Building Survival Game Design Document

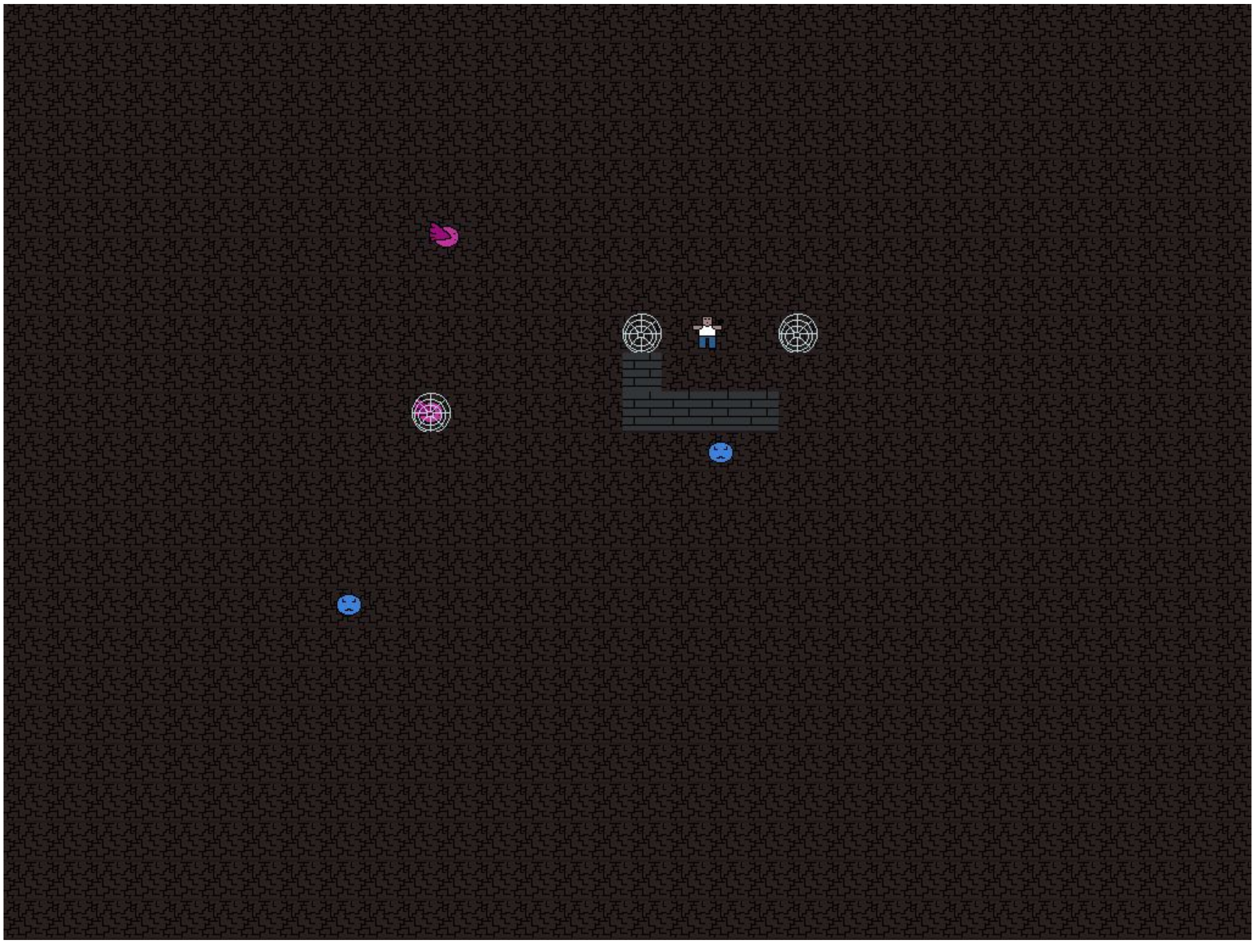
AIE Artificial Intelligence Assignment 2019
Emily Oliver

Introduction:

This project is an endless survival game where the objective is for the player to avoid monsters for as long as possible. The player has the ability to build several objects which will impede the movement of the monsters, with varying efficacy depending on the type of monster. As soon as a monster touches the player, the game will end.

This game takes inspiration from tower defense and survival games, with the objective being to create a simple project that captures the essence of these genres.

Gameplay:



Overview:

The player character will start in a central point of a 32*24 tile map. One monster will also spawn at a random corner of the map. At given intervals, additional monsters will spawn.

