

***FYP Final report***

AR3

Computer Game 1

**Optimal path-finding  
Puzzle-solving game**

By

Cheng Tsz Tsun, Jason  
Tsang Hin Kam, Woody  
Ng Sik Hong, Henry  
To Nok Hang, Tommy

Advised by  
Prof. Sunil ARYA

Date of submission: April 14, 2021

## Abstract

Puzzle-solving is always a favourite genre in the computer games industry. Even in 3A games, puzzle always plays an important role. STEM has been a popular topic in recent years because of its great potential. Creativity is an important characteristic for students who want to succeed in this area. STEM does not teach students the knowledge on the books only but how to apply in real life or create a new product. One of the targets of our game is enhancing the creativity of the player. Solution always is more than one is our principle while developing the game. As a computer student, developing a game using theory learnt from university is an interesting topic for us. Finding optimal path is an important topic in computing. There is a lot of complicated theory to find the optimal path. After playing our game, player can learn the basic of finding optimal path in a more interesting way.

We use Unity to develop a puzzle-solving game with art materials built by blender. Also, we have included machine learning in our project to allow the player to have a look at the potential of AI. Details of our project will be shown in this report.

## Contents

<b>Abstract .....</b>	2
<b>1. Introduction .....</b>	5
<b>1.1 Overview .....</b>	5
<b>1.2 Objectives.....</b>	6
<b>1.3 Literature Survey.....</b>	7
<b>2. Game Description .....</b>	9
<b>2.1 Idea .....</b>	9
<b>2.2 Gameplay .....</b>	9
<b>2.3 Mechanism.....</b>	11
<b>2.4 Endless mode (Random generate game map) .....</b>	16
<b>3. Methodology.....</b>	20
<b>3.1 Design .....</b>	20
<b>3.2 Implementation .....</b>	21
<b>3.2.1 Background/Story (Tutorial).....</b>	21
<b>3.2.2 Sound effect &amp; graphic .....</b>	21
<b>3.2.3 Game mode.....</b>	22
<b>3.2.4 Gameplay .....</b>	23
<b>3.2.5 Grade/score.....</b>	26
<b>3.2.6 User interface.....</b>	26
<b>3.2.7 Save/Load system .....</b>	30
<b>3.3 Testing.....</b>	31
<b>3.3.1 Test the tutorial.....</b>	31
<b>3.3.2 Test the graphic and sound effect .....</b>	31
<b>3.3.3 Test the game mode.....</b>	31
<b>3.3.4 Test the grade and score .....</b>	31
<b>3.3.5 Test the user interface .....</b>	31
<b>3.3.6 Test the save/load system .....</b>	32
<b>3.3.7 Test plan – testing game mode (example) .....</b>	32
<b>3.5 Evaluation.....</b>	35
<b>3.5.1 Comparable system .....</b>	35
<b>3.5.2 Evaluation of objectives.....</b>	36
<b>4. Discussion.....</b>	37
<b>4.1 Game Design .....</b>	37
<b>4.2 Random generating game map (Endless mode) .....</b>	38
<b>4.3 Overall Game performance .....</b>	38
<b>5. Conclusion .....</b>	39

<b>5.1 Summary .....</b>	39
<b>5.2 Further development.....</b>	39
<b>5.2.1 Choice of game engine.....</b>	39
<b>5.2.2 Improvement on Music and Sound Effect.....</b>	39
<b>5.2.3 Improvement on puzzle design .....</b>	39
<b>5.2.4 Combination of 3D and 2D mode.....</b>	40
<b>Project Planning .....</b>	41
<b>Distribution of Work .....</b>	41
<b>GANTT Chart.....</b>	42
<b>Required Hardware and Software.....</b>	43
<b>Reference.....</b>	44
<b>Appendix A: Meeting Minutes .....</b>	46
<b>Appendix B: Game Demo.....</b>	57
<b>Appendix C: Game design document .....</b>	61
<b>Appendix D: Endless mode PowerPoint.....</b>	79

## 1. Introduction

### 1.1 Overview

In recent years, computer games have increased their impact on the society rapidly because of technological advancements, especially for mobile devices. In 2019, mobile games had become a multi-billion business and constitute about one-third of the total mobile application downloads, indicating the influence of the gaming industry [1]. Besides, developments in hardware like the Central Processing Unit (CPU) for desktop computers and consoles also allow a better gaming experience and thus help expand the gaming industry [2].

There are different genres for a computer game such as action, puzzle, and shooting. According to [3], puzzle-solving is the most favoured genre for computer games. It might be because puzzle games do not need specific mechanical skills to be proficient at the game. For example, shooting games might require fast reactions and music games would need a good sense of rhythm. Also, Frazer et al. [4] have mentioned that the puzzle game shines at its logical features with precise goals. Therefore, puzzle games could allow players to train their logical skills like logical thinking and analysing.

Puzzle games could be further categorized into two main types: one with only puzzles and the other one with some puzzles. Pure puzzle games only have puzzles for the content, including games like minesweeper and portal series. On the other hand, games that contain some puzzle elements like the tomb raider series would put their main focus on different genres like shooting and action.

However, most of the puzzles from the games provide only one intended solution. For instance, minesweeper would have only one answer for each map. It could result in the absence of creativity since players could only follow the clues given by the games for that single solution.

The goal of this project is to create a puzzle-solving game with multiple possible solutions for each of the stages, and players would be required to discover the solutions in their ways. By doing this, players could be able to think out of the box and thus help train their creativity. Our game provides 26 handicraft stages and an endless mode for the players to challenge themselves.

## 1.2 Objectives

The goal of our project is developing a puzzle-solving game. The total gameplay time will be about two hours and the planned number of stages will be twenty-six. We want to develop a puzzle-solving game which can provide a brand-new experience for the players which they can enjoy these twenty-six stages and feel excited.

Within the development stage, our project will focus on these objectives:

1. Develop a complete game that is interesting
2. Achieve completeness, multiple elements in the game must come together including stage design, user interface, music and sound effects, and art style
3. Design a core game mechanic using optimal pathfinding/optimal path problem
4. Increase the complexity of the problem with the extra game mechanic for players to solve

To achieve the first goal, we will continue studying previous puzzle-solving games to find out what makes them interesting. Also, we will study from famous indie game developers of developing a good indie game.

To achieve the second goal, we will try out designs that fit the game mechanic and ensure the sound effects, art style can be compatible under the background of the game. Also, we will study applications with a good user interface to design a better UI.

To achieve the third goal, we will study some optimal path problems and design variations of the algorithm to implement in our game.

To achieve the fourth goal, we will research on famous computer games and analyze what elements are interesting that attract players to play. Also, we might use some ideas from other puzzle-solving games to increase the interestingness of our game.

We expect the biggest foreseeable challenge will be the way to design a stage well. Since we previously do not have experience in designing games, we spent our summer holidays to study in computer game design. These experiences, we hope, could be used in our design process. Also, we planned to study game design not only in the scope of computer games but fields such as boardgames design and mechanism, which hopefully could give us some inspiration towards developing a better game.

### 1.3 Literature Survey

There are many masterpieces created in the gaming industry. During the design processes, we have done reviews on prior work, especially those which are puzzle-themed and award-winning. We found out that these games share elements that make them popular and long-lasting.

Mathematical tools in computer games:

Mathematics has been widely used in computer games. While it plays an important role in computer programming, games such as Conway's Game of Life (cellular automaton) also include mathematics as the games' core mechanism [6]. In another game called 'The Witness', large amount of maze-like graph puzzles is presented to the players [7]. The interesting part of the gameplay involve solving the graph puzzles of various difficulties, which also raised discussions among researchers [8][9].

Prior work in the field of graph theory has inspired us a lot as well [10]. We decided to show players the beauty of recreational mathematics with solutions based on mathematical and computational calculations. We look to implement extra game mechanics to increase the complexity of the stages so that they would be challenging and rewarding.

Graph Theory in computer games:

Traditional graph-themed puzzles, such as Icosian [11], use concepts in graph theory, which involves finding the Eulerian path and the Hamiltonian cycle to solve the puzzle. We push it one step forward by introducing special mechanics, which includes allowing players to move objects from a path to adjacent nodes, or enabling color modes where players can only collect specific color objects until they have switched colors, and even more for players to figure out. This not only makes the game more complex and challenging but also makes players know more and think more about graph theory.

Pathfinding and optimal path in computer games:

Pathfinding algorithms are widely used in different types of computer games.

Traditional 2D-MMORPG and MOBA [12] uses A\* algorithm to find shortest path to move characters and mobs. There are also modified algorithms developed to handle specific problems, such as the flocking algorithm in moving a large number of units in StarCraft2. [13] Pathfinding algorithms are also been used in finding exit in mazes. Situationally Depth-First-Search (DFS) and Breadth-First-Search (BFS) will be used in solving optimal path search problems.

In our game, there is an infinite mode where players were provided with many random maps generated by our algorithm. There is a need to develop a algorithm to search for optimal solution. We have used a modified BFS algorithm to search for optimal solution in random maps with push and pull mechanics generated.

While studying previous work, we have analysed some essential elements for our optimization puzzle game, especially those missing in previously released puzzle games. We will focus on creating a complete gaming experience that revolves around puzzle-solving whilst adding depth to the game.

## 2. Game Description

### 2.1 Idea

We want to make a puzzle-solving game that requires players to think and challenge themselves. The number of players in this game will be one. In this game, the objectives of the player are collecting all the stars and arriving on the goal. If the player cannot fulfil the requirement within some specific moves, the player will lose and need to retry. We intended to provide answers that contain some counterintuitive moves end up being the better moves. We also provide stages with the same map and a slightly change in the condition which have a huge impact on the result. We want to develop a game inspired by travel salesman problem to make players think of how to clear using the least rounds or moves. The difficulty of the game would mainly depend on the complexity of finding an optimal path and all of the mechanism we added to the game would provide extra options to players and hence increase the possible number of outcomes that players have to process.

### 2.2 Gameplay

Gameplay:

1. 2D
2. 3D

Game mechanisms:

1. Walk
2. Pull
3. Push
4. Portal
5. Two stars
6. Two worlds

Player action:

1. Select the specific node
2. Push/pull star
3. Build a bridge
4. Move to the node that connects by bridge
5. Portal
6. Change mode

Node:

1. Player can stand on the node
2. Player can build bridge to connect the nearest nodes
3. Can store more than one stars

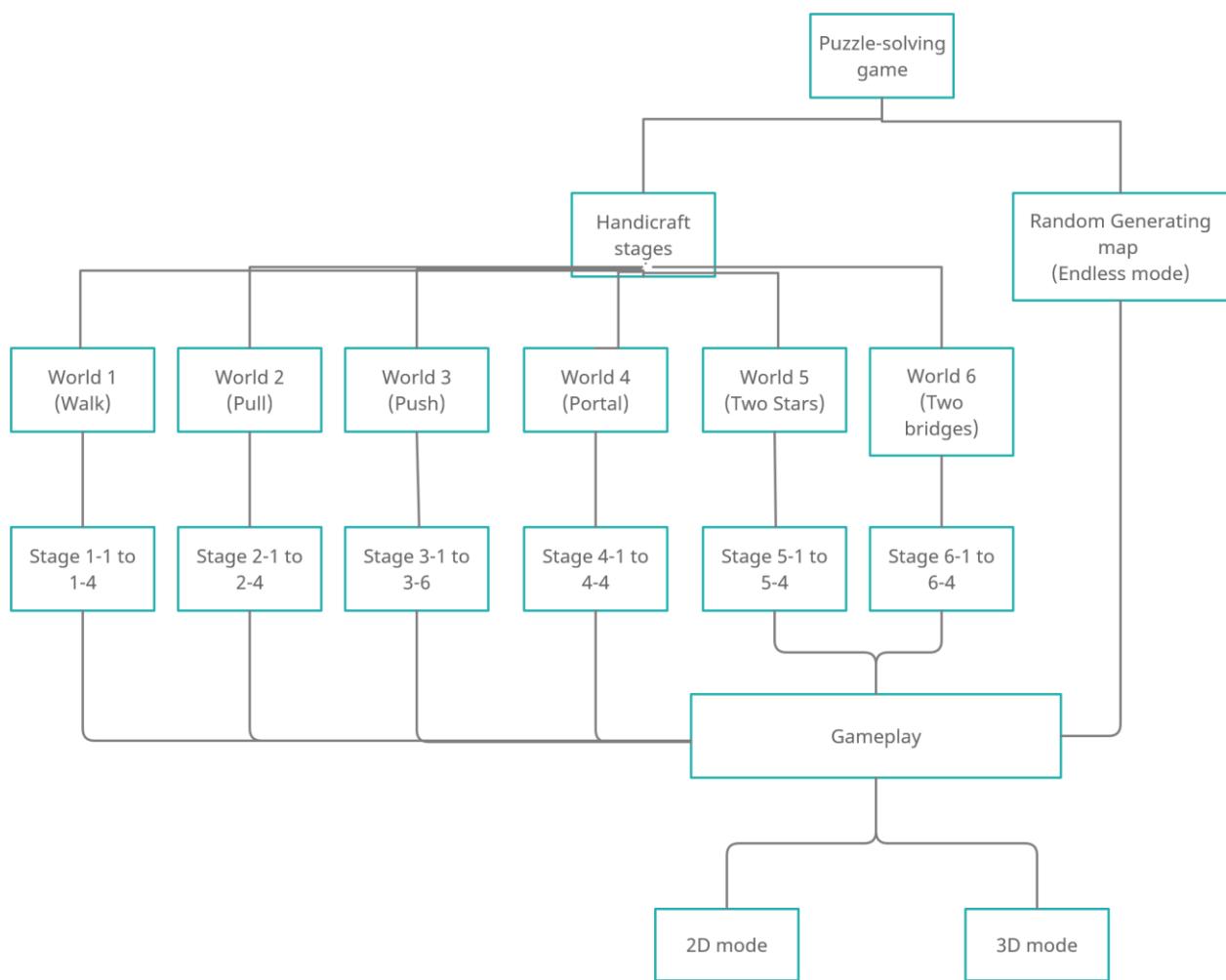
Star:

