Rogue Sweeper

From the Pythonic Pioneers
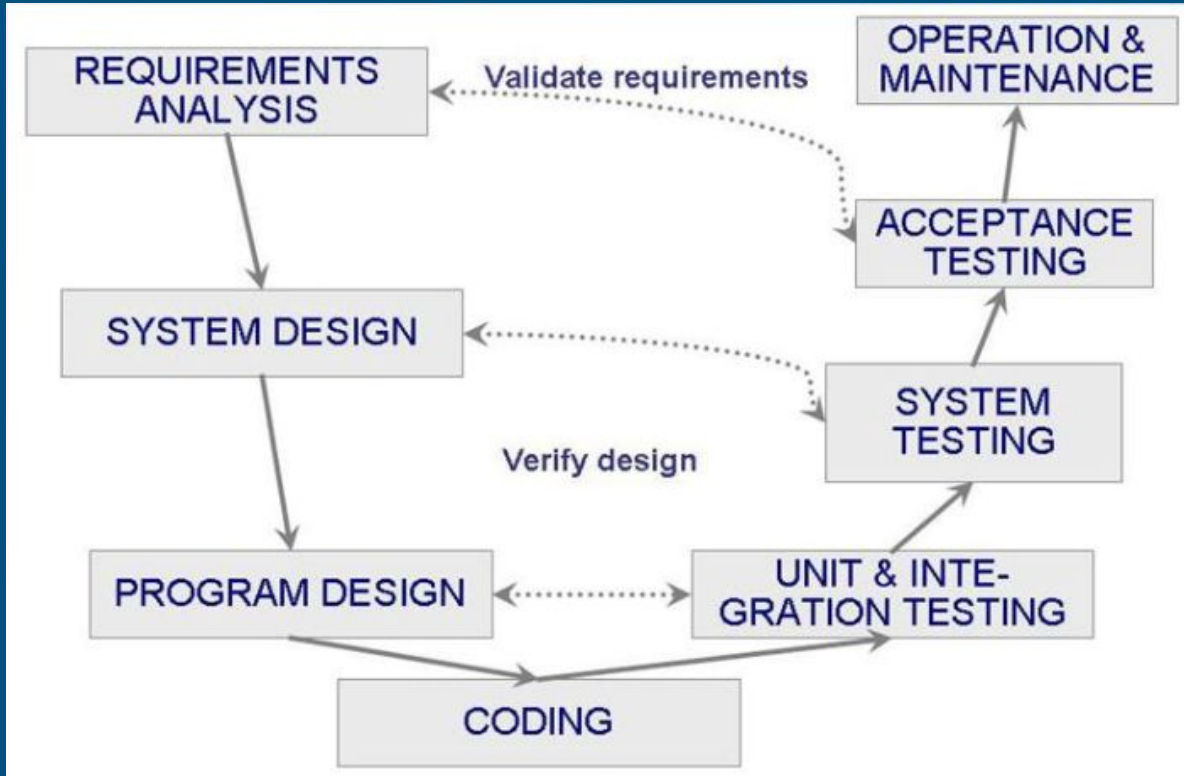By: Josh Wiles, Logan Scheurer, Brody Richards, Aidan Mack

