Progress Report
- Increment 1 -
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, multi-level computer game will allow the player to move about each maze level while collecting timers, health, and powerups. Timers will add 5-10 seconds 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 randomly 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 may 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 time it takes to complete the 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.

Implemented base interface + classes for victory and win conditions, currently item retrieval is the only supported "quest" as a victory prerequisite but later iterations will implement more varied objectives extending the objective interface. Item retrieval is currently implemented through player collision detection and its status is stored in its own ItemRetrieval class. The base functionality of this interface should be able to handle most of the planned "Challenges" and obstacles we will implement in our game, with the more complex interactions possibly having player input required in addition to collision. The current implementation of objectives is functionally independent from the victory timer scripts, as well as extending one base interface, so this current design is modular and can very easily be iterated over and replaced or extended as needed.

Abbey C.

Abbey implemented a possible maze design for the first level and 3 C# scripts for handling player movement, an exit maze trigger, and a timer countdown trigger. Player movement is handled with the keyboard arrow keys, and a player avatar is unable to walk through the walls of the maze. The timer begins at 20 seconds. If the timer reaches 0 before the exit is found, the text "Game Over!" will be displayed. However, if the exit is found before the timer reaches 0, the "You Win!" message will be displayed. The implementation of these features will be helpful as we design other levels with more sophisticated event triggers like enemy movement, health collection, etc. Abbey also implemented camera movement, which causes the camera to move with the player as it moves about the maze. Much of the first iteration was also spent learning the Unity engine and learning how to write C# scripts.

David S.

Studied Unity tutorials in order to gain an understanding of the engine and the tools necessary to construct the project. Built a tutorial level to give user's a rundown of fundamental gameplay including instructions and pause. Centered camera/text around the avatar to follow player movements properly. Implemented level progression into the Event Trigger created by Abbey, allows for the smooth transition of new scenes, with this trigger implemented New Levels can be designed and tested at a much faster rate. Developed Use Case, Class, and Sequence diagrams that outline the current state of our project. Overall Project status is we have developed a strong foundation for our project, and rapid progression in terms of level design and other goals for our second term iteration are promising.

Paul S.

Introduced the scene structure into the game and adding the title screen as the first scene of the game. Clicking 'start game' on the title screen advances the player to the first level of the game. Set the sample scene as the first level of the game, and created the basic pause menu layout to be used during gameplay. The player can pause during the game, stopping the game time. The player can resume the game and quit the game, bringing the player back to the title screen. Also took the time to learn a good amount about unity, what its strengths are as an engine, and how many unity projects are structured.

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

Abbey C.
We initially planned to use Unity/C++ to build the maze game, Django/Python to build the web interface, and MySQL for the backend database. However, we have learned that the Unity engine uses C#, not C++. We have also learned that Django, Python, and MySQL will not be necessary to build the interface or store leaderboard scores. The Unity engine can be used to implement all of these features.

Jason H.
Having little experience with Unity, I had originally planned to design most of the backend features with other systems, but since most functionality is provided by the Unity engine most of the time spent in this first iteration was spent familiarizing myself with all of Unity's features and tools, as it will be easier to build all on one platform in the long run, rather than designing a totally independent leaderboard/web interface.

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., Abbey C., 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

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

Abbey used the following online tutorials to help implement the CountdownTimer.cs, TriggerEvent.cs, and PlayerMovement.cs files. Together, these scripts are used to handle player movement using the keyboard arrow keys, an exit maze trigger for displaying a victory message, and a timer countdown trigger to display a game over message. Abbey also implemented limitations for the player so that the avatar is unable to move through the walls of the maze, and camera movement so that the player can see his/her current position as the avatar navigates the maze.
https://www.youtube.com/watch?v=o0j7PdU88a4
https://www.youtube.com/watch?v=whzomFgjT50
https://www.youtube.com/watch?v=CNNeD9oT4DY

Jason H.
Created basic item collection objective class as well as the base IObjective interface that all future quests/objectives will implement. Modified the victory/goal object to now check the status of all required objectives so victory may not be achieved until all mandatory objectives are completed. Currently there is only one objective type, (ItemObjective) that implements IObjective, however additional types will be implemented before the next iteration. KeyTrigger was also implemented and attached to the "key" object, as a prototype for how objective triggers will be tracked and activated.

Paul S.
Added a basic title screen as the first scene in the project, as well as a pause menu that the user can access during gameplay. The title screen has a start game button that transitions to the first level of the game. The pause menu is accessed by pressing the 'p' key, and gives the player the option to resume the game by clicking 'resume' or by pressing the 'p' key again. The player can go back to the title screen scene by clicking 'quit' on the pause menu.

David S.
Added a Tutorial level to the project, with instructions of the controls. Added level progression to the event checker to trigger change in scenes.

e) *the **video or presentation***

Paul Santora- Video Editing, 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, and Describe goals/plans for the next project increment.

6) Plans for the next increment

By our next Iteration (2nd iteration) our project goals include implementing Challenges (new levels, objectives to complete, powerups), a Scoring System (Way to record players progress), Difficulty (Checkpoints, Saves, Penalties, Shorter timers, Larger Mazes), Settings (Volume, Controls). Some possible goals to complete (Music, Unique Enemies/Powerups, Game speed, More objectives). Implement more varied objectives, as well as status tracker/goal display for the user that will update as goals are completed. Also implement a reset/paused state after victory achieved.

7) Link to video

https://youtu.be/lyzTZ2fD9fc