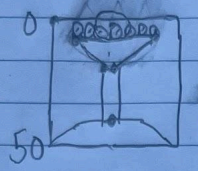
Assignment 3

Creating Models for the Pieces (Queen)

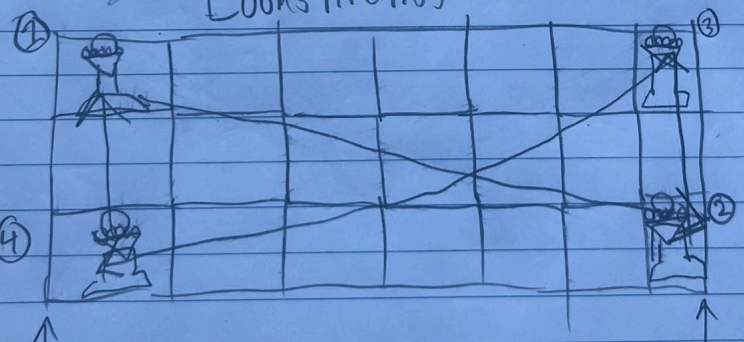


Head → Draw.Circle
→ Draw.Triangle
Crown → Draw.Circle 7 times (Smaller than Previous)
Neck → Draw.Rectangle
Base → Draw.Arc

Queen Movement

- ☑ Goes diagonally to one horizontal edge to another
- ☑ While on the edge, it goes up until it reaches its maximum height
- ☑ once at its maximum height, does this diagonal movement

Looks like this



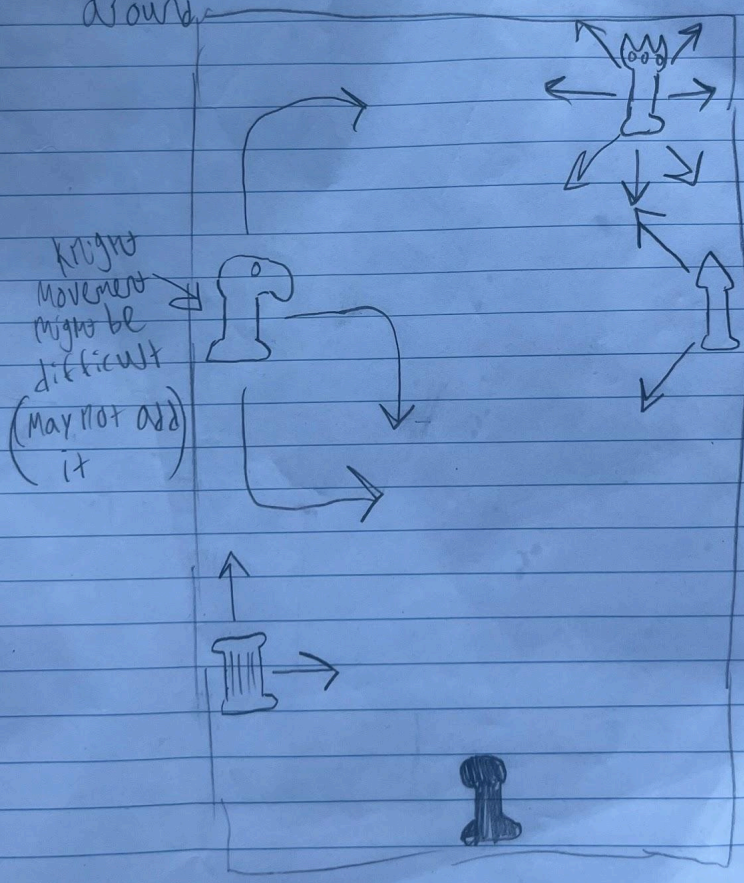
Bool rightedge true

Bool leftedge false

Add how be to track when the Queen needed to move on the edge

Assignment 3

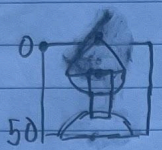
- Frogger clone
- Chess themed
- Pawn that needs to get to the other side to get promoted
  - Can only move forward
  - Pressing right or left with move you go in a diagonal direction
- Player needs to avoid enemy pieces as they're moving around



The pieces will move in a set pattern so that the player can strategize on the best possible move

