

Epic Games | Jonathan Wu (Loki), Jesse Xie (Polly), William Chen (Cheap), Josephine Lee (Kitty)

SoftDev Design Doc

P04 – SoftDev Project 2021-22

2022-05-26

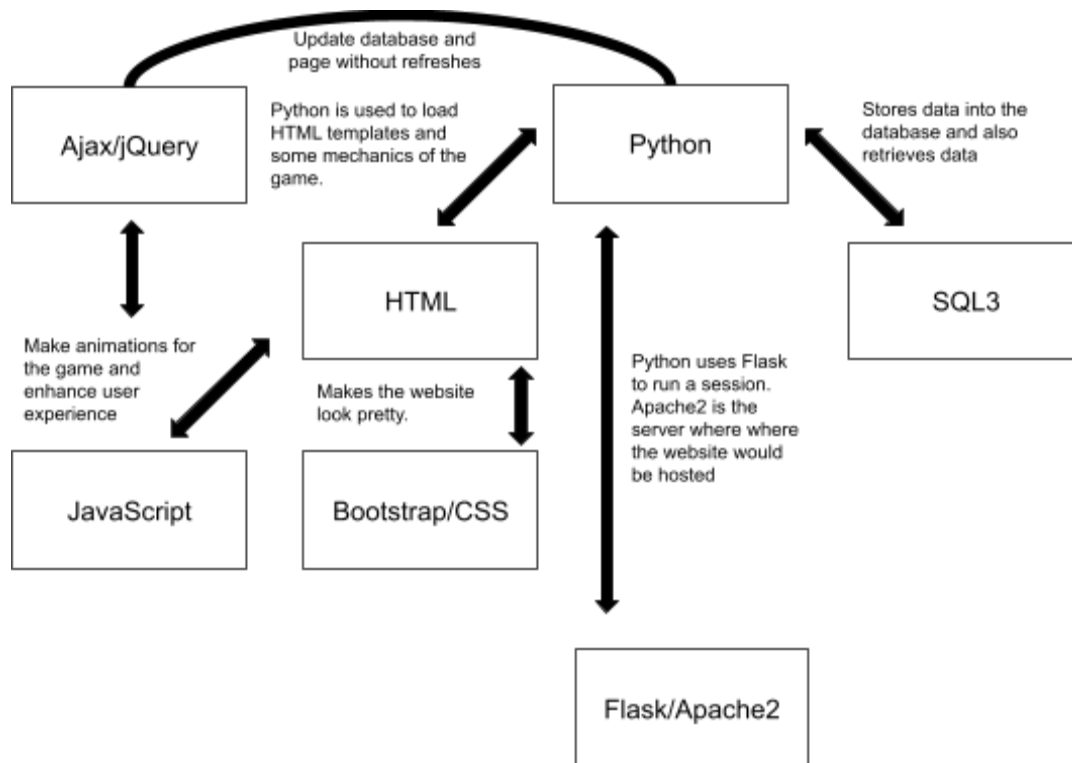


Project description:

Our project, PYTHNX, is our take on the arcade auto-battler game, SNKRX. The player controls a snake-like character made of segments/heroes that can be purchased with currency acquired from the game. The segments automatically attack the enemies approaching the snake, so it is up to the player to keep the snake from harm. Unique builds can be created and bonuses can be activated from assembling different compositions of heroes, as the player combats endless waves of enemies.

- The player can move around in any direction by rotating with the A and D keys
- The player can purchase segments of the snake for a currency, along with units. These units can be rerolled by spending a specific amount of a currency and be locked (they will remain in the shop even if it is rerolled)
- For every level completed, the player will obtain a specific amount of currency that they can use to purchase more units
- The segments of the snake can have different classes that give them different attacks and abilities, and the classes can be upgraded
- Having specific units of the same class in a party will trigger class bonuses
- During wave rounds, pill-like enemies will approach the player's snake
- The different segments of the snake will automatically attack the enemies with an attack cooldown
- The player will only take damage from the enemies, colliding with the wall will change the angle that the snake is approaching it from, and the segments of the snake can pass through each other
- Player will be prompted to provide a username at start, arcade style leaderboard

Component Map:



Project Components:

SQL3

- We will use SQL3 to store user and leaderboard information in collections, and interact with flask accordingly to generate the appropriate pages.

