

Shooty Space Rocks

Game Design Document

By Sam Ward

Contents

| | |
|-------------------------------|----------|
| Overview..... | 3 |
| Description | 3 |
| Feel and Style..... | 3 |
| UI..... | 3 |
| Main Menu | 3 |
| Start | 3 |
| Controls..... | 3 |
| Quit | 3 |
| Pause menu | 3 |
| Continue..... | 3 |
| Controls..... | 3 |
| Main | 3 |
| In-Game HUD | 4 |
| Gameplay..... | 5 |
| Progression | 5 |
| Mechanics..... | 5 |
| Artificial Intelligence | 5 |
| Player Movement..... | 5 |
| Asteroids | 5 |
| Rockets:..... | 5 |
| Bullets | 5 |
| Collision detection | 5 |
| Background scenery..... | 5 |
| Controls | 6 |
| Player Ship | 6 |
| Menu Controls | 6 |
| Cheat Features | 6 |
| Assets..... | 7 |
| Gameplay Assets | 7 |
| Environmental | 7 |
| Sound | 7 |
| Music..... | 7 |
| Sound Effects | 7 |

Overview

Description

The player is lost in an asteroid field and must escape by destroying all the asteroids.

In order to win the player must shoot asteroids while simultaneously manoeuvring to avoid collisions. Once a set number of asteroids have been destroyed, the player is free from the asteroid field and wins the game. If the player takes too much damage from collisions, their ship is destroyed, and the game is lost.

Feel and Style

The feel of the game intends to keep the player on their seat with the rush of near collisions, at times surging to overwhelmingness while still maintaining plausibility and hopefulness.

The visual and audio style, and general control of the game is that retro arcade style but with a modern console approach.

UI

Main Menu

User scrolls through menu items with current option highlighted.

Start

Starts a new game.

Controls

Displays controls for the game.

Quit

Exits the application.

Pause menu

Continue

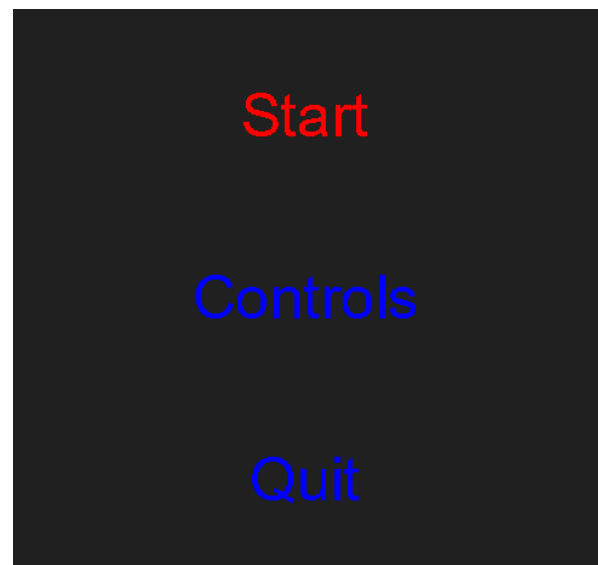
Un-pause the game and continue.

Controls

Displays controls for the game.

Main

Quit current game and return to main menu.



*not final product

In-Game HUD

Information display for ship health, number of missiles available, and number of asteroids destroyed (image example only).



Gameplay

Progression

The game utilises continuous play progression. Rather than using set levels, the difficulty increases over time. This is non-linear in order to create periods of intensity and lulls, adding to the game feel.

Mechanics

Artificial Intelligence

Enemy UFOs will follow a random path and fire bullets at the players current position.

Player Movement

Player movement has x and y movement and rotation. Algorithm using current angle, target angle, and rotation speed in order to give the rotation a more realistic feel, rather than snapping to joystick position.

Asteroids

Asteroids are rotating sprites that follow a linear trajectory (random at creation). When impacted, asteroids will split into two smaller asteroids along a different vector (semi-random – still somewhat same direction)

There are 4 sizes of asteroid with different characteristics:

- Smaller asteroids are faster and bigger asteroids are slower.
- If two asteroids collide, the smaller one is destroyed (split). If both are the same size, then both are destroyed.
- Large asteroid does 100% damage, medium 50%, small 25%, tiny, 2%.
- Tiny asteroids will fade over time.

Rockets:

Rockets are the players secondary weapon. They are limited in number and accumulate over time. The player can load a rocket by pressing a button, then aim the crosshair with the analog stick and fire with trigger. During this process the ability to rotate and fire the blaster will be disabled but can still move to avoid asteroids.