Assignment 3

Creating Models for the Pieces (Bishop)



Head → DrawTriangle = (PositionBishop + NewVector2(X,Y), PositionBishop + NewVector2(X,Y), PositionBishop +

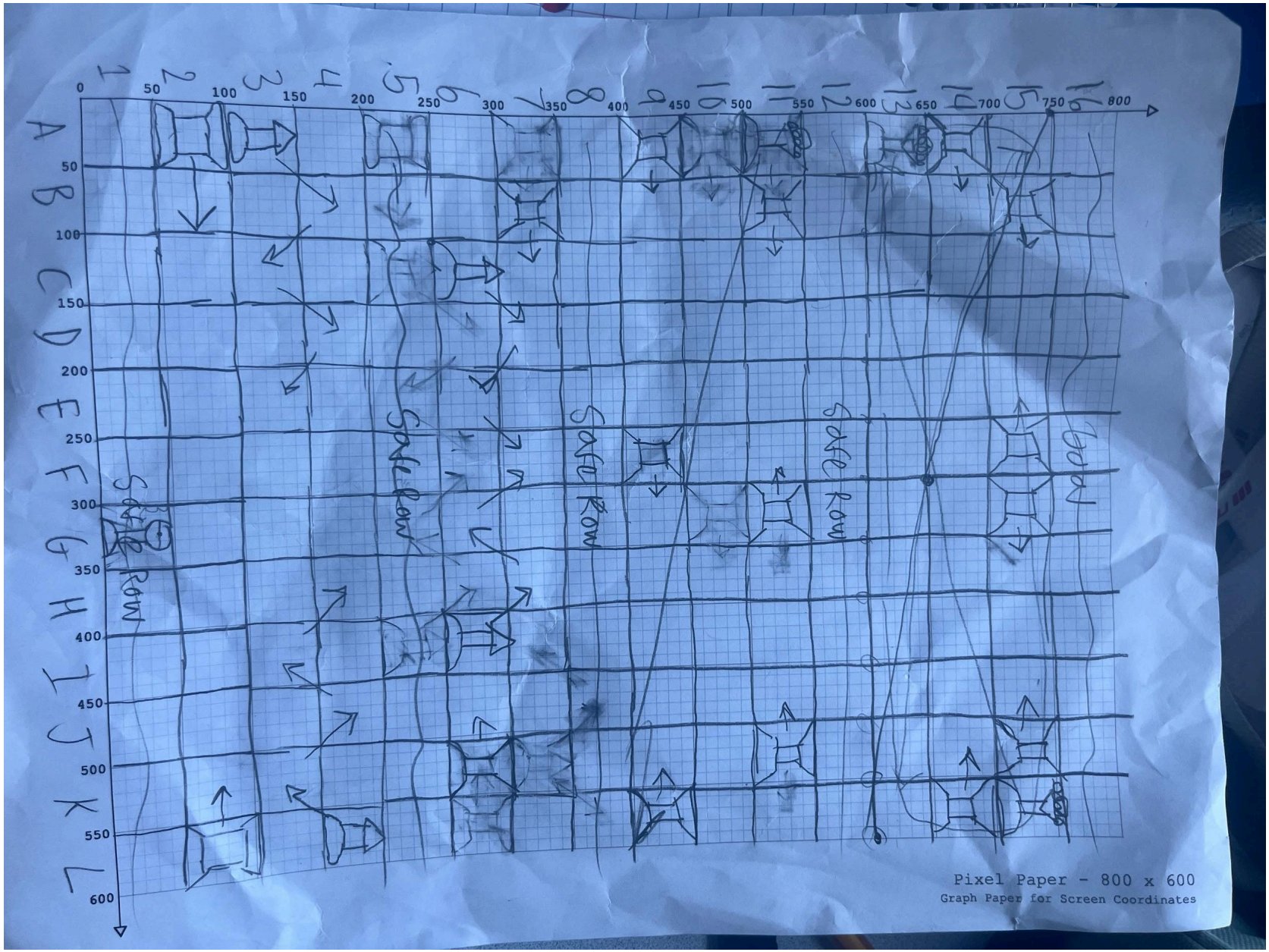
↓ Draw.Arc = ~~

Base → Draw.Arc = ~~

Neck → draw.rectange = ~~

Collision Detection

- [x] Add bool to tell if Player was touched by a Piece
- [x] ~~Add~~ Positions to make a hit box
- [x] Find a way to get those positions to the Pieces
  - Class
  - ~~use return Variable as Public float (Drone the Code)~~
  - ~~use Vector2 return Variable as compare it directly to the PlayPosition~~ (can turn return Variable into Vector2 as X and Y floats)
  - [ ] Seperate position of Rook


