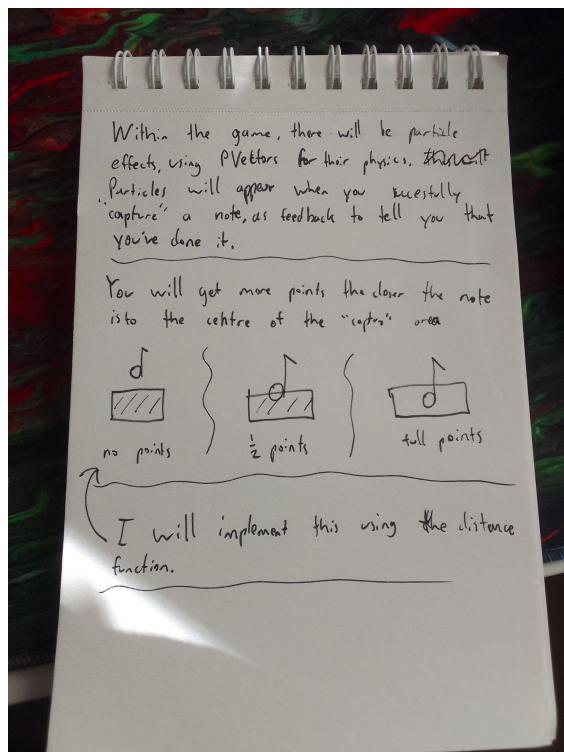
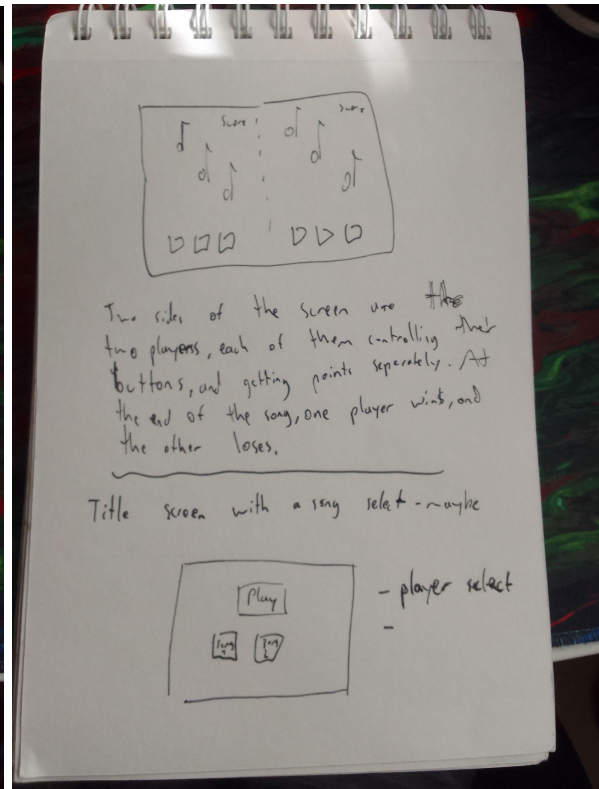
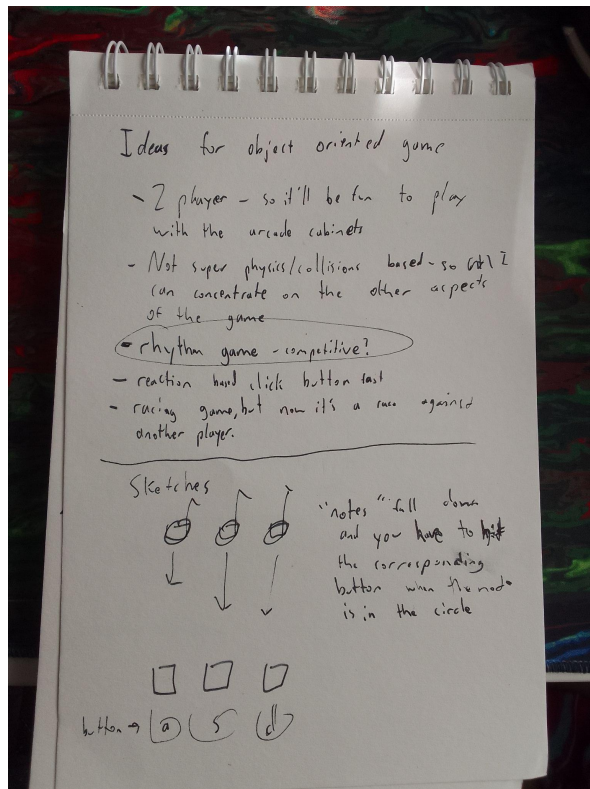


