Asteroids Technical Design Document



Asteroids is a classic arcade game invented in 1979 by Atari.

## Requirements

The task is to create a functional clone for the Windows 10 platform running at fixed frame rate of 60 fps. Sound can be ignored.

## Gameplay

The player controls a spaceship in a 2-dimensional asteroid field. The objective is to clear the screen of all asteroids by shooting them and amass a large a score as possible before dying. Each level starts with a few large asteroids drifting in random directions on the screen. When an asteroid is shot it splits into smaller, faster pieces. When the smallest pieces are shot, they disappear.

The player can rotate left and right and go forward, and fire bullets. If the ship is hit by an asteroid it explodes and the player loses a life. The player starts with 3 lives and earns extra lives by scoring points. The player earns points by shooting asteroids and enemies.

Enemy ships also periodically appear and fire bullets at the player’s ship.

Once a screen is cleared of asteroids and enemies a new level is started with more large asteroids appearing. As the level increases the number and speed of asteroids increases.

The game space wraps so if an object moves off the top of the screen it will reappear at the bottom.

The player can also use a warp ability that randomly moves the player to a vacant spot in space.

Engine

This game will be written in Rust and will use the Amethyst game engine. Rust is a systems level programming language that offers great performance and guarantees memory-safety and thread-safety. https://www.rust-lang.org/

I am choosing Rust for this exercise in order to assess its suitability as a games programming language.

<http://arewegameyet.com/> is a website promoting rust as a games language and provides links to resources for programming games. The top link in its game engines list is for Amethyst.

Amethyst ( <https://www.amethyst.rs> ) is a data driven engine written completely in Rust and makes use of an Entity Component System architecture. It is released under an opensource license: both MIT Licence https://github.com/amethyst/amethyst/blob/master/docs/LICENSE-MIT and Apache Licence 2.0

It is likely overkill for the asteroids task, but I am also considering this as a learning exercise so want to learn this engine and see how it works.

## Tools

Visual Code – lightweight Ide with rust support.

Paint.net

git

## Controls

Keyboard.

Up Arrow = accelerate

Left = Rotate anticlockwise

Right = Rotate clockwise

Space = Fire bullet

W = Warp

## Things to consider

Wrapping- Can we hide the pop when an object wraps from one side of the screen by rendering on both sides of the screen at the same time?

## Game Flow

**Loading State** – Initial state. Loads in all necessary assets and prepares game.

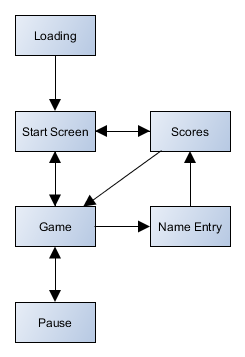
**Start Screen** – Displays game in attract mode and displays press start instructions

**Hi-score screen** – Displays the 10 highest scores. New entry should flash.

**Game** – State responsible for controlling the main game logic.

**Pause** – This state is pushed over the game state to pause it.

**Name entry screen** – If the player has a hi-score then they will be asked to enter their initials.



## Resources

Sprites- I will need to create some sprites for the ship, bullets, saucer and asteroids.

Player ship should have a thruster displaying when accelerating.

Animations for explosions

Different variations of asteroid for variety.

It might be nice not to have a completely black background but possibly this could obscure bullets and rocks flying around.

## Game Objects

**Playfield**

The game takes place in a 2d toroidal space. Moving off one edge brings an object to the other side. Objects move without friction and so keep moving at the same velocity.

**Player ship**.

The player ship can rotate left and right and accelerate forwards and fire bullets. There is no friction, so the ship keeps moving at the same speed and in the same direction unless the player accelerates. The ship will explode if it hits an enemy, enemy bullet or an asteroid. The player can only fire a limited number of bullets in one go.

**Asteroid**

Asteroids come in 3 sizes; large, medium and small. If a small asteroid is hit, then it is removed from world. The bigger sizes will split into 2 of the smaller ones each of which will have a faster speed. Asteroids do not collide with other asteroids but pass through each other.

**Bullet**

Bullets have a limited lifespan and will disappear when its lifetime runs out.

**Enemy**

Periodically a flying saucer will fly onto screen and fire at the player’s ship. As the level increases the accuracy and speed of the saucer also increases.

**Score**

The player’s score is displayed at the top of the screen above the playfield.

**Lives**

The player starts with 3 lives. These are displayed at the top of the screen above the playfield

**Level**

When the player clears the screen the level will increase with faster and more rocks.

**Highscore Table**

The ten best scores are recorded along with the player’s 3 initials. This is displayed on the highscore screen.

## Components

In addition to built in components the additional components are

Lifetime

Kill entity when lifetime runs out

GameTransform represent object’s position in space in game representation.

Pos

Velocity

Rotation

Radius

Rotation Speed

Game Systems

Systems operate on collections of components

Score System

This award’s points to the player

InputSystem

Update player ship’s direction and acceleration

Movement System

Update game objects->transform

Make sure that objects wrap.

Update vel

Vel = unit vector in angle direction

Pos += vel \* delta time

Collision System

Checks for collisions

## Possible Extensions

Audio

I have ignored audio for this exercise. Sound is an integral part of the asteroids experience and so this is the first thing that should be addressed in extending the game.

Configuration

Move game constants to external data files.

Power-ups

To make the gameplay more varied different power-ups could randomly drop from exploding asteroids. These could include

* Invulnerability – player can’t be killed for a short time.
* Smart bomb – Destroy all the small asteroids on screen.
* Stupid bomb – split all asteroids into small asteroids.
* Rapid fire – Player can fire more bullets more rapidly.
* Homing bullets – bullets seek out asteroids for a short time.

Multiplayer

Allow multiple players to control ships on screen at the same time.

Network play

We could extend the playing field to allow online co-operative or competitive play. I imagine it would end up looking something like xpilot. <http://www.xpilot.org/>