1. The target to be transported to the goal
2. Player needs to collect all of them to clear the stages

Bridge:

1. Can be used once only
2. Only build near the player's location (3D gameplay)
3. Build anywhere (2D gameplay)
4. Limited amount of bridges can be built

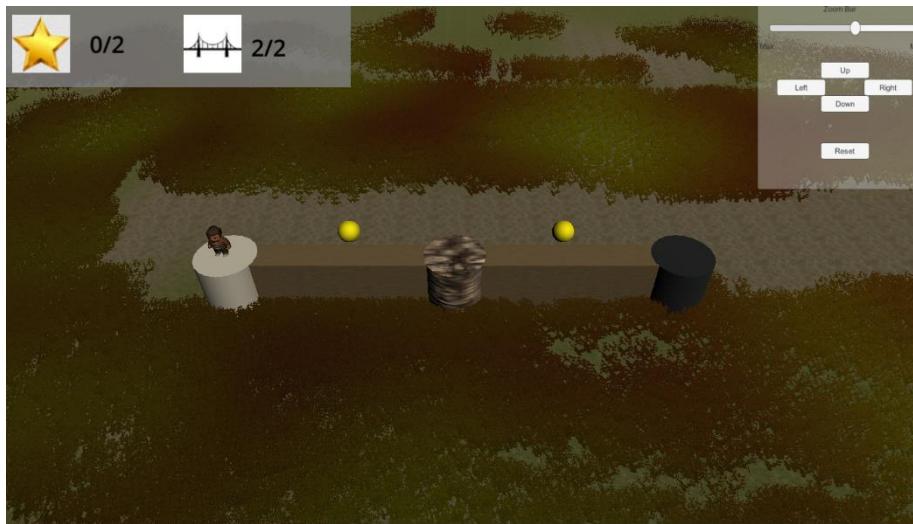
## Game structure:



## 2.3 Mechanism

### Walk (World 1):

In World 1, the player can only clear stages with walking only. This world is a tutorial for players to familiar with the mechanism such as building bridge, collect star and moving the character.



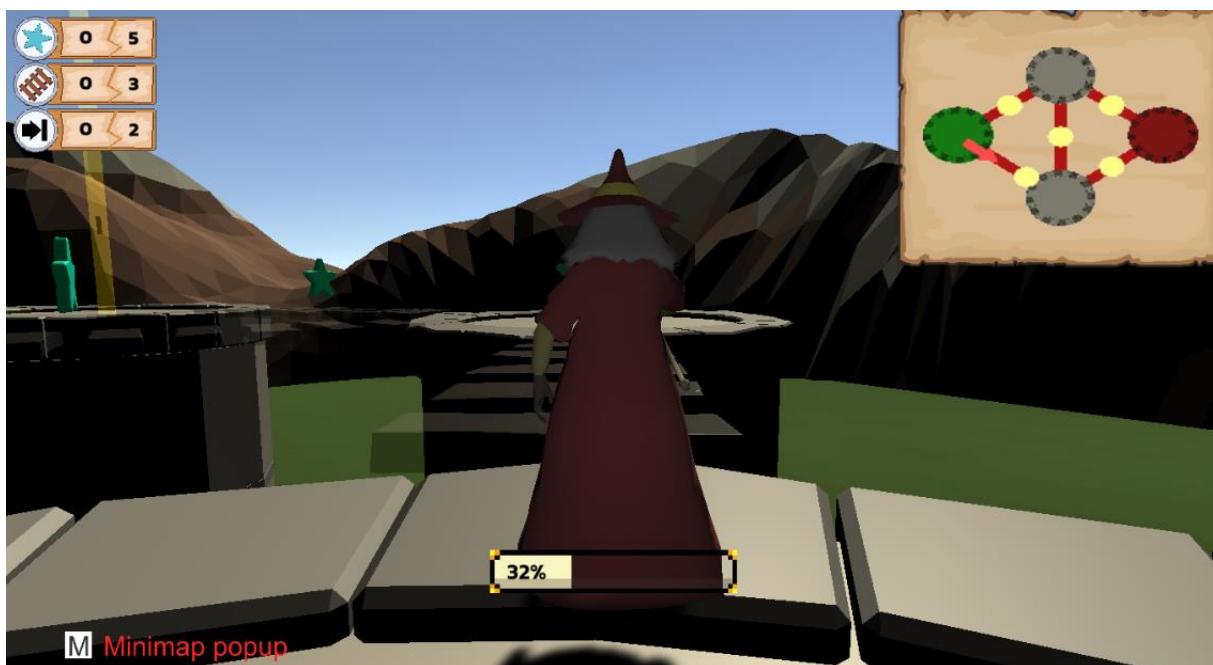
### Pull (World 2):

In World 2, we have introduced a new mechanism – pull, a new way for player to interact with stars. Player can pull the star nearby to the player's current position and collect it. Using pull means player can get star without building bridge. With this mechanism, player will also have to consider how they would like to collect every star rather than just consider about the order in the previous world.



**Push (World 3):**

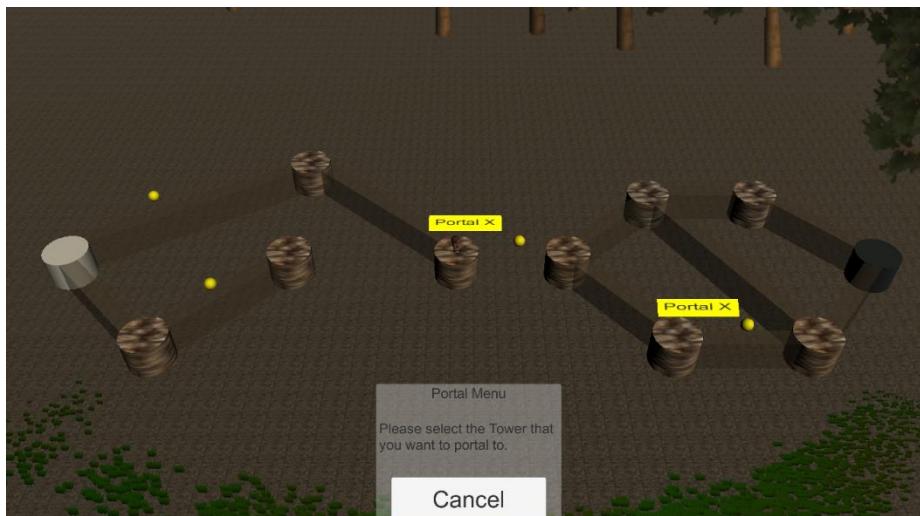
In World 3, push is the new mechanism for the player. Push is the opposite to pull which the player will move the star to the node on the other side. Then, the player needs to walk to that node to collect the star. Player needs to use an extra move to collect star but sometimes it can reduce the number of bridges. Therefore, player needs to think carefully before pushing the star because it may waste one bridge. Similar to pull, push also provide an extra depth of complexity by asking the player which way to collect every star. Both push and pull will be in the new fundamental move set since next world.



**Portal (World 4):**

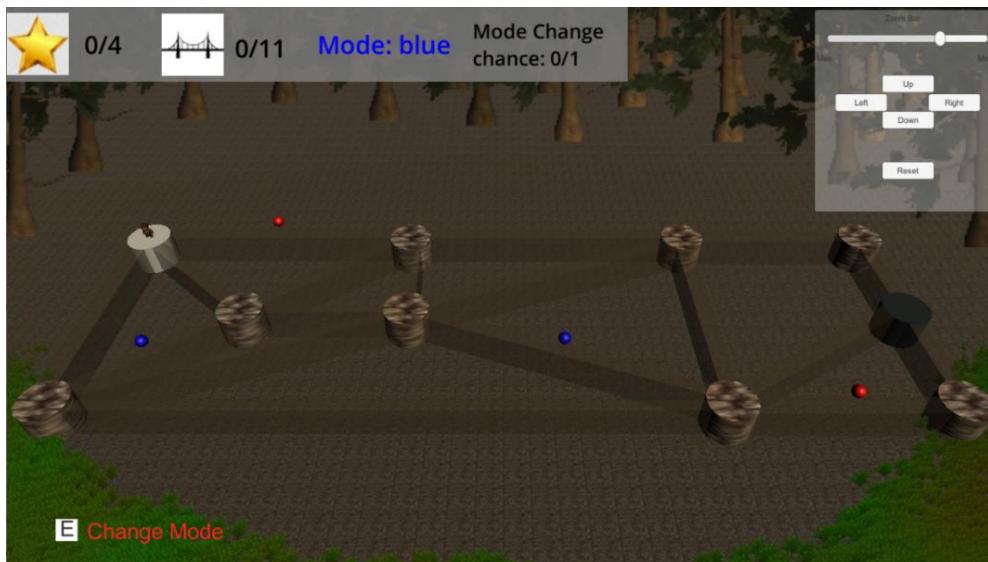
In World 4, unlike the above two worlds, we have introduced a mechanism – portal that interact with nodes. When player arrives at nodes with portal, player can choose to teleport the node with the same type of portal. To utilize portal, one of the two ways is to produce a big map, but we think that this approach may not add too much depth to the game play and make solving the fun puzzle become a tedious job.

Because of that, we create new maps that are separated for the players, so they will have to use portal nicely, while connecting those small part of map with a star so it would not feel like solving some small level at the same time. We also intentionally make the map less connected to increase the importance of portal. In this world, player is allowed to use push to solve the puzzle. Since portal is a free resource with limitation on the location, player will have to consider when and where to use the portal. After combining the two mechanisms, the difficulty of this world is much higher than the first three worlds. Photo below shows the portal stage in 2D gameplay:



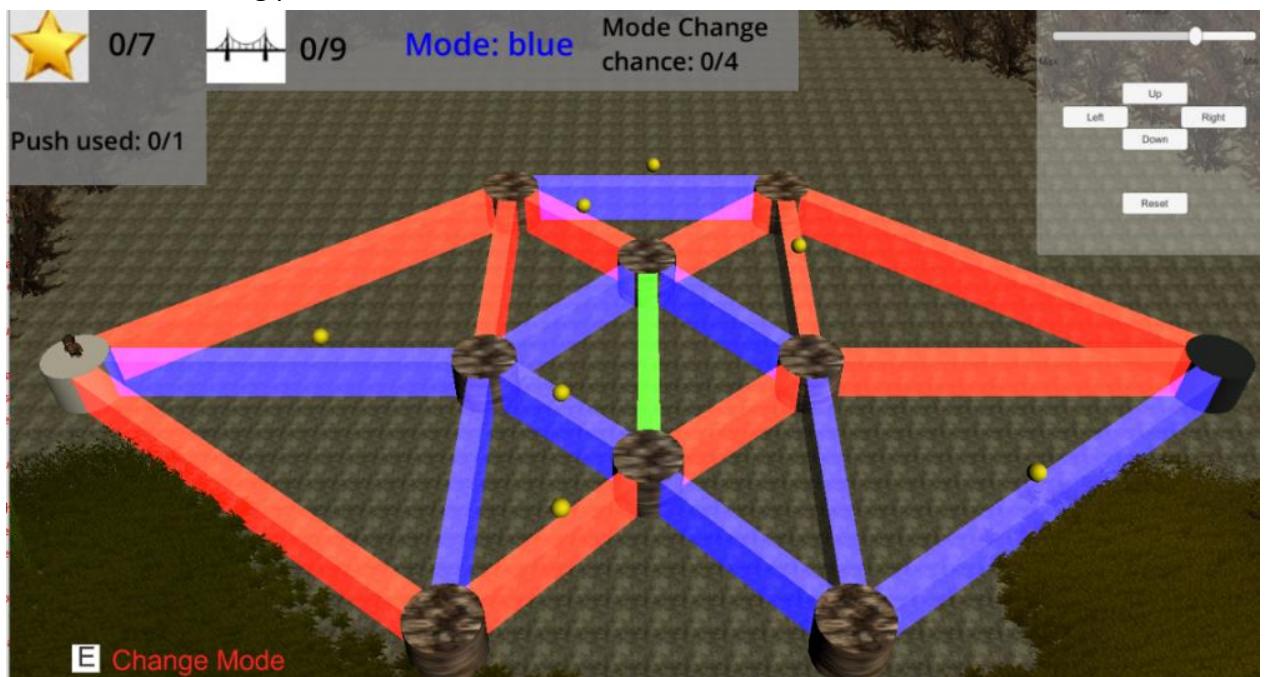
Two Stars (World 5):

In World 5, a new mechanism – two stars have been introduced. From this world, we start implementing different game items to increase the complexity of the puzzle. Every star has corresponding color and player can only collect or interact with the stars when player is in the same mode (color). The chances of changing color are limited. Also, we have included pull and push in the last two stages. How to use resources and when to use them are the big challenges for player.



