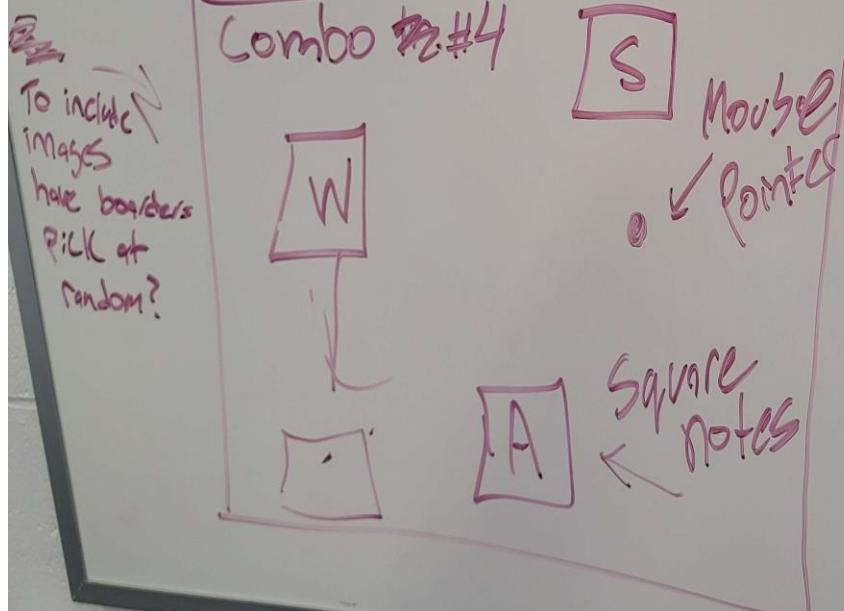


## IN CLASS BRAINSTORMING

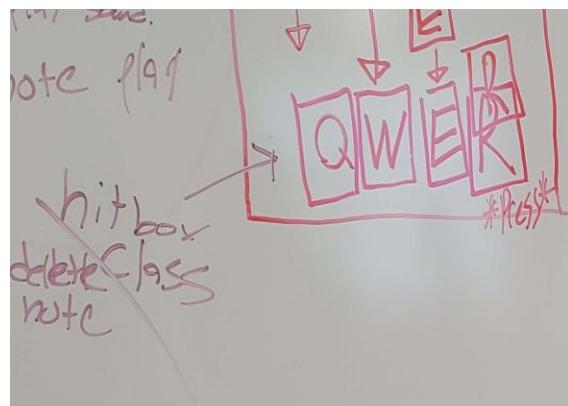
- If note hits bot' screen delete - pass  
Rhythm Game ntc

track Notes pressed correctly in a row - Square



To include images have borders pick at random?

- Maybe che
- Make nice
- = After 3 s



row - Square on screen for 3 seconds? - How to do this?

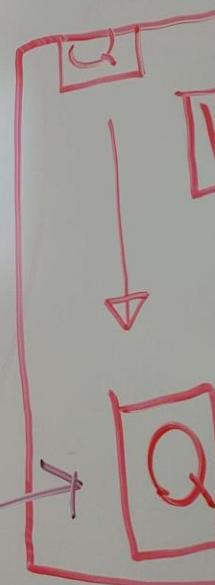
- Maybe check if the user is pressing the correct key or use mouse to click square
- Make nice notes play when key is pressed correctly?
- After 3 Seconds make note disappear and play ~~note~~ music or a bad noise

