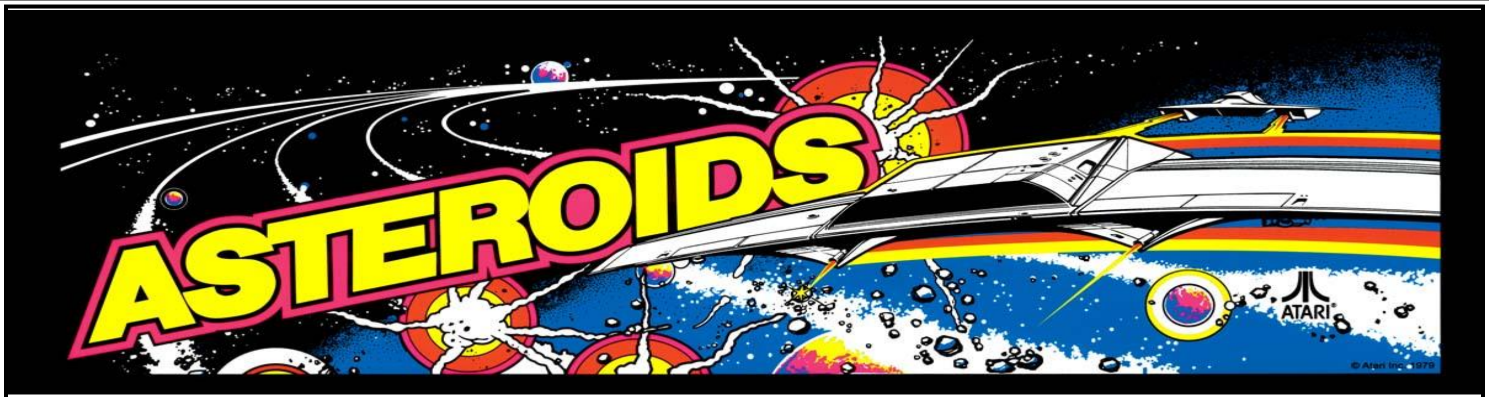


NAME	JUST VAN DER LINDE
CLASS	GDV2
DATE	10-15-2017
GAME	REVERSED ASTEROIDS
DOC	TECHNISCH DOCUMENT

THE GAME

Intro.
The game seems like the original Asteroids arcade game, but the originally player controlled spaceship is now controlled by artificial intelligence. While the spaceship is controlled by the AI, the player can instantiate asteroids and fire them at the AI controlled ship. The player only gets 1 minute of time, and gains points for hitting the ship with asteroids.

Why this design?
I chose this design because I really wanted to experiment with artificial intelligence, and this was a good excuse to try it.
I chose for the Asteroids game because it was one of my favorite games in my early years and I hold some good memories to it.
I didn't chose for boids because I want to safe that for my next experiment with artificial intelligence, and this method worked better for my current design/ideas.

How did the project change through the process?
I started off with huge expectations from myself, giving the AI accurate controls, making it fire projectiles at the asteroids and creating more different kinds of enemies.
I noticed that I have to limit myself if I want a project to be viable and realistic. That in mind I changed the concept and limited the mechanics, resulting in a fun game with possibilities to expand.

What would you do differently next time?
I would make more realistic terms so I cannot put a lot of time and energy in researching mechanics and ideas I'm not going to use anyway.
The process went very smoothly and I learned a lot of things, so there's not a lot I would change on the next project.
I would like to expand this project to match the concept I primarily thought of, but that's for a time when I'm not too busy grinding c++ homework.

PATTERNS

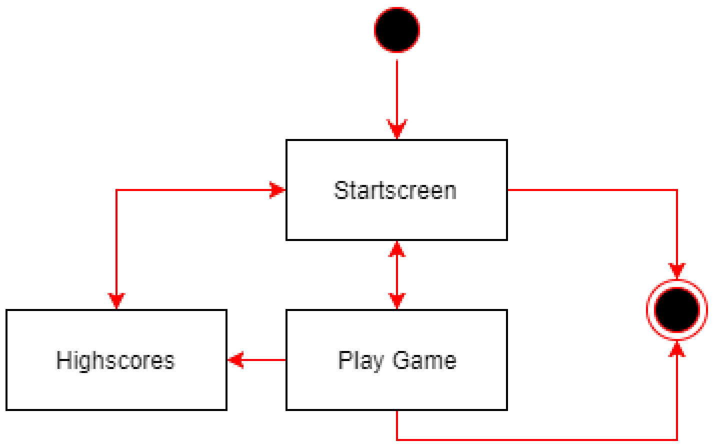
OBJECT POOL

Used for keeping a fixed amount of asteroid clones which can be spawned, preventing the constant instantiating and destroying of clones.