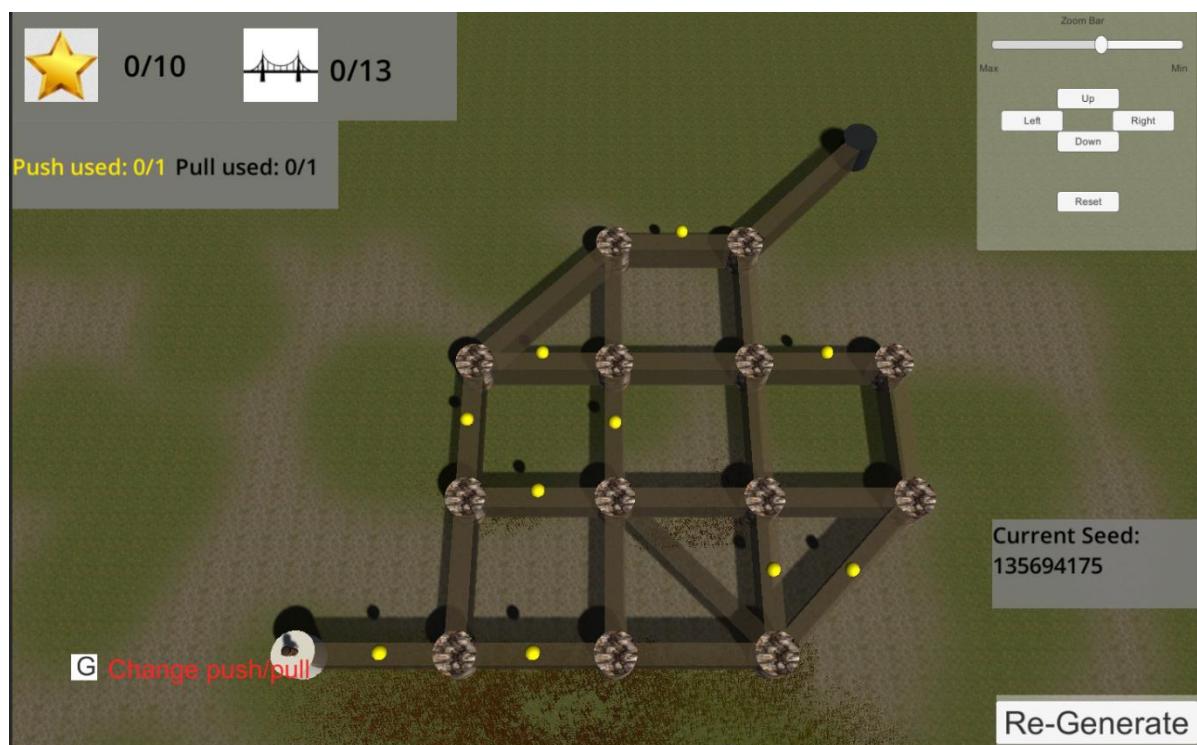
### Two worlds/bridges (World 6):

In World 6, the last mechanism – two worlds/bridges are ready for the player. This mechanism is similar to two stars but this time is changing the mode on bridges. Player can only walk on the bridge with the same color when the same mode is activating. There are some normal bridges that player can walk in both modes. With this mechanism, players will have some control of the map and the map will be separated into two part while also being very connected for players to move around. And it will be like solving two interconnecting puzzles at the same time.



## 2.4 Endless mode (Random generate game map)

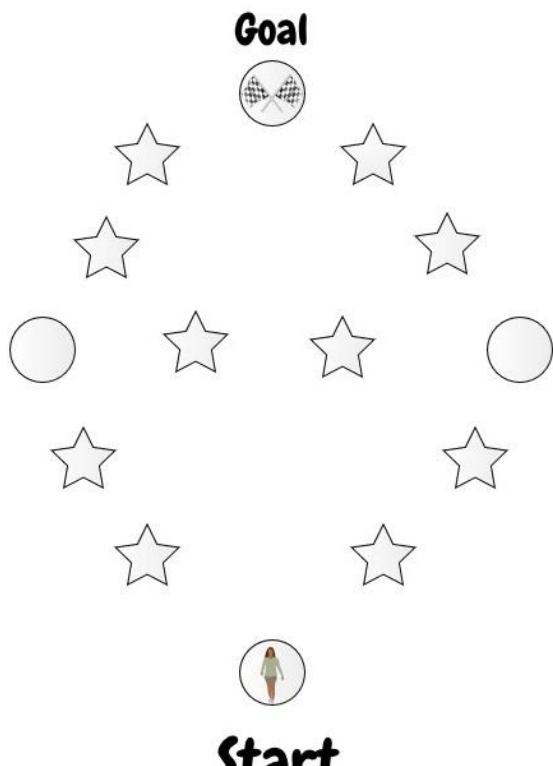
After those 6 worlds, there will be an extra world for endless mode. In the endless mode, players will be presented with random generated maps with both the push and the pull mechanisms. One of the objectives of the game is to provide a platform for players to develop their creativity by solving puzzle with different conditions. In order to do that, we provide a map generator so players will have more than enough stages for them to practice their creativity. Since players will have overcame all 26 stages to get to the endless mode, the algorithm in the generation code will ensures a certain level of complexity in the gameplay, which means that there should not be very easy stages that players only have to follow the stars to reach the goal.



## 2.5 Simple stage gameplay (Push)

This is the initial idea of gameplay which is different from the current gameplay. The core mechanism is the same.

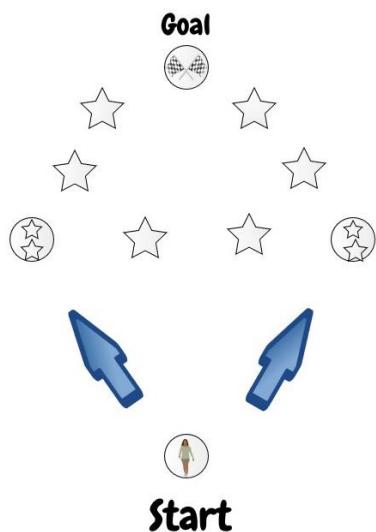
Step1:



This photo shows the basic setting of a wave with basic elements including:

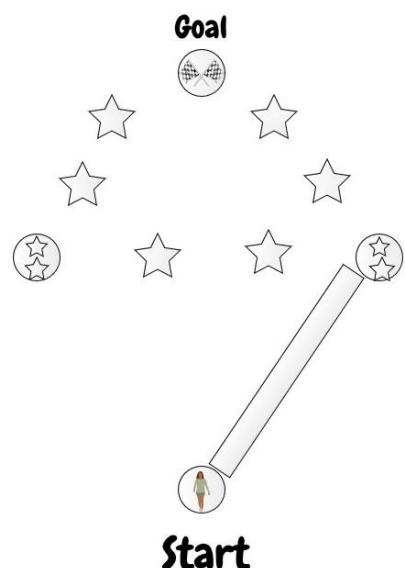
1. Start point
2. Goal
3. Star
4. Node (foothold)
5. Player

Step2:



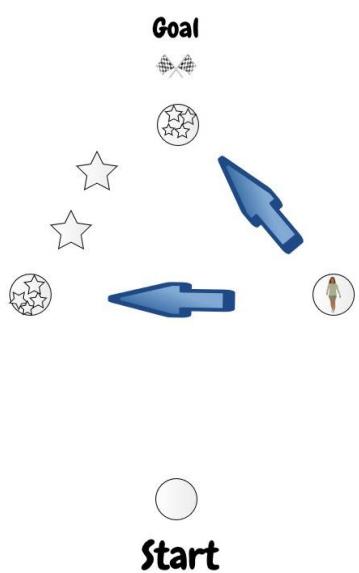
Player is pushing the star to the nearest nodes.  
The arrows are the action of pushing.

Step3:



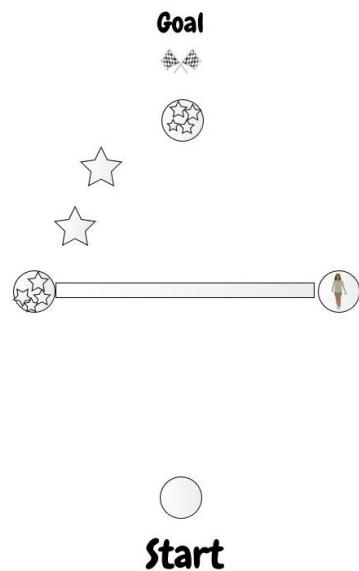
Player build the bridge to connect the right node and move to that node.

Step4:



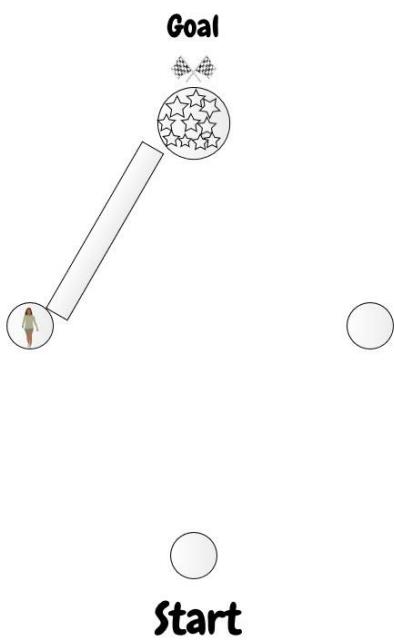
Player pushing the stars to the nearest nodes.

Step5:

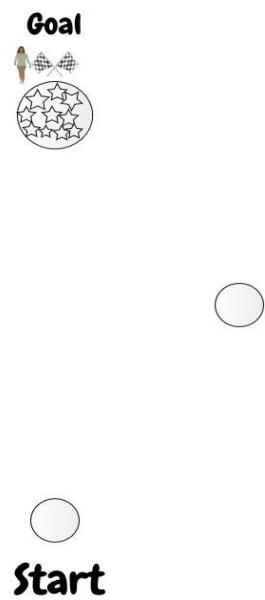


Player build the bridge to the left node and move to that node.

Step6:



Step7:



Player push the remaining stars to the goal and build the bridge to the endpoint.

Player reach the goal and win the game.

In this stage,

Number of pushes: 5

Number of pulls: 0

Number of bridges built: 3

Number of moves: 3

Number of stars collected: 10

### 3. Methodology

#### 3.1 Design

We have started designing the project since June and we are working on these aspects:

- Background (Tutorial)
- Sound effect & graphic
- Game mode
- Gameplay
- Grade/score for stage
- User interface
- Save/load system

Game title: puzzle-solving game

Genre: puzzle solving

Number of players: 1

Platform: Microsoft Windows

Target users: Players like puzzle-solving games

Controller: Keyboard and mouse

Total stages: 26 stages + endless mode

Gameplay: 2D or 3D

Playtime: 2 hours

Game engine: Unity

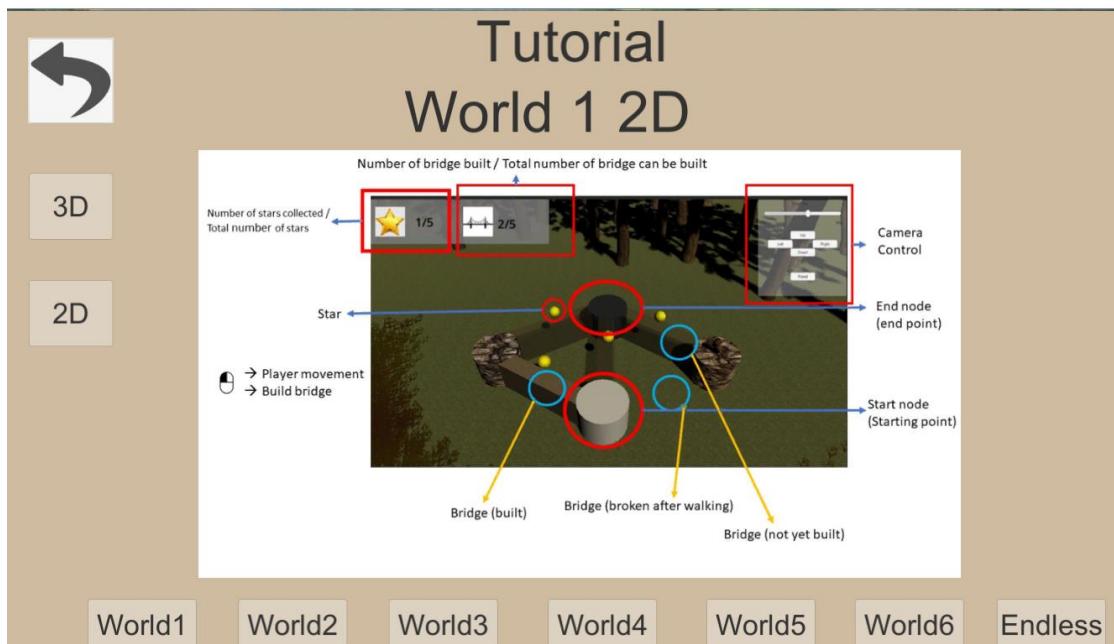
We choose Unity as our game development engine because we think it is the best free game engine. Also, all of us have experience in coding using C++. We think it is easy for us to learn how to do coding with C#. Unity has good performance among different platform so we can use either Windows or macOS to develop our game. There are a lot of free assets in Unity for us. All of us are not good at design or music so we can develop the game easily using these assets.

## 3.2 Implementation

### 3.2.1 Background/Story (Tutorial)

In the proposal, we wanted to develop a background that the player will play as a magician and the background will be shown in the game through dialogue when the player enters the first stage. However, we decided to cancel the background story and replace it with a simple tutorial due to limited time. Now, the game provides a tutorial of each world with screenshots for the players. This can allow the player to familiar with the game in the shortest time and focus on solving puzzle. We implemented the tutorial in the main menu so the player can read the tutorial before start playing our game.

Example of tutorial:



### 3.2.2 Sound effect & graphic

Most of the graphic and sound effect will use the online free assets in the Unity asset store. We may use other resources and the resources will be properly acknowledged. In 2D gameplay, we mainly use free online resources from the Unity asset store. In 3D gameplay, we use blender to create our art design and materials because 3D gameplay can display better graphic to attract the players playing.

