

Readme

Mehrad Haghshenas

2022-11-14

Asteroids

This project was created with “cabal” and the dependencies “gloss” and “random” have been added. There are four haskell files which are in the app directory.

Main

This file contains the main function: `main :: IO()` The images have been loaded in this file. Note that the images were taken from openGameArt.org

Model

This file contains the data types which have been defined to create this game. The data types include: 1- GameState: to encapsulate the characteristics of the game in each state (like imperative programming). 2- Phase: to check whether the game has ended, is pauses, is playing, or has just started. (The game has 4 phases). 3- Bullet: the bullets which the user shoots via the ship. 4- Collide: each collide has a velocity, location, and radius. This data type was created to check whether two things have collided or not. There are two types of collision in the game namely: 1) collision between asteroids and the ship. When this happens the game is over. 2) collision between bullets fired and asteroids. When this happens the user gains score. 5- Location: Location of each collision. 6- Velocity: velocity of movement. 7- Asteroid: the enemies in this game are the asteroids. 8- Particle: when explosion happens, these particles are created. 9- Explosion: a list of particles are created through explosion. Thus, this is just an alias of [Particle]

Finally, we have defined the initial state of the game in this module.

View

This file contains the function in turning the game state to a picture. All the characteristics of the game state such as, the ship, asteroids, bullets, and particles are pictured in this module.

Controller

This file contains the interaction and mainly impure functions of the game. Birefly, this first function is one step on the game. This is followed by several functions checking the collision of the game and procuding asteroids. Then the event handling functions are created. These functions are the response of the game when the user clicks a keyboard button. The only interaction in this game is done through the keyboard and the mouse is not used. Furthermore, there are functions related to updating the gamestate features (ship, asteroids, bullets, and particles). At last, there are the functions related to updating the file “scores.txt” which scores a string of the highest scores in a descending order.

How to play

open the file and within the directory in the terminal type “cabal run”. The game will run. In the game there is randomness in the location of the asteroids and the speed of them. The scores of the user will be shown to

the score.txt text file. You can pause and restart the game with the “p” and “r” button respectively. Finally, the asteroids are the enemies. The game also includes explosion when killing an asteroid. The game will end if an asteroid hits the spaceship. As the game proceeds the game will become harder. Specifically after score 1000, the asteroids will become more intelligent and moves towards the spaceship instead of wandering around. So I guess, they are kind of different enemies than the enemies seen in the start of the game.