

Project Proposal

Project Idea 1: PyGame Missile Shooter Game

- This will be a 2D runner, where the objective is to obtain a high score
- The user is controlling a tank and you endlessly run across screen
- Movement includes jumping and rolling to avoid obstacles
- Power ups that will be available include, but are not limited to:
 - Shield- allows to take 1 extra hit
 - Jet- can stay in the air longer
 - Rapid Fire- shoot faster

Languages to use: Python

- Using the PyGame Library, which allows for graphic generation, as well as the importing of images into the project
 - This would handle most of the graphical aspects of our planned game, along with the assets it would include

Group Member Names: *Emmanuel Salcedo, Michael Rojas, Mario Linares*

Concept Images:



Above is an example of one the game inspirations, the Google Chrome Dinosaur Game. We will be using the gameplay of an endless running in a 2D space while avoiding things from the side.



Above is the other inspiration for space invaders. The gameplay we will be using is the idea of shooting aerial enemies.

Tentative Schedule:

1. N/A
2. N/A
3. Finalize Project Proposal
4. Planning out a complete concept of our games mechanics / design
5. Setup our team's workspaces, which include setting up PyGame / Github workflow
6. Start of Production cycle: Establish a skeleton of our game's models / physics
7. Refine any physics/mechanics that cause initial issues
8. Begin modeling and designing with visuals, audio, etc.
9. Play test for any bugs/ gameplay exploits that may hinder a release
10. Spare time to squash any bugs, begin game refinement
11. Final play testing / makeup week for refinement
12. Finalize Github, branches, workflow
13. Prepare for presentation

Project Description:

The project at it's core will be a 2D game that will utilize Python and the PyGame library to create essentially the entire game. Our intended game is meant to mimic those such as the Google dinosaur game, with some mechanics inspired by space invaders. Given our groups experience with PyGame and 2D game development, it is very possible to accomplish our intended game mechanics. On the creative side of development, there are various websites/software available that will make the creation of modes/sound effects/etc. much easier to create and import. A project such as this one heavily relies on proper workflow to ensure

productivity and communication throughout our entire team. Below is a rough sketch on how we would like our game to function. The green in the sketch represents the bounds of the game, the red represents obstacles the user encounters, and the blue represents the user-controlled tank and it's canon shooting. Like any other game's development process, we are currently in it's pre-production and features are subject to change as we progress.

Functional Requirements:

Requirement ID	Statement	Priority	Comments
FR01	Include a heads-up display	High	The HUD will include crucial information such as inventory, lives remaining, and scores.
FR02	A functional live game score	High	The player will be able to view there progress by a numerical score system that grows the further they go along with enemies defeated.
FR03	A working player inventory for the available ammunition for the tank's cannon	High	This will also show any powered-up ammunition the player may have.
FR04	A working player inventory for the available powerups	Medium	Potential power ups include an extra hit, larger projectile
FR05	At least 2 different types of aerial obstacles (what the cannon will shoot)	Medium	
FR06	At least 2 different types of land obstacles (what the cannon will avoid)	Medium	
FR07	Workable start/exit processes	Medium	
FR08	Respectable	High	

	gameplay mechanics/physics, with minimal bugs/exploits		
FR09	Custom program icon for the executable file	Low	
FR10	Audio on and audio off	Low	
FR11	Functioning leaderboard	Low	This will potentially be similar to an arcade style leaderboard.
FR12	Customizable player models for the user	Low	
FR13	The aerial enemies will be controlled with a simple ai that will move the aerial target left to right.	High	This ai will be similar/inspired by the <i>Space Invaders</i> targets