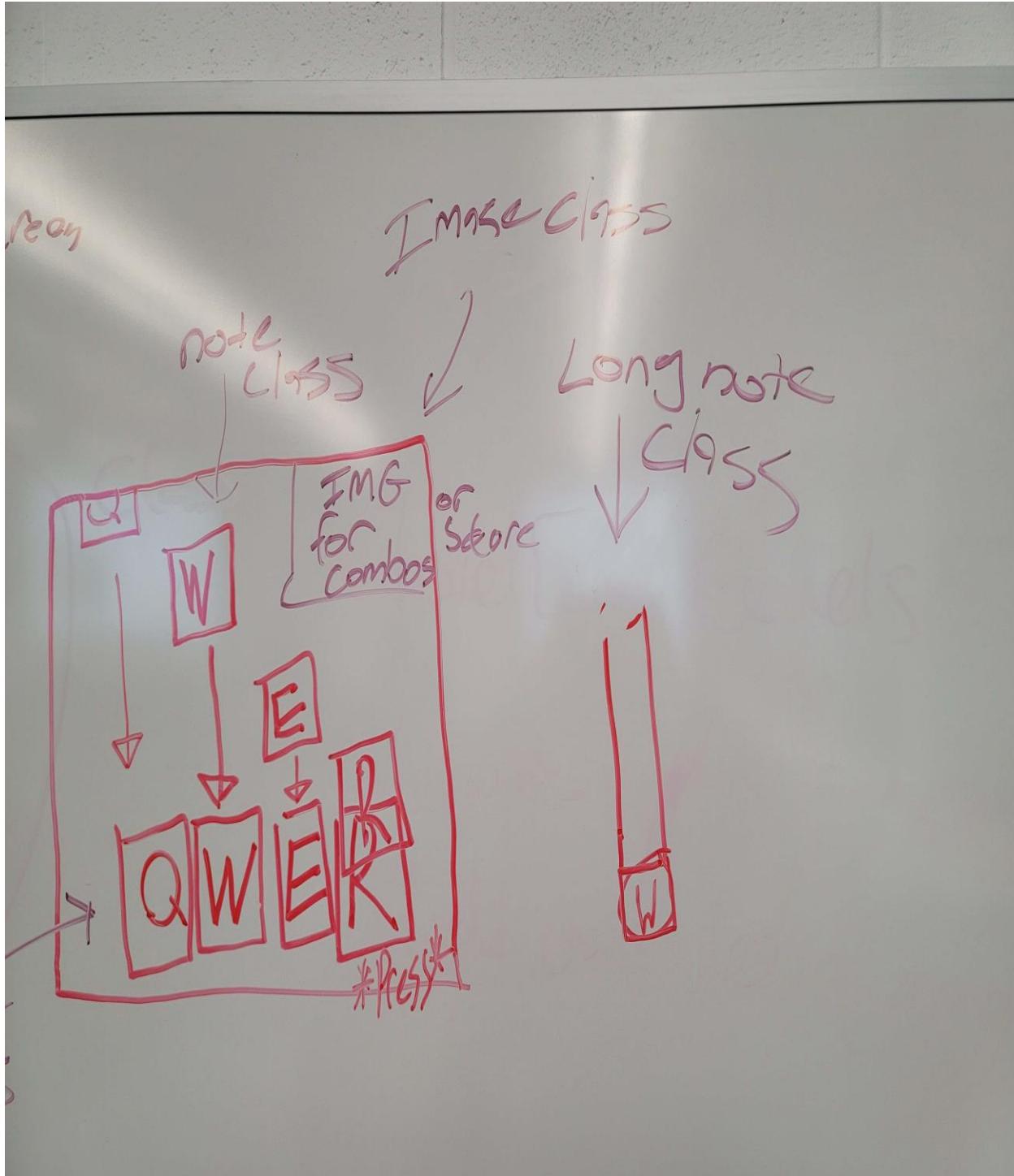
- Scale to Small?

## Rhythm Game #2

- Squares drop from the top of the screen
- Each key has text of "Q, W, E, R"
- At bottom of squares hit boxes/bottoms with text in boxes "Q, W, E, R"
- When the dropping square hits the hitbox at the bottom play at note.
- If player hits note correctly play sound.
- If player doesn't hit note play bad sound.

