mesmos chegarem à superfície, impedindo o jogador de conseguir aquela pontuação;
- A possibilidade de se acelerar a linha de pesca enquanto um peixe está fisingando, apertando um botão;
- Variações entre padrões de movimentação do tubarão e dos peixes;
- Possibilidade de se jogar em um ou dois jogadores;
- O objetivo do jogo, que era chegar a 99 libras de peixe pescado;

## 4 A Recriação do Jogo

### 4.1 Escolha de Componentes Gráficos

Durante o início do projeto, foi decidido que se utilizaria componentes gráficos presentes de forma livre na internet, e após pesquisas em diversos sites, o grupo optou por utilizar o pacote “Free Fishing Pixel Art Pack”, disponibilizado de forma gratuita pela “Craftpix.net” através da plataforma “itch.io”.



Figura 4: Exemplo de Assets do pacote

<https://free-game-assets.itch.io/free-fishing-pixel-art-pack>

### 4.2 Configuração Inicial

Utilizando a ferramenta “gdx-setup.jar” disponibilizada pela biblioteca, o grupo criou a primeira base onde o jogo seria construído, utilizando os parâmetros:

- Nome do Projeto: Fishing Derby Clone;
- Nome do Pacote: com.grupo7;
- Classe do Jogo: FishingDerbyClone;
- Plataformas Suportadas: Desktop (LWJGL 3).

Foi realizado o download dos componentes gráficos escolhidos, os mesmos foram colocados na pasta “Assets” e a estruturação padrão da libGDX foi mantida e utilizada.

### 4.3 Implementação

A partir da configuração inicial, o grupo começou desenvolvendo a lógica de inicialização de todos os componentes audiovisuais. Dentro da classe principal

‘FishingDerbyClone’, a função *create* é utilizada para carregar as texturas, sons, música, os objetos padrões do jogo (como barcos, pescadores, peixes e tubarões), além da inicialização da câmera, música e fontes de pontuação.

Com a ideia de ter um jogo com animações, o grupo partiu para a criação da classe “AnimationController”, que posteriormente foi remodelada para comportar mais características além das pensadas inicialmente, e passou a se chamar “GameObjectController”. Essa classe é responsável pelo funcionamento das animações frame a frame, além de armazenar sons e hitboxes (áreas de colisão dos objetos).

Também foi criada uma classe abstrata chamada “Movement”, que possui métodos que seriam comuns à maior parte dos objetos do jogo. A mesma foi estendida por “GameObjectController”.

Para objetos que estão na tela do jogo, foram criadas as classes “Fish”, “Shark”, “Player”, “FishingRod” e “FishingHook”, todas essas estendendo “GameObjectController”. “Fish” controla a movimentação dos peixes, as variáveis para definir se ele foi pego por algum jogador ou não, identificações de peso, altura no mar e ID, além da função de correr da vara de pesca. “Shark” controla a movimentação do tubarão e sua função de começar e parar um ataque, que faz sua velocidade de movimento aumentar. “Player” não guarda nenhuma informação importante, e apenas serve para a exibição dos jogadores na tela. “FishingRod” controla unicamente a movimentação da vara de pesca na parte superior da tela. Já “FishingHook” é responsável pela principal interação do jogador com o próprio jogo, controlando o anzol responsável por fisgar os peixes.

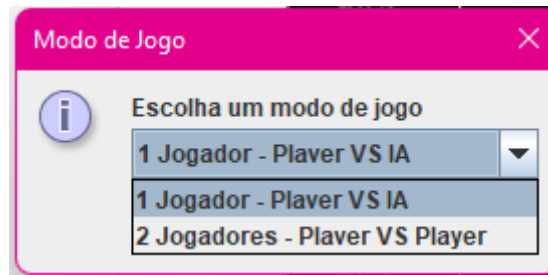
#### **4.4 Lógica de Funcionamento**

Ao inicializar o jogo, o jogador é apresentado a uma tela de escolha, onde ele deve optar por uma das opções:

- 1 Jogador - Player VS IA: O jogador irá competir contra uma inteligência artificial básica programada pelo grupo. Aqui temos duas diferenças principais em relação ao jogo original. Primeiramente, a inteligência artificial no jogo original tem a capacidade somente de pescar da forma mais ‘eficiente’, buscando sempre o peixe de maior valor ao fundo, já a nossa funciona de forma mais randômica, buscando peixes que ela considera dentro das suas possibilidades de fisgar. E em segundo, para compensar uma leve incapacidade e velocidade de processamento que levam a defasagem da inteligência artificial em relação ao jogador, ela ganhou a capacidade de se

movimentar em diagonal, opção não disponível para o jogador, que deve sempre se mover em um eixo por vez.

- 2 Jogadores - Player VS Player: Como o próprio nome indica, o jogo será entre duas pessoas utilizando o teclado. O primeiro jogador usa como controle as teclas WASD, e o segundo jogador deve utilizar as setinhas do teclado.



**Figura 5: Tela para a escolha do modo de jogo**

Arquivo / Reprodução Pessoal

Quando o jogador confirmar a escolha, ele irá para a tela onde poderá jogar, seguindo o modo de jogo escolhido.

A lógica de funcionamento das mecânicas do jogo é bem simples:

**Tubarão:** Se desloca repetidamente entre as laterais da tela, próximo aos barcos onde os jogadores ficam. Tem a chance de iniciar ataques em certos intervalos de tempo, e durante os mesmos, sua velocidade de movimento cresce. Se o tubarão encostar em algum anzol com um peixe fígado, ele come o peixe, roubando assim a pontuação que aquele jogador iria obter.