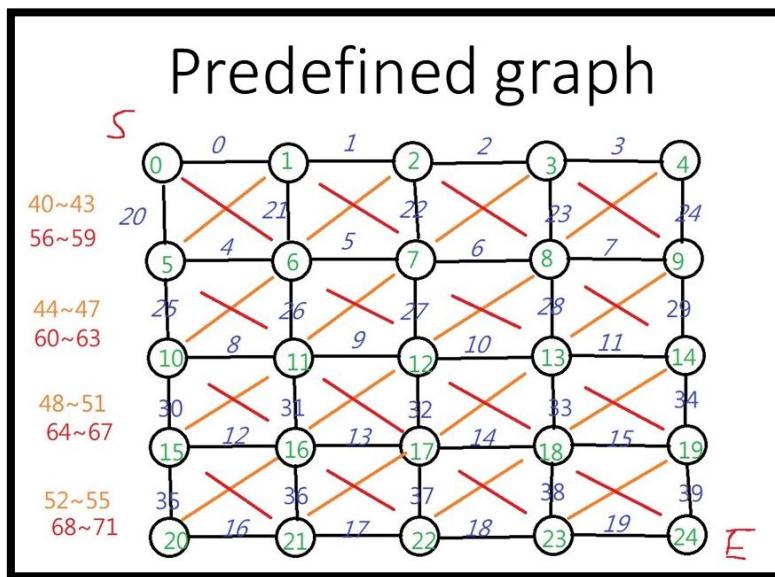
### 3.2.3 Game mode

We developed 6 worlds (6 game mechanisms) with 26 stages for the players and an endless mode with generated maps with both the push and the pull mechanisms.

**World selection – player will choose and enter different world in this scene**



For the endless mode, we have created a random map generating system. In order to easily incorporate the map into unity, we have a predefined square-like map.



(The label for each vertices and edges.)

Since having overlapping edges may cause confusion with stars, we will remove one to two edges for the crosses. To guarantee an answer, we will first construct a long random path by pseudo-randomly selecting three points. Then we will once again pseudo-randomly put stars on that path. We also put two extra stars next to nodes on that path

in order to increase the complexity on the map. Since the random generated path seldomly contain loops of path with length 3-4 (back to the same location after some moves), which is one of the very important structures for our mechanics to work. Adding the extra two stars will help solve this problem and our guaranteed answer will be walking that path and using 2 pushes.

We will protect these edges during the modification in the later steps. We will then randomly drop some edges to increase game complexity. It will then check against dangling vertices as long as dead-end paths with no stars, which will be deleted from the map.

One of the huge questions for this part is to search for the solution. Since every mechanics are designed to increase the possible state of a map, and so mechanics are more powerful than other, we have to define our standard of good answer rather than walking less edges. We end up choosing to brute-force in a smart way. The first thing we do, is to try take mechanics out of the search problem into a checking condition. So, instead of tripling the number of possible states of ever star (with mechanics, player will have more option on a star), we actually reduce it to a small checking case only for those states reached the goal. In this case, the search tree will order by number of walked edges while the rating of the solution will also increase by the mechanics used. So, we will have to search until all the remaining solutions must be worse than the current solution. Because of this approach, we can only have bad estimation of remaining step to reach the goal. Causing other searching algorithms like a\* search end up spending much more time on sorting the states than reducing the number of states that have to be search

The generation code also has some tuning parameters to increase the difficulty of the maps and balance the number of appearance of stages with “no push/pull”, “push only”, “pull only”, and “push + pull”.

The resulting map will look very similar to a handcrafted map, except the angles between edges are the same. This can be modified in the future by using force-directed graph techniques to dynamically generate the positions of entities in the stage.

### **3.2.4 Gameplay**

We implemented 2 gameplay mode for players – 3D and 2D. We implement world selection and stage selection with 2D gameplay because it is more convenient for the player to choose the stages or worlds.

In 3D gameplay, players will use keyboard (WASD) to control the character to solve the puzzle. The characteristic of gameplay is players can enjoy the beautiful world during

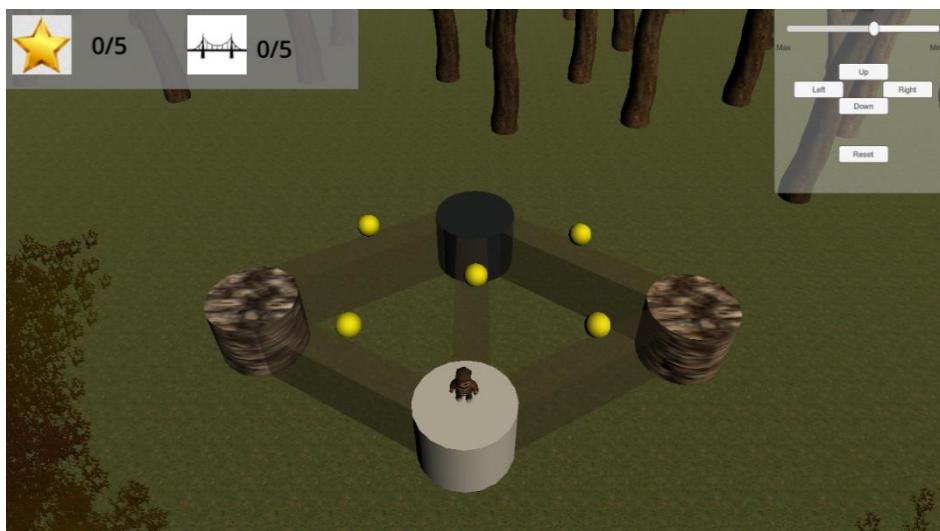
solving the puzzle. In this gameplay, the camera will follow the character so players can see more beautiful art materials and animation in this gameplay like pushing star and falling bridge. Also, a mini map is provided for players to read the whole puzzle from top view.

In 2D gameplay, players will use mouse to control the character to solve the puzzle. The characteristic of this gameplay is players can focus on solving puzzle and not distract by other elements. In this gameplay, the camera is from the top view and players can see the whole puzzle clearly so players are easier to find the solution and clear the stage in the shortest time.

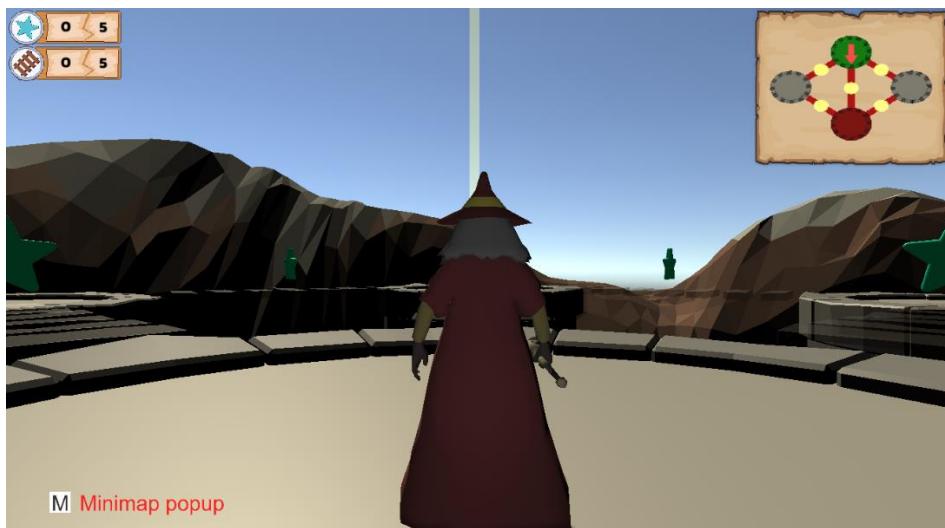
**Gameplay selection menu – player choose the gameplay (way to play the game) on this menu:**



**2D gameplay:**



**3D gameplay:**



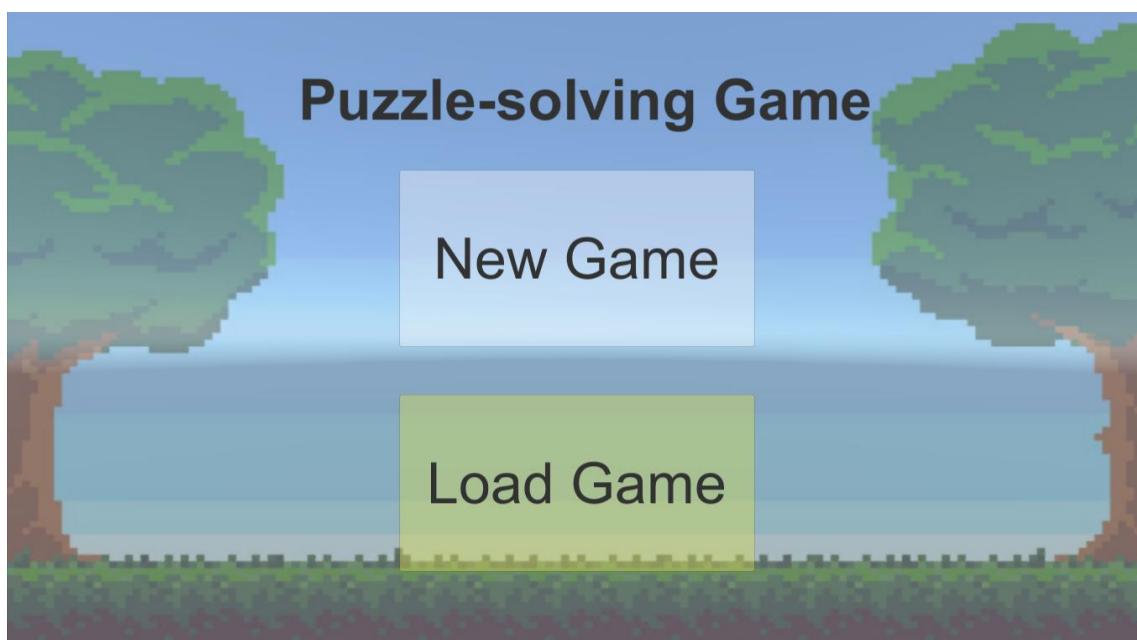
### 3.2.5 Grade/score

The original proposal was giving players different grades depending on the resources they have used. However, we hope the players always finish the stages with the best solution. As a result, we decided to give minimum resources to ensure the solution is the best. Now, the stages only have two states: cleared or not yet cleared.

### 3.2.6 User interface

When the players enter the game, they will see two options.

1. New game
2. Load game



When the players enter the start menu, they will see four options on the menu.

1. Start game
2. Options
3. Control
4. Quit
5. Tutorial
6. Credit



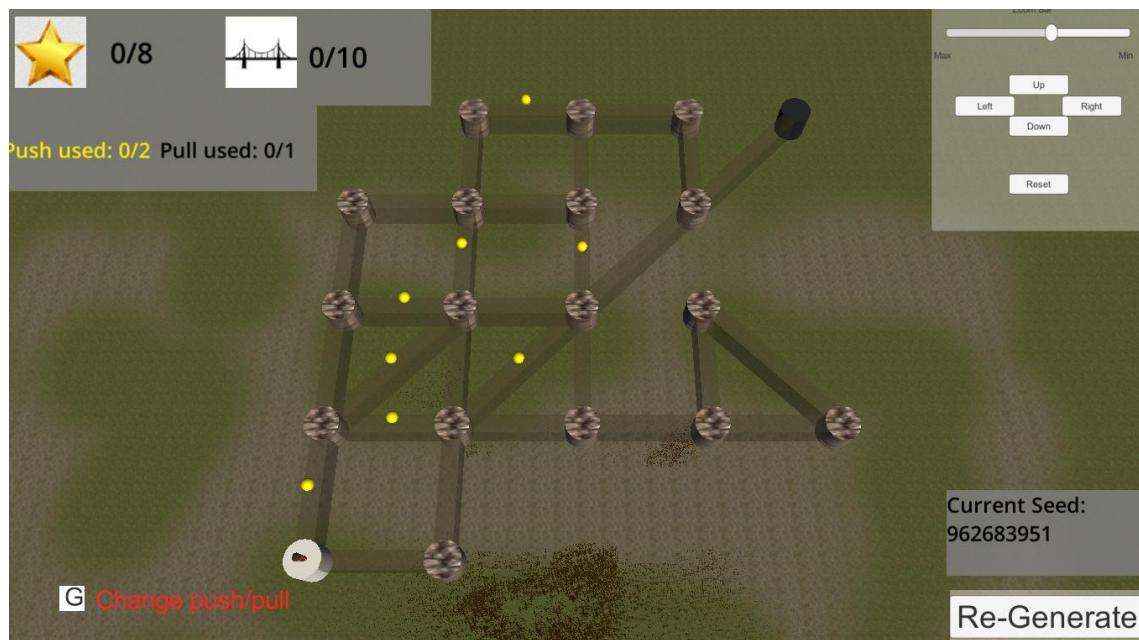
When players select **Start Game** in start menu.

1. They will enter **World selection** scene and they can choose to enter which world. In this scene, players will use similar control in 2D gameplay. There are 6 worlds and endless mode.
2. If players choose to enter one of the six **Worlds**, they will see the stages in that world. Then, they can choose to enter which stage and they will see the nametag become green if they have cleared the stages. Also, they can select to return to World selection by pressing the button on the top-left corner. When players select a specific stage, they will see a popup window to select **2D** or **3D** gameplay. Players can choose to clear stages with 2D or 3D gameplay.

