Ideation 1

Deep Sea Fishing adventure

Players will control a character sitting on a boat on the sea to catch variety of fish while avoiding obstacles like sharks or torpedoes by controlling the timing of casting the fishing line. The players will lose their health if they hit obstacles.

Features will be designed:

1. Main character
2. Fishing rod
3. Boat
4. Sea water
5. Various of fish (at least 5 sorts)
6. Shark and torpedoes

Game mechanics:

1. Player casts a fishing line with a mouse click.
2. Fish and obstacles randomly swim around.
3. The timing of casting the fishing line is important.
4. When hitting the obstacles, player will lose health.
5. When the health bar reaches 0, the game is over and start over again.

Additional features that can be add:

1. Time limitation: catches target number of fish within the time limit.
2. Include water wave effect using Perlin Noise

Challenges may face:

1. Implement the fishing line mechanics and detecting successful catches.
2. Managing the random movements of fish and obstacles.
3. Balancing the win and lose condition by setting requirements for numbers of fish players should catch.

The overall experience for player will be great. Especially those who like to play fishing simulator. However, it is a challenge to balance certain numbers and mechanics during programming. The effect may not be so good as expected. I also want to draw the fishes by myself, the time may not be enough…

Ideation 2

Farm Simulator

Players have a small farm. Their daily mission is planting, watering, and harvesting crops just like Stardew valley. The game will end when the player harvests a set number of crops. Players will experience monsters like slime or craw. They will attack your crops. The players need to click and get rid of them.

Features will be designed:

1. Character
2. A piece of farm land
3. Three types of crops, each type of flower, vegetable and fruit
4. Monsters: slime or craw (insert animation images)

Game mechanics:

Players move the farmer and press specific keys to perform actions

- Plant seeds ("p")
- Water the crops ("w")
- Harvest mature crops ("H")
- Click to get rid of the invaders ("mouse clicks")

Additional features:

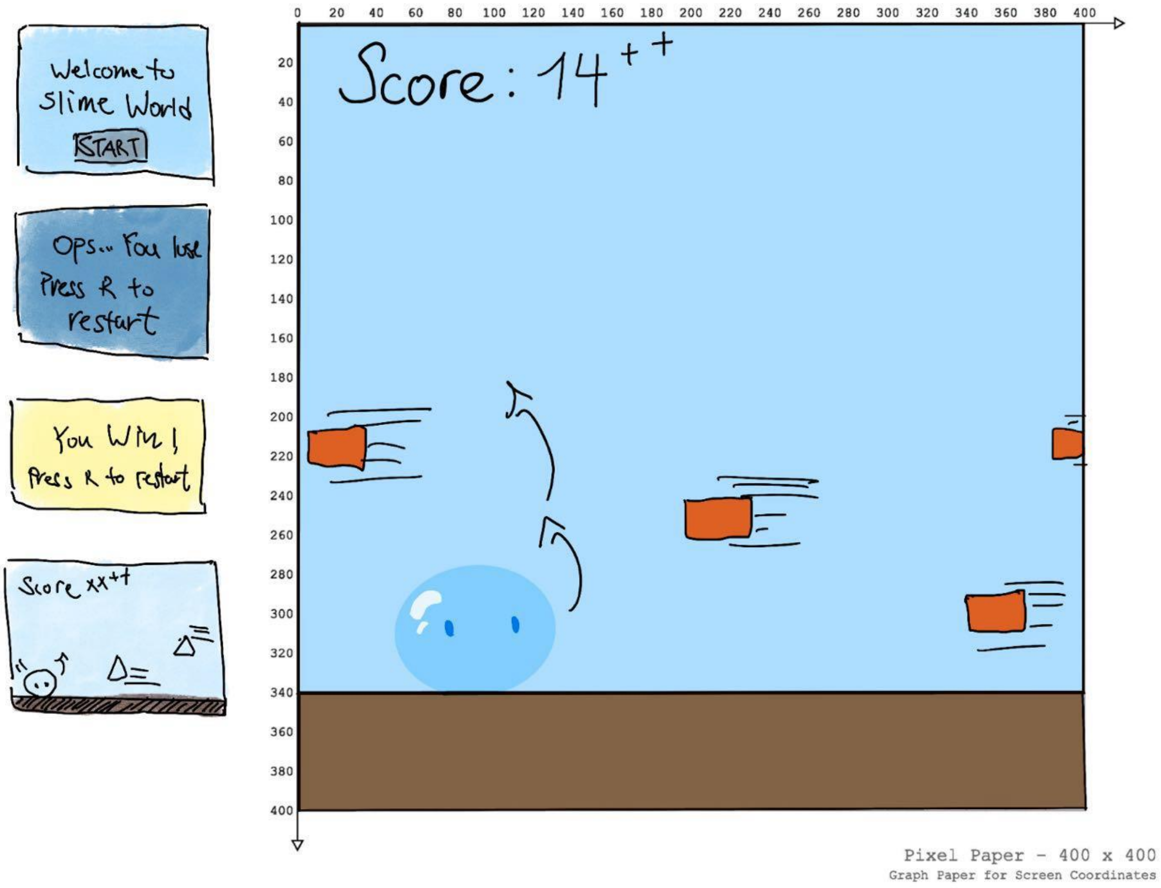
- Add different types of crops with varying growth times and values
- Include weather effects (rain speeds up growth, effect plant growth)

Challenges may face:

- Manage crop growth stages with time and player interaction
- Creating a smooth and intuitive control system for planting, watering and harvesting.

I personally find this idea is quite good. The players will have a good play experience with a mix of strategy, time management and quick reflexes to create an engaging. However, due to the time limitation, I decide to do an easier one.