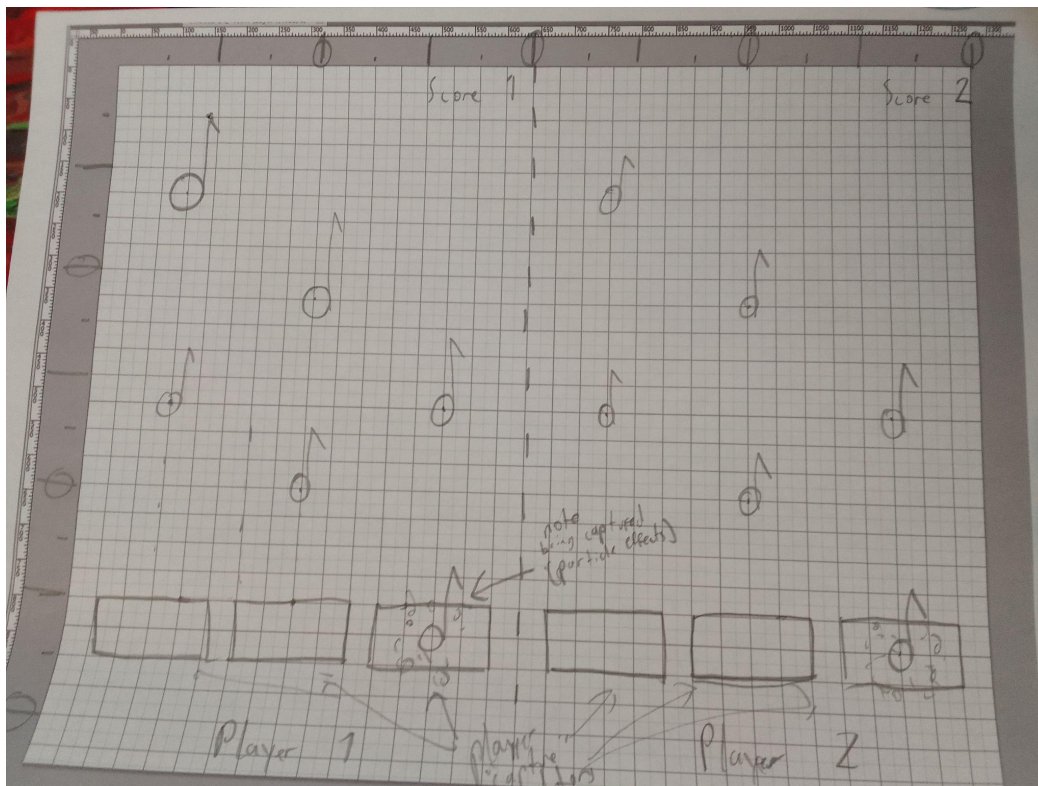
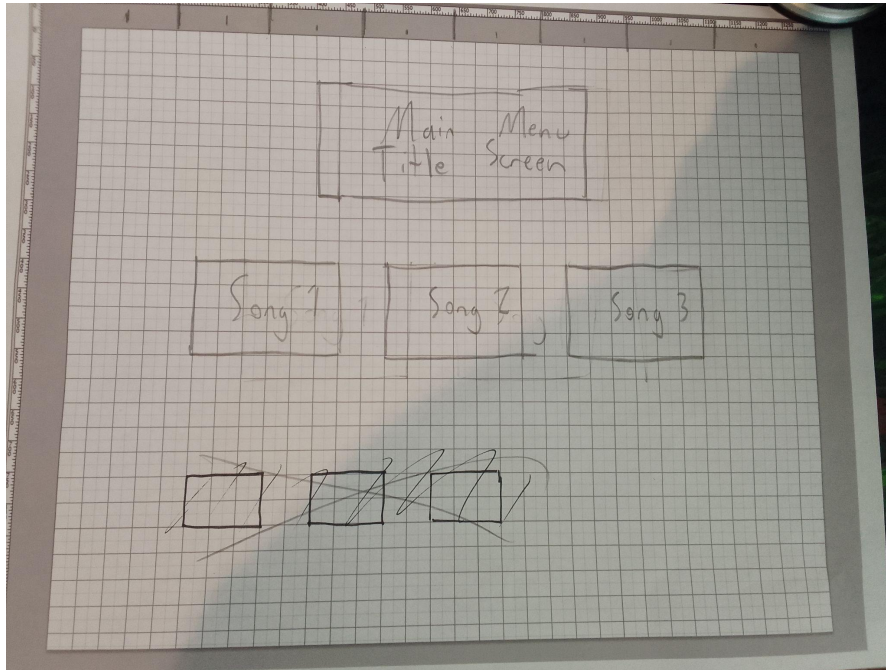
Github username: JacobMunterSGD

