Ian Jeannin
11/30/2020

Proof of Concept Log:

- Create Player and PlayerController. PlayerController handles input, Player handles movement variables such as speed and speed multiplier (for when score is eventually implemented)
- Create a basic arena script. Holds a prefab for the arena. Player will check if the player is in the bounds of this arena. (TODO: Generate a grid for the arena, to allow for different maps or changing the map mid-game)
- Change Player sprite color when off of the map. This simulates losing.
- Add Wall and WallManager scripts. The Wall script contains movement functionality, while WallManager handles spawning. Walls randomly spawn from one of the directions and move towards the opposite side.
- Change player color when crushed between walls (also simulates losing). To do this, include a TriggerCollider inside the wall prefabs that the player can only hit when crushed.
- Add a ScoreManager with static variables that any class can access. (Add an object that only contains a TriggerCollider to the prefab. Player checks for the ScoreTrigger tag to add a point when hitting this TriggerCollider).
- Instead of changing color, have the player reset to the center of the arena and all walls delete/start spawning again when the player dies.
- Increase player and wall speed whenever the player

Vertical Slice Changes:

- Add grid creation in Arena script. This allows for individual pieces of arena that can be broken off, to make the play area more interesting for the player.
- Add ArenaSquare script. This controls individual pieces of the arena grid. Contains an enum SquareState that determines if the player is currently on the square, if it's in the process of breaking, or if it is regenerating. This forces the player to also think about their pathing in order to try and conserve as much of the arena as possible.
    - As a side effect of this, the player now checks if they are on an ArenaSquare game object instead of an Arena game object. If they are not (went onto a broken square) the game resets.
- Added color transitions on individual squares. While the player is on the square and its durability is wearing away, it will slowly transition into red. While the player is off of it for a time, it will transition back to green. This gives the player visual feedback to show them how the mechanic works and what areas they need to avoid.
    - TODO: Regeneration is slightly bugged, and squares sometimes stop changing visually after regenerating. Cause to be determined.
- Add a button (Escape) to close the application on build. This makes it easier for the leaver to quit the game.
- TODO: Determine how to change Wall movement to make it as enjoyable as possible. Make the randomization formula more interesting. Add more variations of walls.

Final Changes:

- Prevent walls from progressively spawning faster each reset, eventually leading to infinite walls. (Reset the spawnTimer back to its initial value.)
- Fix ground colors not correctly showing to Improve visual clarity/feedback. (Regeneration Lerp wasn't taking into account the regeneration rate, leading to a discrepancy between the durability and the color; changed formula to be more directly related to durability value)
- Add more possible walls for variety (Created more prefabs).
- Prevent reset when player hits a wall with sufficient speed (Made hitboxes smaller and adjusted speed values to reduce cases of this. Still possible if player makes it far enough)
- Add a diminishing growth to speed/wall-spawn-rate values, with a cap at 50. (Should fully prevent player colliding with a wall and resetting.
- Scale UI with screen size (Using Unity canvas settings.
- Add Panel for Menu (Rather than having a different scene for just the menu, have a panel that opens/closes whenever the player dies or pauses.
- Add multiple difficulties, with different speeds/wall cooldowns/square durability. (Accumulated debt, difficulty should be moved out of the menu class to gain more separation of concern)
- Change ResetWalls/Reset methods so the game can function through the menu multiple times. (Pausing the game while in menus caused errors in the reset methods, necessitating adjustments)
- Change square regeneration and durability loss to be determined by OnTriggerStay to better work when the game is reset. (Same as previous issue)