Maze Champion – Final Project Report

Team members

1. Piyush Mungre
2. Tejas Sane
3. Sriharsha Srinivasa Karthik Kaipa

Gameplay Video Links

http://youtu.be/zZ7lrd181iA

http://youtu.be/4OrT0-LecRY

Introduction

The Maze Champion is like any other typical maze solving game that expects the player to start from a certain position on the map and reach the destination point maneuvering the maze that has both obstacles and dead ends. However it also implements additional mechanics that make the gameplay experience much more challenging and immersive.

Mechanics are the basic building blocks of any game. These include setting of the ground rules, game objects that live within the game space, capabilities and limitations of the player etc. Thus mechanics have the capacity to influence the overall pace and operation of the game. The following aspects of the Maze Champion can be categorized under game mechanics:

Multiplayer Mode*:* Maze Champion supports multiplayer mode i.e. two players can play simultaneously. Maze solving games that involve multiple players rather than a single person are more interesting as they encourage competition between players by giving them the additional challenge to finish the level before the other player does.

Enemy AI: Although solving the maze is a challenge in itself, implementation of AI based enemy that portray certain characteristic behavior make the gameplay experience even more challenging. These enemies are displayed as black blocks moving in the maze. Apart from the basic challenge to reach the goal, players also needs to adjust their strategy so as to avoid enemies patrolling the maze. If an enemy collides with the player, he/she is re-spawned at the starting point.

Lock & Key: Maze Champion also implements lock and key mechanism that expects player to capture his/her colored key (either green or red) in order to unlock the goal which is hidden at the start of the game. The placement of the key also depends on the player activity (such as preferred direction). Also once the gate (or lock) is unlocked i.e. visible, enemy objects start guarding the gate by altering their patrol area.

Blocks: Players during gameplay have the power to place blocks in order to stop both the opponent player and AI controlled enemies. If an opponent player or AI controlled enemy steps onto a block placed by the other player, they become immovable for 5 seconds. After the 5 seconds, opponent player can move as before but the AI controlled enemy goes out to seek and kill the player that placed the block. Each player gets 5 blocks to place at any position in the maze that he/she desires.

Procedurally Generated Maze: Players have a novel gameplay experience every time they play as the generated maze is different for each new game. This avoid boredom caused by repetition in the game due to the lack of newer content to explore and play with.

Motivation

Procedurally generated games aim at providing their players with near infinite content that gives them newer experiences while exploring the game. This aspect of re-playability is an area of interest to both researchers and game developers alike. This motivated us to develop a game that uses simple procedurally generated content such as mazes.

Maze Champion also intends to give its players a new and challenging experience with every gameplay session. Integration of AI based enemies into a world that is procedurally generated, encourages the player to learn newer ways to play and improve his/her strategy to get better. Also the multiplayer aspect encourages players to change their strategies based on the opponent's strategy too.

Related Work

The other projects that we saw during the prototype demonstration class also had their game world procedurally generated, although these projects may have used different algorithms to generate content. From the point of view of aesthetics, these projects used Unity game development platform as compared to our Processing based display which surely gives more options to the developer.

The main objective of using PCG in all our games was to give the player something different and exciting each time. While one project like us had the procedurally generated maze others made a dungeon like world with interconnected rooms which can further be developed into a fully operational game.

Also while making the AI enemy for the Maze Champion, we were inspired by the ghost AI used in the Pacman framework.

Implementation

1. Growing Tree Algorithm: The game world is a maze of (35 x 35 cells) which is generated using the Growing Tree Algorithm. Growing Tree Algorithm ensures that the generated maze has accessibility from every node in the maze to every other node in very short time with relatively fast performance and reliable results.
2. A-star Algorithm: For the implementation of enemy AI we chose to use A-star algorithm, which finds the path from the start position of the enemy object to its goal position. The enemy keeps moving between the start and the goal giving a sense of a guard patrolling an area.
   Although once the player acquires a key, the patrolling area of the AI enemy is altered and then it guards the gate instead. This effect is achieved by simply setting one of the patrol points of the enemy to the gate of the player.
3. Breadth first Search: For the lock and key mechanics we chose to use the Breadth First Search algorithm. It takes into account the position of player when he/she has travelled 20 steps in the maze. Taking this position as the start point, the key is placed at a location obtained by the output of breadth first search algorithm.