Idea



A multiplayer rhythm game, where you have to press the button in time with musical notes falling from the top of the screen.

Pixel Paper



Pseudocode - a rough idea of what needs to happen in the game

gameOver variable set to true // this will become false when the game starts

An array for the notes for each player

Score for each player

Set up the sound library

Setup()

Set all the variables to whatever you want them to be - this will be a different function, for when the game gets reset after finishing a level.

Setup the background, and all necessary beginning graphics for the title screen

Draw()

If gameOver is true

if they're in the title screen

Draw the title screen and all the necessary things there

If they're in the level end screen

Draw the level end screen

If gameOver is false

If the game just started

Setup up the game

The song

when the notes are going to fall

Add all the notes in the level to an array, and then we can target all of them at once. Then duplicate it for the other player.

The player scores get reset

Draw all the graphics on the screen

Create the player button objects

If the game hasn't just started

Update the game

Move all the notes

If there's a note getting dropped at this time, select a random location (of the three possible slots) for it to fall to

Check for player input

If they correctly "captured" a note, add score

Draw particle effects to give feedback to the player

If they failed in capturing a note, remove score

Update the score display

Check if the song has ended

Update any other visuals

Player buttons class

Constructor

Is this player one or two?

Set up the player accordingly

Input

Check which of the 3 inputs the player has done, then check the distance from the notes in that column, and update accordingly

Display

Draw the buttons

Notes class

Constructor

Add this note to the array of notes

Function to display

Function to check position of a note

Checks where it is, when this function is ran in draw, it will run it for each note in the array

Particle Class

Constructor

Add the array of particles

Move

Move the arrays with some sort of physics thing - fly outwards, then slowly fade out

Skills inventory

Shapes

- Basic shapes
- Shape properties
- Shape modes

System

- Setup() and draw()
- background(), random(), noise()
- constrain(), dist()
- keyPressed(), keyReleased(), etc
- Increment operators
- Local variables
- Global variables

Debugging

- println(), stop()

Control Flow

- Conditional statements
- Boolean expressions
- Logical operators
- Switch statement

Loops

- for/while Loop
- Nested loop
- break()

Functions

- Declare and call with no parameters and no return type
- Declare and call with a return type
- Declare and use a function that takes a primitive as an argument
- Declare and use a function that takes an object as an argument

Classes/objects

- Write a class with a constructor function
- Use the keyword new to instantiate an object
- Write a constructor function with parameters

Lists

- Initialize and populate an array
- Initialize and populate an ArrayList
- Manage a set of objects with an array or ArrayList
- Use an ArrayList method

Vectors

- Use the PVector class
- Do basic physics
- Find the direction and distance between two points

- Create a random 2d vector
- Use a new way to use a PVector

Optional

- Use a timer
- Switch between game states
- Make a button or toggle switch with a roll-over highlight
- Create a drag and drop object
- Do animations with images
- Use collisions detections between objects
- Use game control plus controller library

