**Project Description**

Name: Mundane Man

Description: A stickman game that combines elements of a first-person-shooter, platformer, and parkour, that is based off an indie game called Boring Man Online Tactical Stickman Shooter. The goal of the game will be to reach the top of a map while obstacles and enemies stand in your way.

**Competitive Analysis**

The game is based off the indie game Boring Man OTSC, so the game will have very similar movement and projectile physics to Boring Man. Unlike Boring Man, it will not have online servers for multiple players to shoot each other. Instead, it will only have bots to fight against. There will only be one map and one gamemode, unlike the multiple maps and gamemodes in Boring Man. Instead, Mundane Man will be a mix between a ‘climb map’, where you try to reach the end of the map, and ‘deathmatch mode’ in Boring Man. Additionally, there will not be the full selection of guns and grenades available in Mundane Man as there is in Boring Man.

**Structural Plan**

The code for the project will all be in one file, separated in the file by classes. This file will be put in a folder that also contains the sprites and images to be used in the code, as well as this design proposal and storyboard.

**Algorithmic Plan**

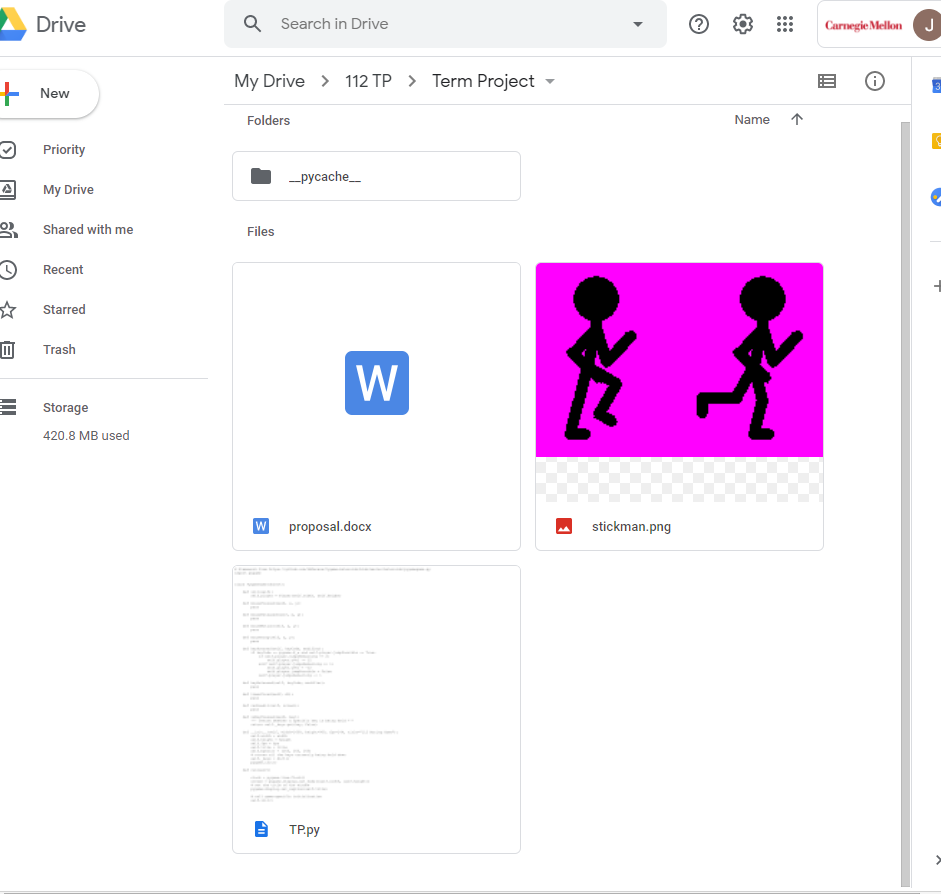
* Enemy AI – Will use Djistrika’s algorithm in order to find the shortest path to the player upon first sight of the player. When the player is in the fireline of the enemy, they will shoot at the player. To do this, will first implement Djistrika’s algorithm so the enemy can follow the player well. After that, will figure out how to make the enemy automatically shoot upon seeing the player with above average accuracy
* Gun Recoil –
  + Some guns will push the player in the opposite direction of firing while in the air. The recoil will be strongest when moving in the opposite direction that you fire the gun and if you fire a split second after jumping. It will be weakest when moving in the same direction of gunfire.
  + Additionally, some bullets upon hitting the player will push the player in the direction of the bullets if they’re in the air.
    - In order to do all this, I will need to account for all the current conditions the player is under by keeping track of a bunch of different physics variables, e.g. current velocity, acceleration, friction coefficient, and change them based on the force vectors the gun or bullet produces.
  + Some guns will also not have perfect accuracy and will instead have a spread around your reticle.

**Timeline Plan**

* Should achieve at least MVP by TP2, which will have all the movement physics and collisions accounted for, as well as gunfire and basic enemy AI. Hope to also have gun recoil physics by then also.
* By checkpoint 7, will make formal map and better enemy AI, as well as extra features like more guns and grenades
* By TP3, will have full game finished

**Version Control Plan**

Backing up my Term Project folder to my Google Drive account on my school email.



**Module List**

Pygame

**TP2 Update**

The player will only have 1 gun instead of multiple options. All the enemies will use a sword and be melee attackers that follow you on sight that will kill in 1 or 2 hits instead of having guns themselves. There will be no recoil added to recoil-heavy guns, as the player only has a pistol. The goal of the game will still be to reach some ‘flag’ at the top of the map with obstacles in the way. Most of the algorithm complexity is now based within the movement, scrolling, and firing rather than recoil or enemy AI.

**TP3 Update**

Drone enemies that fly and shoot at you with random accuracy. Sound effects and music. Better sprite animation flipping and rotating. Fire rate implemented. Flag at top of map as goal, with time it took to reach it. Respawn at bottom when you die. Sword zombie enemies randomly spawn. Better walljumping physics. Improved/harder map. Sloped tiles added. Ammo and reloading added. Gun fire explosion animation added. Second ‘impossible’ level starts after you complete the first level; has random drones as well as more zombies. Invulnerability added when respawn. Fixed libpng errors that popped up in terminal as enemies were added, using ImageMagick. Slowed down time when you reached flag. Fixed all lag in the game through optimization of rendering. Optimized enemy movement speeds.