Similarly, the gate is also placed using the breadth first search algorithm that takes the position of the key as the start point and places the gate at the location obtained from the breadth first search.

4. Collision Detection: In the maze there are walls that mark the unreachable locations. To avoid the player going to those places, we check if the player is standing right next to the wall. In such a position, if player tries to move through it by pressing any arrow key, that key press has no effect and the player effectively stays at the same position. Also we check for collision between AI enemies and the players by checking if the location of enemy and the player in the maze is same then the player is killed and is re-spawned back at his/her starting point.
5. Freezing opponent and enemy: If an opponent or AI controlled enemy steps onto a block placed by the other player they are frozen for 5 seconds. This is achieved by keeping the location of player fixed for those five seconds and ignoring any movement keystrokes made by the player.

Enemy AI Architecture

In a broader sense, this project uses Finite State Machines (FSM) as the base architecture for the enemy AI. These states are shown in the below diagram.

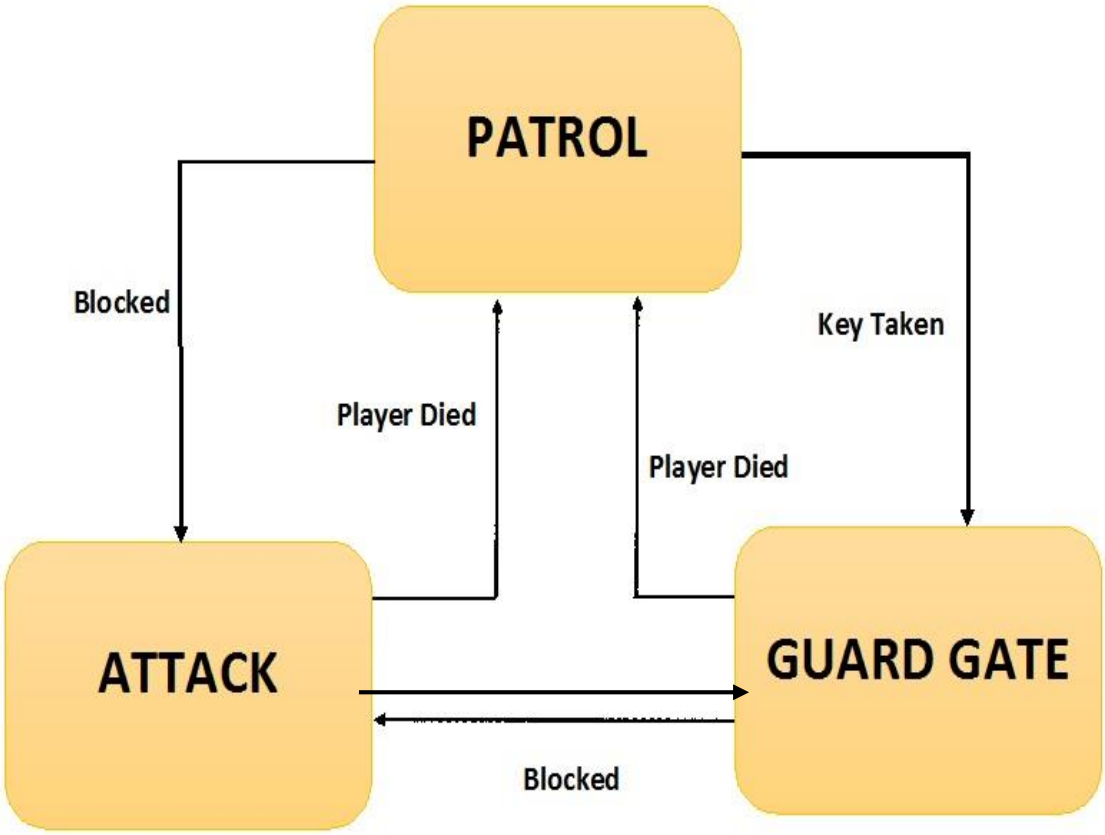

Fig 1.1 Enemy AI Architecture

Based on different trigger conditions, the AI enemy transitions among three states of patrol, attack and guard. For example: When the AI enemy is in the patrol state and player takes the key, the enemy would alter its path and guard the gate. Thus, the event of player taking the key is the trigger that makes the AI enemy transition from the patrol state to guard state. Now suppose that the enemy is guarding the gate and the player in pursuit of the gate collides with the enemy and dies, the AI enemy would again transition from the guard state to patrol.

Screenshots

Fig 1.2 Home Screen

This screenshot shows the screen that players see at the start of the game. It explains all the rules to the player and also the control keys for navigation and placing blocks.

