**Project Name:** 112 Dash – It will be a recreation of Geometry Dash, an endless-runner style game where the character must avoid obstacles on its path by jumping. The path also gets faster the longer you play as well. The obstacles on the path are also randomly generated but the path that these obstacles form is still winnable.

**Competitive Analysis:** Some of the similar Geometry Dash projects I saw online used scratch and/or tried to make an exact recreation of the original game. What I intend to do as an extra feature is implement a split screen local multiplayer functionality so that two players can play the game at the same time.

**Structural Plan:** Begin with a menu screen which will allow you to select difficulty and whether you want single player or multiplayer. Then it will transition into the game. Once a round of the game ends, you get the option to start again or go back to the main menu screen.

**Algorithmic Plan:** The trickiest parts of my project I think are implementing the randomly generated (yet winnable) path and applying the game in a split screen style for multiplayer.

* Randomly generated path:
  + Create sprites for the different obstacle sets the level would have.
    - Store these sprites in a list and call them randomly but with certain restrictions (like excluding certain obstacles from the list after a particular obstacle to ensure the path remains winnable)
  + Call them to the screen by switching out the image attached to the obstacle surface and updating the rectangles for those obstacles.
* Split screen multiplayer:
  + Create a line for the split screen in the middle of the screen and use that line to mark the start/end point of each player’s screen
  + Perhaps call on separate sprite groups for player 2 so that collision detection works according to the player.

**Timeline Plan:**

* **By 18/11/2021:**
  + Set up menu and options for player to select (maybe set up multiple levels?)
  + Completed a working path that calls on random obstacles
    - Maybe make it winnable too
  + Be able to set difficulties and play accordingly
* **By TP2:**
  + Get split screen functionality for multiplayer working
* **After TP2 before TP3:**
  + Music for levels, start looking at online multiplayer or introduce character customization.

**Version Control/Backup Plan:**

* Will be using a GitHub repository as a backup space to store all my files needed for the project
* Can also split different versions of my code with branches (which I have not needed to use yet)
* Graphical user interface, text, application, email, website

  Description automatically generated

**Module List:**

Only using one external module: Pygame

**TP2 Update**

Structural plan was missing some information which is mentioned here:

* One file contains the GameObject class, a parent class which is used to make the sprites for characters, floors, backgrounds, and obstacles
  + One file is also dedicated to each of the subclasses mentioned above:
    - Player.py for Player class
    - Floor.py for Floor class
    - Wall.py for moving background class (called Wall)
    - Obstacle.py for Obstacle class
      * Block.py for Block class which is used to make each individual block sprite per obstacle
* Another file is used solely to store the data for the different types of obstacles made called Obstacles.py
* Main game is run on Game.py, which handles obstacle generation and sprite interactions including collisions.
  + Game.py uses a base framework class called Game from the pygamegame.py file.

**Design Changes:**

* Randomly generated path:
  + Each obstacle consisted of sprite groups of block sprites as opposed to single sprites, found that these were easier to use when removing obstacles once they reach the end of the screen. However, each of the obstacle sprite groups are separately stored in an array within Game.py which gets regularly appended to or removed from.
  + Each obstacle sprite group is loaded onto the screen as they get appended to the obstacles array and new obstacles are added to the array after a set time has passed since the last obstacle appeared on screen. This time between obstacles changes depending on the obstacle on screen.
* Making a winnable path:
  + Due to the extensive time taken to create a traversable random path, I will change the game format from a winnable path to an endless runner style game where the longer you last, the more points you gain.
* Online multiplayer:
  + As opposed to making online multiplayer for TP3, I will implement the split screen multiplayer as part of TP3 for a similar reason for not making my obstacle path winnable, just traversable. It would take too long and I would not be able to get it working by TP3.