JavaScript

- We will use JavaScript to add animations to the website with canvas (showing the snake moving around) to enhance the user experience. It will also be used for the bulk of the functionality whenever the user makes keypresses.

Ajax/jQuery (using HMart's guide at

https://github.com/michelle-lo/p02/blob/d11e9ca75b022e4b41d9985183e1314846254cfd/howto_use_jQuery-and-AJAX.md)

- We will use these JavaScript libraries to update our database and our html pages with the appropriate data without needing to refresh or change the page we are on.

Python

- This will be used to add functionality to the website by enabling flask/html to create our pages and also parse any files that we will need

HTML

- Provides the text/skeleton of the website, also used for audio.

Bootstrap/CSS

- We will use bootstrap as our framework to create a visually pleasing and user friendly website. Any buttons and elements outside of the canvas can be produced with bootstrap.

Flask/Apache2

- Flask will allow the use of templates and session variables, and apache will be the server that our website will be hosted on.

Database Organization:

- SQL3
 - Leaderboard collection
 - username TEXT, waveReached INTEGER, teamComposition TEXT
 - teamComposition will be comma separated values (ex. wizard,archer,healer)

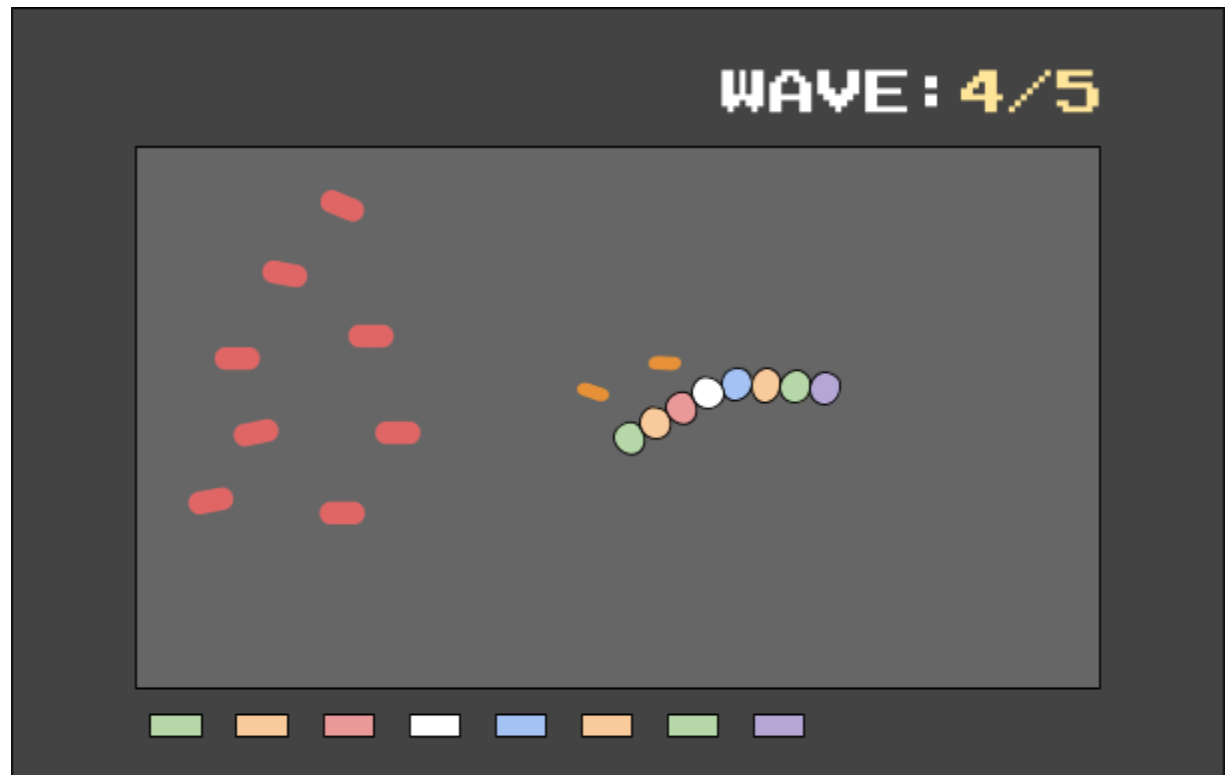
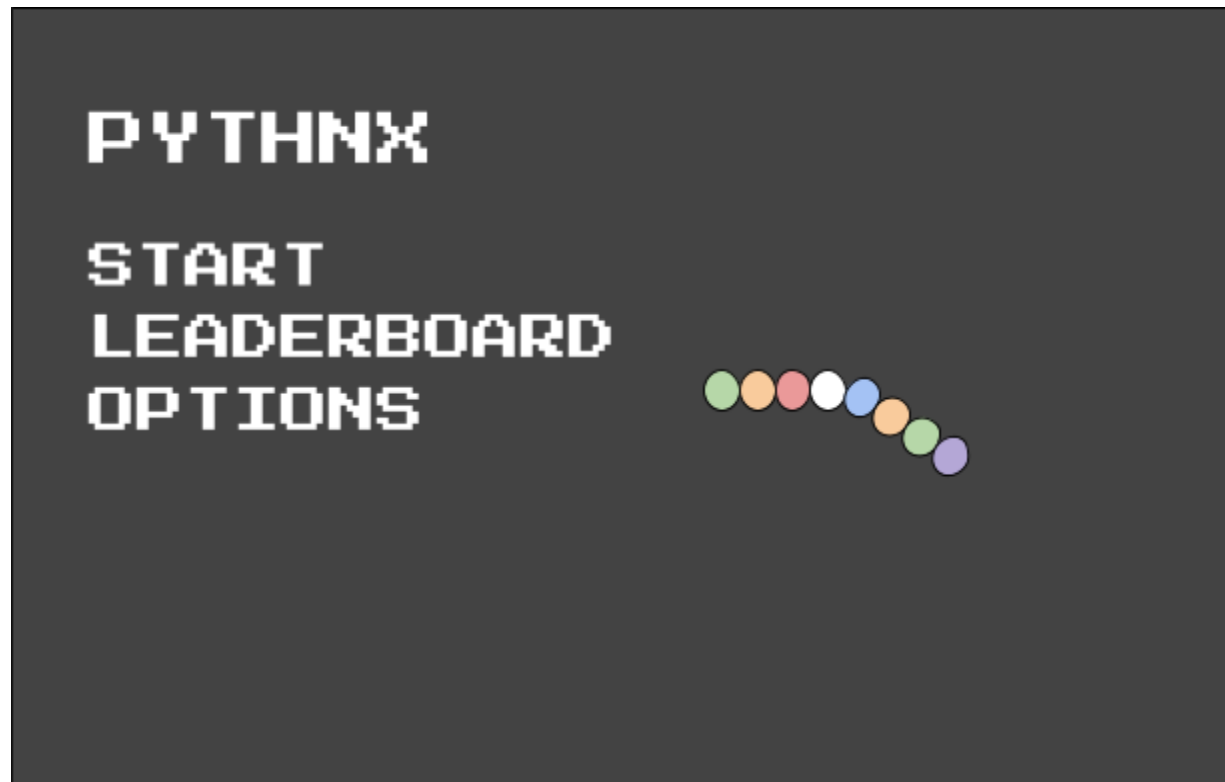
Site Map:

