Sprint 1 Requirements Artifacts

Team Number: 12

Team Members: Kai Achen, Andrew McFerrin, John Newman,

Aiden Patel, Landon Pyko

Ordered by Requirement ID

1: Choose language/game engine:

Game engine: Godot

Languages used: GDScript and C#

2: Learn language/game engine (if necessary):

- Download game engine and set up environment with GitHub.
- Decide on what skills need to be learned to create the game.
- Break down skills needed by who needs to do what.
- Have people learn the skills needed for the requirements and potentially teach other members later if it would be quicker than them learning on their own.
- Create coding conventions for the new language that can be followed throughout the software
- Learn how to execute the code and modify the running conditions depending on the platform if performance capabilities vary.

3: Choose game format (2D, 3D, top-down, etc):

- Graphics presentation of game is decided
- One of [2d, 3d, top down, isometric, first person, third person]
- Clearly labeled design choice
- All members have input to the format

4: Decide Game rules (lives, rounds, turn-based or live, etc):

Lives:

- A player gets a certain amount of lives
- Each time a player gets his, they lose a life

- Once the player is out of lives, they lose
- If the player wanted a “hardcore” mode, they could start with one life

Rounds:

- A player’s victory could be determined by how many rounds they won
- Once they are out of lives, they lose one round
- Once they have lost a predetermined number of rounds, they lose the game

Stages:

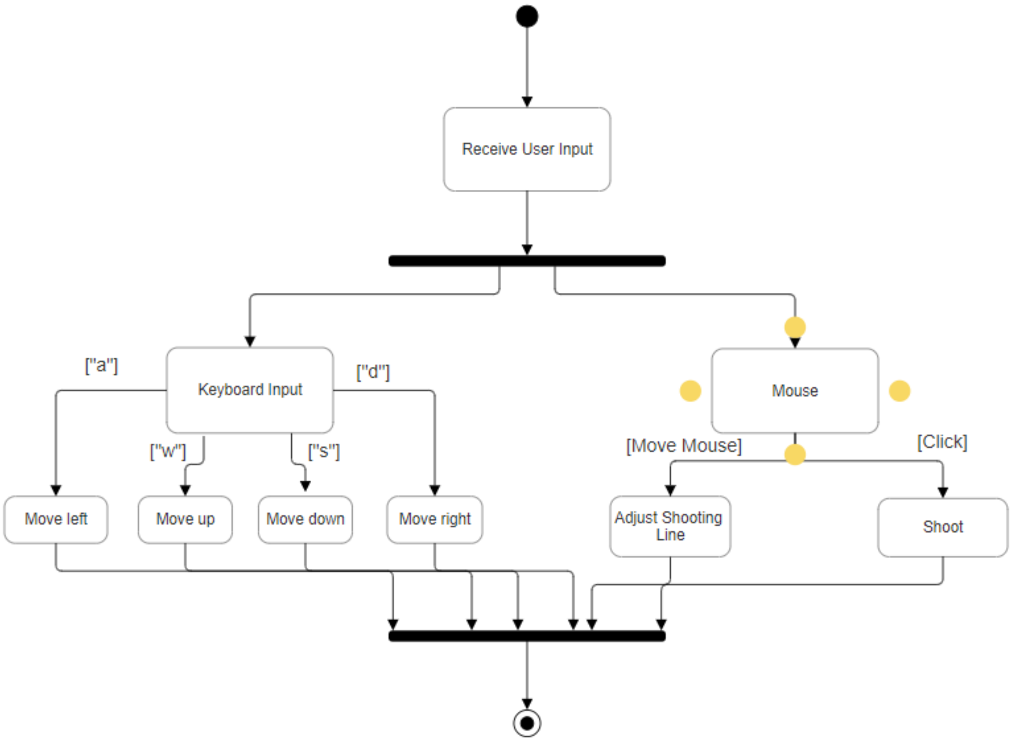
- There will be multiple stages, each presenting some sort of challenge for the player
- Each stage will look different, and have different enemy tanks