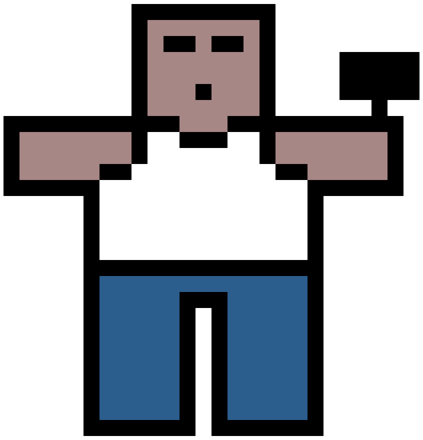
Monsters will wander around the map until coming close to the player, where they will change to pursue behaviour. If the monsters reach the player, it will result in a game over.

The player can avoid monsters by placing down walls and traps. Walls prevent ground monsters from passing and traps freeze monsters in place for an amount of time, whilst also giving the player an opportunity to have the monster lose sight of them.

After the game is over, the player will receive a score based on how long they were able to survive.

Agents:

Player:

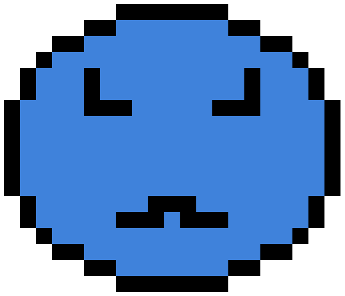


The player can move freely around using the WASD or arrow keys on the keyboard. Pressing Q or E will place down an object in front of the player, locked to one of the tiles in the map. This will incur a cooldown that will prevent the player from placing the same object down again.

Monsters:

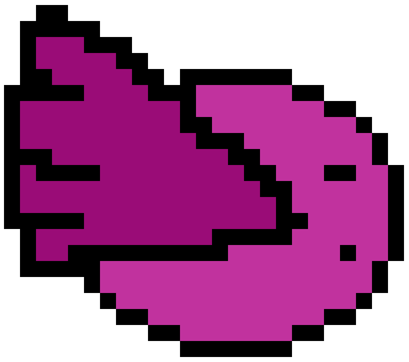
Monsters will spawn at set times during the game. They will wander around until getting close enough to the player, where they will switch to pursue behaviour. If they are caught in a trap while in pursue and lose sight of the player, they will switch to search behaviour once they are freed, where they will either switch back to pursue or wander behaviour depending on whether they find the player.

Gel:



Basic enemy type. This monster will switch to pursue behaviour once it gets within a certain distance to the player and has line-of-sight.

Aeroplane Gel:

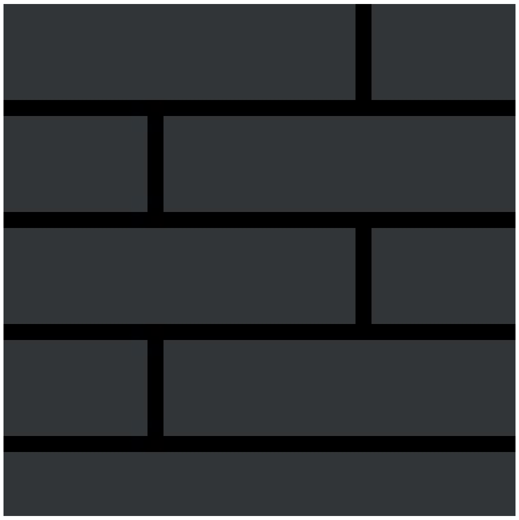


Ignores walls, but spawns less frequently than the first monster and switches to pursue behaviour at a closer distance.

Objects:

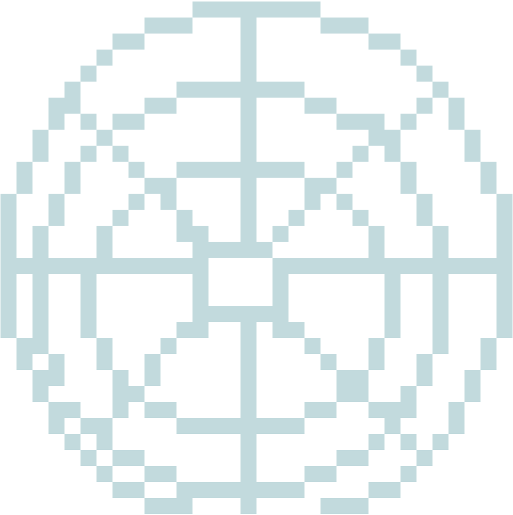
There will be two objects in the game, a wall and a trap. Both can be placed by the player on a 32 * 24 grid. Only one object can be on a tile at a time.

Wall:



Will prevent both the player and the first monster from passing. However, has no effect on the second monster. Incurs a 5 second cooldown when placed.

Trap:



Locks the monster in place for 5 seconds.Incurs a 10 second cooldown when placed. If the monster was previously in pursue behaviour and it has lost sight of the player, it will switch to search behaviour once it is free.
After the monster is freed from the trap, the trap will disappear.

Technical Information:

Will be coded in C++ using Visual Studio 2017

AI Mechanics:

Keyboard Controls:

The player will be able to move with the arrow keys or WASD keys, as well as place objects down on the tile in front of them with Q and E.

