Universidade Federal do ABC

Haskell Breakout

Paradigmas de Programação, A Noturno, Santo André.

Luiz Felipe Leal Gomes Azzolini RA: 11048715

Introdução.

O projeto visa implementar um clone do jogo Breakout, de 1976 da plataforma Atari, de modo a explorar conceitos de programação funcional e específicos da linguagem Haskell, dita como puramente funcional

Estrutura do projeto.

O projeto foi dividido em vários arquivos de modo a manter sua organização, são eles:

1. **Main.hs**: responsável por centralizar renderização e carregamente de assets
2. **Breakout.hs**: mantém a estrutura de mundo, seus parâmetros gerais e valores iniciais do jogo
3. **KeysController.hs**: Define teclas de controle do jogo
4. **PaddleController.hs**: controla a movimentção da raquete
5. **TileModel.hs**: estrtura e renderização dos tijolos
6. **BallController.hs**: gerencia colisões e movimentação da bolinha e decremento de estado dos tijolos

Conceitos utilizados.

Foram utilizados vários conceitos de programação funcional e haskell como Pattern Matching, Data types, Imutabilidade, High order functions, Paralelismo

Como utilizar.

Para utilizar o programa basta executar a linha de comando abaixo, dentro da pasta raiz do projeto:

```
stack run
```

Para a execução no WSL2 usando a distro **Ubuntu18.04** foi necessário instalar os pacotes freeglut3 freeglut3-dev.

Dificuldades.

Durante a implementação algumas dificuldades foram encontradas. Uma delas foi a introdução de bitmaps externos no programa, entender como é feita a leitura de arquivos externos é um desafio, porém utilizando de modo correto sua implementação e utilização é bastante simples.

Outro ponto de dificuldade foi lidar com as colisões de forma concisa.

Pontos de melhoria.

Faltaram pontos desejados a serem implementados, como *"poderes"* que cairiam dos tijolos ao serem quebrados, um deles já foi implementado e é acessível através da tecla *espaço* para testes.

Além deste foram planejados mais 3 poderes, desaceleração da velocidade da bolinha, triplicar o numero de bolinha em jogo e bolinha com maior dano aos tijolos e seriam implementadas da seguinte forma:

1. **velocidade da bolinha**: já existe um parâmetro reservado para este propósito dentro do tipo *Game*, dessa forma ao dar esse poder bastaria altera-lo em tempo de execução.

2. **triplicar o numero de bolinhas**: seria implementado um vetor de bolinhas dentro do tipo *Game*, de forma similar a que foi feita com os tijolos, colocando assim parâmetros que estão em *Game* atualmente, dentro de um outro tipo chamado *ball*.

3. **bolinha pesada**: esse poder seriaum parâmetro dentro do tipo *ball* utilizado como o dano causado ao estado do *Tile*, decrementando 2 ou mais por colisão.

Links.

Github

YouTube