SINGLETON

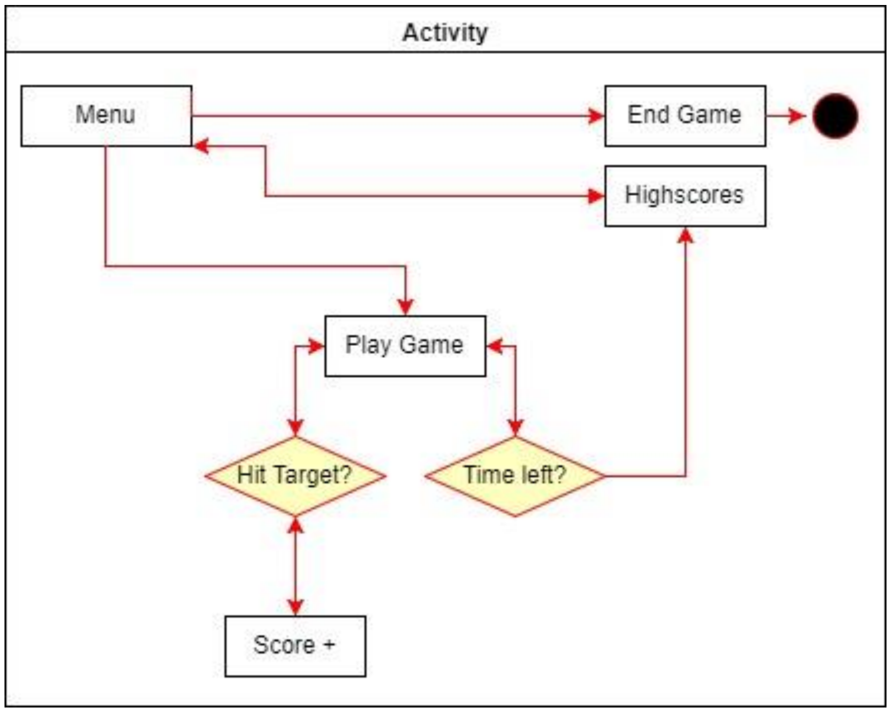
Since I’m using DontDestroyOnLoad() for my background music, I need to use a Singleton to prevent the music from duplicating.

SCENE FLOW DIAGRAM



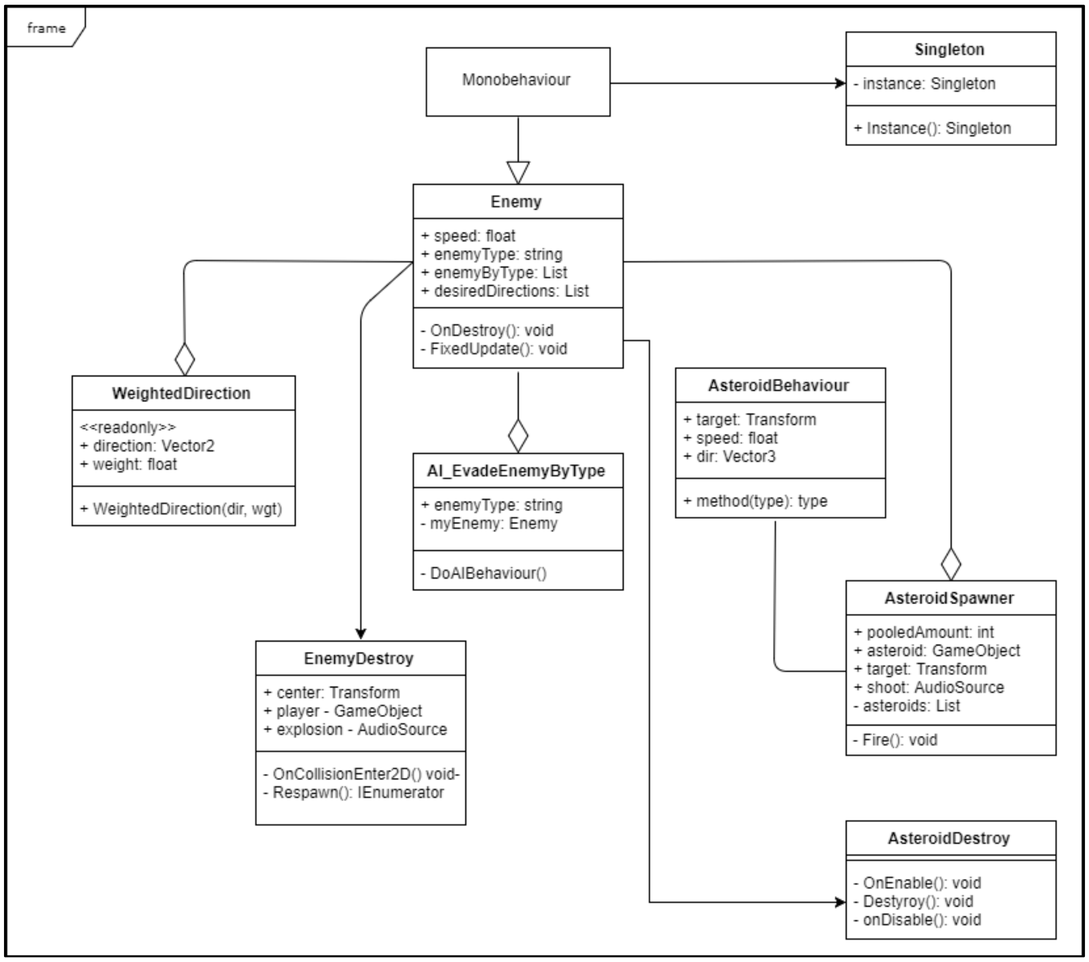
When starting the game, the player will enter a main menu screen. From there the player will have the options to visit the high scores, quit the game and to start the game.

ACTIVITY DIAGRAM



When the player entered the game, the countdown of 60 seconds start to count and the player has to score as many points as possible. Each time a successful hit is accomplished, a point will be added to the score.

CLASS DIAGRAM



The main element of the game is the class Enemy.
Every instance in the game is an enemy with an enemyByType and desiredDirections List.
The enemyByType list is used to check other enemies, and the desiredDirections is used to calculate the direction to go.
The direction to go is calculated in the WeightedDirection class by looping through the float weight and Vector2 direction. The asteroids spawned by the user get their behavior from the class AsteroidBehaviour

PLANNING

WEEK	GOAL	NOTES
1	Concept	
2	Player controls, start AI	
3	Finish AI, UI, Design patterns	
4	Fix and finalize	Process feedback in game
5	Submit	