**Mission: Labyrinth  
Impossible Mazes(?)**

# Our Idea

* Game has multiple levels which gets harder as it progress
* Revolves around controlling an avatar clearing puzzles in a maze that can be different
* 2D game
* A combination of a puzzle in a maze.
* The game restarts whenever the player quit halfway through or Game Over.
* To complete a level, player have to be quick and avoid obstacles such as cannonball and monsters to be able to get the keys to unlock the gates to the exit.

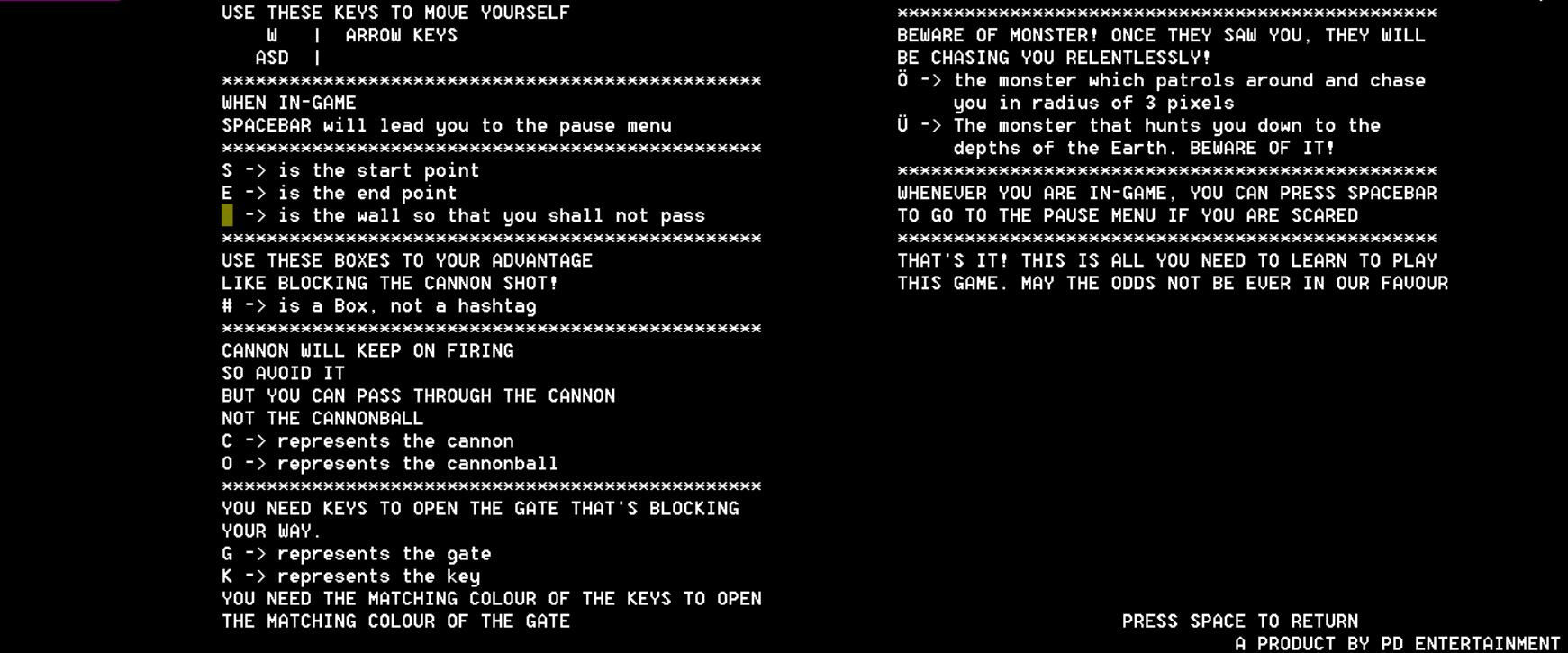
Project Schedule and timeline of the task breakdown

* First week:
  + Creation of the core gameplay which is the maze. Timeline of the task: 1 week.
  + Creating of the character’s controls and movements. Timeline of the task: 1 day.
  + Creating of the UI to make the transition of the between different states to be possible such as from menu to starting the game. Timeline of the task: 5 days.
  + Creation of the Level Editor with all the features that will be implemented. Timeline of the task: 2 weeks.
  + Creating of the cannon and cannonballs. Timeline of the task: 2 days.
  + Interaction between the things such as cannonball and the player, Starting point and the player, and the End point and the player. Timeline of the task: 2 days.
  + Collision detection between player and certain objects such as wall. Timeline of the task: 2 hours.
* Second week:
  + Inventory system
  + Creating of the level editor and Saving of the level editor. Timeline of the task: 1 day
  + functional levels with correct Interactions. Timeline of the task: 2 days
  + creating new levels and use all the available things. Timeline of the task: 1 day
  + Integrating pressure plate and hatch into the maps. Timeline of the task: 1 day
* Third week:
  + Debugging of certain codes to ensure that the game is bug-free. Timeline of the task: 4 days.
  + Creating of a teaser for our game. Timeline of the task: 4 days.
  + Creating of the presentation slides. Timeline of the task: 2 days.
  + Ensure there are music features inside the game and sound effects for different kind of interactions such as monster and player, and Cannonball and player. Timeline of the task: 4 days.