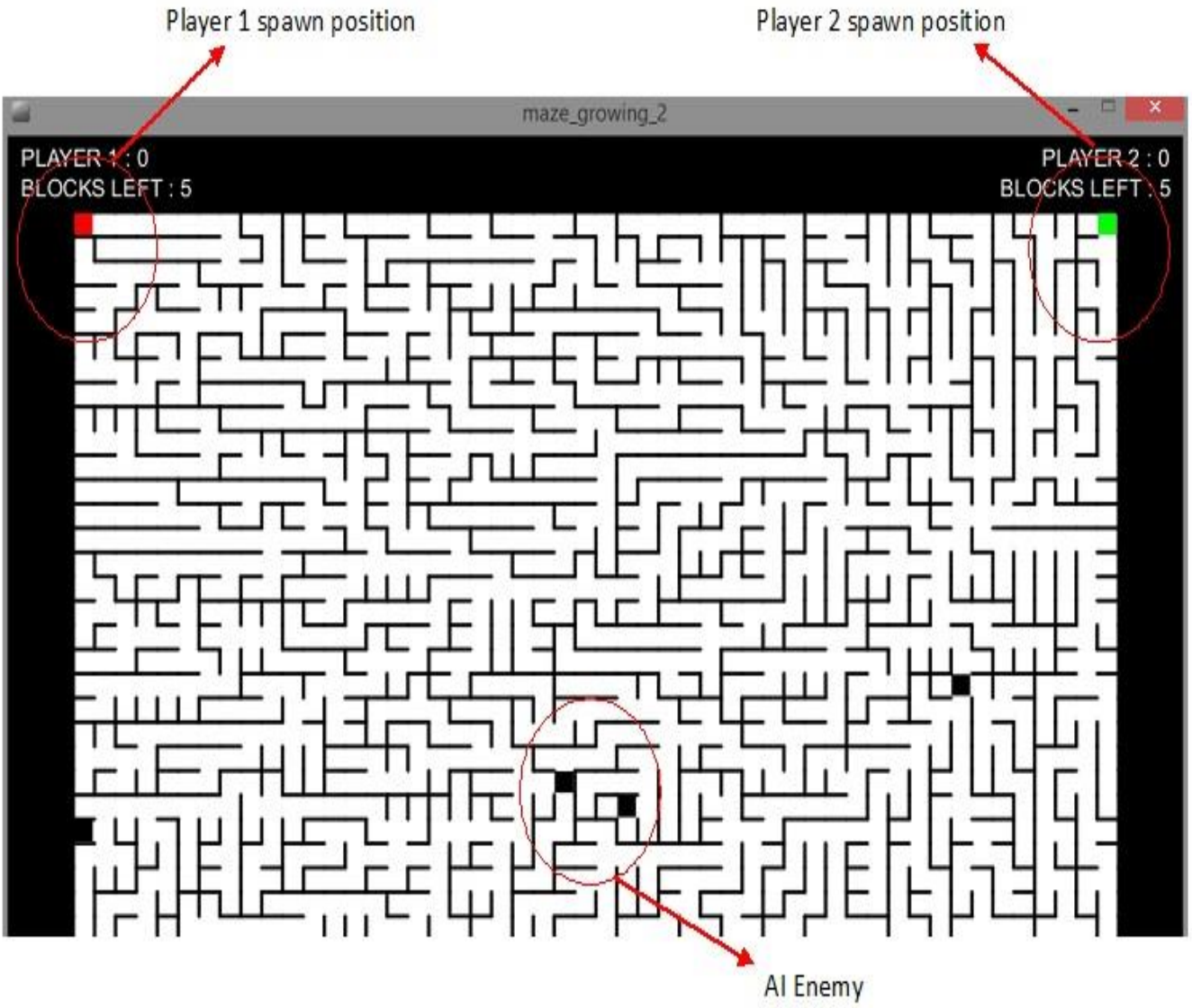

Fig 1.3 Gameplay Screenshot

This screenshot shows the two players displayed as the red and green squares and the patrolling enemies in the black color. The players are initially at their respective spawn positions.

