Project Name: Galaga                               Author: Divyaansh Sinha; Andrew ID: divyaans

**TP3 Updates:**

Flocking behavior:

Enemies now follow a leader led flocking behavior. This was done by implementing a modified version of the boids algorithm. It not only accounts for maintaining cohesiveness, speed, and distancing between the boids (red enemies in my game), it also accounts for the heading of the flockLeader. This causes the rest of the enemies to follow the flockLeader (kind of like geese in a flock).

The flockLeader has truly random motion, so the flock travels in any general direction on screen. The random motion means that the flockLeader is a lot more adept at dodging, and this outcome of the natural passive interaction between the boids algorithm and the evasion function evadeShots. Killing the flockLeader will lead to a new enemy (as long as there is one) to be chosen as the flockLeader.

This modified boids algorithm is implemented using the changeDirection function and its sub-functions in timerFired. The algorithm was implemented using pseudo code and explanation from <http://www.vergenet.net/~conrad/boids/pseudocode.html>

Normal enemy changes:

The number of enemies was tripled. However, the number of lives each enemy had, was reduced from 3 to 1.

Only a certain number of enemies are designated firers at any given time, this is to prevent too many shots coming at the player at once

Enemies evade more intelligently. Essentially it is a slightly more intelligent dodge. If the laser approaching is left of the enemy's centre, it dodges a certain angle within a certain range to the right. If the laser approaching is right of the enemy's centre or at the centre, it dodges similarly to the left.

Boss fight:

Visualization issues have been fixed. Mechanics are easier to see and understand

Bomb explosion radius was increased from 50 to 100

You can now shoot the bomb, as it flies toward you, to detonate it

Explanation of new rules in boss fight:

1. Shooting the bomb when the boss is holding/charging it will cause it to explode. The boss’ shield will go down, or it will lose a life if the shield is not up.
2. Shooting the bomb while it is traveling towards you will cause it to explode at that spot, with the full radius. Anyone caught within will lose a life, or their shield, if they are shielded and the boss.
3. Rest of the rules are the same as described in the TP2 update. However, rules 1 and 2 directly above, supersede any similar rules in the TP2 update

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

Controls:

SPACE – Fire laser. There can only be 2 player shot lasers on screen at any one time

RIGHT – Move right

LEFT – Move left

‘c’ – When the round finishes, press ‘c’ to begin the next round (displayed on screen)

‘r’ – During a round, press ‘r’ to reset the round, carrying over hit ratio. Costs 1 life

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 versions have the player's spaceships combatting enemy space ships 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 centered on random points of the board (This ensures randomness each round)

a. 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)

a. The game can made harder using thresholds. Above a certain ratio, enemies become more sensitive to player fire, dodging quicker.

b. 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:

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

Modules Used:

Nil

TP2 Updates:

Design Updates:

No major design changes. No unforeseen problems or ideas. Just some annoying bugs and glitches.

Boss Fight Instructions:

The boss is shielded (represented by the light blue), and while shielded, being shot will not cause them to lose lives

The boss regularly shoots a bomb (red circle) towards the player

Once the bomb reaches the y-value of the center of the player square, it explodes (I am having issues displaying the bomb explosion as intended)

Whoever is caught in the blast radius will be affected. If shielded (for the boss), they will lose their shield temporarily, and if not, they lose a life

Another way to make the boss lose the shield, is to shoot them while they are holding the bomb. This deactivates their shield, and delays the firing of the bomb

The issues with the boss fight, as far as I can tell, are mostly display related. The mechanics seem to be working as intended

Runtime:

If the game is running too slow, try setting app.numStars (the number of stars on the canvas) in appStarted to a smaller number. Or set it to 0 and comment out the starProcesses function in timerFired.

TP3 Updates:

Flocking behavior:

Enemies now follow a leader led flocking behavior. This was done by implementing a modified version of the boids algorithm. It not only accounts for maintaining cohesiveness, speed, and distancing between the boids (red enemies in my game), it also accounts for the heading of the flockLeader. This causes the rest of the enemies to follow the flockLeader (kind of like geese in a flock).

The flockLeader has truly random motion, so the flock travels in any general direction on screen. The random motion means that the flockLeader is a lot more adept at dodging, and this outcome of the natural passive interaction between the boids algorithm and the evasion function evadeShots. Killing the flockLeader will lead to a new enemy (as long as there is one) to be chosen as the flockLeader.

This modified boids algorithm is implemented using the changeDirection function and its sub-functions in timerFired. The algorithm was implemented using pseudo code and explanation from <http://www.vergenet.net/~conrad/boids/pseudocode.html>

Normal enemy changes:

The number of enemies was tripled. However, the number of lives each enemy had, was reduced from 3 to 1.

Only a certain number of enemies are designated firers at any given time, this is to prevent too many shots coming at the player at once

Enemies evade more intelligently. Essentially it is a slightly more intelligent dodge. If the laser approaching is left of the enemy's centre, it dodges a certain angle within a certain range to the right. If the laser approaching is right of the enemy's centre or at the centre, it dodges similarly to the left.

Boss fight:

Visualization issues have been fixed. Mechanics are easier to see and understand

Bomb explosion radius was increased from 50 to 100

You can now shoot the bomb, as it flies toward you, to detonate it

Explanation of new rules in boss fight:

1. Shooting the bomb when the boss is holding/charging it will cause it to explode. The boss’ shield will go down, or it will lose a life if the shield is not up.
2. Shooting the bomb while it is traveling towards you will cause it to explode at that spot, with the full radius. Anyone caught within will lose a life, or their shield, if they are shielded and the boss.
3. Rest of the rules are the same as described in the TP2 update. However, rules 1 and 2 directly above, supersede any similar rules in the TP2 update