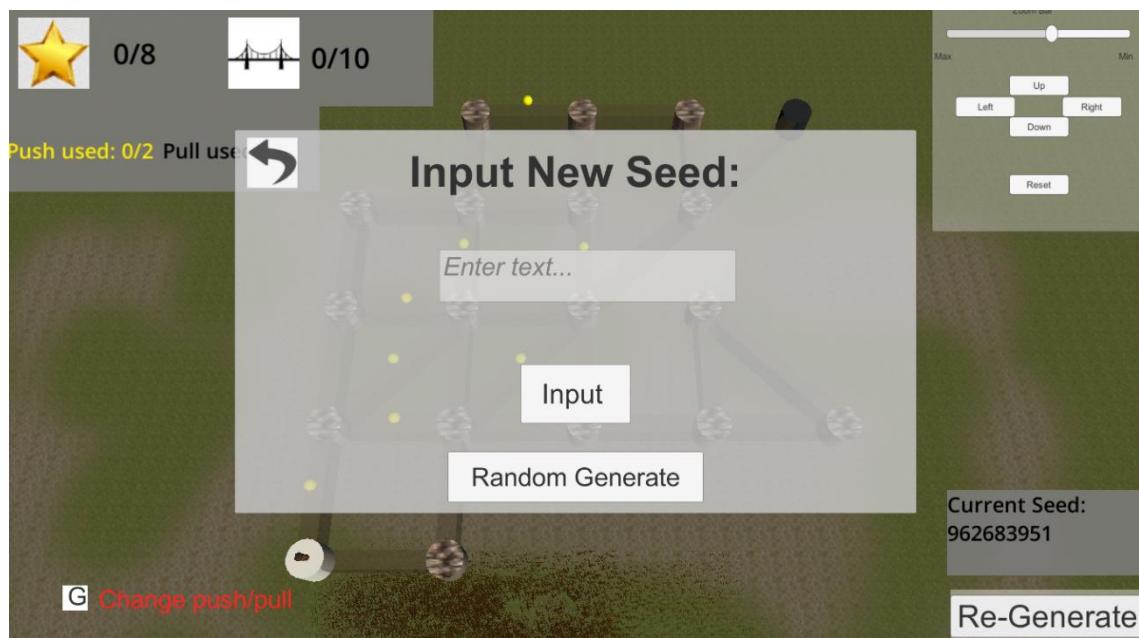
3. If players choose to enter **Endless mode**, they will see a generated game map based on a random seed. If the player wants to generate another map, the player can press the re-generate button on the bottom right corner to input value or generate based on a random seed. Also, the current seed is displayed at the bottom right corner and the player can mark the seed number if the player finds the stage is interesting.

4.

Endless mode (2D gameplay):



Popup menu for re-generate:



When the players are in the stages in 3D gameplay,

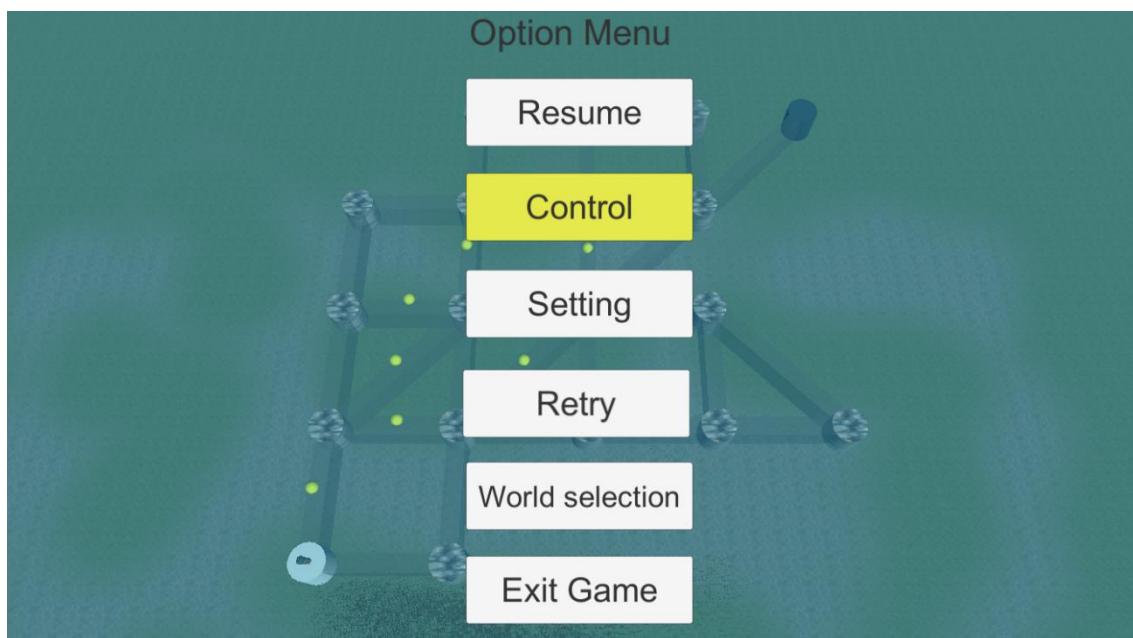
1. At top-left corner, game stages such as number of stars, number of bridges, number of pushes, etc. will be displayed
2. At top-right corner, a top-view camera is showed and this camera is following the character

When the players are in the stages in 2D gameplay,

1. At top-left corner, a Top UI will display game states such as number of stars, number of bridges, number of pushes, etc.
2. At top-right corner, a Camera control will be provided for players to adjust the camera
3. The camera is not fixed in this gameplay and players can view the stages on top view

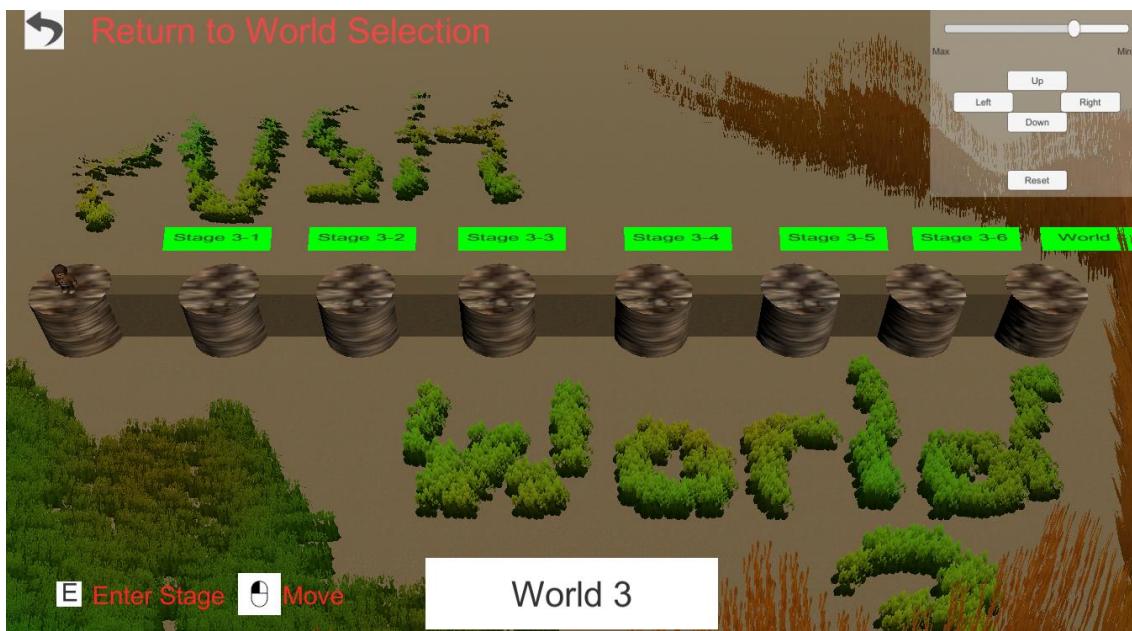
When the players press Esc in the stages, they will see these options

1. Resume
2. Control
3. Setting
4. Retry
5. Current world / world selection
6. Main Menu
7. Exit game



### 3.2.7 Save/Load system

When the players finish the stages, the game will save automatically. Players cannot use manual save in the game. If the players leave the game or return to the main menu while playing stages, the process will not be recorded. We developed a simple save/load system using PlayerPrefs in Unity because we do not need to store complicated data. When players enter the game, they can choose New game or Load game. New game means deleting all recorded data and starting the game from the beginning. Load game means continuing the game with the previous save data. Now, the player can only have one save data and we may add more save data slots for the player in the future. Also, the preferred setting will be stored with PlayerPrefs so that player do not need to adjust the setting every time.



**Label with green color means that stage or that world has been cleared  
(photo above)**

### **3.3 Testing**

After developing the game, we will have several testing plans on different aspects to ensure the game is working properly.

#### **3.3.1 Test the tutorial**

We will ensure every world have at least one tutorial photo for the players. The instruction of the game needs to be clear enough so one members will write the instruction and others will check for clarity. Also, the photo must be clear in different resolution and in both editor and game mode. We build the game with medium quality in Unity to ensure the photos having clear display.

#### **3.3.2 Test the graphic and sound effect**

We will play every stage to ensure the graphic and sound effect can display properly. In 3D gameplay, we provide more animation for players so we need to ensure the animation will not make bugs while playing. Also, we will test the game in different computers with different specification to see if the animation works properly in an older computer.

#### **3.3.3 Test the game mode**

We will play every stage in both 2D and 3D gameplay to ensure the player can play the game and there are not bugs while playing the game. We will have at least 3 test cases in every stage. For example, we will first clear the stages with the best solution to see if the stages end properly. Then, we will play not following the path in solution to test if the players can walk to every node or tower properly. We have 6 mechanisms in total and there will be more than one mechanism existed in some stages. We will test if different mechanisms working properly. For example, players can use push, pull and change mode in Stage 5.3 and 5.4 so we need to ensure these functions can be used and do not have errors. Also, collecting stars is an important function in our game and we need to see if the character can collect every star while walking pass or pulling them.

#### **3.3.4 Test the grade and score**

We will play every stage in both 2D and 3D gameplay. If the result has been stored properly, we will see the nametag of that stage becoming green and the players can walk to the next stage or next world.

#### **3.3.5 Test the user interface**

In Start Menu, we will click on every button to see if they work functionally. Also,

Option menu in the stage is important and we will have a similar testing approach to test every button in every stage. Players can adjust their setting through the Setting scene and we will test if the setting options work properly. During playing the game, we will have different popups such as option menu and mini map. We will see if any popup crashed with others while playing. In 3D gameplay, we provide a mini map and we will test if the mini map display properly in different window size. In 2D gameplay, we provide a camera control for players so we need to ensure the buttons are working functionally. In both gameplay, we provide a top UI for players to read their current state including collected star, total star, the number of push or pull has been used, the number of push or pull can be used, etc. We will play every stage to test if the top UI can update the information correctly and correctly display the number of resources.

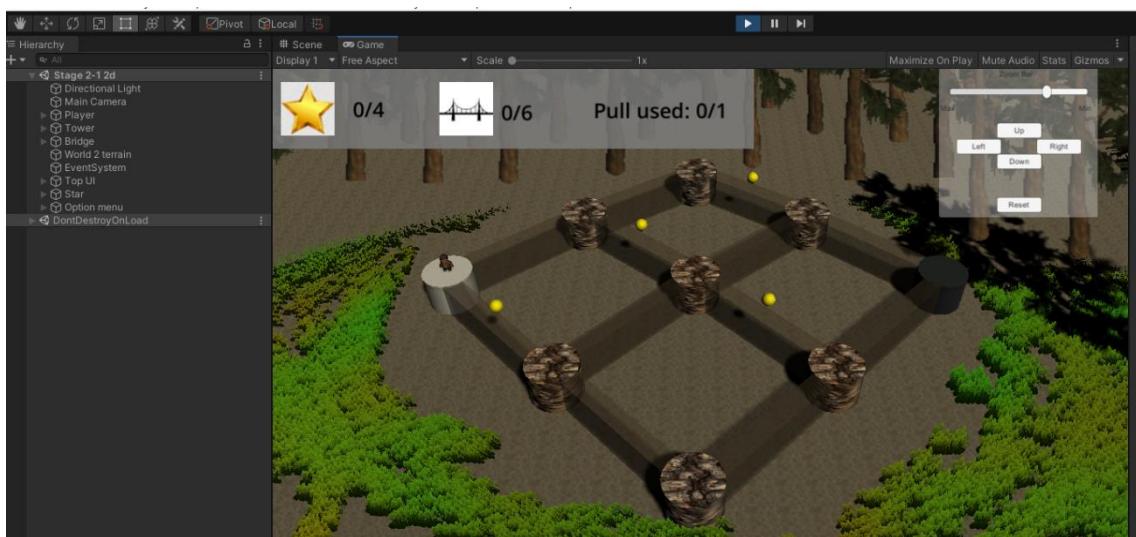
### 3.3.6 Test the save/load system

We provide two gameplays for the players and we need to ensure the player can clear the stage in either 2D or 3D gameplay. We will play every stage in both 2D and 3D gameplay to ensure the save system is working properly. Also, we will test the New game and Load game buttons in the New game scene. We will test if the New game button can clear all save data and if the Load game button can load the previous save data.

