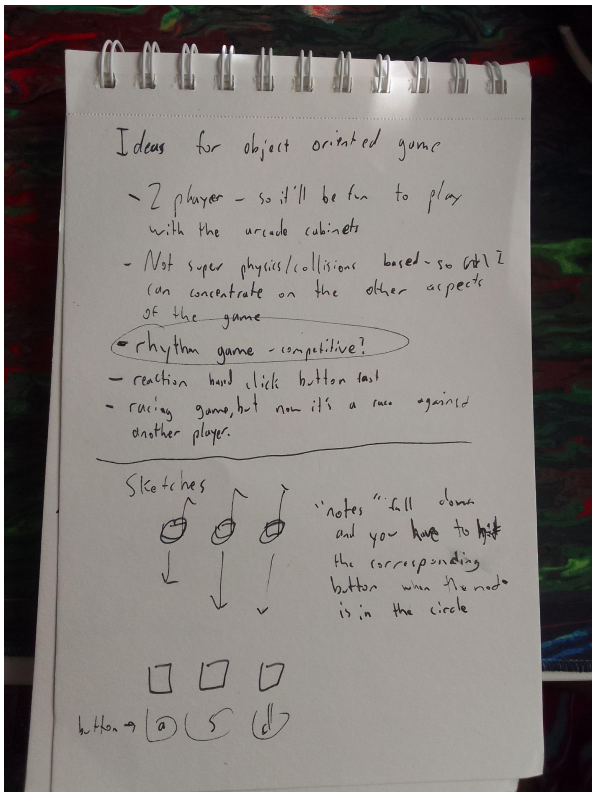
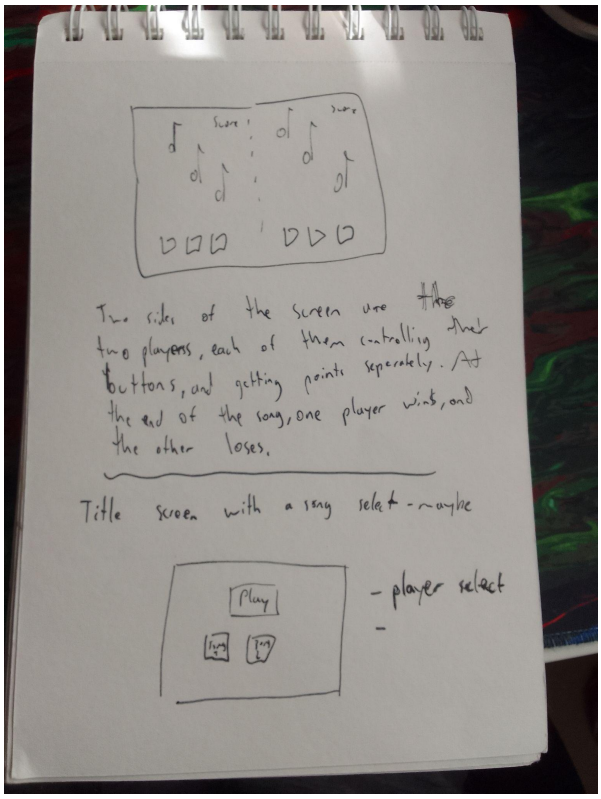
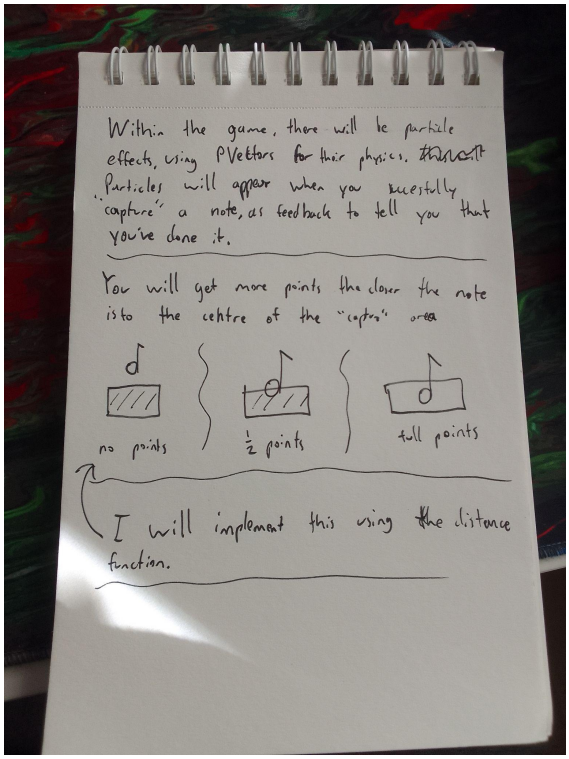
Github username: JacobMunterSGD