A Rocket will destroy any asteroid or UFO in one hit.

Rockets do not wrap screen or fade; they move to the target location and explode.

Bullets

Bullets will fade over time and wrap screen. There is a fixed number of bullets that can be on the screen at a time, this allows the player to fire faster at closer objects.

Collision detection

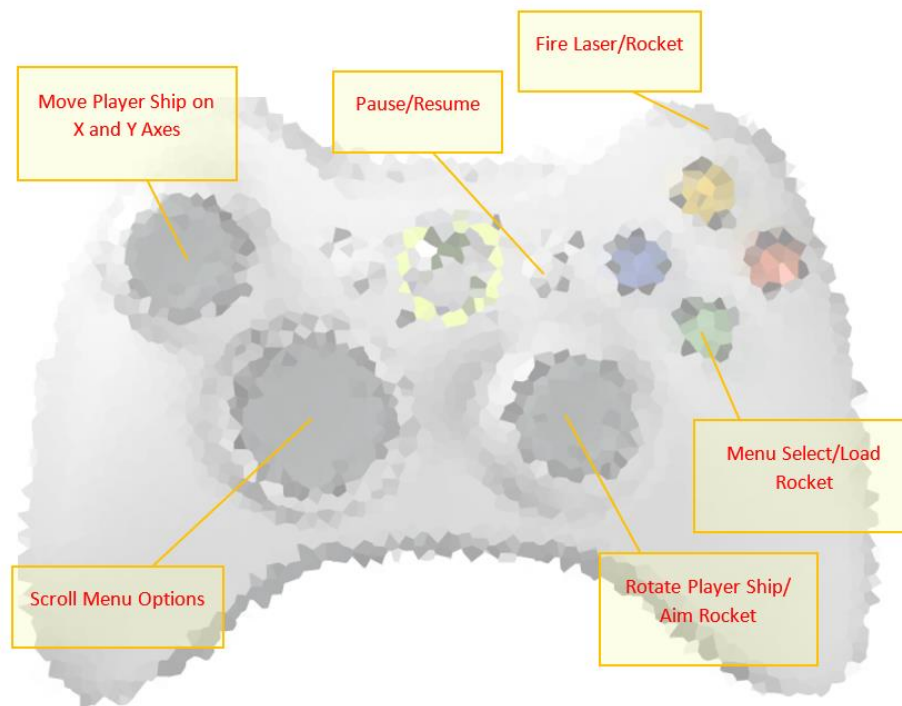
Bullet to asteroid, asteroid to ship, asteroid to asteroid, enemy bullet to ship, enemy bullet to asteroid.

Refined algorithm to allow more precise collision detection for the varying shapes of asteroids and ship.

Background scenery

Dynamic space environment consists of various non interactive entities. Stars, planets, and asteroids will move in the background to create the scene.

Controls



Player Ship

- Moves along x and y axis using left analog stick.
- Right analog stick controls ship rotation, faces analog stick direction.
- Has movement speed, acceleration, deceleration, and rotational velocity.
- Press A button to load rocket.
- Right bumper to fire bullet/rocket.

Menu Controls

- D-Pad Up/Down to scroll through menu options.
- A to select option

Cheat Features

Toggleable cheat features:

- Unlimited health
- Unlimited rockets
- Increase max number of bullets on screen

Assets

Game entities consist of non-animated and animated sprites, in the form of 2D PNG images.

Gameplay Assets

Player Ship: Animated Sprite, UFO with laser blaster on top. Image TBD.

Bullets: Sprite, fired from the Player Ship's blaster. Possibly particle trail. Image TBD.

Rockets: Sprite, fired from the Player Ship. Image TBD.

Asteroids: Rotating Sprite, 4 sizes, few different images but can be reused and scaled. Images TBD

Enemy UFO: Animated Sprite, can be same as Player UFO but different colour. Image TBD.

Enemy Bullet: Sprite, same as Player bullet.

Environmental

Background entities to add to the feel and look of the game. These are non-interactive and consist of stars, planets, and asteroids in the distance.

Text Font – TBD for the game menus and HUD.

Sound

Music

The game will be using basic audio. Possibly music for Menus but not for gameplay.

Sound Effects

Sound effects for the following:

- Explosions (collisions)
- Firing the blaster
- Firing the rocket
- Possibly for ship movement