Tanks:

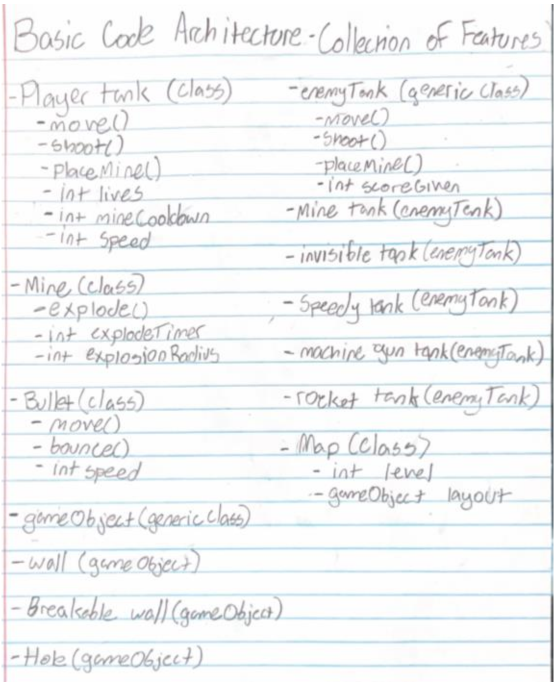
- Different tanks will have different abilities, which can be discerned through the color of the tank
- One tank will have a normal shooting function
- One tank will place mines
- One tank will shoot fast

7: Player functions (move, shoot, etc):

UML Activity Diagram



10: Basic code architecture (classes and inheritance structure):



Basic Code Architecture - Collection of Features

- Player tank (Class)
  - move()
  - shoot()
  - PlaceMine()
  - int lives
  - int mineCooldown
  - int Speed
- Mine (class)
  - explode()
  - int explodeTimer
  - int explosionRadius
- Bullet (class)
  - move()
  - bounce()
  - int speed
- gameObject (generic class)
- wall (game Object)
- Breakable wall (gameObject)
- Hole (gameObject)
- enemyTank (generic Class)
  - move()
  - Shoot()
  - placeMine()
  - int scoreGiven
- Mine tank (enemyTank)
- invisible tank (enemyTank)
- Speedy tank (enemyTank)
- machine gun tank (enemyTank)
- rocket tank (enemyTank)
- Map (Class)
  - int level
  - gameObject layout

11: Bullets and Mines:

Collection of Features:

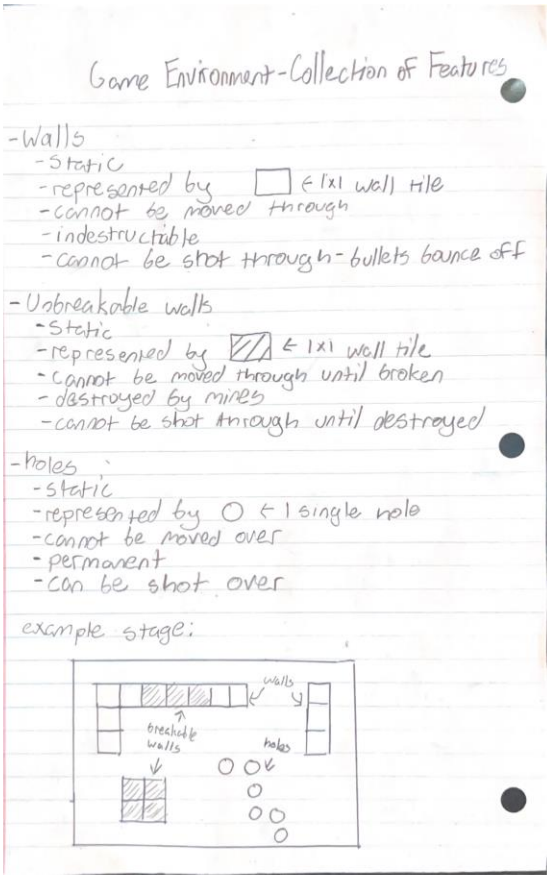
Bullets:

