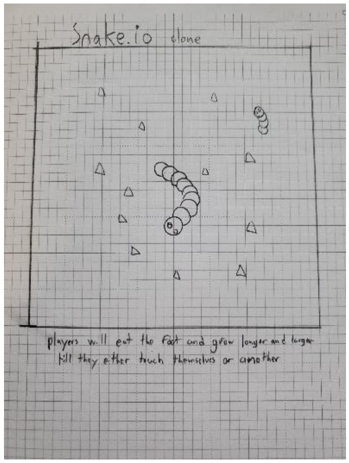
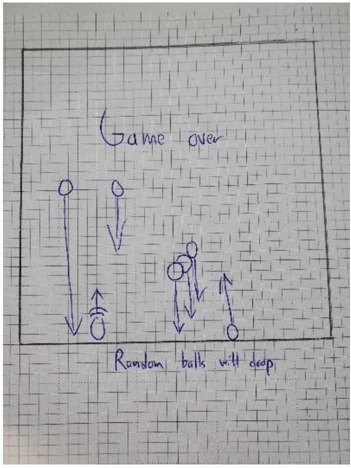
My idea – Joe Place

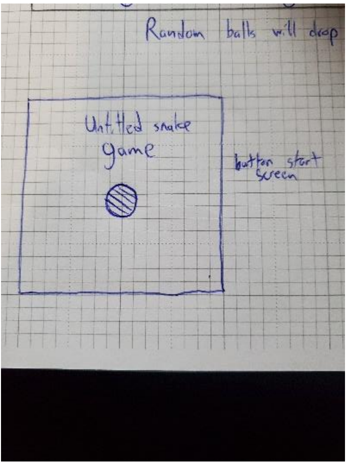
Use this to summarize your idea, plan it using sketches, notes and pseudocode as needed

GitHub Username: LungMilk
Create a game inspired by slither.io where snakes grow eating food in a two player environment. Players will increase in size after eating food and can eliminate the other player or themselves by contacting the other players' body. The game is survival based with the food spawning through the use of a timer with the players score being monitored by how much food they have eaten.

Player will eat food depending if the snake head is intersecting any piece of food
To check the intersections the distance between the two objects center point will be compared to their radius or half the width and if the value is within the limit then the food will be eaten or player will loose
The snake will be a separate entity where circles are drawn in the main head's last known position using an array.
Eatig food will increase the amount of segments in the body
Using classes the food and snake will be separated to allow for easy modifications and randomization of attributes (color, eye color)
The end screne will spawn circles that can bounce to simulate physics and the start screen will have a button that will start the game
The starting state will be saved to allow for easy replay ability
**the menu screen will have the player be able to rotate between colors (this will be an array)


Snake.io clone
players will eat the Foot and grow longer and longer till they either touch themselves or another





Where will the inventory skills be demonstrated? List every one to be sure you’ve included them.

Shapes – all the shapes will be used to draw the graphics of the snakes and food==

1. line, ellipse, rect, triangle, quad, arc, curve
2. fill, stroke, strokeWeight, noFill, noStroke, color – determine the attributes of the drawings
3. Modes: CORNER, CORNERS, CENTER, RADIUS – circles are from the center

System==

4. setup(), draw() – to have the program loop to create animations
5. background(), random(), noise() – position of the food and amount present for their initial spawn
6. constrain(), dist() – dist() will be used to check if the snakes are interesting as well as the food
7. keyPressed(), keyReleased(), keyPressed, mousePressed(), - The “joystick” will be keypressed and determine the direction of movement
mousePressed
8. increment operators: ++, +=, --, -=, *=, /= - used to alter the attributes of the snakes as well as within loops to ensure they properly increment
9. declare and use a local variable – variables used within the classes of the snake and food for malluability
10. declare and use a global variable – Booleans to determine game state

Debugging==

11. println(), stop() – println used to communicate both the instructions, game state and polish

Control flow==

12. conditional statements: if, else if, else - Check for intersections and determine game state or effects of players actions (which direction they move)
13. Boolean expressions: ==, >=, <=, >, <, != - ^^^ used in if statements to initialise elements of the game based on the state of the boolean
14. Logical operators: &&, || - used to specifically determine an if statement (if the snake is intersecting an object)
15. switch statement – determine various circumstances as in how the player dies

Loops ==

16. for loop, while loop – initialize and create objects and the snakes body
17. A nested loop – used to determine if each individual object is intersecting over another object
18. break() – end state (game over)
19. What’s the difference between a for loop and a while loop? – for loop will loop based on parameters when it is established containing a variable, when the loop stops, and how long it lasts. A while loop will go forever if the conditional state is true

Functions ==

20. Declare & call a function with no parameters and no return type - void functions will be used to draw the various graphics
21. Declare & call a function with a return type – the intersection checker of the snake and food will return a Boolean value
22. What’s the difference between parameters and arguments? – Parameters are variables defined when the function is defined in its creation, while arguments are values declared when it is called. So parameters are a functions variables that can be altered while arguments are values passed that are then used in the functions process
23. Pass by copy (value): declare and use a function that takes int,
float, char, etc as an argument – used to check the position of the snake against the other objects for intersections.
24. Pass by reference (objects): declare and use a function that – Pass the values of a vector object to check its position and state
takes an object as an argument