**Peixes:** Se movimentam entre as paredes incessantemente, e possuem a chance de observar uma vara de pesca se aproximando e desviar para o outro lado. Se forem fígados, eles acompanham o anzol enquanto o mesmo sobe em direção ao jogador.

**Anzóis:** Podem se movimentar dentro do seu espaço, que vai da borda mais próxima do jogador, até um ponto próximo ao centro da tela. Se pescarem um peixe, se movimentam automaticamente para cima, em direção ao jogador, possuindo a única capacidade de acelerarem o movimento apertando uma tecla de ação.

**Vara de Pesca e Linha de Pesca:** Acompanham o movimento dos anzóis.

**Jogadores, Barco e Background:** São estáticos, e servem como ambientação.

**Pontuação:** Cresce de acordo com o peixe fígado, podendo subir 2, 4 ou 6 pontos. Se chegar a 99 pontos, o jogador ganha o jogo.

**Inteligência Artificial:** Só disponível no modo de um jogador, ela foi programada pelo grupo para seguir um certo padrão de movimentos. Primeiramente ela busca na tela um peixe

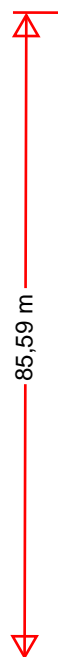
que considere possível de ser fisgado, e se movimenta no menor caminho possível até ele, depois, sobe com velocidade máxima até o barco do jogador, e repete o processo novamente.

Áudio: Como música de fundo, temos um loop da música “A Lonely Cherry Tree”, do canal “Pix”. Já para o áudio, temos alguns efeitos sonoros que tocam em determinadas situações, como um peixe sendo fisgado, ou o tubarão comendo um peixe que o jogador estava puxando, a maior parte dos efeitos sonoros foi gerada utilizando o site “jsfxr”.



**Figura 6: Tela de jogo padrão**

Arquivo / Reprodução Pessoal



## 5 Considerações Finais

Esse trabalho teve por objetivo realizar a recriação de um jogo simples e *Old School*, aperfeiçoando a capacidade dos membros de se trabalhar com a Programação Orientada a Objetos em um ambiente mais prático e lógico, presente na vida de grande parte dos alunos, os jogos.

Para isso, foi estudado e utilizado a libGDX, uma biblioteca em Java, adquirindo mais conhecimentos da sua documentação e colocando em prática diversas funções da biblioteca, aprendendo juntamente disso a estrutura de um jogo, e o melhor funcionamento das etapas que são realizadas na criação do mesmo.

Como objetivo futuro, o grupo poderia visar em construir melhorias para o jogo, como uma atualização na inteligência artificial e correção de pequenos bugs, além de melhorias visuais.

## 6 Referências

DESENVOLVENDO Meu Primeiro Jogo - Java, LibGDX e Inkscape. Youtube, GuiiCS, 2020. 1 vídeo (05:13). Publicado pelo GuiiCS. Disponível em: [https://youtu.be/C24nIOJU\\_kc](https://youtu.be/C24nIOJU_kc). Acesso em: 26 jun. 2022.

HOW to Setup a libGDX Project in 1 Minute (2020). Youtube, ACCountNine38, 2020. 1 vídeo (01:05). Publicado pelo ACCountNine38. Disponível em: <https://youtu.be/upB47t8h888>. Acesso em: 26 jun. 2022.

LIBGDX. **libGDX Wiki**. [S.l.]. libGDX, 2022. Disponível em: <https://libgdx.com/wiki/>. Acesso em: 22 jun. 2022.

LIBGDX. **libGDX Source & Documentation**. [S.l.]. libGDX, 2022. Disponível em: <https://libgdx.com/dev/>. Acesso em: 22 jun. 2022.

LIBGDX. **2D Animation**. 2D Animation Documentation. [S.l.]. libGDX, 2022. Disponível em: <https://libgdx.com/wiki/graphics/2d/2d-animation>. Acesso em: 25 jun. 2022.

LESTAT. **Stack overflow**. How to draw a BitmapFont in LibGDX?. [S.l.]. John Conde, 2013. Disponível em: <https://stackoverflow.com/questions/12895822/how-to-draw-a-bitmapfont-in-libgdx>. Acesso em: 26 jun. 2022.

MARIO ZECHNER. **GitHub**. Official source-code of libGDX. [S.l.]. Mario Zechner, 2022. Disponível em: <https://github.com/libgdx/libgdx>. Acesso em: 4 jun. 2022.

TUTORIAL how to do 2D animation in libGDX. Youtube, Zereb Zereb, 2021. 1 vídeo (04:33). Publicado pelo Zereb Zereb. Disponível em: <https://youtu.be/VHaFoym7OBg>. Acesso em: 26 jun. 2022.

XOPPA, JRENNER, TITOVMAXIM, NATHANSWEET, TAVOMACIEL, MGSX-DEV, SIMONIT. **GitHub**. AnimationController.java. [S.l.]. libGDX Community, 2022. Disponível em: <https://github.com/libgdx/libgdx/blob/master/gdx/src/com/badlogic/gdx/graphics/g3d/utils/AnimationController.java>. Acesso em: 26 jun. 2022.