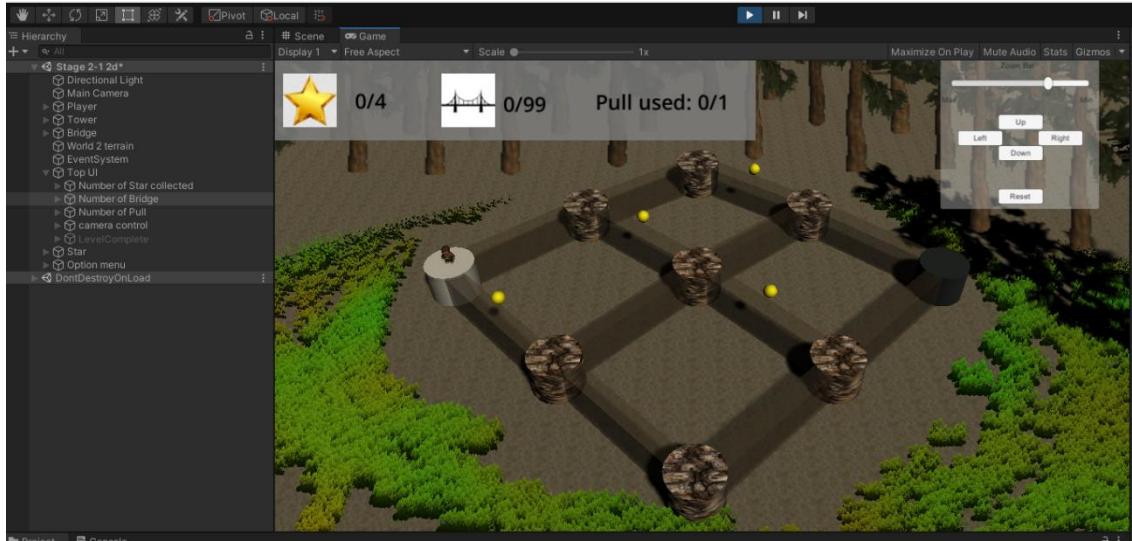
### 3.3.7 Test plan – testing game mode (example)

We will use Stage 2-1 (2D gameplay) as an example of testing plan.

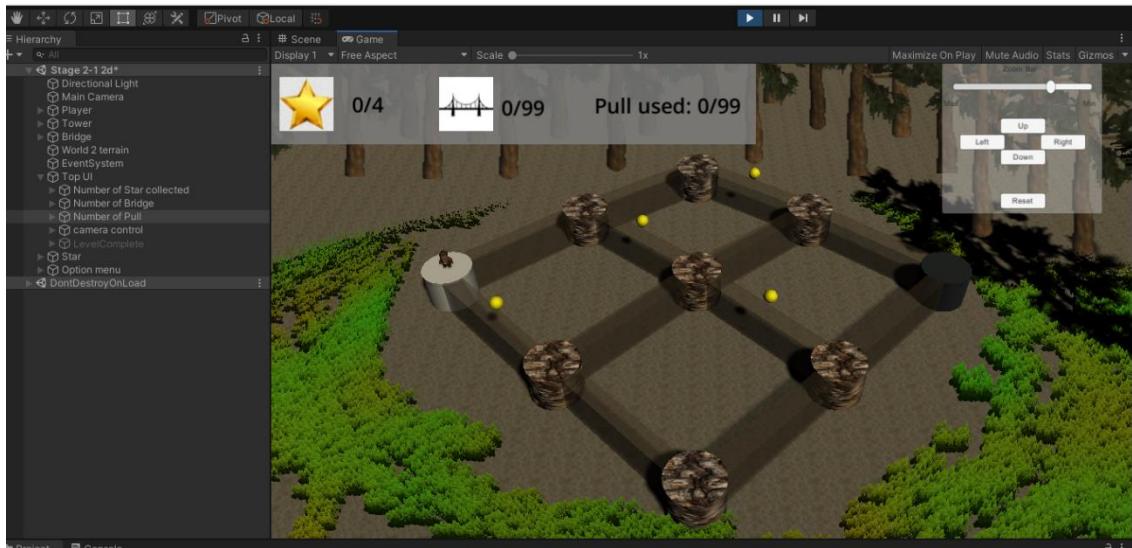
Step 1: Clear the stage with the best solution. We will ensure the player can clear stage and get proper result like changing the nametag color and save the result of clearing this stage. Also, we will ensure there is no red warning icon while running the game in Unity editor because the red warning sometimes will cause game crashing.



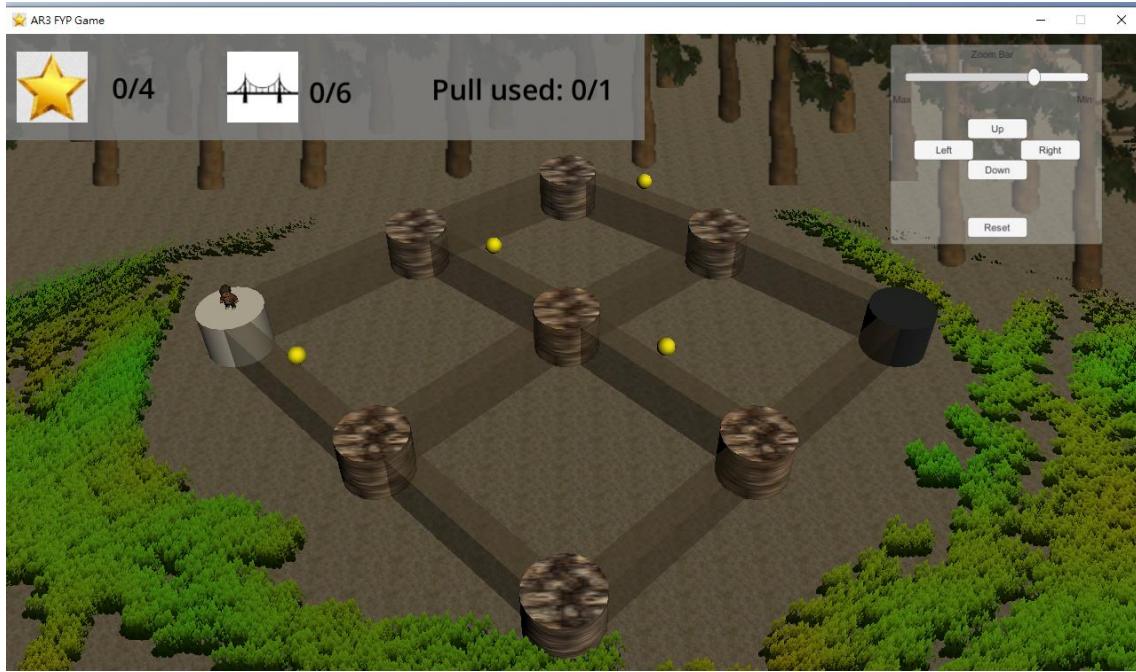
Step 2: Set the number of bridge can be built to be 99 and walk pass every node to see if the bridge and tower function work properly. This step will be repeated several times until every node and bridge has been walked pass at least one time.



Step 3: Set the number of pull can be used to be 99 and try to use pull at different nodes on different stars. We will test if the stars are pulled to correct nodes and is the collect star function working properly. This step will be repeated several times as every star can be pulled in two directions. We will repeat the testing until pulling every star in two directions.



Step 4: We will repeat the above steps in the game (.exe file) because sometimes the result will have difference between game and editor. As a result, we need to test the game in both exe file and editor mode. Also, we will test is there any delay on control and animation after publishing the game.



We will have similar testing approach in every stage (both 2D and 3D gameplay). This is our testing plan for game mode or gameplay.

### 3.5 Evaluation

After the testing, we will evaluate the game's performance. There are some objectives for our evaluation.

1. Can the game start properly? Is there a memory leak problem when playing the game?
2. Can the user interface provide enough information for the players?
3. Do the graphic and sound effect use properly in different stages and user interface?
4. Is the game challengeable and fun to attract players to play?
5. Is the curve of difficulty appropriate for the player? Do we provide the best solution?
6. Do the automatic save function work properly? Are there any bugs when the players save the games in different stages?

To develop endless mode, we have used the concept learnt from courses in HKUST such as BFS and AI. In endless mode, we have pre-built a map with nodes and bridges. Then, we will use a random seed to generate different game maps based this pre-built map. We need to make sure the generated game map has an existing solution so AI is used during this process. Also, we need to use the solution to set different constraints like the maximum of bridges can be built. After that, a randomly generate game map is ready for players to play.

#### 3.5.1 Comparable system

Our game can be divided into two parts. First one is the handcraft stages. We have developed 26 handcraft stages to provide different challenges for the players. The second one is the endless mode which is a random generating game map using AI. In this mode, players can play a new game map every time with different seeds.

Handcraft stages are designed for a long time so the players may feel more challenging. There are several stages in every world and the stages have increasing difficulty. Players can be familiar with the new mechanisms in the first stage and then they need to use their creativity to solve the harder stages. Also, the stages are tested for several times to ensure there are no bugs.

Stages in endless mode are randomly generated based on the pre-built and AI only concern on the solution exist or not. As a result, the difficulty is less compared with the handcraft stages. Also, the stage has a lot of possibilities and it is difficult to test all stages so bugs may exist in endless mode. However, players can have new experiences

whenever the seeds are different. Also, we can apply more computer science knowledge while developing an endless mode.

### **3.5.2 Evaluation of objectives**

In the previous reports, we have stated four main objectives.

#### **1<sup>st</sup> objectives: Develop a complete game that is interesting**

We have developed a game that is playable and challengeable. All of us think the game is interesting because of the mechanisms and 2D, 3D mode. We may find some tester to give us feedback afterwards. Now, we are satisfied with the game.

#### **2<sup>nd</sup> objectives: Achieve completeness, multiple elements in the game must come together including stage design, user interface, music and sound effects, and art style**

Our game has included all the components mentioned above. Stage design and user interface are designed ourselves. Music and sound effects are using free online assets. We have used Blender to create all art materials in 3D mode. In 2D mode, we use the character model from Unity asset store. Therefore, we think we have fulfilled this objective.

#### **3<sup>rd</sup> objectives: Design a core game mechanic using optimal pathfinding/optimal path problem**

26 handcraft stages and the random generating stages in Endless mode is designed with optimal pathfinding problem. Player need to use minimum resources to clear the stages and this is the same as the principle of optimal problem. Therefore, we have done well in this part.

#### **4<sup>th</sup> objectives: Increase the complexity of the problem with the extra game mechanic for players to solve**

We have a total of six mechanisms and the difficulty is increasing every world and every stage. In the first three worlds, the player will play in relatively smaller maps to familiar the basic flow of the game. From World 4 to World 6, we have included more than one mechanism in one stage to increase the difficulty. Also, the game map is larger including more nodes and stars to have higher complexity. Therefore, we think we did a good job in this part.

## 4. Discussion

We are going to have a brief analysis of our project. We have done well in some parts but our game still has room for improvement because of limited time and lack of knowledge.

### 4.1 Game Design

Our game can be divided into two parts: handcraft and endless. In this part, we will discuss the handcraft stages. We have designed the stages and worlds with increasing difficulty to provide a better playing experience. The game maps correlate with the mechanisms successfully. Some stages are a variation of another stage with different constraints or starting position to give a new challenge for the player. Player can practice creative thinking while playing the game.

At first, we want to design a puzzle with multiple solutions. However, we find that it is difficult to keep multiple solutions optimal while being challengeable. Now, player can still clear stages in different ways but the variation is not as much as we expected. Solution of some stages is not optimal because we do not have a strict mode including all restrictions to find the solution. The designing process of our game still can be improved.

Also, if you have played our game in both 2D and 3D mode, you will find that the art design and style have a lot of difference between the two modes. This is a problem that will make the player cannot enjoy our game. A good game should have a theme that lasts for the whole game so the game can create a good atmosphere for the players to enjoy. However, you can see that two modes were implemented by two different groups of people. One of the reasons is we do not have a consensus on the theme before the implementation. This part still has room for improvement if we want to develop a better game or software. The developing team should decide on a unified standard before assigning the tasks to the programmer. At the same time, this will be easier for testing and debugging in the maintenance period.

#### **4.2 Random generating game map (Endless mode)**

At the same time, this straightforward design of puzzle is an advantage for us to use machine learning for developing Endless mode. We pre-built a map with nodes, stars, bridges and character for generating a map in Endless mode. Then, we use the random seed to generate the game map with BFS. After testing, we do not find bugs that cause the stage cannot be solved. Although the difficulty is lower in endless mode, we are satisfied with the result because we can apply machine learning in our game.

Now, we only implement three mechanisms into endless mode. Portal, two stars and two worlds are more complicated and we need to spend more time to ensure no serious bug. Also, we have pre-built objects to generate game map by a random seed. If we have more knowledge, we may generate the map from zero without a pre-built map.

#### **4.3 Overall Game performance**

UI, music and art materials engaging player to play our game. Player can have a simple adjustment on the game setting in menu and every stage will have a corresponding control method provided in menu too. Although sometimes there are bugs during playing time, most of the time our game can run normally. Having more mechanisms or better art materials may make our game more interesting.

## 5. Conclusion

### 5.1 Summary

In this project, we developed a complete and interesting puzzle-solving game for the players. We got the chance to practice the knowledge that learnt from university such as software engineering and machine learning. Also, we can be familiar with Unity and blender that we did not study at university. After all, we developed a game as we expected and we are glad about the product.

### 5.2 Further development

We have several ideas to improve our game if we want to continue working on this project.

#### 5.2.1 Choice of game engine

We used Unity as our engine in this project because we can use the free education version and it is easier to get started. However, the UI and texture in Unity are not good as Unreal Engine. If we want to create better art materials, we may use Unreal to develop in the future.

#### 5.2.2 Improvement on Music and Sound Effect

We are using online resources for music and sound effect because none of us has experience in creating music. There are a lot of online tools for us to create our music. If we want to be a game developer later, we may study basic knowledge and then create music in the future.

#### 5.2.3 Improvement on puzzle design

There are many ways to design a good puzzle. For example, if we are expert in machine learning, we can develop a model to create the best puzzle. Or we may use mathematical model to design the puzzle. This may be the task for us as game developer in the future.

#### **5.2.4 Combination of 3D and 2D mode**

One problem of 3D mode is hard to have detail plan. One solution is implementing 2D mode as a popup panel into 3D mode. Player can plan their route on the popup panel in a way similar to 2D mode. Then, they can follow the planned route to clear the stage and there will be some visible assistance for the player such as the planned route has different color and the bridge in the route has been built already. This is a task that need to take a lot of time so we cannot implement it into the final product. It is a interesting idea that can combine the strengths of two modes which will make our game becoming better in further development.