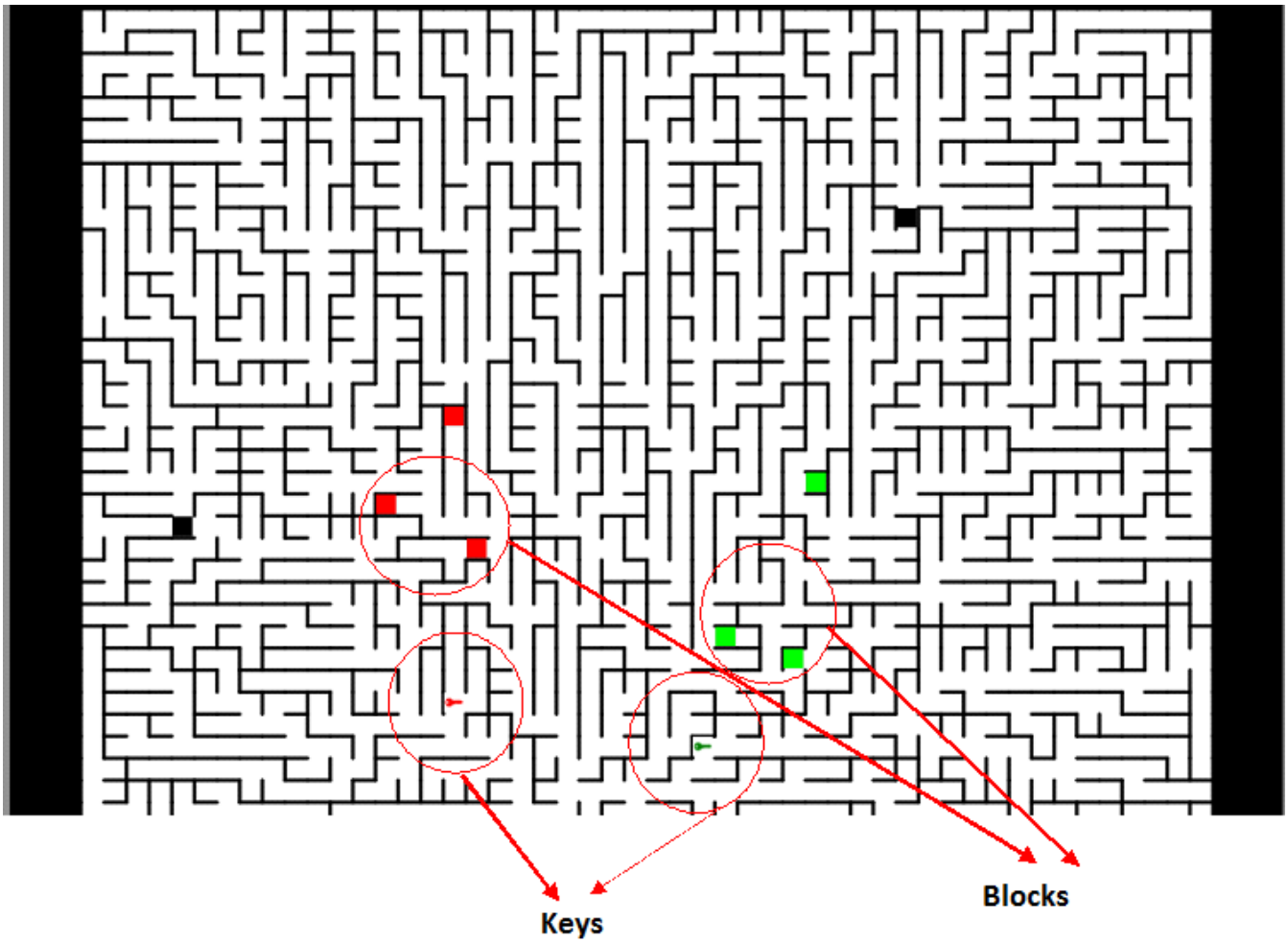

Fig 1.4 Player Keys and Blocks

This screenshot shows the player blocks that are placed to freeze the enemy for 5 seconds. The key is for each player is placed after the player has moved 20 steps in the maze. Taking the position of the 20th step of the player as the start point, the key is placed using the breadth first search algorithm.