Safe Row
Safe Row
Safe Row
Safe Row
Goal
Pixel Paper - 800 x 600
Graph Paper for Screen Coordinates

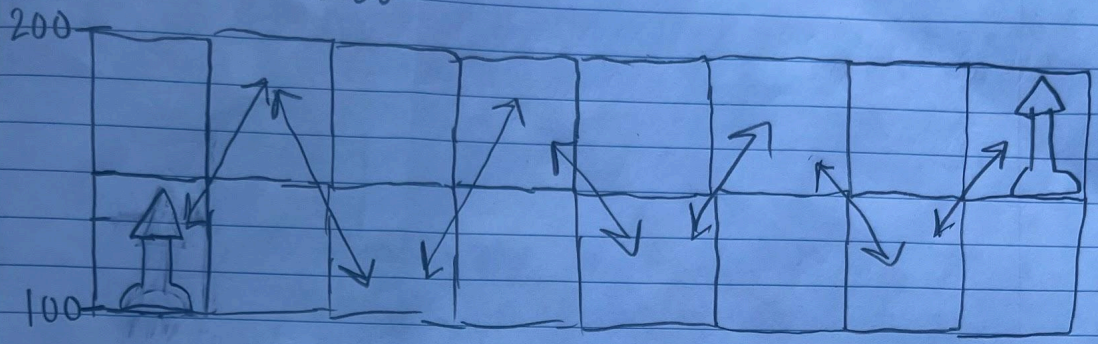
Assignment 3

Piece Design

- Create 3-4 unique enemy pieces (Knight is a stretch goal)
  - [x] Rook (Moves Left and right)
  - [x] Bishop (Moves in a diagonal Patten from Left to right)
  - [x] Queen (Moves Like a Bishop and a rook but fast in a 3x12 square box)
  - [ ] Knight (Moves in an L-shape)
- create hitboxes to cause the Player to die if touched
  - [x] Player (Pawn)
  - [x] Rook
  - [x] Bishop
  - [x] Queen
  - [ ] Knight
- Make the actual tokens
  - [x] Player (Pawn) [circle top, rectangle body and semisphere bottom]
  - [x] Rook [rectangle body and trapazoid bottom and top]
  - [x] Bishop [semisphere Bottom, triangle top and rectangle body]
  - [x] Queen [Inside down triangle plus tiny circle for head, rectangle body and semicircle bottom]
  - [ ] Knight [

Assignment 3

Bishop movement



- ✓ A Bool to determine if they're touching the left side (similar to Rook's code)
- ✓ A Bool to determine whether or not they're touching the top or bottom
- ✓ 2 Floats to determine the Max and minimum Y the bishop can go
- ✓ Construct code to switch seemlessly

Assignment 3

Creating models for the Pieces (Pawn)

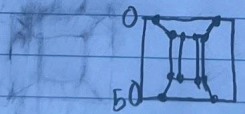
Already the position is tracked using a square, so using it as a base should work well

To create head → Draw.Circle(PositionPlayer + newVector2(x,y), radius);

To create neck → Draw.Rectangle(PositionPlayer + newVector2(x,y), newVector2(w,h));

To create Base → Draw.Arc(PositionPlayer + newVector2(x,y), newVector2(size), 0, -180);

Creating models for the Pieces (Rook)



To create Base (arc to Past) → Draw.Quad(PositionRook + newVector2(x,y), PositionRook + newVector2(x,y), PositionRook + newVector2(x,y), PositionRook + newVector2(x,y));

To create Neck (Do twice for detail) Draw.Rectangle(PositionRook + newVector2(x,y), newVector2(w,h))