	Milestone 1	Milestone 2	Milestone 3	Milestone 4
What will I deliver	<p>Get the basic gameplay working, notes falling from the top, the processing sound plugin working, music syncs up with the notes. The player can capture notes. Nothing looks pretty at all yet.</p> <p>Finish a quarter of the answers for the skills inventory questions.</p>	<p>Create an additional player, create the title screen and end level screen, fix bugs in the gameplay, refine the gameplay</p> <p>Finish half the answers for the skills inventory questions.</p>	<p>Add visuals, work on particle effects (that use PVectors), playtest a little bit, refine gameplay if needed.</p> <p>Finish the answers for the skills inventory questions.</p>	<p>Refine visuals, make the game look pretty, complete the final level of polish. Set up the game controller plus library.</p>
What skills	<p>Basic shapes</p> <p>Shape properties</p> <p>Shape modes</p> <p>Setup() and draw()</p> <p>background(), random(), noise()</p> <p>constrain(), dist()</p> <p>keyPressed(), keyReleased(), etc</p> <p>Increment operators</p> <p>Local variables</p> <p>Global variables</p> <p>Conditional statements</p> <p>Boolean expressions</p> <p>Logical operators</p> <p>for/while Loop</p>	<p>println(), stop()</p> <p>Switch statement</p> <p>Nested loop</p> <p>break()</p> <p>Declare and call with a return type</p> <p>Declare and use a function that takes a primitive as an argument</p> <p>Declare and use a function that takes an object as an argument</p> <p>Write a class with a constructor function</p> <p>Use the keyword new to instantiate an object</p>	<p>Use the PVector class</p> <p>Do basic physics</p> <p>Find the direction and distance between two points</p> <p>Create a random 2d vector</p> <p>Use a new way to use a PVector</p> <p>Make a button or toggle switch with a roll-over highlight</p>	<p>Use game control plus controller library</p>

	<p>Declare and call with no parameters and no return type</p> <p>Initialize and populate an array</p>	<p>Write a constructor function with parameters</p> <p>Initialize and populate an ArrayList</p> <p>Manage a set of objects with an array or ArrayList</p> <p>Use an ArrayList method</p> <p>Switch between game states</p>		
--	---	--	--	--

Questions from skill inventory (these will get finished over the course of the project)

19. What's the difference between a for loop and a while loop?

A "for" loop has all the information for repeating the loop in a single line (local loop variable, what value that variable has to be for the loop to run, and how much that variable will increase every time the loop repeats), while the "while" loop simply checks a boolean expression, and while repeat until that expression returns false.

22. What's the difference between parameters and arguments?

Parameters are the local variables stored within the function, these are different every time the function is called, and are determined by the arguments that you put in for that given time that you call the function. The parameters go between the parentheses when you create the function, you write the arguments whenever you call it.

25. What's the difference between a class and an object?

The difference between class and object is kind of similar to the difference between parameter and arguments. A class is like the instructions to create an object, and the object is the actual thing itself. Either thing doesn't make sense without the other, you need the instructions to make something, but you need the thing itself to actually use it.

26. What is a constructor function? What does it do and when?

The constructor function is only called when a new object of that class type is being created. It sets up the object, and (sometimes, but necessarily) uses the arguments that were passed when you made the object. Often, the specific variables for an object are made within the constructor, as the constructor gets called everytime an instance of that object is created, so it's a good spot to set things up.

27. Why should each class have its own tab in processing?

When each class has its own tab, everything is more organised, which makes reading the code much easier, and finding the source of the problem can be localised more easily. Having a separate tab also differentiates (to whoever's writing the code) what happens within the class, and what happens outside, which is useful when coming across class related problems.

31. What's the difference between an array and an ArrayList?

32. Why would you want to go through an array list backwards, decrementing the index?

37. When should you use PVector instead of float variables?

You should use PVectors whenever you want to store two related floats, such as the position of something on a cartesian plane having “x” and “y” coordinates. This can apply to things like position, velocity, acceleration.

42. What is a normalised vector, why is it useful?