~~Rhythm Game~~ ~~Screen~~ ~~hit box~~ ~~note~~ ~~delete class~~  
~~track Notes Pressed~~





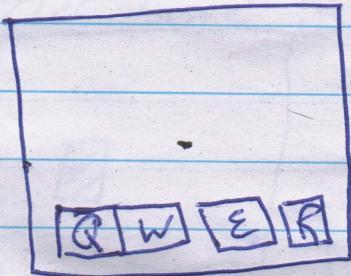
Picked Game #2

# [Rhythm Game]

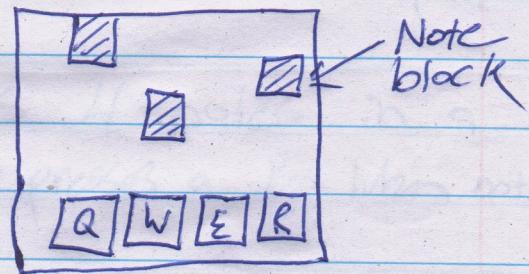
- Group assignment will be a Rhythm game/ piano tiles clone

## [Rules]

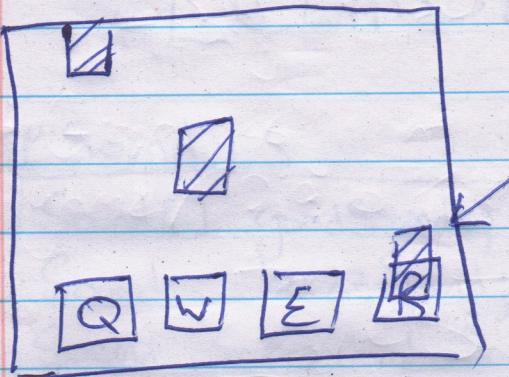
- 4 Blocks at bottom of the Screen titled 'Q' 'W' 'E' 'R'



- Note Blocks will come down from top of the Screen.

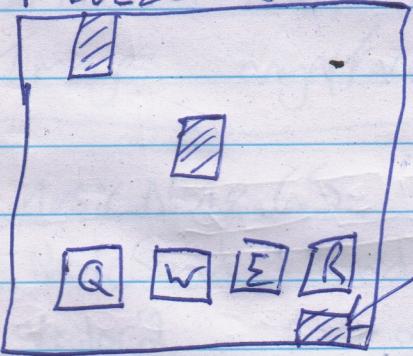


- If User presses the Correct Key  
When the note collides with ~~the~~ the corresponding block at the bottom of the Screen a sound will play & score will increase



User presses 'R' key  
When blocks collide.  
Music note sound plays.

- If player misses the music note and it hits the bottom of the screen a buzzer sound is played



Music note hits bottom of the screen.  
buzzer sound is played.

- If player misses 4 notes in a row game is over or presses a key when note music note is colliding.
- 50 random notes are in a level. Game ends after the 50 notes ~~are~~ are played.
- If player hits 5 music notes correctly in a row the score will increase x2 when next notes are played correctly.

