

Name: Mr. Healy

Course: Intro CS Honors

Sample Game Proposal

This is a proposal for a game. It demonstrates the content I'd like you to include in your own proposal. You may use this as a template and provide the details for your own game.

Overview

My game is called **Asteroids**, and is based on the 1980s arcade classic. The object of the game is to pilot a spaceship around on the screen and shoot at asteroids that are in your way so they don't collide with your ship and kill you.

A sample screen from the game might look like this (you'll probably want to include a hand-drawn sketch since you haven't written your game yet; since this is just an example I'm using a screenshot):



Gameplay

The player's ship is represented by a small triangle on the screen. The player controls the ship by using the left and right arrow keys (to turn), and the up arrow key (to accelerate the ship forward). There are no brakes! To slow down, the user must turn the ship around and apply the thrusters to counteract any existing velocity.

The ship also has a gun, which can be fired with the "f" key, and a shield that can be activated using the "down" key. There is a limited amount of power for the shield, so the player cannot use it indefinitely.

The asteroids are represented by circular objects on the screen. When a player's bullet collides with an asteroid, the asteroid breaks into smaller asteroids and points are added to the player's score. When the asteroids are small enough, they're destroyed completely.

If an asteroid collides with the player's ship, then the ship is destroyed and the player loses a life (or the game).

If an asteroid collides with the player's shield then it is destroyed in the same way as a bullet. Objects on the screen "wrap around". For example, if something flies off the right side of the screen, it reappears on the left side.

Basic Features

The basic features of the game include:

- The player's ship, and the ability to apply steering and thrusters
- The ability to fire bullets from the ship
- Asteroids that appear on the screen and can be destroyed by the player's bullets
- The player's ship can be destroyed by colliding with an asteroid
- The player's score should be displayed and go up when an asteroid is destroyed

Advanced Features

- A new level starts when all the asteroids are destroyed
- The player's ship has a shield, with a limited power supply and a display of the remaining power on the screen
- The player has multiple lives, and can choose when to start their next life and reappear on the screen
- The asteroids break into smaller asteroids when shot
- The asteroids look more like rocks instead of just circles
- Sound effects
- Additional animations for the ship or asteroids being destroyed

Objects

The game will use the following major objects:

Ship

This object represents the player's ship on the screen.

Variables

- x coordinate
- y coordinate
- rotation
- x velocity
- y velocity
- boolean for whether shield is active or not
- amount of shield power remaining
- boolean for if the ship is turning left
- boolean for if the ship is turning right
- boolean for if the ship is alive (on the screen)

Functions

- overlaps(Asteroid): returns true if the ship overlaps with a particular asteroid
- spawn(): make the ship appear on the screen
- die(): make the ship disappear
- render(): draw the ship on the screen
- update(float): move the ship based on the elapsed time
- shieldOn(): make shield true if there's enough shield power
- shieldOff(): turn the shields off
- thrust(boolean): increase forward velocity if true
- left(boolean): start turning left if true
- right(boolean) start turning right if true

Bullet

The player's ship can generate bullets. Once created, the bullets move on the screen independently of the ship.

Variables

- x coordinate
- y coordinate
- x velocity
- y velocity
- time to live (bullets disappear after a short time)

Functions

- render(): draw the bullet on the screen
- update(float): move the bullet based on the elapsed time

Asteroid

Asteroids move with a constant velocity on the screen. If they are shot by a bullet, they split into smaller asteroids with their own velocities.

Variables

- x coordinate
- y coordinate
- x velocity
- y velocity
- rotation
- rotational velocity

Functions

- render(): draw asteroid on the screen
- update(float): move the asteroid based on the elapsed time
- split(): split the asteroid into smaller asteroids
- overlaps(Bullet): see if this asteroid overlaps the given Bullet

ScoreBoard

The scoreboard keeps track of and displays the score, number of lives, and level for the player.

Variables

- score
- level
- lives

Functions

- render(): draw the scoreboard on the screen
- update(float): update the scoreboard based on the elapsed time
- reset(): reset all values back to start
- hit(): register an asteroid being destroyed
- die(): register the player losing a life

Explosion

An explosion is created when an asteroid is destroyed. It's just a short-lived animation to make things look cooler. The explosion is just made up of a bunch of ellipses that radiate from a central point. All the variables are arrays, where the array holds the type of value for each ellipse.

Variables

- array of x coordinates
- array of y coordinates
- array of x velocities
- array of y velocities

Functions

- render(): draw this explosion on the screen
- update(float): update the position of this explosion based on the elapsed time provided

ShipExplosion

ShipExplosion is an effect when the player's ship is hit and dies. The types of variables and functions are nearly identical to those of the Explosion class, but the size and color of the particles are different.

Variables

(Same as Explosion.)

Functions

(Same as Explosion.)