Project Name: Galaga

Description:

I plan to make Galaga, a space based shooter. The player will directly control a spaceship, and face enemy spaceships in a series of rounds, culminating in a final round where the player will face a boss ship. The player wins if they manage to complete all rounds while making sure the number of lives they have left never falls to zero

Competitive Analysis:

To my knowledge, my version of Galaga will differ from most others in these ways:

1.Enemy ships dodge player fire

2.The levels will be randomized, but increasing in difficulty

3.The difficulty will scale based on the player's performance

While scouring the web for other similar projects, I came across many versions of Galaga. Each version was slightly different. Some had a few different ships, some only had lateral movement and firing, some were almost copies of the original Galaga. Those versions, and mine are similar in the core essence of the game. All of our versionshave the player's spaceships combatting enemy spacehships while trying to stay alive. However, this is where most of the similarities end.

During my research I never found a version of Galaga where enemies were able to dodge player fire or where difficulty scaled with progress and performance. So these features, which are included in my version of the game, will differentiate my project from theirs.

Algorithmic Plan:

In my view, the trickiest part of the project is scaling the difficulty based on the player's progress and performance, while at the same time maintaining an an aspect of randomization when generating a round.

Algorithm planned:

1. Each round, generate new enemies centred on random points of the board (This ensures randomness each round)
   1. The number of enemies increases as the rounds go by (To ensure difficulty always goes up)
2. Track the player's shots fired to hit ratio. Above a certain ratio, the game gets harder (Difficulty is scaled with performance)
   1. The game can made harder using thresholds. Above a certain ratio, enemies become more sensitive to player fire, dodging quicker.
   2. Above another threshold, the enemy's fire rate increases

Timeline:

18 Nov 2021 - Complete the dodging algorithm, test it on a singular case; Debug and clean up any previous code

20 Nov 2021 - Add ability to dodge even when in groups (for enemy ships); Create boss ship's program and test

22 Nov 2021 - Add in the difficulty scaling as described in the Algorithmic Plan

23 Nov 2021 - Bring everything together, touch up code, and add in small features (if wanted); Submit TP2

Version Control Plan:

Graphical user interface, application

Description automatically generated

Image 1: Copies of work are uploaded to and kept updated on Github and OneDrive

Modules Used:

Nil