High Concept Statement

An idle game where the just start a game leave it alone for minutes and come back for rewards. A game which allows players to either to see the outcome of a war or only reap the rewards at the end.

Features:

- Start and stop a randomly generated simulation.
- A world full of rangers and hunters that seek to claim the hunting grounds as their own.
- 2 teams 1 player who picks a side who they think will win.
- Just press start and watch as these 2 factions battle it out or start it off and put your phone down and do what you want.
- Unlockable rewards that will allow players to show off to their friends.
- No need for the player to learn the game.
- Only 1 level but different each game.
- The hunters can only attack from up-close, but they deal significantly more damage than rangers
- The rangers attack from a distance giving them a significant advantage over melee units.
- The wilds have many dangers and in these hunting grounds the wolves are the most prominent dangers in these forests.

Overview:

Player Motivation:

This game offers players a hands-off experience where they can watch a game play by play or leave it, do other things and check back later for your rewards

Genre:

Gambling, simulator, fighting.

Target Customer:

Casual gamers who download games from the app store.

Licences:

None

Competition:

- It is not highly competitive.
- Player versus player.
- Player versus environment.

Unique Selling Point:

A game that plays for you where you earn rewards and you don’t even know it.

Target Hardware:

Smartphones.

Design Goals:

A low tension, highly addictive game that gives the player satisfaction with the choices they make.

Game Treatment:

High Concept Statement:

An idle game where the just start a game leave it alone for minutes and come back for rewards. A game which allows players to either to see the outcome of a war or only reap the rewards at the end.

You don't need to participate to have fun all you need to have fun is bragging rights so test your luck with random rewards and test how reliable your team is.

Genre:

A top down 2D RTS that rewards players depending on the choice they made with a realistic art style that attracts players. With its fully automated simulation the player can do whatever they want hands free and with no worry for real stakes.

Hooks

Barely any player input needed with the only player input required being to start or pause the program.

Promotes the players use of thinking and micromanagement of units to make their decisions.

Short games that are always different and requires no need to learn the controls.

Lots of collectible in-game items that promotes replay ability.

Licence:

No Licence required.

Online Highlights:

There are plans to take this game onto an online setting where players bet against each other

Gameplay Highlights:

Each game is different in each new game.

No learning required.

Better suited to new phone users or busy people.

Hardware:

iOs devices

Competition:

Logging Quest:

This game is another idle game where players click a start button and leave it alone until they come back the biggest benefits of this game is that it can be played during meals and provide entertainment in both cases our game also aims to do these things. Our game will be a direct competitor on the app store as our game offers the same value but appeals to a different fantasy which is where our markets diverge from each other our game also has a better visual atheistic as players can watch the simulation play by play while it occurs on their devices giving visual stimulus .

Characters:

Rangers – range warriors that are known to reside in forests and protectors of nature but if the situation calls for it they will shoot to kill and in turn spend their life seeking redemption. Top down view wearing a cape using greens uses a bow and arrow.

Hunters – melee warriors that seek what is best for their respective village these hunters fight in the name of village and seek to destroy those who hunt in what they consider their domain. Top down view wearing a hood of some kind uses a machete or a knife.

Wolves – neutral enemies who just seek to defend their territory and hunt. Larger than the hunter or ranger grey.

World:

A world where there is only the hunt where hunting grounds are the only source of food amongst many villages. These hunting grounds are competed against by each village sending out their best hunters and rangers to dominate and monopolise the hunting ground.

User Interface:

Start Button to both initialise or continue the scenario.

Pause Button gives the player the ability to pause the game.

Save Button saves the data to text files.