Pixel Paper drawing



Pseudocode

Add CLASS for:

- Slime
- Obstacle

Add ARRAY for slime and obstacles

Void set up: canvas set to 800,400

Background setup

- Start screen: light blue
- Game over: dark blue
- Win screen: yellow
- Play screen: light blue

Initialize variables:

- slime
- obstacle
- score

Boolean:

- game Start
- game Over
- start page

Function draw:

- Display start screen
- Display game over screen
- Display win screen
- Update and display slime and obstacle
- Increase score
- Check for collision between slime and obstacle

Control input:

- Mouse Pressed
- mouse clicks
- key Pressed
- key Released

Display screens:

- Game Over
- Win
- Reset Game

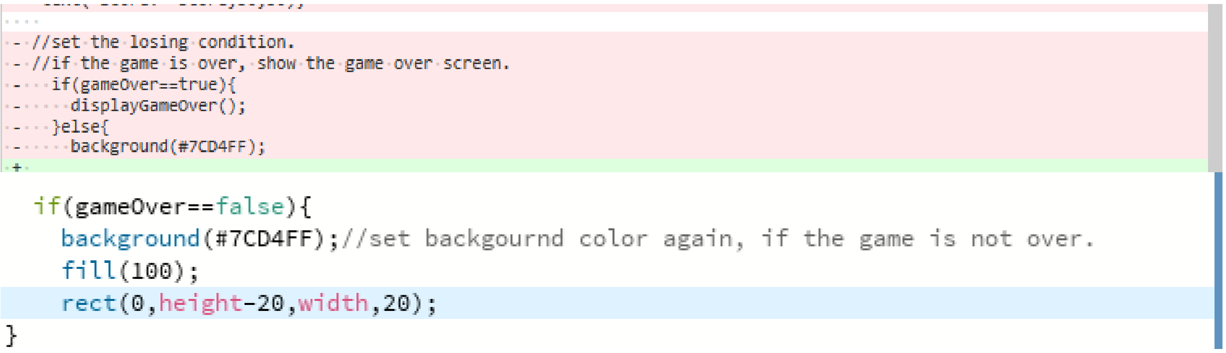
Reset status:

- Score
- Obstacle Height

Implementation Log

Problem 1: Condition page couldn't appear in the right order

The game doesn't show the start screen in the right order. For instance, processing did not show the game over page when the slime touches the obstacles. It continues displaying the game play page.

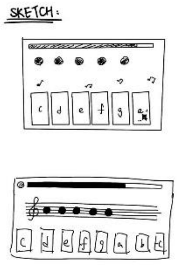


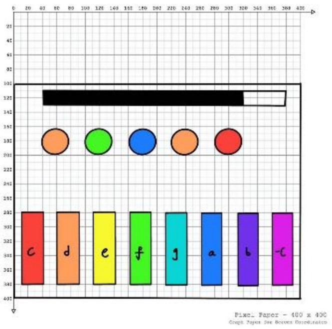
```
- //set the losing condition.
- //if the game is over, show the game over screen.
-   if(gameOver==true){
-     displayGameOver();
-   }else{
-     background(#7CD4FF);
+

  if(gameOver==false){
    background(#7CD4FF);//set backgournd color again, if the game is not over.
    fill(100);
    rect(0,height-20,width,20);
}
```

Solution: There are some logical mistakes with the if and else statement. After testing for several times, my solution is to separate the statement into two, which makes the whole logic easier. I deleted the else statement and added a rectangle to cover the whole screen.

Problem 2: changed the topic of this project

I was planning to do a music game. The basic mechanic is that the players need to remember the music notes appeared on the top of the screen. Then play the instrument (xylophone) to realize the rhythm. However, the determinations for each key to match the notes appearing above is too complicated and caused a lot of BUGS.

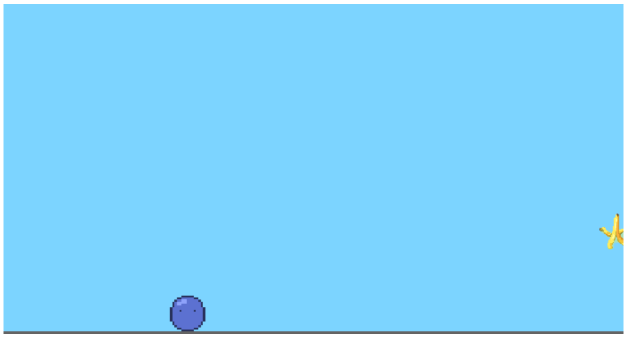


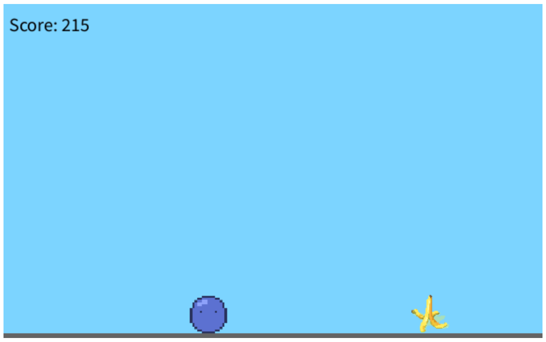


Solution: I designed to change a theme with easier mechanics like jumping, moving and avoiding obstacles. Since, I want to avoid using to much supporting tools for my coding.

Problem 3: The score didn't display and reset after game over

When the game starts, the score does not reset to 0, and didn't display on screen at all. Thinking from player's perspective, it will cause confusion since they might not see their final scores or how to start a new game with fresh score.




Score: 215

Solution: I come up first with the order of displaying scores. First, the score should be included in reset functions and being displayed in draw functions. The score code was above the displaying codes, so it was covered. So, I moved it down, so that it can be displayed. I also changed the Text size of score to make the whole visual more comfortable.