

O projeto consiste em recriar o jogo Asteroids, lançado pela Atari em 1979 para arcades, em linguagem Haskell. Para esta finalidade, foram utilizadas as bibliotecas Graphics.Gloss e System.Random. Em Asteroids, o jogador controla uma nave representada por um triângulo isósceles. Os controles incluem acelerar a nave, rotacioná-la em sentido horário e anti-horário e atirar. Os asteroides são representados pelas demais formas geométricas na tela. O asteroide se move sempre em linha reta com velocidade constante e pode colidir contra a nave, reduzindo o número de vidas do jogador. O jogador possui três vidas e o jogo encerrará após perder todas as vidas. O jogador pode atirar contra os asteroides para destruí-los. Cada asteroide grande destruído dará lugar a dois asteroides menores com direções e velocidades aleatórias, e os asteroides pequenos não formam novos asteroides quando destruídos. O objetivo do jogo é destruir todos os asteroides na tela para avançar de nível. Em cada nível há um número de asteroides e ao destruir todos o jogador avança de nível e uma nova leva de asteroides surgirá com número inicial de asteroides incrementado. Asteroids funciona com um sistema de coordenadas toroidal, isto é, o lado de cima está conectado ao lado de baixo da tela, assim como o lado esquerdo e o lado direito. Dessa forma um objeto deixando a tela por um lado poderá ser visto reaparecendo do lado oposto em tempo real.

O código pode ser compilado com o comando `stack build` e executado com o comando `stack run`. Uma vez executado o código, uma nova janela será criada com a tela inicial do jogo. A partir desse momento o programa passará a capturar entradas fornecidas pelo usuário. Apertando a barra de espaço o usuário dará início ao jogo. Neste momento o usuário ganha controle da nave e três asteroides são gerados em posições aleatórias, sendo garantido que nenhum asteroide aparecerá em cima da nave. Para controlar a nave são usadas as teclas direcionais, direcional para cima acelera, direcional para esquerda rotaciona em sentido anti-horário e direcional para direita rotaciona em sentido horário, e a barra de espaço para atirar, sendo que no máximo cinco tiros podem estar na tela simultaneamente. O número de vidas do jogador e o nível atual estão demonstrados no canto superior esquerdo da tela nos campos “Lives” e “Waves”, respectivamente. Inicialmente o número de vidas é três e o número da onda é um. A janela poderá ser fechada a qualquer momento com a tecla Esc.

O maior destaque do código é o módulo Wrap onde foi feita a implementação do sistema de coordenadas toroidais. Como o pacote Gloss não suporta nativamente esse sistema de coordenadas, foi necessário criar uma função wrap que corrige a posição de um ponto saindo da tela para que reapareça do lado oposto. Com essa função implementada, foi possível escrever funções que desenhavam pontos, linhas e polígonos bem como funções que realizam transformações lineares neste sistema de coordenadas. Outra destaque é o uso de conceitos cinemática vetorial para atualizar as transformações dos objetos na tela.

Uma dificuldade encontrada foi a utilização de números aleatórios com o pacote System.Random. Como Haskell é uma linguagem com funções puras, não é possível modificar uma variável global para ser usada como semente de um gerador de números aleatórios, o que torna o uso de números aleatórios mais inconveniente comparado a uma linguagem imperativa como C. Contudo, essa dificuldade pôde ser superada criando um gerador inicial na função main com a função `getStdGen` do pacote e escrevendo funções que recebem um gerador como parâmetro e devolvem um gerador diferente com a função `split`.

Link para a apresentação do projeto em vídeo: <https://youtu.be/U1cN7DABW9k>