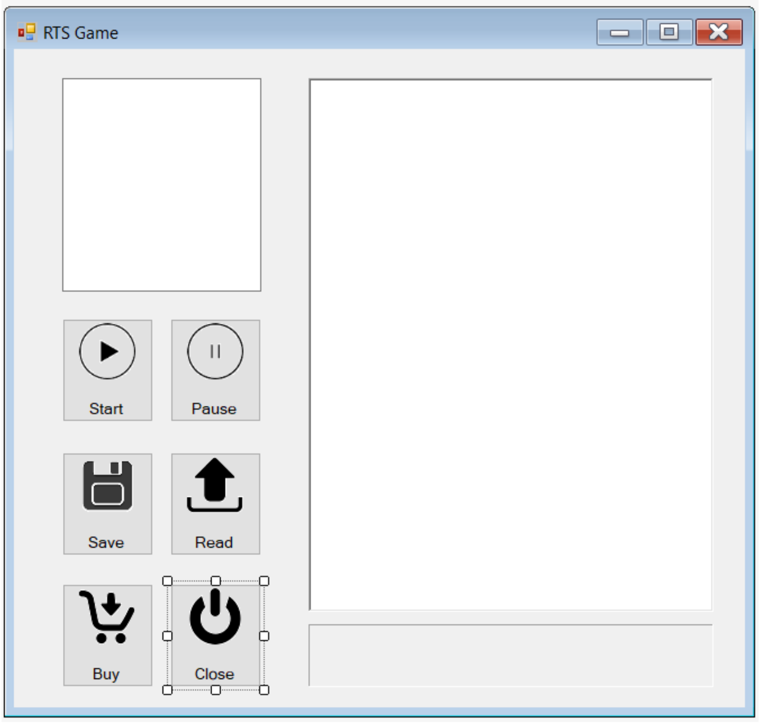
Read Button reads the data from text files.

Buy Button allows the player to extend the game further in intervals of 50 seconds.

Close Button exits the program after a prompt.

Mouse Click the player can tap on a unit to check its current status.

Timer that the player can use to track events.



Level Progression:

Units will gravitate towards the closest unit. When they get into attack range they will stop and start attacking. But if the unit they are attacking is not necessarily targeting that unit and can still move. When a unit is under 25% health they will move in a random direction. Towers deal damage to any enemy units in attack range that get too close to the spawn point of the factory building. The resource building generates money that can be used at a certain point in the game to regenerate health.

Game Script:

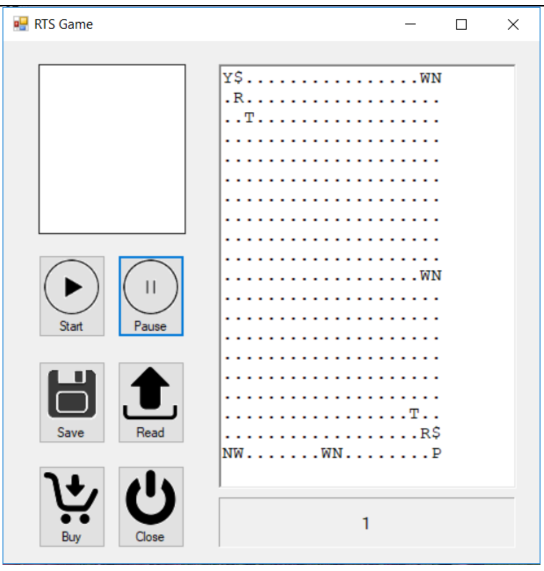
This is a sample from my Tower Class should you wish to see the rest of the code please refer to the github link below.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using System.Diagnostics;

namespace Munro17603375Task1
{
    class Tower : Building
    {
        const int DAMAGE = 10;
        private bool attack;
        private int range;
        public Tower(int x, int y, int health, string faction, string symbol, bool 
attack, int range)
        : base(x, y, health, faction, symbol)
        {
            this.attack = attack;
            this.range = range;
        }
        public Unit closestUnit(List<Unit> list)
        {
            Unit closest = null;
            int attackRangeX, attackRangeY;
            double shortestRange = 1000;
            double AOE;
            foreach (Unit u in list)
            {
                attackRangeX = Math.Abs(this.X - u.X);
                attackRangeY = Math.Abs(this.Y - u.Y);

                AOE = Math.Sqrt(Math.Pow(attackRangeX, 2) + Math.Pow(attackRangeY, 
2));
```

```
                if (attackRangeX < shortestRange && u.Faction != faction)
                {
                    shortestRange = attackRangeX;
                    closest = u;
                }
                if (attackRangeY < shortestRange && u.Faction != faction)
                {
                    shortestRange = attackRangeY;
                    closest = u;
                }
            }
            return closest;
        }
        public void combat(Unit enemy)
        {
            enemy.Health = enemy.Health - DAMAGE;
        }
        public bool attackRange(Unit enemy)
        {
            if (Math.Abs(this.X - enemy.X) <= this.range || (Math.Abs(this.Y -
    enemy.Y) <= this.range))
                return true;

