Progress Report
- Increment 3 -
Group #12

1) Team Members

Abigail Centers -- alc16j, GitHub ID: abbeyLC

Jason Hamilton -- jvh16, GitHub ID: Jakuu

David Song -- ds15g. GitHub ID: Sawwas99

Paul Santora -- ps15f, GitHub ID: LeavepaulS

2) Project Title and Description

Project Title: Maze Game Using the Unity Engine

Description:
This 2D computer game will allow the player to move about each maze level while collecting timers, health, and powerups. Timers will add more time to the allotted time for that level and health will allow the player to recover from an enemy encounter. The player must dodge any enemies that move about the maze and power-ups will increase the player's overall score. Every level will begin with the player avatar at the start position of a new maze. With limited visibility of the maze, the player's main objective is to use the arrow keys to navigate the maze and find the exit within the allotted time while also completing any challenges for that level.

For example, to successfully win the game, the player might be prompted to collect a hidden item. If this objective is not completed first, the player cannot win. A scoring system will also be used to measure the player's overall performance for each level, which will be primarily based on the number of power ups collected during that level. A leaderboard will also be used to store the top 10 highest scores.

3) Accomplishments and overall project status during this increment

Jason H. (Increment 3)

Added support for additional objective types, there are now scripts that support "constrained" objectives, for example the levers in a level must be activated in a certain order for completion. Added objectives to each level, increasing in complexity from level 1 to level 4. Playtest/bug fixes in preparation for final presentation. Fixed level completion to once again check for objective status. Fix layout scaling for objective tracker, the tracker now properly handles adding more objectives, and supports self-populating text for different objective types.

Abbey C. (Increment 3)

Implemented a leaderboard that can track the top ten scores for each level. This feature operates like an arcade high score table by allowing the player to enter his or her username if the score is high enough to be listed in the top ten. If the player's score is not high enough for that level, the username prompt does not appear. High scores are stored in the project using PlayerPrefs and are reloaded between runs. Also wrote two enemy scripts to automate enemy movement. Enemies can now be easily configured to move either horizontally or vertically and will change direction if a collision is detected between the enemy and a wall of the maze. Created four complete levels (including the bonus level) by designing the layout of the maze walls and the position of each collectable item. The main difference between the bonus level and a regular level is the added gravity component, which allows the player to push against the walls and cause the entire structure to move. Other items scattered throughout the maze (enemies, clocks, and timers) are also affected and will begin to drift and bounce around the maze. A significant amount of play testing and debugging was also completed during this increment.

David S. (Increment 3)

Added layer collision during health damage so Player can ignore Rigidbody collisions, will also flash red on damage. Used Coroutines to set up wait periods so player can time out certain actions (invulnerability, respawn). Some added mechanics were not used in the final version of the project, added to projectile branch. These include 2 new Enemy types, a turret that fires projectiles at the player, and an enemy which track players position. Uses Vector3 manipulation to control turrets positioning and transform to track the players movement. Also created multiple lives/respawn mechanic.

Paul S. (Increment 3)

Completed all sprites for game including title screen and level backgrounds, walls, player, enemies, coins, clocks, levers, goals, safezones, and game controls. Created animations for player, enemies, coins, clocks, levers, title menu background, and goals as well. Added original background music that loops throughout the entire game. Put walls sprites into a tile palette that can be reused in new levels. Completed all menus, including adding controls to title menu, and a menu for when the player wins and if they lose for each level. Created a level template that includes the basic layout and design to be used in each level. Helped edit levels 2 to 4 and created level 1. Set the build settings for the game including the order levels are played in.

4) Challenges, changes in the plan and scope of the project and things that went wrong during this increment

This iteration we have not seen a major change in scope other than limiting some animation/visual development in order to focus more on gameplay

- Reduced the number of music/animations/powerups focus
- Kept minimum amount of levels. Added a restart and respawn mechanic
- We created more objectives to complete per level (levers, gates).
  We added a health/damage tracker, removed some extra enemy types from presentation.

5) Team Member Contribution for this increment

Discussed and Wrote Progress Report, RD document, and IT document as a group.