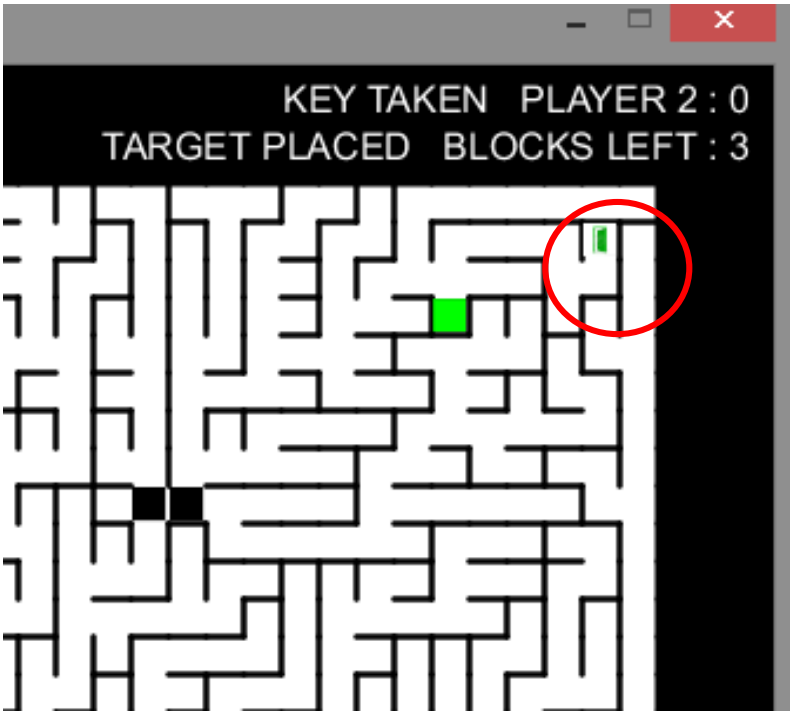

Fig 1.5 Player 2 Gate Unlocked & Visible

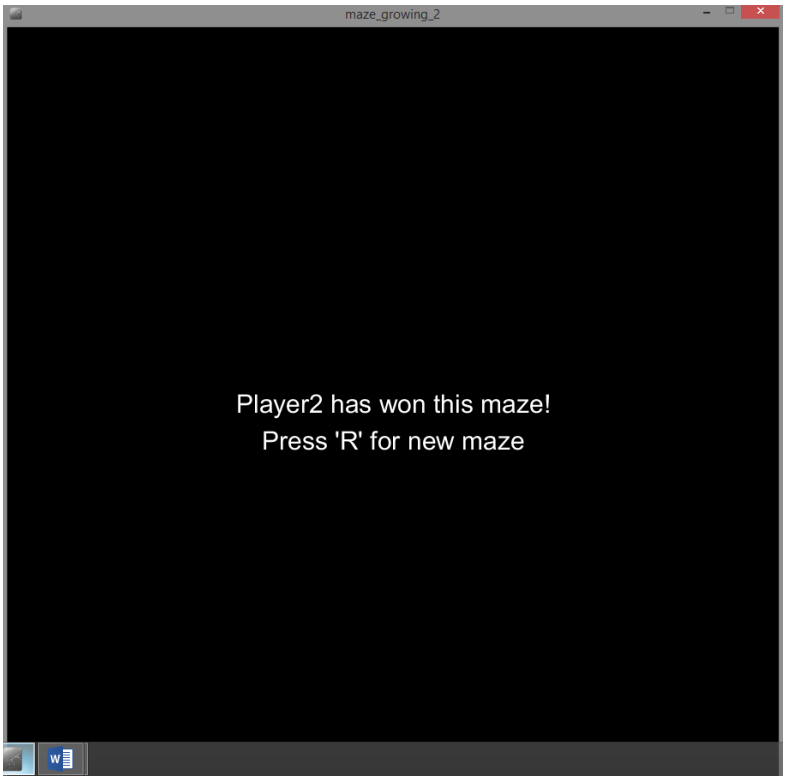

Fig 1.6 Player 2 Won!

Key Learnings

1. How to effectively plan ahead and manage the work.
2. How to overcome setbacks while achieving the goal and still manage to create a finished product of good quality.
3. Different types of maze generation algorithms like Growing tree, Prim's, recursive backtracker, breadth first search, A-star algorithm etc.
4. Handling collision detection and interaction of 2 players in the game.

Strengths and Weaknesses

Strengths

1. Efficient use of maze generation algorithm instead of hardcoding the level. Use of the growing tree algorithm for content generation creates a new maze every time the game is played.
2. Use of simple line drawing instructions to generate maze walls that can be scaled to any level. The current game window is of size (35 x 35) which can be changes easily by tweaking a few parameters to adjust game difficulty.
3. Maze accessibility from each node to every other node ensures players do not get frustrated by maze that does not have full accessibility.
4. Reliability of the maze accessibility from each node to every other node ensures players do not get frustrated by maze that does not have full accessibility.
5. Implementation of Enemy AI that have characteristic behavior making the game more challenging and fun.
6. Multiplayer mode makes the game more engaging by encouraging the competitive gameplay.
7. Game mechanics of allowing the players to block the opponent make the gameplay more interesting and fun.
8. Use of music gives vital feedback to players and makes the game aesthetically pleasing.

Weaknesses

1. Lack of good game assets for key and door sometimes makes it hard to find them in the maze.
2. This game can be developed using a stronger and efficient game development framework such as Unity.
3. Currently the game has static maze size (35x35) and cannot be changed.
4. The blocks placed by players are similar to the player in appearance which can sometimes get confusing.

Future Scope

1. The game is currently a 2D maze that can be converted to a 3D game.
2. Adaptable difficulty based on player's performance.
3. Customizable enemy and player assets based on player's choice.
4. Multiple players playing against each other online and online leaderboard.
5. Extension to other platforms such as mobile phones.

Conclusions

This was the first time that we created a game. Also this has been a learning experience for us in terms of how one can plan effectively and take decisions beforehand for completing a project. Also, it taught us how we can overcome hurdles in designing games and finding innovative solutions to tackle these problems. Using the AI architecture such as Finite State Machine, NPC behavior, PCG and path finding algorithms such as A-star in our game helped us reinforce the knowledge of these game development realms taught in this class.

Thanks for providing us with this opportunity to learn so much in very short period of time.

References

www.freesound.org

www.freeSFX.co.uk

http://www.redblobgames.com

http://code.compartmental.net