## Project Planning

### Distribution of Work

Task	Jason	Henry	Woody	Tommy
Research on computer game	✓	★	★	★
Propose idea	✓	★	✓	★
Game planning	✓	★	✓	★
Write proposal	★	★	★	★
Write monthly reports	★	★	★	✓
Write progress report	★	✓	★	★
Write final report	✓	★	★	★
Prepare for presentation	★	★	✓	★
Coding	★	✓	★	✓
Game testing	★	✓	★	✓
Debugging	★	✓	★	✓
Game improvement	★	✓	★	✓
User interface	★	✓	★	✓
World 1	✓	★	★	★
World 2	✓	★	★	★
World 3	✓	★	★	★
World 4	★	★	✓	★
World 5	★	★	✓	★
World 6	★	★	✓	★
Endless mode	✓	★	✓	★

✓ Leader ★Assistant

**GANTT Chart**

Task	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Research on computer game										
Propose idea										
Game planning										
Write proposal										
Write monthly reports										
Write progress report										
Write final report										
Prepare for presentation										
Coding										
Game testing										
Debugging										
Game improvement										
User interface										
World 1										
World 2										
World 3										
World 4										
World 5										
World 6										
Endless mode										

## Required Hardware and Software

### Hardware (for Unity)

Development PC:

- Operating system: Windows: 7 SP1+, 8, 10, 64-bit versions only
  - macOS: 10.12+
  - Linux: Fixed at: Ubuntu 16.04, 18.04 and CentOS 7
- CPU: SSE2 instruction set support.
- GPU: Graphics card with DX10 (shader model 4.0) capabilities.

Testing PC:

- Operating system: Windows: 7 SP1+
  - macOS: 10.12+
  - Linux: Ubuntu 16.04+
- CPU: SSE2 instruction set support.
- GPU: Graphics card with DX10 (shader model 4.0) capabilities.

### Software

Unity	Game development
Blender	Art materials, character and animation

## Reference

[1] O. Kaplan, "Mobile gaming is a \$68.5 billion global business, and investors are buying in,"

Tech Crunch , Aug 23, 2019. [Online]. Available:

<https://techcrunch.com/2019/08/22/mobile-gaming-mints-money/>. [Accessed Sep 17, 2020].

[2] "How the CPU Affects Your Gaming Experience", n.d. Accessed on: Oct. 11, 2020. [Online].

Available:

<https://www.intel.com/content/www/us/en/gaming/resources/how-cpus-affect-your-gaming-experience.html>

[3] K. Oktriono and H. Chong, "Developing 3D action puzzle game application "The Mechanic"

using UNITY," 2015 IEEE Conference on e-Learning, e-Management and e-Services (IC3e), Melaka, 2015, pp. 120-124, doi: 10.1109/IC3e.2015.7403498.

[4] A. Frazer, D. Argles and G. Wills, "The Same, But Different: The Educational Affordances of

Different Gaming Genres," 2008 Eighth IEEE International Conference on Advanced Learning

Technologies, Santander, Cantabria, 2008, pp. 891-893, doi: 10.1109/ICALT.2008.228.

[5] M. Gardner, "MATHEMATICAL GAMES", Scientific American, vol. 223, no. 4, pp. 120-123, October 1970.

[6] Thekla, Inc. (2016). *The Witness official blog* [Online]. Available: <http://thewitness.net/>

[7] I. Kostitsyna, M. Löffler, M. Sondag, W. Sonke and J. Wulms., (2018), *The hardness of Witness puzzles* [Online]. Available: [https://conference.imp.fu-berlin.de/eurocg18/download/paper\\_67.pdf](https://conference.imp.fu-berlin.de/eurocg18/download/paper_67.pdf)

- [8] Z. Abel, J. Bosboom, E.D. Demaine, L. Hamilton, A. Hesterberg, J. Kopinsky, J. Lynch and M. Rudoy., (last revised Feb 2019), *Who witnesses The Witness? Finding witnesses in The Witness is hard and sometimes impossible* [Online], Available: [https://erikdemaine.org/papers/Witness\\_FUN2018/paper.pdf](https://erikdemaine.org/papers/Witness_FUN2018/paper.pdf)
- [9] WOLFRAM Demonstrations Project. (March 2011). *Traveling Salesman Game* [Online]. Available: <https://demonstrations.wolfram.com/TravelingSalesmanGame/>
- [10] S. Rabin, “Introduction to Game Development (2.ed)”. USA: Cengage., 2010, pp. 70.
- [11] Johann, Nature Loves Math (2011). “Icosian: A graph theory game” [Online]. Available: <https://www.naturelovesmath.com/en/games/icosian-a-graph-theory-game/>
- [12] ForgeableSum, Strike Tactics Dev Blog (2017). “STARCRAFT 1 PATHFINDING: A TECHNICAL ANALYSIS”. [Online] Available: <http://striketactics.net/devblog/starcraft-1-pathfinding-technical-analysis>
- [13] Jay Scott, Starcraft AI blog (2016). “pathing 6 - flocking” [Online]. Available: <http://satirist.org/ai/starcraft/blog/archives/68-pathing-6-flocking.html>

## Appendix A: Meeting Minutes

### Minutes of the 1<sup>st</sup> Project Meeting (Online)

Date: April 24, 2020

Time: 10:00 pm

Software: Discord

Present: Jason, Henry, Woody, Tommy

Absent: None

Recorder: Jason

#### 1. Approval of minutes

This was the first formal group meeting, so there were no minutes to approve.

#### 2. Report on progress

2.1 All team members have read the instructions of FYP and have prepared ideas to discuss.

2.2 All team members have read the project list and have chosen the project.

2.3 We have a WhatsApp group and discord group for our FYP.

#### 3. Discussion items

3.1 In this meeting, we need to choose the project that everyone accepts to do in the coming year. We form groups through the online platform so we have talked about ourselves in this meeting.

3.2 All team members are interested in a topic related to gaming. Some team members have experience of developing board games before. After the discussion, we choose making a computer game as our FYP.

3.3 We have done researches on gaming engine during the meeting and we have chosen Unity as our development tool.

3.4 We decided to apply for the project: computer game on the project list.

3.5 We wanted to make a 3D open-world game with the combat system.

#### 4. Goals before next meeting

4.1 Tommy will communicate with Prof. Sunil ARYA and get the approval of applying for the project.

4.2 Jason, Henry and Woody will research developing games.

4.3 All team members will share opinions of computer games using an online platform.

5. Meeting adjournment and the next meeting

The meeting was adjourned at 12:00 am.

The next meeting will be after the examination.

## **Minutes of the 2<sup>nd</sup> Project Meeting (Online)**

Date: June 5, 2020

Time: 10:00 pm

Software: Discord

Present: Jason, Henry, Woody, Tommy

Absent: None

Recorder: Henry

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Our group has got the approval of Prof. Sunil ARYA and apply for the project successfully.

2.2 All team members have filled the form of their opinions on computer games.

2.3 All team members have researched different type of computer games.

3. Discussion items

3.1 After sharing the opinion using online platform, all team members have agreed to make a puzzle-solving game.

3.2 In this meeting, we watched FYP video of the previous year and shared our thoughts. We agreed that making a 3D open-world game will be difficult because of limited time.

3.3 We thought about making 2D platform game but the difficulty is hard to adjust to the appropriate level when we don't have experience of developing the game.

3.4 We have shared the experience of playing puzzle-solving games. We decide to implement the games using computer algorithms that we have learned before like travel salesman problem.

4. Goals before next meeting

4.1 All group members will learn how to use Unity in the summer holiday

- 4.2 All group members will study and research puzzle-solving games.
  - 4.3 All group members will write the proposal using Microsoft Word and Google Drive.
5. Meeting adjournment and the next meeting  
The meeting was adjourned at 12:00 am.  
The next meeting will be at the end of the summer holiday.

### **Minutes of the 3<sup>rd</sup> Project Meeting (Online)**

Date: September 16, 2020

Time: 10:00 pm

Software: Discord

Present: Jason, Henry, Woody, Tommy

Absent: None

Recorder: Tommy

1. Approval of minutes  
The minutes of the last meeting were approved without amendment.
2. Report on progress
  - 2.1 All group members understand basic knowledge of Unity.
  - 2.2 We have finished drafting proposal using the information from research.
3. Discussion items
  - 3.1 All group members shared their timetable and agree to arrange the meeting with supervisor at mid-October. Tommy will be responsible for communication with professor.
  - 3.2 All group members agreed to implement a simple stage in this month for demonstration in the meeting.
  - 3.3 We divided the progress report for lang4030 in four parts. Each group members will response for one part and Henry will be responsible for proofread after finishing the report.
  - 3.4 Woody suggested that we should start working on individual essay and we could decide the topic in this meeting. However, we found it is difficult to choose an appropriate topic for the essay after reading the previous work on the FYP website. Therefore, we agreed to do some research first and discuss again in the next meeting.

4. Goals before next meeting

- 4.1 All group members will research on topic of individual essay.
- 4.2 All group members will start writing progress report for lang4030.
- 4.3 Jason and Henry will start coding for a simple stage.
- 4.4 Tommy and Woody will start designing more complex stages.

5. Meeting adjournment and the next meeting

The meeting was adjourned at 11:00 pm.

The next meeting will be on next week.

### **Minutes of the 4<sup>th</sup> Project Meeting (Online)**

Date: September 23, 2020

Time: 10:00 pm

Software: Discord

Present: Jason, Henry, Woody, Tommy

Absent: None

Recorder: Woody

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

- a. All group members have completed researching on individual essay's topic and started working on progress report.
- b. Jason and Henry have started coding on some basic elements in the map.
- c. Tommy and Woody have designed part of the complex stages

3. Discussion items

- a. All group members agreed that the progress of the game development is good and we should keep this trend.
- b. Jason suggested that we can work on the map first so that we can use the map to demonstrate our game to the supervisor in the coming meeting. All group members agreed.
- c. Tommy faced difficulties when working on the progress report. Woody and Henry proofread the overview and gave opinion on how to continue the writing.

4. Goals before next meeting
  - a. All group members will work on the map first to demonstrate the map to professor in the coming meeting.
  - b. All group members will continue working on the progress report and the deadline will be 9<sup>th</sup> October in our group.

5. Meeting adjournment and the next meeting

The meeting was adjourned at 10:30 pm.

The next meeting will be arranged later.

### **Minutes of the 5<sup>th</sup> Project Meeting (Online)**

Date: October 7, 2020

Time: 3:30 pm

Software: Zoom

Present: Dr. Sunil Arya, Jason, Henry, Woody, Tommy

Absent: None

Recorder: Tommy

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 A simple game demo have been completed and demonstrated to Dr. Sunil Arya during the meeting.

2.2 The progress report have been done and we need to start proofreading

3. Discussion items

3.1 We have demonstrated our game demo and the core design of the game to professor. However, there are still room for improvement of both parts.

3.2 The professor suggested us to add some advanced function into the game after we have completed the basic game.

3.3 We told the professors that we will create our own art design and professors agree. We will start working on the art design in the next semester.

4. Goals before next meeting

4.1 The coding team will continue developing the game and find more appropriate art material.

4.2 The design team will think of adding more features and more complicated

design of the stages.

4.3 All group members need to finish the individual essay before 23/10.

5. Meeting adjournment and the next meeting

The meeting was adjourned at 4:15 pm.

The next meeting will be arranged later.

## **Minutes of the 6<sup>th</sup> Project Meeting (Online)**

Date: October 29, 2020

Time: 10:00 pm

Software: Discord

Present: Jason, Henry, Woody, Tommy

Absent: None

Recorder: Henry

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

2.1 Implement team has developed better demo for the game.

2.2 Design team has started developing more game mechanism which has 5 main mechanisms now.

3. Discussion items

3.1 Jason and Woody has discussed about the data structure of endless mode but did not have a conclusion.

3.2 We decided to set up a private repository in Github to have better version control within the implement team.

4. Goals before next meeting

4.1 Henry will try to develop the gameplay in 3D way and Tommy will develop in 2D way.

4.2 Henry will use blender to create better art materials.

4.3 Jason and Woody will keep finding a better solution for the endless mode.

5. Meeting adjournment and the next meeting

The meeting was adjourned at 11:00 pm.

The next meeting will be arranged later.

## **Minutes of the 7<sup>th</sup> Project Meeting (Online)**

Date: November 18, 2020

Time: 10:00 pm

Software: Discord

Present: Jason, Henry, Woody, Tommy

Absent: None

Recorder: Woody

### **1. Approval of minutes**

The minutes of the last meeting were approved without amendment.

### **2. Report on progress**

- 2.1 Henry has developed 3D gameplay with better graphic using blender.
- 2.2 Tommy has developed the game stage selection UI.
- 2.3 Jason has tested the game demo and discovered a few bugs.
- 2.4 Woody has finished designing all major mechanisms for our game.

### **3. Discussion items**

- 3.1 Jason talked about the bugs that he has find and the implement team try to fix some of them during the meeting.
- 3.2 We agreed that the animation need to be improved to be smoother.
- 3.3 Woody talked about the final mechanism – two-layer world. Implement team discuss about the possible difficulties during implementation.

### **4. Goals before next meeting**

- 4.1 Henry and Tommy need to develop a better game demo before the next meeting.
- 4.2 Jason and Woody need to prepare documents for demonstrating our game design and ask professors questions about the endless mode.

### **5. Meeting adjournment and the next meeting**

The meeting was adjourned at 11:00 pm.

The next meeting will be arranged at 20/11(Fri) with professor.