*a) the **progress report**, including the sections they wrote or contributed to*

Project Title and Description- Abbey C., Accomplishments and Overall Project Status- Jason H. Abbey C., Challenges changes in the plan and scope of the project and things that went wrong- Abbey C. Jason H., Plans for the next increment- Everyone

*b) the **requirements and design document**, including the sections they wrote or contributed to*
Overview- Abbey C., Functional Requirements- Everyone, Non-Functional Requirements- Everyone, Use Case Diagram- David S. Abbey C., Class Diagram- David S., Abbey C., Sequence Diagram- David S., Operating Environment- Everyone, Assumptions and Dependencies- Everyone

*c) the **implementation and testing document**, including the sections they wrote or contributed to*
Programming Languages- Abbey C., Other technologies used- Everyone, Execution-based functional testing- Everyone, Execution-based non-functional testing- Everyone, Non-execution based testing- Everyone.

*d) the **source code** (be detailed about **which** parts of the system each team member contributed to and **how**)*

Abbey Centers (Increment 3)
Contributed to the following scripts during Increment 3:

1. BonusLvl_TriggerEvent.cs
2. Clock_Weighted.cs
3. PowerUp_Weighted.cs
4. EnemyMvHorizontal.cs
5. EnemyMvVertical.cs
6. HighScoreTable.cs
7. TriggerEvent.cs.

The first three scripts are needed to accomodate the gravity aspect of the bonus level, and the enemy movement scripts are needed to configure the direction that an enemy should move in (horizontal or vertical). HighScoreTable.cs uses PlayerPrefs and JSON strings to load and store leaderboard content and maintain that data between runs. TriggerEvent.cs was modified to prompt the user for a username if their score made it to the leaderboard (for displaying the top ten highest scores for each level).

Jason Hamilton (Increment 3)
Contributed to: KeyTrigger, ButtonTrigger, ItemObjective, ObjectiveTracker, GetQuests, Multi and Seq Objectives, Objective Collection, Sequence Tracker and Sequence Lever scripts. Finished objective display implementation, it now properly handles text population for different objective types, and now scales up with additional objectives in each level properly. Key and Button trigger scripts were updated to handle the additional objective types and fixed interactions between these scripts and the objective tracker. Seq and Multi objective/tracker scripts handle the interaction between the UI quest tracker and the new objective types added.

Paul Santora (Increment 2)
Contributed to the scripts -- MenuButton, MenuController, PauseMenu, PauseMenuButton, TitleScreen, PlayMusic, CountdownTimer, PlayerMovement, and TiggerEvent.

The first four scripts had to be updated with the additions of new menus, adding winMenu, deadMenu, and the player controls menu. PlayMusic sets the music object to “do not destroy” meaning that it will not be destroyed between scenes as Unity automatically does. This way the music will keep looping until the

application is closed. For the last three scripts, added exit conditions for in level, when the player loses and wins. Specifically when they hit an enemy, run out of time, or reach the goal.
David Song (Increment 3)
Contributed to: PlayerMovement(lives, layer collision ignorance, temporary invincibility), Turret (creates and destroys projectiles, follows player), Projectile(aims at player), ChaserEnemy(tracks player's position). The new Turret type Enemy uses several Vector3 variables to track and constantly face the player's direction, it will automatically create and destroy projectile type objects when the player is too close to it. Projectile object will track the player's last known Transform.position and move towards it. Added several new functions to PlayerMovement, Invincibility, done through layer collision detection, after colliding with an object from enemy layer Player layer will ignore all objects of that layer type for several seconds. Player avatar will also flash red when this happens (taking damage) to show they are in a period of invulnerability. Created lives to respawn the player at each death at the set Spawn point. After final life is gone The Game Over screen will appear on death that will redirect to different options (Restart level, Quit game, etc).

e) the **video or presentation**

Paul Santora- Gameplay Demo, Jason Hamilton- General Overview, Abigail Centers- Short description of Project state and what was accomplished during this increment**.** David Song- Describe any changes in the scope from the initial proposed project plan, talked about code that was unused/not included.