## [Group parts]

Setup GitHub Repo : Justin  
& Invite members

NoteSound Class : Keaton

GameImages Class : Dylan.M

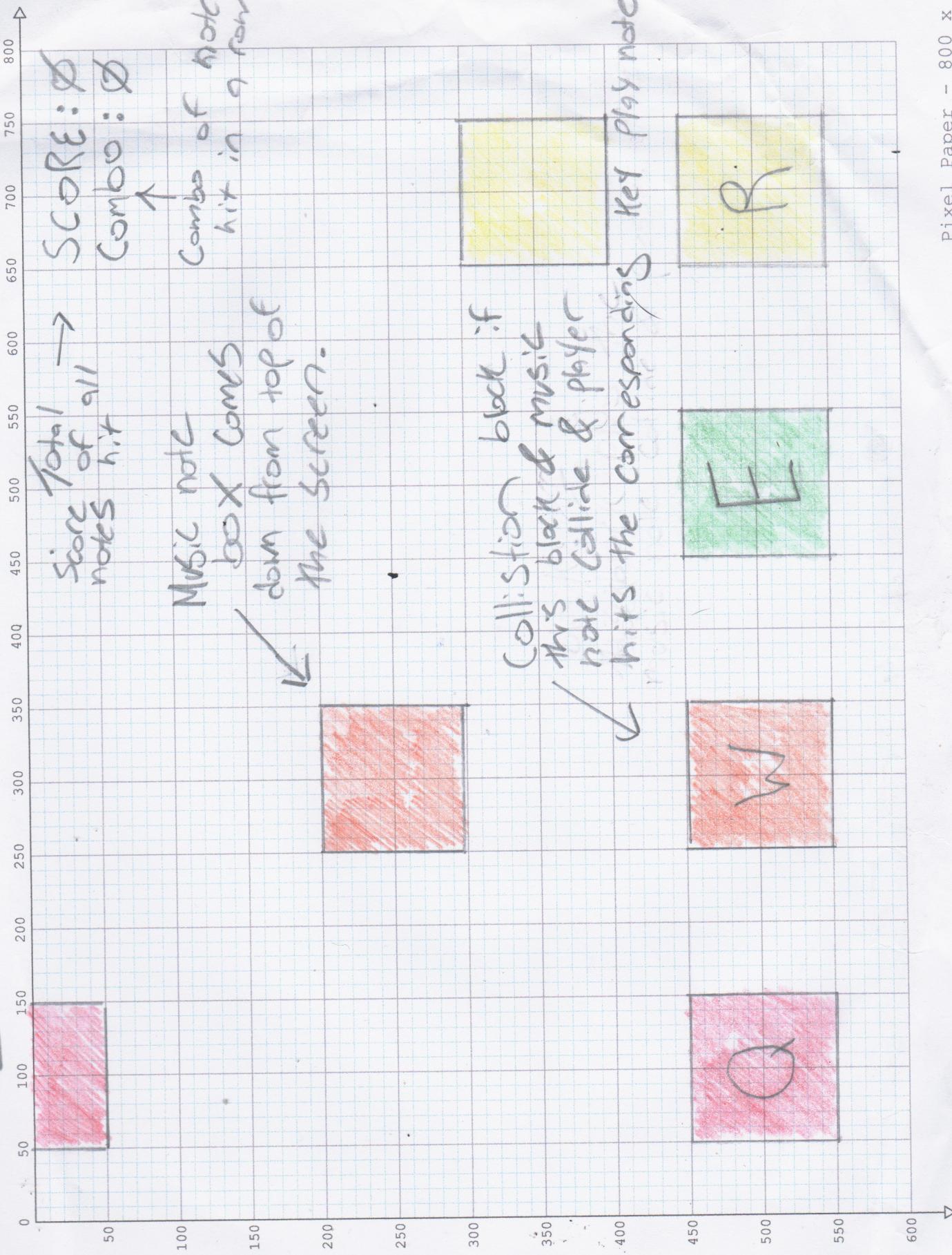
CollisionBlock Class : Ahseb

MusicNote Class : Justin

Main program : ?