- Bullets are shot by tanks in the direction of the cursor (the direction of the barrel of the gun on the tank).
- If a bullet connects with an enemy tank, the enemy tank will be destroyed.
- If the bullets connect with a wall, it will reflect off the wall.
    - The angle of reflection will be based off the angle of incidence in the same way that a billiard ball would bounce off the wall of a billiards table.
- Players can shoot a maximum of five bullets at a time.
    - That is, five bullets from a single player can be on the screen at any given time. If a player tries to fire a bullet when five are already on screen, it will not fire anything.
    - Once a bullet disappears (connects with an enemy tank or a wall for a second time), the player can once again fire another bullet.

Mines:

- Mines are placed by tanks on the ground beneath the tank.
- Mines are not armed until a small timer has run out.
    - This helps prevent tanks from placing mines and immediately exploding.
- Different tanks can place a different number of mines.
    - For example, the player may place at maximum two mines at a time.
- Mines will detonate under any of the following conditions:
    - An enemy tank touches the mine.
    - A bullet touches the mine.
    - The timer on the mine runs out.
- If a mine explosion touches a tank, the tank will be destroyed.

<u>12: Game Environment:</u>

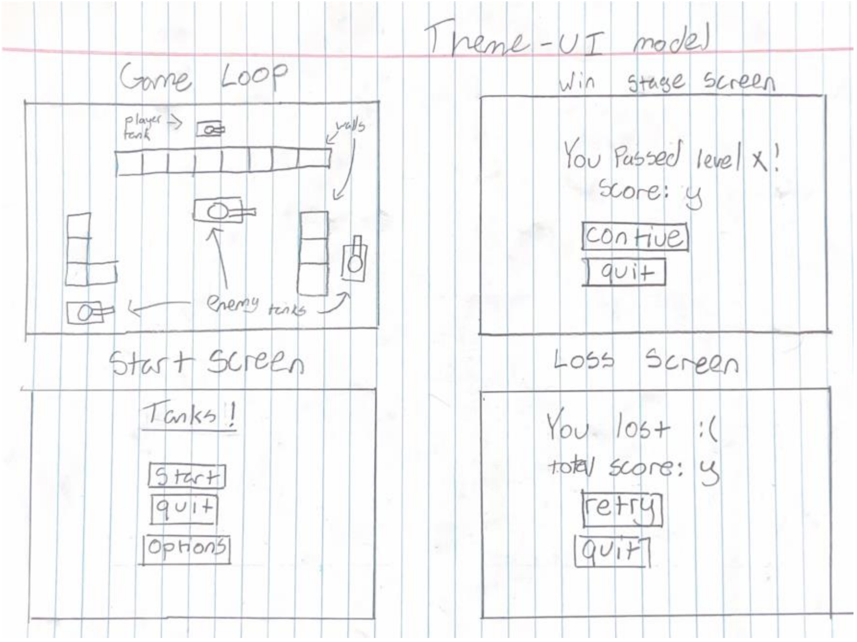


Game Environment-Collection of Features

- Walls
  - Static
  - represented by □ ← 1x1 wall tile
  - cannot be moved through
  - indestructible
  - cannot be shot through - bullets bounce off
- Unbreakable walls
  - Static
  - represented by ▨ ← 1x1 wall tile
  - cannot be moved through until broken
  - destroyed by mines
  - cannot be shot through until destroyed
- holes
  - static
  - represented by ○ ← 1 single hole
  - cannot be moved over
  - permanent
  - can be shot over

example stage:

21: Decide on theme:


Theme - UI model
Game Loop
player tank
walls
enemy tanks
Win stage screen
You Passed level x!
Score: y
contiue
quit
Start Screen
Tanks!
Start
quit
Options
Loss Screen
You lost :(
total score: y
retry
quit