**Project Proposal: Smile**

Smile is a pixel-art roguelike, action-adventure video game developed using Pygame. It works for Windows, Linux, and Mac OS. The player plays as a character called Smile who’s able to shoot, and dash.

The world, called Spirit, has different rooms. Spirit is filled with dangerous enemies, all waiting to kill you. You have to survive this dangerous world with the help of your trusty gun, Lirma.

**Dependencies:**

The game uses pygame module for game development and pytmx module for generating the map. To run the game on your device, you’ll need to install these dependencies. To install Pygame and Pytmx from pip, type the command given below in command prompt and press enter

pip install pygame

pip install pytmx

**Competitive analysis**:

There’s a coding video on YouTube titled “Creating a Zelda style game in Python [with some Dark Souls elements]” by Clear Code which is similar to the game I’m creating. The logic I use for spirit groups and camera movement will be similar to the way he implements it. However, shooting, dodging, enemies’ AI, and a lot of other stuff is completely different.

Here’s the link to the video: <https://www.youtube.com/watch?v=QU1pPzEGrqw&t=13764s>

There’s another video that I found recently. The game the person is trying to create is also very similar to the one I’m creating. He’s trying to create the whole game in exactly 10 lines of python code, and is a bit less complex than what I’m creating.

Here’s the link to the video: <https://www.youtube.com/watch?v=LgvLbahJOTA>

**Structural Plan:**

The game file is divided into two folders:

* source code – contains all the .py files
* graphics – contains all the images, textures used for the game
  + character - contains the enemies, player and world elements
  + tilemap – contains the Tiled maps and the textures of the game
  + tileset – contains all the tileset used in the game
* sound – contains all the sound for the game

Each element in the game world will have their own class. So far we have the following class:

* Game
* Level
* Player
* Projectile
* Creature
* Tile

**Algorithmic Plan**:

The trickiest part of my code will be animating, moving, and managing all of the enemies, generating new enemies, keeping of the interaction between each of the bullets and the enemies all at the same time.

To tackle this, I’ve created several sprite groups -- visible sprite, obstacle sprite, killable sprites, damaging object sprites. . All the elements that’s visible on the screen is put inside the visible sprite group, that includes all the enemies, bullets, player, and all other graphics. The obstacle sprites manage the interaction between the walls of the map and the player and enemies. The damaging object sprites is for weapons and the killable spire group is for all the enemies that are destroyable.

**Timeline Plan:**

Oct 23: Develop the world, implement player movements.

Oct 27: Create animations for player and enemies.

Oct 29: Implement Enemies AI. Fill the world with enemies. (MVP)

Nov 1: Submit TP2

Nov 3 : Create start menu, develop new maps, and new enemies. Change gun animation and add different types of bullets. (A big boss with different AI and a puzzle map if time allows)

Nov 5: Finish new maps, polish animation and fix all new bugs.

Nov 7: Submit TP3

**Version Control Plan:**

Git and GitHub

Link to the repository: https://github.com/SunayaUpadhyay/sm11e

**Module List:**

Pygame, Pytmx and other general python libraries like random, date, etc.