            return false;
        }
        public override string toString()
        {
            string output = "";
            output += "x: " + X + Environment.NewLine;
            output += "y: " + Y + Environment.NewLine;
            output += "Health: " + Health + Environment.NewLine;
            output += "Faction: " + Faction + Environment.NewLine;
            output += "Symbol" + Symbol + Environment.NewLine;
            return output;
        }
        override public bool isDestroyed()
        {
            if (health <= 0)
                return false;
            else
                return true;
        }
        public override void save()
        {
            FileStream outFile = null;
            StreamWriter structure = null;
            try
            {
                outFile = new FileStream(@"Files\StructuresTower.txt",
    FileMode.Append, FileAccess.Write);
                structure = new StreamWriter(outFile);
                structure.WriteLine(X);
                structure.WriteLine(Y);
                structure.WriteLine(Health);
                structure.WriteLine(Faction);
                structure.WriteLine(Symbol);
                structure.WriteLine(attack);
                structure.WriteLine(range);
                structure.Close();
                outFile.Close();
            }
            catch (Exception e)
            {
                Debug.WriteLine(e);
            }
            finally
            {
                if (outFile != null)
                {
                    structure.Close();
                    outFile.Close();
                }
            }
        }
    }
}
```

Flowboard:

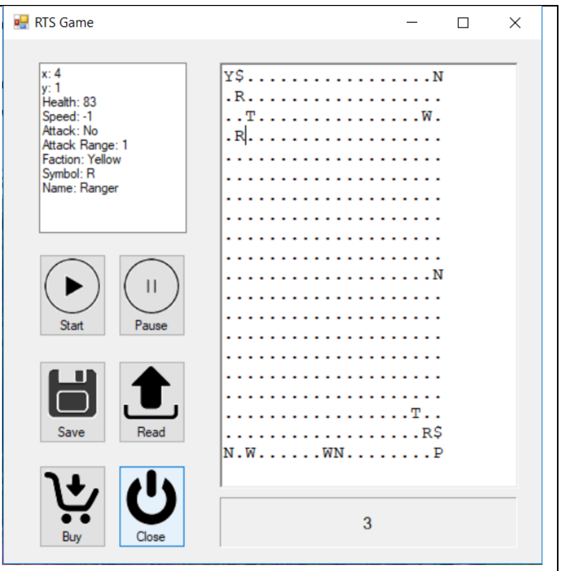
1.



```
Y$................WN
.R..................
..T.................
....................
....................
....................
....................
....................
....................
....................
..................WN
....................
....................
....................
....................
....................
....................
.................T..
..................R$
NW.......WN........P
```

```
1
```

2.

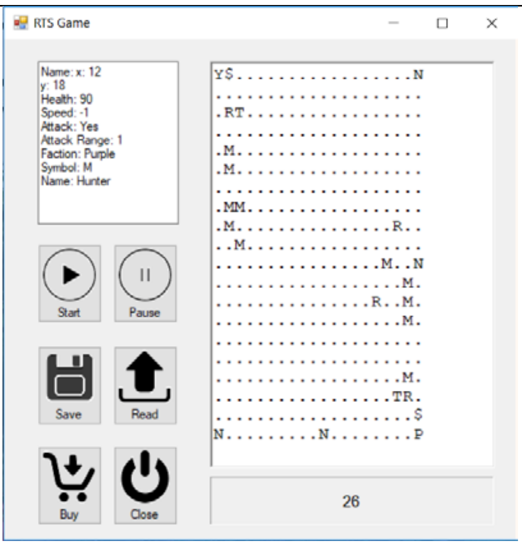


```
x: 4
y: 1
Health: 83
Speed: -1
Attack: No
Attack Range: 1
Faction: Yellow
Symbol: R
Name: Ranger
```

```
Y$.................N
.R..................
..T...............W.
.R..................
....................
....................
....................
....................
....................
....................
...................N
....................
....................
....................
....................
....................
....................
.................T..
..................R$
N.W......WN........P
```