Classes/objects==

25. What’s the difference between a class and an object? – A class is the logical information that can be referenced as an object where as an object is the physical entity

of a class.
26. What is a constructor function? What does it do and when? – a constructor function creates the entity of a class. It sets the properties of the new object when the class is initialized
27. Why should each class have its own tab in Processing? – having their own tab allows for easier modifications and isolations for debugging to create an easy method of understanding and identification of classes.
28. Write a class with a constructor function – the snake and food will each be a separate class
29. Use the keyword new to instantiate an object – to use the classes of the snake and food new will be used to control the logic within the game.
30. Write a constructor function with parameters – the snakes will be passed color and player identification to determine which is being moved

Lists==

31. What's the difference between an array and an ArrayList? - Arrays are a data structure initialized with a fixed length where arraylist is a variable length structure. Array lists are able to be modified in program while it is operating while an array is limited to its initial length.
32. Why would you want to go through a list backwards, decrementing the index? - When you want to remove the final items in a list or delete the objects at the end. Ex. The first to be removed would be the last
33. Initialize and populate an array – the array will control the food position
34. Initialize and populate an ArrayList – the arraylist will be used to elongate the snake body and shrink if necessary
35. Manage a set of objects with an array or ArrayList – to enlongate or modify the snake after it eats food the array will be modified
36. Use an ArrayList method: size(), get(), remove(), contains() – be able to reference the length of a snake or the values to determine score

Vectors

37. When should you use PVector instead of float variables? – Pvectors can be used in more complex equations and functions such as Position, acceleration, or velocity
38. Use the PVector class – pvector will control the movement of the snake
39. Do some basic physics: use position, velocity, and acceleration
(due to gravity) vectors – an end graphic will have balls bouncing on the position of the player who died
40. Find the direction and distance between two points – used to determine if the snake is intersecting with food or the other player
41. Create a random 2D vector – used to randomise the position of the food
42. What is a normalized vector, why is it useful? – Vectors numerical value is very specific with numerous decimal places. By normalizing a vector it stabilizes the vector reducing the fluctuation of the variables. Normalized vectors are more accurate and will ensure proper values
43. Using the Processing documentation look up a method in the
PVector class that's new to you and use it in your code. – I will look up a new pvector method most likely to be used at the end screen

Milestone 1	Milestone 2	Milestone 3	Milestone 4
What will I deliver? The snakes will be able to increase and move The food will be able to randomly spawn	Implement the intersection checkers to have end states and link the snake increase to the amount of food eaten Add a menu screen and end screen	You are strongly encouraged to deliver your finished game at Milestone 3. Polish Add 2 player functionality and hopefully joystick compatibility	
Which inventory skills will this demonstrate? List them.			
1. line, ellipse, rect, triangle, quad, arc, curve 2. fill, stroke, strokeWeight, noFill, noStroke, color 3. Modes: CORNER, CORNERS, CENTER, RADIUS	31. What’s the difference between an array and an ArrayList	37. When should you use PVector instead of float variables 38. Use the PVector class	
setup(), draw()	constrain(), dist(	11. println(), stop	

declare and use a local variable 10. declare and use a global variable	keyPressed(), keyReleased(), keyPressed, mousePressed(), mousePressed ncrement operators: ++, +=, --, -=, *=, /=	Find the direction and distance between two points 41. Create a random 2D vector	
conditional statements: if, else if, else 13. Boolean expressions: ==, >=, <=, >, <, != 14. Logical operators: &&, ||	32. Why would you want to go through a list backwards, decrementing the index 33. Initialize and populate an array 34. Initialize and populate an ArrayList	42. What is a normalized vector, why is it useful?	
background(), random(), noise()	35. Manage a set of objects with an array or ArrayList – to enlongate or modify the snake after it eats food the array will be modified 36. Use an ArrayList method: size(), get(), remove(), contains()	43. Using the Processing documentation look up a method in the PVector class that's new to you and use it in your code	
25. What's the difference between a class and an object 26. What is a constructor function? What does it do and when? 27. Why should each class have its own tab in Processing? 28. Write a class with a constructor function 29. Use the keyword new to instantiate an object 30. Write a constructor function with parameters	20. Declare & call a function with no parameters and no return type 21. Declare & call a function with a return type 22. What's the difference between parameters and arguments 23. Pass by copy (value): declare and use a function that takes int, float, char, etc as an argument 24. Pass by reference (objects): declare and use a function that takes and an object as an argument	16. for loop, while loop 17. A nested loop 18. break() 19. What's the difference between a for loop and a while loop? 15. switch statement 39. Do some basic physics: use position, velocity, and acceleration (due to gravity) vectors	
You should deliver approx. 10 skills at this milestone	You should deliver approx. 10 skills at this milestone	**You must deliver 30 inventory skills by this milestone.**	