Rogue-Sweeper Overview

We are taking the classic minesweeper game and turning it into a roguelike.
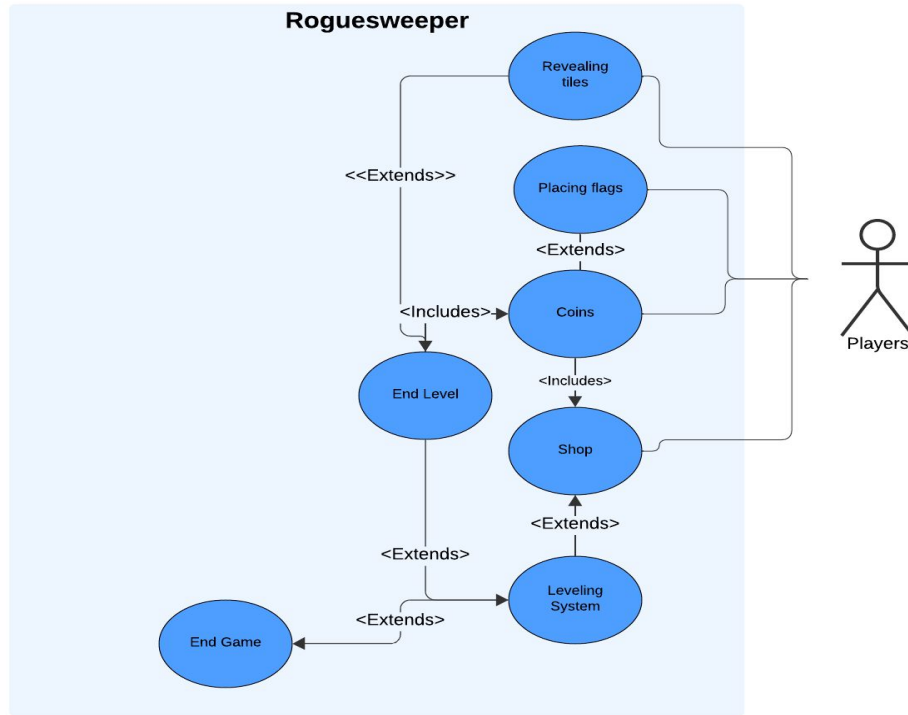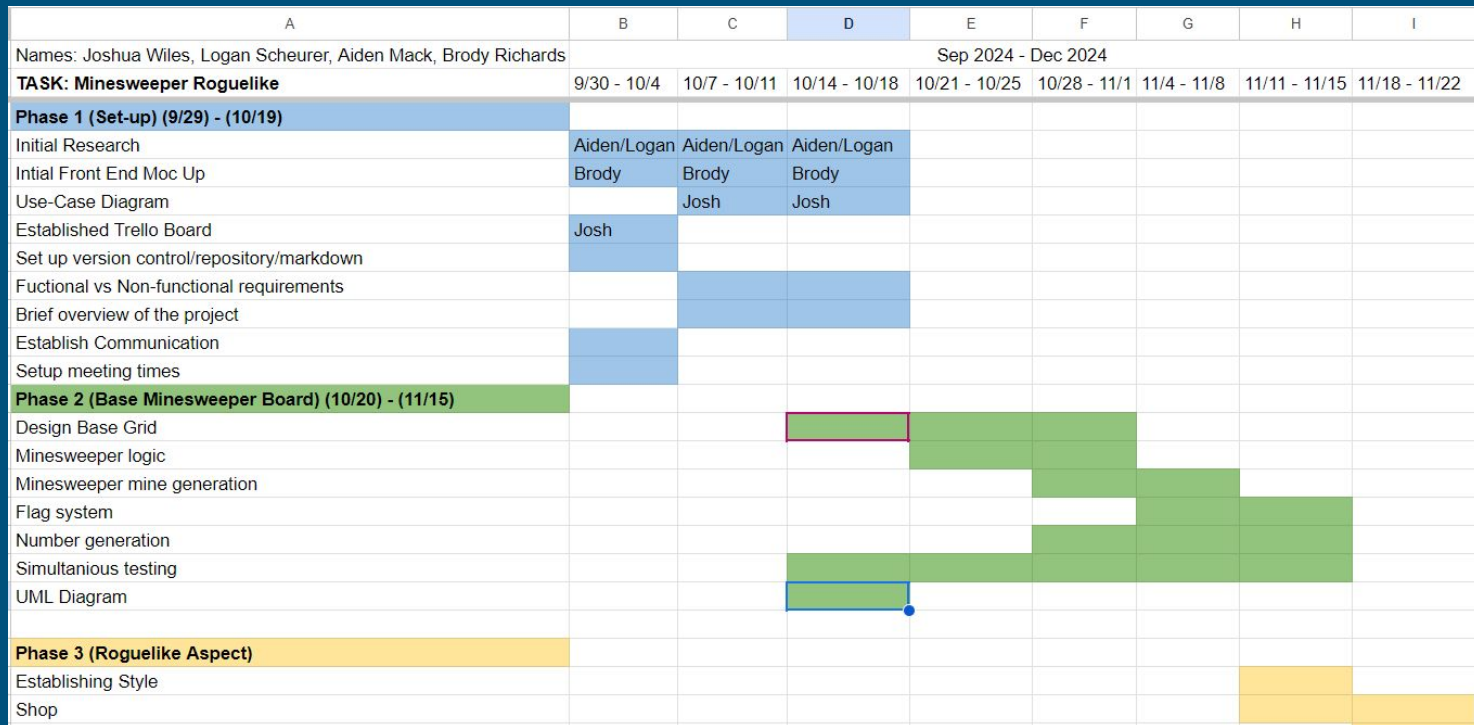
Our Process Model: V-Model

Technologies

For our project, we will be using python and pygame to create the board and implement the logic.

Use Case Diagram

Project Timeline - Gantt Chart

	A	B	C	D	E	F	G	H	I
	Names: Joshua Wiles, Logan Scheurer, Aiden Mack, Brody Richards				Sep 2024 - Dec 2024				
	TASK: Minesweeper Roguelike	9/30 - 10/4	10/7 - 10/11	10/14 - 10/18	10/21 - 10/25	10/28 - 11/1	11/4 - 11/8	11/11 - 11/15	11/18 - 11/22
	Phase 1 (Set-up) (9/29) - (10/19)								
	Initial Research	Aiden/Logan	Aiden/Logan	Aiden/Logan					
	Intial Front End Moc Up	Brody	Brody	Brody					
	Use-Case Diagram		Josh	Josh					
	Established Trello Board	Josh							
	Set up version control/repository/markdown								
	Fuctional vs Non-functional requirements								
	Brief overview of the project								
	Establish Communication								
	Setup meeting times								
	Phase 2 (Base Minesweeper Board) (10/20) - (11/15)								
	Design Base Grid								
	Minesweeper logic								
	Minesweeper mine generation								
	Flag system								
	Number generation								
	Simultanious testing								
	UML Diagram								
	Phase 3 (Roguelike Aspect)								
	Establishing Style								
	Shop								

What we have so far

```python
class GUI:

    def __init__(self) -> None:
        self.printed = 0
        pg.init()
        self.screen = pg.display.set_mode((1280, 480), pg.SCALED)
        pg.display.set_caption("RogueSweeper")

        # Create The Background
        self.background = pg.Surface(self.screen.get_size())
        self.background = self.background.convert()
        self.background.fill((170, 155, 187))

    @classmethod
    def load_images(cls):
        def load_image(count):
            SS = pg.image.load('./images/MS_Sprites.png')
            width = 17
            height = 16
            surf = pg.Surface((width, height), pg.SRCALPHA)
            surf.blit(SS, (0, 0), pg.rect.Rect(width * (count % 8), height * (count // 8), width, height))
            surf_scaled = pg.transform.scale(surf, (IMAGE_SIZE, IMAGE_SIZE)) # Change to image size
            return surf_scaled
```