**Asteroids – Design Document**

# Introduction

Asteroids is a 2D game where you pilot a spaceship, avoiding and destroying asteroids that fly around the screen.

# Player

## Movement

The player has a Vector2 component named ‘p\_speed’ and a float component named ‘p\_rot’.

In order to move the player, using sine, we convert degrees to radians and multiply the players rotation by the returned value, then we multiply that new rotation value with the players speed on the X axis.

It is pretty much the same for the Y AXIS as well, however, instead using cosine rather than sine.

And then finally, we multiply the players speed on the x and y axis by the players acceleration (increased by the up-arrow key) and add the x axis values to the players position on the X AXIS and subtract the y values from the players position on the Y AXIS.

## Firing Projectiles

First, we check if the player has pressed (and released) the spacebar, and then we begin to check through the array of player projectiles (whether or not any have been fired).

If a projectile hasn’t already been fired, we spawn in a projectile, and set its position to be in front of the player, followed by activating the current projectile, then we move the projectile in the direction that the player is facing.

## Projectile Logic

If a projectile is currently active, we increase its (int) lifespan value while it is still active.

If a projectile’s position reaches the edge of the window (screen), or the lifespan reaches a value of 60, we deactivate that current projectile and reset its lifespan value to 0.

# Collision

## Player vs Asteroids

Using the ‘**CheckCollisionCircles**’ function from **raylib**, we check the current position of the player, and an asteroid, using a vector 2 collider for the player, and the asteroid’s position. If the players position reaches an active asteroid’s position + its radius, set the ‘**gameOver**’ bool to true to finally end the game.

## Projectile vs Asteroids

Using the ‘**CheckCollisionCircles**’ function from **raylib**, we check the current position of an active projectile + its radius, and the current position of an asteroid + its radius, and if they are colliding, we deactivate that projectile and set its lifespan to 0, then we deactivate that asteroid.

If the asteroid was a large asteroid, we spawn two medium asteroids where the large asteroid was and create a faded red circle where the asteroid was.

If the asteroid was a medium asteroid, we spawn two small asteroids where the medium asteroid was and create a faded green circle where the asteroid was.

If the asteroid was a small asteroid, we just deactivate the projectile (resetting its lifespan) and the asteroid, then create a faded yellow circle where the asteroid was.