Work on it together

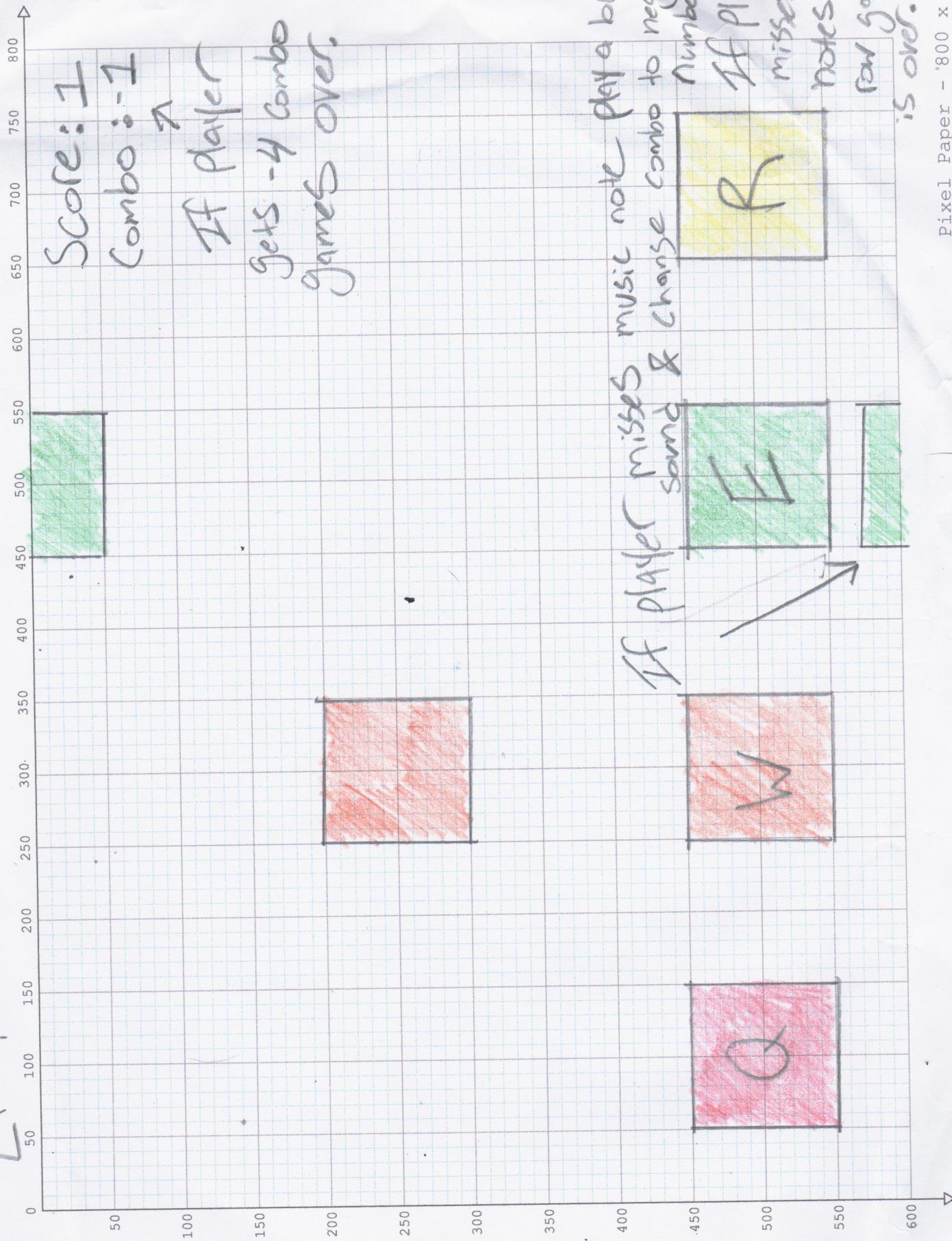
# Music Notes Falling Down



# Player Hits Music Note



# Player Misses Music Note



# [CLASSES]

## NoteSound.cs

```
Sound[] goodSounds;  
Sound buzzSound;  
Sound comboSound;
```

## GameImages.cs

```
Image happyFace;  
Image neutralFace;  
Image angryFace;
```

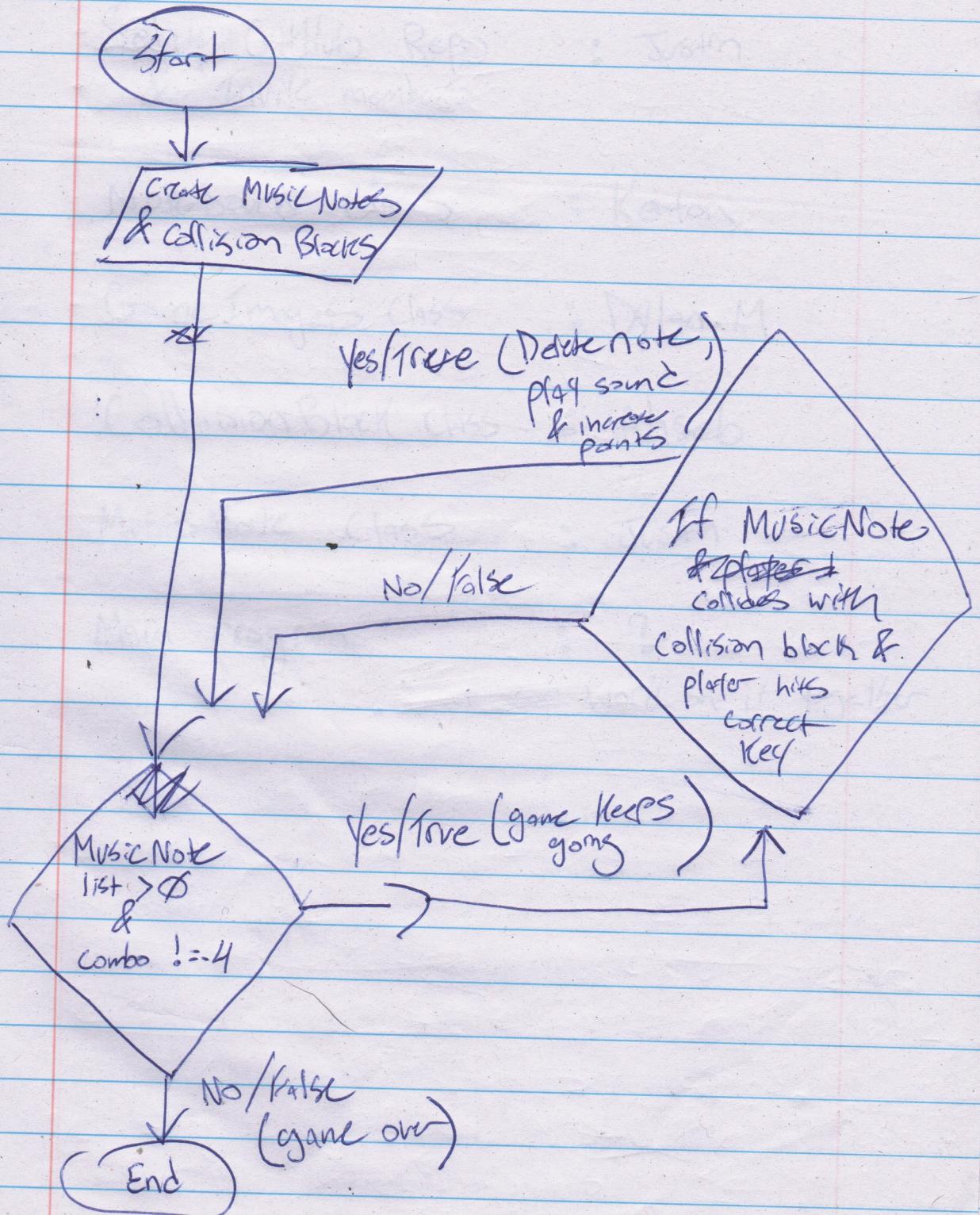
## MusicNote.cs

```
NoteSound sound;  
Vector2 position;  
Vector2 size;  
Color color;  
int points;
```

## CollisionBlock.cs

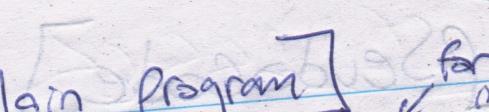
```
Vector2 position;  
Vector2 size;  
Color color;
```

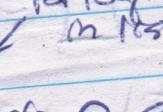
# Flow Diagram

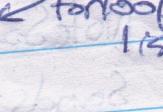


## [Pseudocode]

- Add NoteSound class.
- Import Sounds to NoteSound class.
- Add GameImages class
- Import images to GameImages class.
- Add CollisionBlock class.
- Create constructor that takes int block.  
Draw function. Switch case to create pre 'Q' 'W' 'E' 'R' blocks  
Values based on int block value  
Ex. If the int block value = 1. Create block at  
 $X = 5\phi$ ,  $Y = 45\phi$ , Color = Red
- Create function called "CollideWithNote"
- Input is MusicNote object. Output is bool for if  
a note has collided. ~~Set Value~~
- Add MusicNote class.
- Create constructor ~~MusicNote~~
- Create Random class object
- Randomly create note object from 1 to 4 with  
Random class object Rn in MusicNote constructor.  
Switch case in MusicNote constructor. If value = 1  
Set ~~R~~ X, Y, Color, points, NoteSound.
- Create deleteNote function. Play sound, & return notes points
- Create Draw, & Move functions.
- Create IsPostBottomSide function. Return true or false bool

[In Main program]  farloop store  
in list

• Set up 4 Collision blocks  farloop store in list

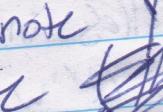
• Set up 50 Music notes  farloop store in list

\* [In update function]

• Move music first music note

• when : • Move first music note

• Loop list of notes

• If 1 note  is below 300 more  
2nd note ~~the~~ (only have 2 notes on board)

\* Check if note has collided with  
Collision block or hit the bottom of  
Screen