Idea

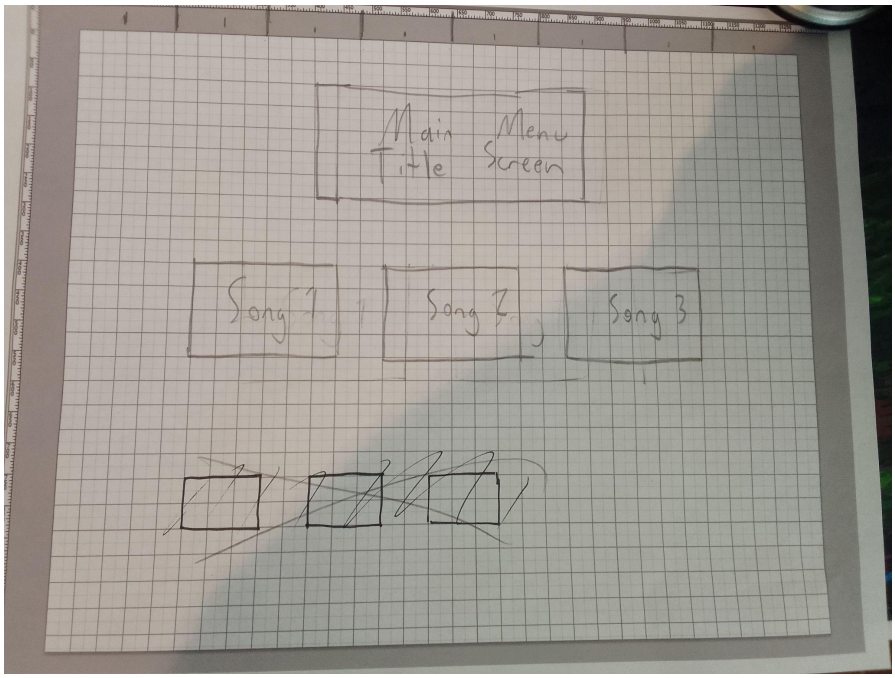


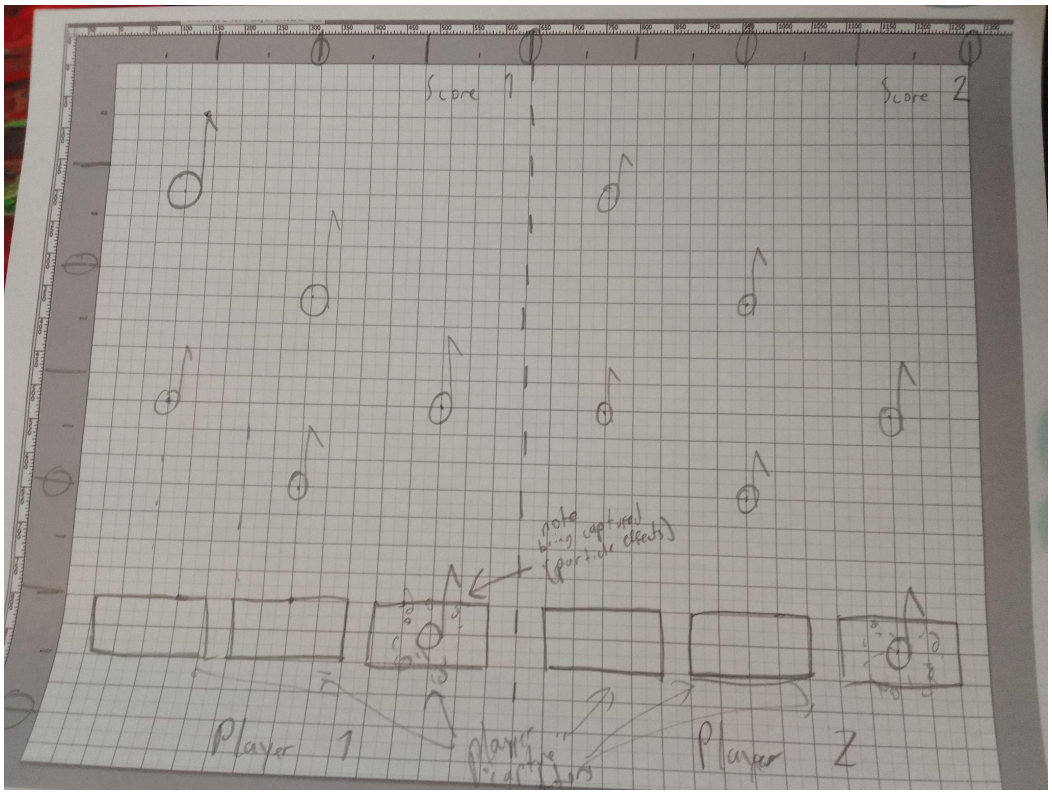




A multiplayer rhythm game, where you have to press the button in time with musical notes falling from the top of the screen.

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

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

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	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. Finish a quarter of the answers for the skills inventory questions.	Create an additional player, create the title screen and end level screen, fix bugs in the gameplay, refine the gameplay Finish half the answers for the skills inventory questions.	Add visuals, work on particle effects (that use PVectors), playtest a little bit, refine gameplay if needed. Finish the answers for the skills inventory questions.	Refine visuals, make the game look pretty, complete the final level of polish. Set up the game controller plus library.
What skills	Basic shapes Shape properties Shape modes Setup() and draw() background(), random(), noise() constrain(), dist() keyPressed(), keyReleased(), etc Increment operators Local variables Global variables Conditional statements Boolean expressions Logical operators for/while Loop Declare and call with no parameters and no return type	println(), stop() Switch statement Nested loop break() 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 Write a class with a constructor function Use the keyword new to instantiate an object Write a constructor function with parameters	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 Make a button or toggle switch with a roll-over highlight	Use game control plus controller library

	Initialize and populate an array	Initialize and populate an ArrayList Manage a set of objects with an array or ArrayList Use an ArrayList method Switch between game states		

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.

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

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

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?

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