Wander:

This is the default behaviour for enemies. They will wander between randomly determined waypoints. They will need to use collision detection and slimes will need to use pathfinding during this.

Pursue:

Enemies will switch to pursue behaviour if they are close enough to the player and can “see” them. Fliers’ sight isn’t impeded by objects, so they will always switch to pursuit behaviour while in range. Slimes, however, will need to use a raycast to check if any walls are blocking line-of sight.

While pursuing the player, the enemy will try to catch them, taking their momentum into account. Slimes will need to use pathfinding to avoid walls.

Search:

Enemies will switch to search behaviour if caught in a trap whilst in pursue behaviour and without . They will head to the position the player was in when they were caught in the trap, then hold that position for a few seconds before switching to wander behaviour. If the enemy finds the player, they will instead switch to pursue behaviour.

Collision Detection:

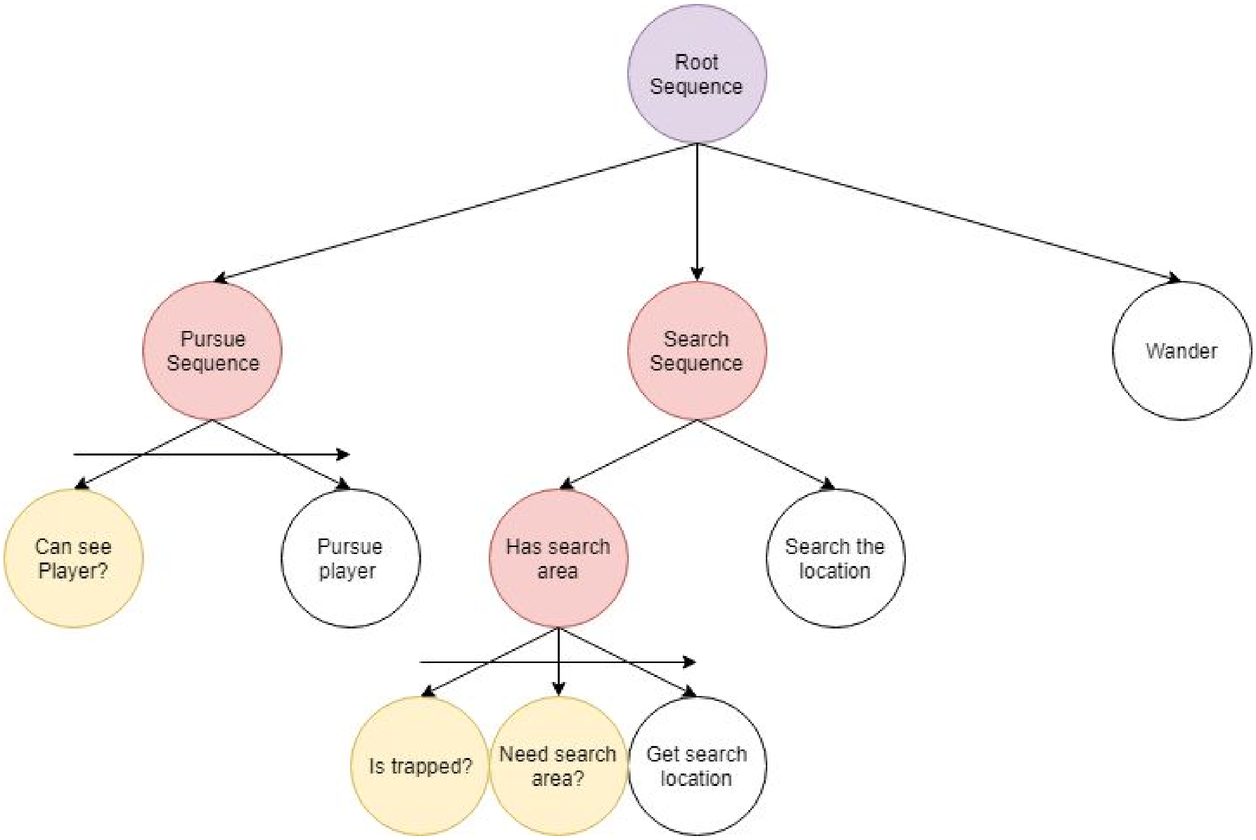
All actors will need rectangle-to-rectangle collision detection. More complex collision detection will not be necessary due to the grid-based nature of the game. Slimes and players will need to check collisions with walls, whilst both enemies will need to check collisions with traps and the player.

Pathfinding:

Slimes will need pathfinding to avoid walls. A* pathfinding will be used due to its simplicity.

Enemy Behavioural Tree:

Emily Oliver - Introduction to AI Design Document


Root Sequence
Pursue Sequence
Search Sequence
Wander
Can see Player?
Pursue player
Has search area
Search the location
Is trapped?
Need search area?
Get search location