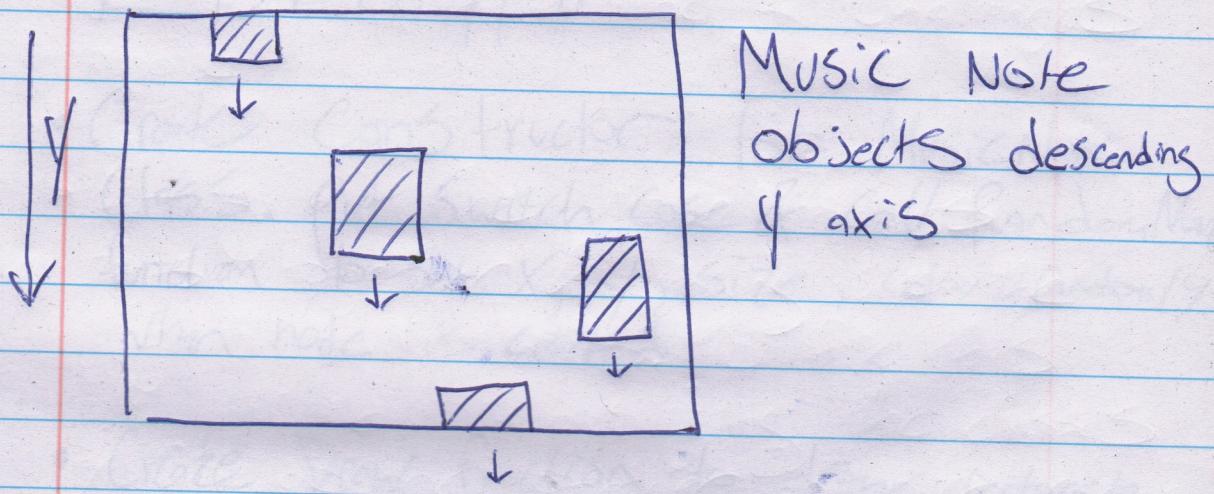
• Update Score & combo depending if play  
hit note or missed note

• When Note List is empty or combo = -4  
end game

• Display score and highest combo

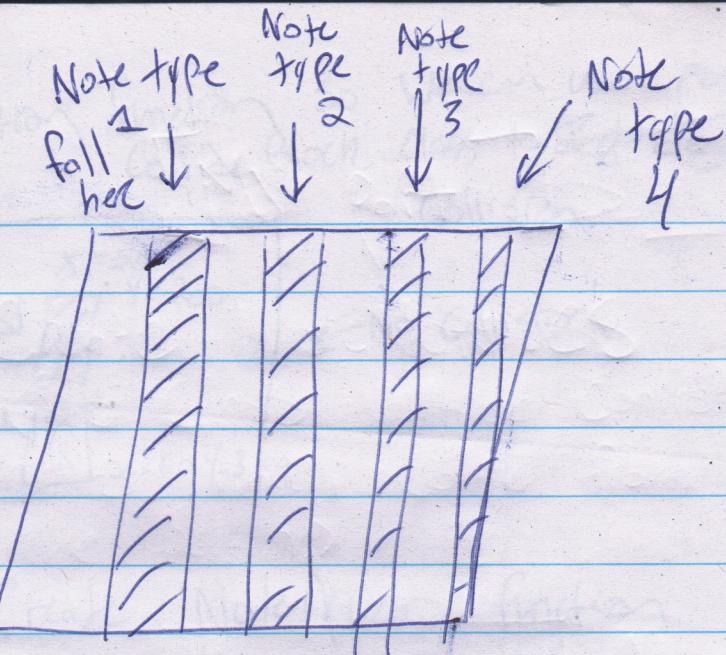
## [MY Part]

- Create MusicNote.cs
- (This Class will be controlling the notes which come from the top of the Screen.)

