**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