Week 9

Progress and design reports

Table

[Background 2](#_gjdgxs)

[Process 2](#_30j0zll)

[Goal 2](#_1fob9te)

[Rules of the Game 3](#_3znysh7)

[Data 3](#_2et92p0)

[Weekly Progress 3](#_tyjcwt)

# Background

The inspiration for the game came from my classmates. When I was taking this course, NM2207, I knew from the beginning that I would eventually need to make a front-end project to perform. I had a quick preview of the weekly course outline. But the front end was so broad and deep that I wasn't sure I would be able to achieve a nice presentation for the final project with so much freedom. So I asked the instructor about it very early on. , but at that time, there was not enough support for the final project, so I wasn't inspired enough. One day in class, I saw my classmate playing a very jerky game of Gluttony, and she told me it was that she was also looking for inspiration for the final project. And at that point, Snakebite stuck in my mind because it was a relatively simple game that could be picked up slightly quickly and without too many logical errors. At the same time, there are level games in my consideration. However, it was clear that the level game involved multiple CSS switches and I wasn't confident enough to handle the CSS pollution and transitions between levels comfortably. At the same time, as a level game, if there are monsters and props, then their complexities are considered to require back-end support. I was completely at a loss as to how to connect the front and back ends, so I decided to make a game that could be implemented on the front-end scope.

After the lecturer put the resources of the final project on the website, I found that a link to an excellent project was provided at the same time, so I tried to play the excellent project and I found that at the beginning it had text animations that played and introduced me to that scene quickly. So I plan to have that animation in my final project as well. I am likewise more adept at making algorithms for maze games. But games that were too similar to the example seemed too uninspired, so Gluttony became my first choice at this point.

# Process

So I started by learning the basic logic of PPT JS and Gluttony. After that, I created two separate projects. The first one was about Canvas animation playing text and the second one was about the basic logic of Gluttony Classic Gluttony.

The two projects were not too challenging to create separately, but the biggest challenge was how to combine the two projects into one, since they had different CSS and the PPT CSS wouldn't disappear after I brutally put them together. So it would stay there and the game would not play properly later. Thankfully, this problem was finally solved by using style.display.

# Goal

The goal of the game is to win the snake game after watching a short animation.

# Rules of the Game

The interval between each feeding must not exceed 100 seconds and the snake will die of hunger. The snake will march automatically and cannot hit the border or eat itself.

Also, there are two additional types of food compared to a traditional snake game. In addition to the normal food that increases the length of the snake by one section, there are also poisons and drugs. Drugs increase the length of the snake by two sections, while poisons decrease the length of the snake by one section, and the snake becomes faster and faster as the length increases.

When the snake reaches a certain length it will win, which is what the game calls escape from the darkness.

# Data

The saved user data are:

1. The name of the user
2. The location of each point of the snake
3. The location of the three different food items
4. The direction the snake's head is facing
5. The position of the snake's head
6. Time left in the countdown
7. The current speed of the snake at this point

# Weekly Progress

| **Week** | **Concept** | **How I've used it** | **Line number** | **Filename** |
| --- | --- | --- | --- | --- |
| Week2 | A basic introduction to CSS and detailed usage. | The format and style of the entire project is defined by CSS, so CSS is a very important part of this project. | 1-the end | css file |
| Week2 | Detailed description of how HTML presents web pages | The HTML will be the home page for my final project, so it will be the place where all the modules will be displayed. | 1-the end | index.html |
| Week2 | Learns how to add links to HTML in this week's exercise. | Call CSS and js files in HTML, call images in CSS, and call music links in html. | 78,198-208 | css/snack.css |
|  |  |  | 10,11,27 | index.html |
| Week3 | How to use js | Throughout the project, js is the main functional and logical file. | all | js/ |
|  | Grasshopper has a lot of tutorials about loops and how to write logic. | It is used in several places throughout the document | all | all |
| Week4 | alert pops up an alert that can be fetched to display the message | Pop up a prompt at the end of the game (win or loss). | / | js/snack.js |
|  | Function is a piece of code that can be encapsulated into a better call. | js logic calls a function multiple times to make the logic smooth | all | js/ |
| Week5 | Learned about different types, such as input and button. | Types are the basis for the whole project to work, such as the input to start, and the button to start the game. | 20 | index.html |
|  | Class and CSS work together to define a type | In Snake, the start button is defined as follows. | 43-45 | index.html |
|  |  |  | 84-111 | css/snack.css |
|  | Listening for event clicks | When the snake game starts, clicking the button will actually start. | 565-588 | js/snack.js |
| Week6 | Uses of loops | The motion and body of the snake are achieved by looping through the nodes. Including removing snakes at the end of the game. | all | js/snack.js |
| Week7 | Implementation and usage of canvas | Before the game starts, a slideshow is shown, and they use canvas to demonstrate it. | all | js/ppt.js |