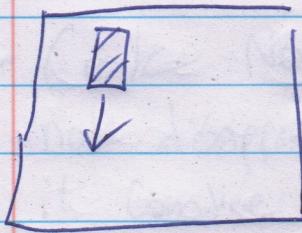


MUSIC Note  
objects descending  
Y axis

- Will need attributes for speed, position, size, colour, pointValue, and sound for each note.
- Create a RandomNumber function to randomly generate number from 1 to 4 to decide what X position the note is. (Can be easily change for more notes)

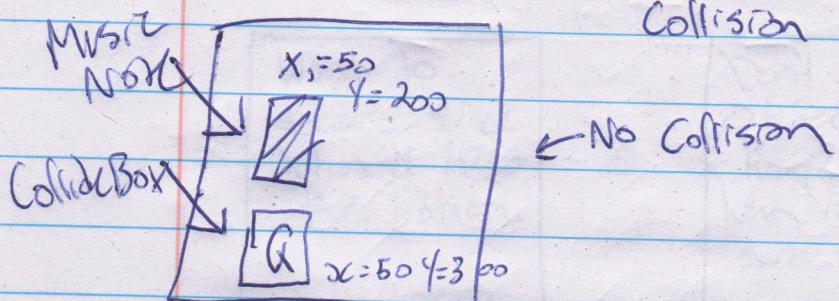


- Create Constructor for MusicNote Class. Put switch case & call RandomNote function to set x, y, size, colour randomly when note is created.
- Create Draw function to draw rectangle to screen.
- Create Move function to make 3 rectangle drop from top of screen

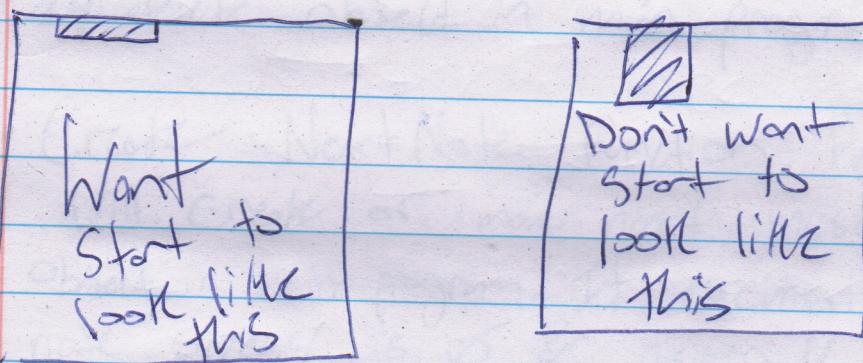


- Create GetPosition function so CollideBlock class can compare the Music Note object position to its own

Get Position function so we can use position in  
Collide Block class to detect collision



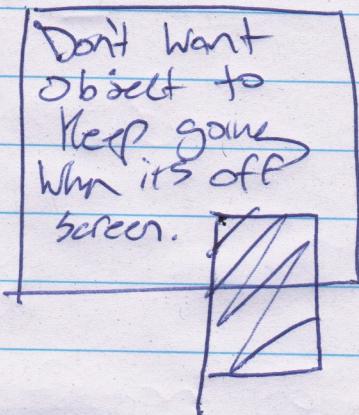
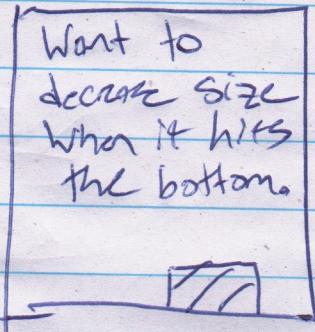
- Create NoteAppear function to make note look smooth when appearing from top of screen



Will need to increase music note size when its  $\text{position}$  is less than 100 as that's the size of the music note y.

- Create NoteDisappear function to make note disappear off screen instead of just letting it continue off screen

Boards  
↓



When the objects size.y == 0 & it hits bottom of the screen we will del delete object in main program.

- Create NextNote function. Returns bool  
Will create or move next music note object in main program. If current music note size.y is 0 & position.y  $\geq$  565 means object is off screen so return true  
So we can create next object.

A Music Note

## [Challenges]

- Using GitHub. Creating the repo.
- ~~forgetting~~ Remembering the steps to download the updated project & uploading changes
- Had to watch the class Demo for GitHub Setup. ~~so~~ for the 2 challenges above
- Putting all the group members parts together & making sure the program still works. To fix this we worked on designing the classes & pseudocode together in voice comms to try and make sure game would work before coding & wasting time.