```
3
```

3.

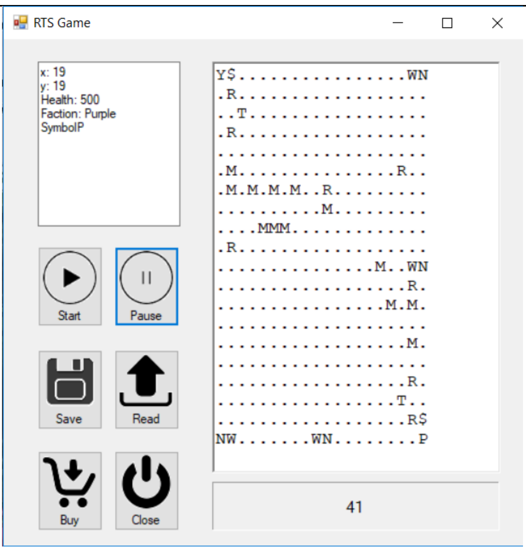


```
Name: x: 12
y: 18
Health: 90
Speed: -1
Attack: Yes
Attack Range: 1
Faction: Purple
Symbol: M
Name: Hunter
```

```
Y$.................N
....................
.RT.................
....................
.M..................
.M..................
....................
.MM.................
.M...............R..
..M.................
................M..N
..................M.
...............R..M.
..................M.
....................
....................
..................M.
.................TR.
...................$
N.........N........P
```

```
26
```

4.

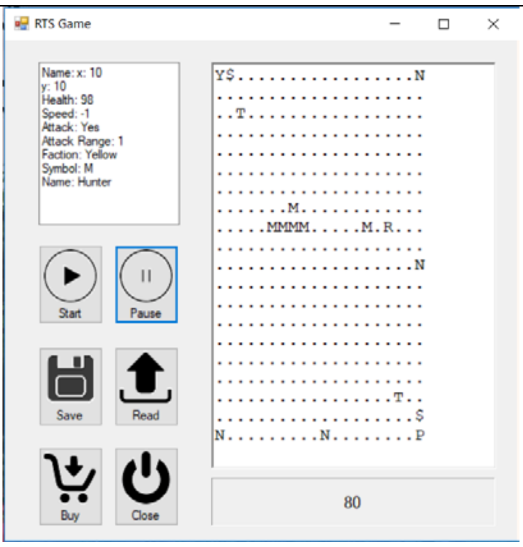


```
x: 19
y: 19
Health: 500
Faction: Purple
SymbolP
```

```
Y$................WN
.R..................
..T.................
.R..................
....................
.M...............R..
.M.M.M.M..R.........
..........M.........
....MMM.............
.R..................
...............M..WN
..................R.
................M.M.
....................
..................M.
....................
..................R.
.................T..
..................R$
NW.......WN........P
```

```
41
```

5.

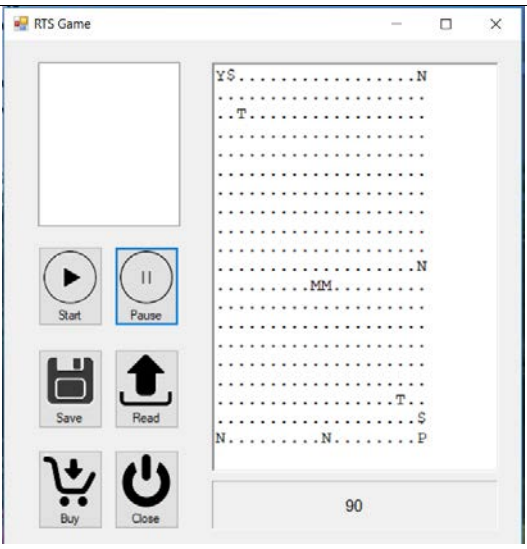


```
Name: x: 10
y: 10
Health: 98
Speed: -1
Attack: Yes
Attack Range: 1
Faction: Yellow
Symbol: M
Name: Hunter
```