- One page
 - Game tab
 - Leaderboard tab

Frontend Framework Selection:

We'll be using Bootstrap for our front-end framework. We'll be taking advantage of Bootstrap's overall visually appealing styles of CSS/templates for the user to gaze at adoringly.

Site Templates:



SHOP - GOLD: 4/4 REROLL: 2 LOCK PARTY 4/4

2 WIZARD 2 OUTLAW 1 ARCHER

1 2
2
1 2
2

LV. 6 (EASY)

GO!

WAVE: 5/5

YOU DIED...

LEVEL REACHED: 15

submit to
leaderboard? (y)

restart run? (r)



LEADERBOARD

1. LEONARDO

LEVEL REACHED: 21

3 3 2 3 3 3 2 2

2. DONATELLO

LEVEL REACHED: 18

2 3 3 2 2 2 2 1

3. RAPHAEL

LEVEL REACHED: 13

1 2 1 2 1 1 2

4. MICHELANGELO

LEVEL REACHED: 7

2 1 1 1

Breakdown of Tasks:

- Jonathan (PM): html/css/js/sql3 frontend stuff
- Jesse: html/css/js/sql3 frontend stuff
- William: python/js/flask backend stuff
- Josephine: python/js/flask backend stuff

Target Ship Date: June 10, 2022