# Screenshots

## Start Menu

## Help Screen



## Level One



# 

# Features

* Alternate Controls
  + Players can now use WASD on the keyboard along with the arrow keys to control the player character. This was done to let people more comfortable or accustomed to FPS games choose between WASD and the arrow keys.
  + Both WASD and the arrow keys may be used to control the player character at the same time.
* Collision Detection and Reaction
  + The Collision Detection and Reaction system allows the player to walk through the maze without passing through the walls.
* Multiple Levels
  + Mission: Labyrinth features several levels throughout the game, not just a single maze.
* Block
  + Blocks are represented by a “#” symbol in the game. They can be pushed around the map by the player and can be used to trigger pressure plates or block cannon balls.
* Pressure Plate and Hatch
  + Pressure plates are represented by a “P” in the game, when the character steps on it or when a block is placed on it, the Hatch represented by a “H” will open, allowing the player to pass.
* Enemies and Traps
  + There is a cannon trap which can shoot up, down, left or right in intervals. It will also not pass through walls or blocks.
  + Monsters with basic AI are able to move around the mazes and kill the player.
  + One monster will constant follow the player based on the coordinate.
  + The other monster will move randomly based on random number generation, but when the player comes within it 3 units range radius, it will go towards the player.
  + The monster have collision detection and will not go into walls or blocks, so players can trap monsters by moving blocks strategically.
  + The speed of the cannonball and the 2 monster is slower compared to the player to make sure the player cannot run with the cannonball at the same frame rate and to nerf the monster since it is moving too fast.
* Gate and Key System
  + This feature is to provide an obstacle in the player’s path and player may have to collect all the keys to unlock the gate.
  + To unlock the gate, player have to collect the matching colour of every key that is scattered around the map.
* Customizing your own levels
  + Players are able to recreate their own puzzles with the things inside the game
  + Player will be able to try out their very own level they created in the “play custom level” option in the menu screen
* Music Features and Sound Effects
  + The Background Music (BGM) will be played and looped throughout the lifetime of the program. However, it will be lowered down when the player is not on the main menu so that the player can focus more on the game itself and the BGM does not interfere with the sound effects.
  + There will be sound effects which when the player is hit by a cannon, an explosion sound can be heard.
  + Another sound effect is when the player is chased by a monster, a scary sound will be played to warn the player that he/she is being chased by a monster, and the sound can only be stopped when the player is not within a certain range of the monster. The range varies on the type of the monster.
  + There will be 2 different sound effect being played when the gate is open or the hatch is opened.
  + The BGM and Sound Effects is taken from these sources:
    - <https://www.youtube.com/watch?v=9VPZAEtq2_o>
    - <http://www.freesoundeffects.com/free-sounds/doors-10030/>
    - <http://dova-s.jp/>

# 

# Knowledge Applied

* Condition Statements such as if
* Enum
* ifStream/ofstream
* switch statements
* Nested Loops
* 2D arrays
* Pointers
* Structs
* Loops
* Array and Pointer Array

# Problems Encountered/Problems Solved/Lessons Learnt

* While adding WASD to the controls, it was discovered that if the player pressed the same direction with both WASD and the arrow keys, the player would move at double of his proper speed. This has been fixed.
* The C-Style String just keeps on crashing the program./ Solved by adding a null to the end of the string./ Always remember to add the null whenever we are using C-style string.
* Trying to immigrate the code into another cpp file crashing program
* Trying to read and print the data in the txt file
* Due to the change of framework, new codes were needed for the monster and traps to work.
* Using IrrKlang as an external music library source to add it into our codes generate errors for the command prompt/ The problem is solved by going to the page properties of the project and add the file’s library location to the C++ and Linker.
* The use of IrrKlang causes multiple sounds to be repeated and can causes lag if there are too many sounds being played at the same time/ The problem is solved by reading through the documentation of the irrKlang carefully, implementing *engine->isCurrentlyPlaying(themusicname)* can help to check whether is the music playing. If not, it will be played.
* There was a problem of not able to spawn multiple of the same cannon types, and have to change codes.
* There was a problem where monsters, cannons, and block were not reseted after changing level. Solved by having a new function to reset the coordinates of the objects.
* Difficulty in merging the different things together and making them work.Solved by passing coordinates of the map to the different items
* There was a problem with reading what the player typed directly into the notepad when creating the level editor.Solved by putting whatever the player type into a 2d array and then transfer it to a notepad
* There was a problem with having more then one of the same thing like cannons.solved by using coordinates array for the cannons
* There was a problem with player not interacting with the End point going to the next level. Solved by reloading the map after incrementing the level variable so it loads the next level
* There was a problem loading the victory level screen where the screen hangs and becomes non-interactive. Solved by changing the way we code and being able to call the screen at the correct time.
* There was a problem with the level in the main game and customise level having crashes and wrong saves. This is fixed by using different variables and checking for the correct requirements
* There was a problem with reading the text file and converting whatever read to become the level, Solved by creating a converter

# Future Improvements

* Artificial Intelligence(done)
  + Although the monsters already have basic AI, Zhao Yuan will continue working on improving them.
* More Enemies and Traps
  + There will be more varieties of enemies and traps in the final version of Mission: Labyrinth.
* Level Editor
  + The player will be able to create his own levels for Mission: Labyrinth in the future.
* Velocity
  + This will allow us to make the player, monsters and cannon balls move at different speeds, making the game more interesting.
* Inventory System
  + The player will be able to pick up a key from the maze and use it to open a gate.
* Music Features
  + Music will be played depending on the state that the player is currently in. If the player is at the main menu, the theme song of the game will be played.