```
Y$.................N
....................
..T.................
....................
....................
....................
....................
.......M............
.....MMMM.....M.R...
....................
...................N
....................
....................
....................
....................
....................
....................
.................T..
...................$
N.........N........P
```

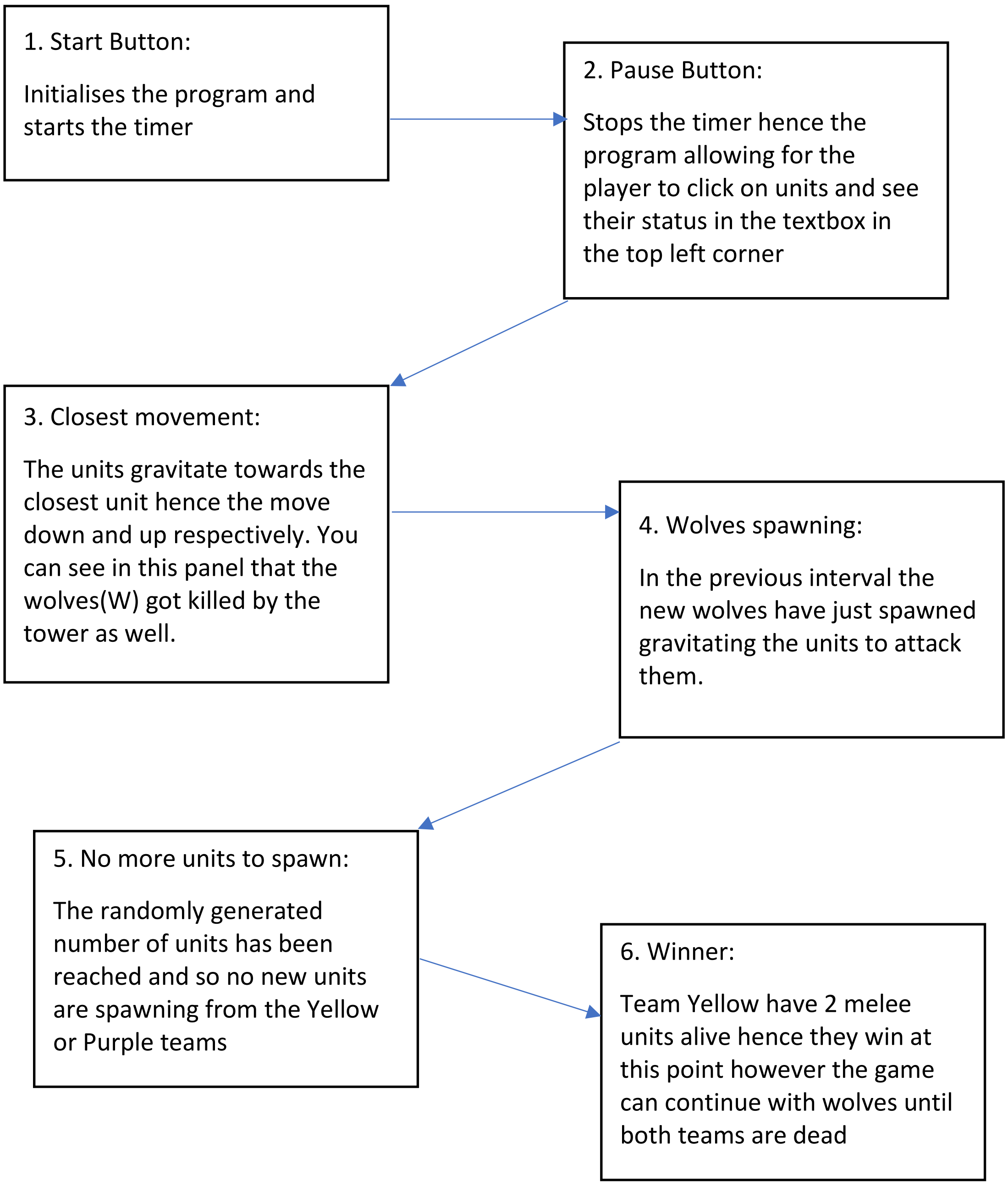
```
80
```

6.



```
Y$.................N
....................
..T.................
....................
....................
....................
....................
....................
....................
....................
...................N
.........MM.........
....................
....................
....................
....................
....................
.................T..
...................$
N.........N........P
```

```
90
```



1. Start Button:

Initialises the program and starts the timer

2. Pause Button:

Stops the timer hence the program allowing for the player to click on units and see their status in the textbox in the top left corner

3. Closest movement:

The units gravitate towards the closest unit hence the move down and up respectively. You can see in this panel that the wolves(W) got killed by the tower as well.

4. Wolves spawning:

In the previous interval the new wolves have just spawned gravitating the units to attack them.

5. No more units to spawn:

The randomly generated number of units has been reached and so no new units are spawning from the Yellow or Purple teams

6. Winner:

Team Yellow have 2 melee units alive hence they win at this point however the game can continue with wolves until both teams are dead

7. Optional Save or Read:

A player can press Save and Read from any point in the game.

8. Optional Buy:

The player can decide to press the buy button to increase both team health by 50 should they have enough resources from their building

Repository link:

https://github.com/SaibaChimera/GADE6112-POE