## **Minutes of the 8<sup>th</sup> Project Meeting (Online)**

Date: November 20, 2020

Time: 2:00 pm

Software: Zoom

Present: Dr. Sunil Arya, Jason, Henry, Woody, Tommy

Absent: None

Recorder: Jason

### **1. Approval of minutes**

The minutes of the last meeting were approved without amendment.

### **2. Report on progress**

2.1 Implement team has prepared the game demo to demonstrate in this meeting

2.2 Design team has prepared the google document to demonstrate the game design.

### **3. Discussion items**

3.1 Henry and Tommy demonstrated the gameplay in two ways and with basic UI.

3.2 Jason and Woody discussed the game design with professor. Also, talked about the difficulties in developing auto-generating game map system.

3.3 We agreed that the developing speed is a bit slow and need to be faster in December.

### **4. Goals before next meeting**

4.1 The implement team will continue developing the game in two ways.

4.2 The design team will write detailed instruction of the world and mechanism.

4.3 Before the next meeting with professor, implement team need to develop all stages in World 1 and World 2 (if time is enough). Also, the UI for players to select 2D stage or 3D stage.

### **5. Meeting adjournment and the next meeting**

The meeting was adjourned at 2:30 pm.

The next meeting will be arranged later after the final examination period.

## **Minutes of the 9<sup>th</sup> Project Meeting (Online)**

Date: December 23, 2020

Time: 10:00 pm

Software: Discord

Present: Jason, Henry, Woody, Tommy

Absent: None

Recorder: Tommy

### 1. Approval of minutes

The minutes of the last meeting were approved without amendment.

### 2. Report on progress

2.1 Implement team has prepared the game demo of world 1 and world 2.

2.2 Design team has found some problems during developing endless mode.

### 3. Discussion items

3.1 Henry and Tommy showed the game demo and got feedbacks from Jason and Woody.

3.2 Jason and Woody talked about the difficulties of developing endless mode. All of us agreed that developing endless mode in 3D gameplay is easier.

### 4. Goals before next meeting

4.1 The implement team will develop stages for the first 4 world.

4.2 Implement team will combine the unity project of 2D and 3D gameplay.

4.3 Design team will keep researching on Endless mode and use Unity to develop some simple stages.

### 5. Meeting adjournment and the next meeting

The meeting was adjourned at 11:30 pm.

The next meeting will be arranged at 8/1/2021 (Fri) with professor.

## **Minutes of the 10<sup>th</sup> Project Meeting (Online)**

Date: January 8, 2021

Time: 2:00 pm

Software: Zoom

Present: Dr. Sunil Arya, Jason, Henry, Woody, Tommy

Absent: None

Recorder: Henry

1. Approval of minutes

The minutes of the last meeting were approved without amendment.

2. Report on progress

- a. Implement team has prepared the game demo of world 1 to world 6.
- b. Design team has developed the random generating map system using C#.

3. Discussion items

- a. We agreed that the 3D gameplay need to have better UI (Minimap).
- b. Jason and Woody demonstrated the generating map system and were thinking how to implement the system into Unity

4. Goals before next meeting

- a. The implement team will develop all stages.
- b. Implement team will combine the unity project of 2D and 3D gameplay after developing all stages
- c. Design team will keep researching on Endless mode and how to implement it into Unity.
- d. Start working on progress report.

5. Meeting adjournment and the next meeting

The meeting was adjourned at 2:30 pm.

The next meeting will be arranged at 5/2/2021 (Fri) before submitting the progress report.

### **Minutes of the 11<sup>th</sup> Project Meeting (Online)**

Date: February 5, 2021

Time: 4:00 pm

Software: Zoom

Present: Dr. Sunil Arya, Jason, Henry, Woody, Tommy

Absent: None

Recorder: Woody

1. Approval of minutes

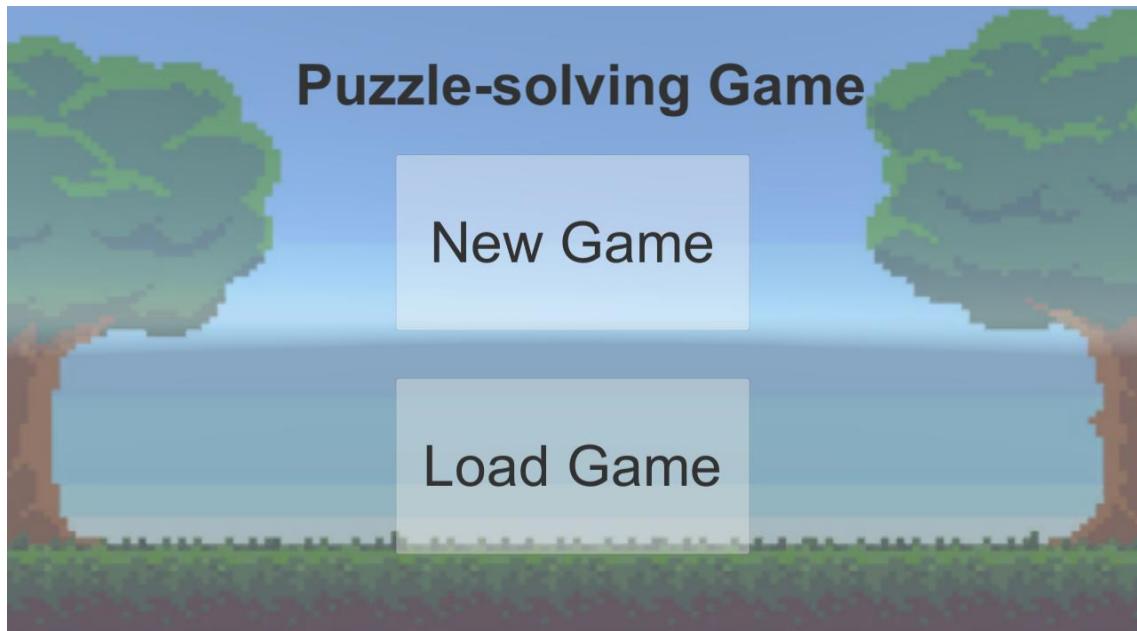
The minutes of the last meeting were approved without amendment.

2. Report on progress

- a. Implement team has developed all stages (World 1 to World 6) and combined their projects.
  - b. Design team has developed the 2D gameplay Endless mode (testing stage).
  - c. The progress report is almost finished.
3. Discussion items
- a. Jason and Woody have demonstrated how to generate game stage of push or pull using AI.
  - b. All of us have discussed our future working plan with professor.
4. Goals before next meeting
- a. Finish the progress report before the deadline.
  - b. Implement the endless mode with more mechanisms.
  - c. Improve UI.
  - d. Game debugging.
5. Meeting adjournment and the next meeting
- The meeting was adjourned at 4:30 pm.
- The next meeting will be arranged later.

## Appendix B: Game Demo

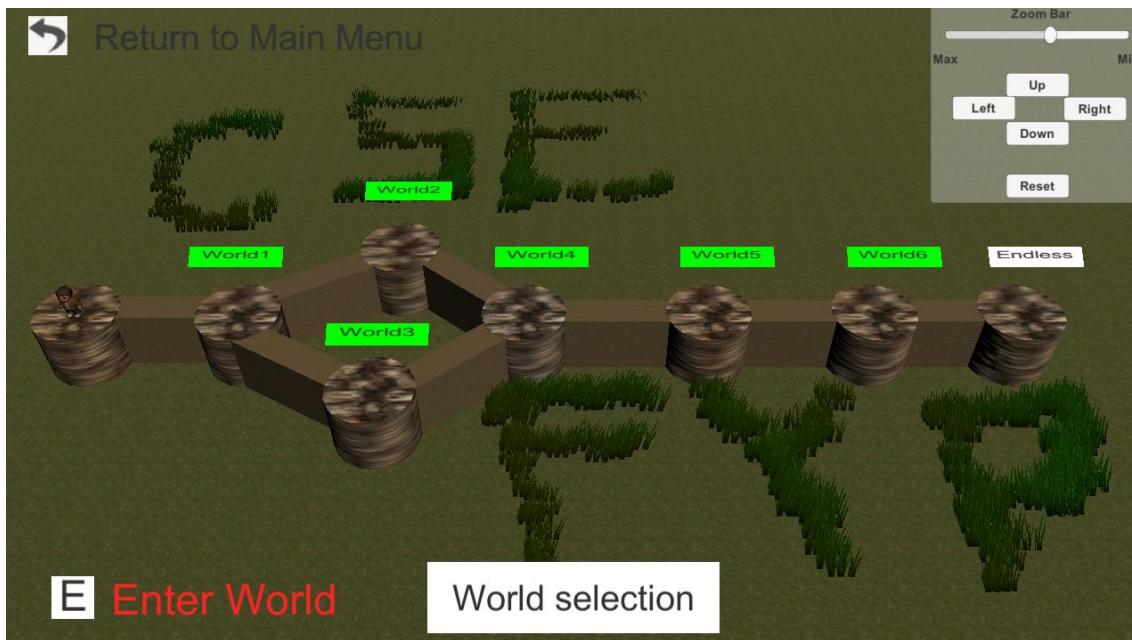
Start menu



Main menu



World selection



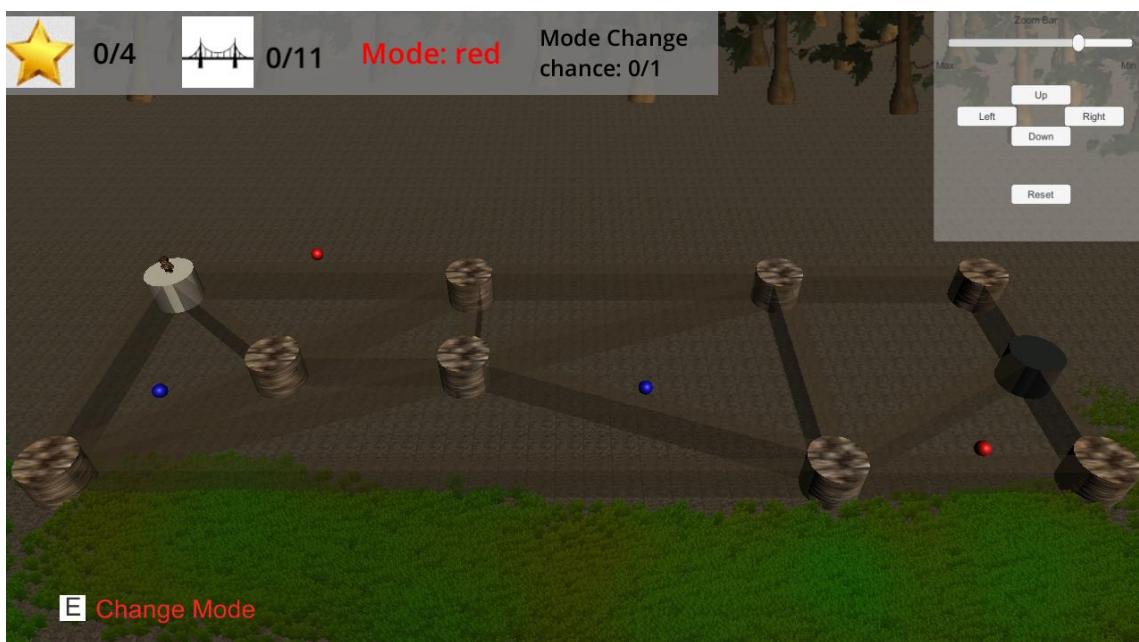
Stage selection



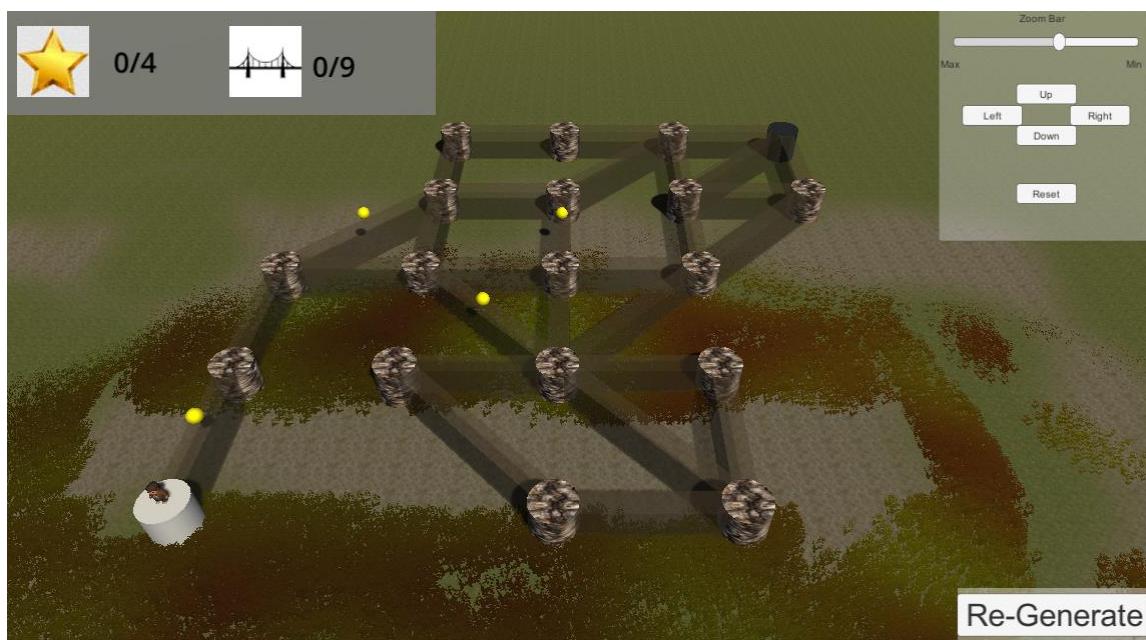
3D gameplay



2D gameplay



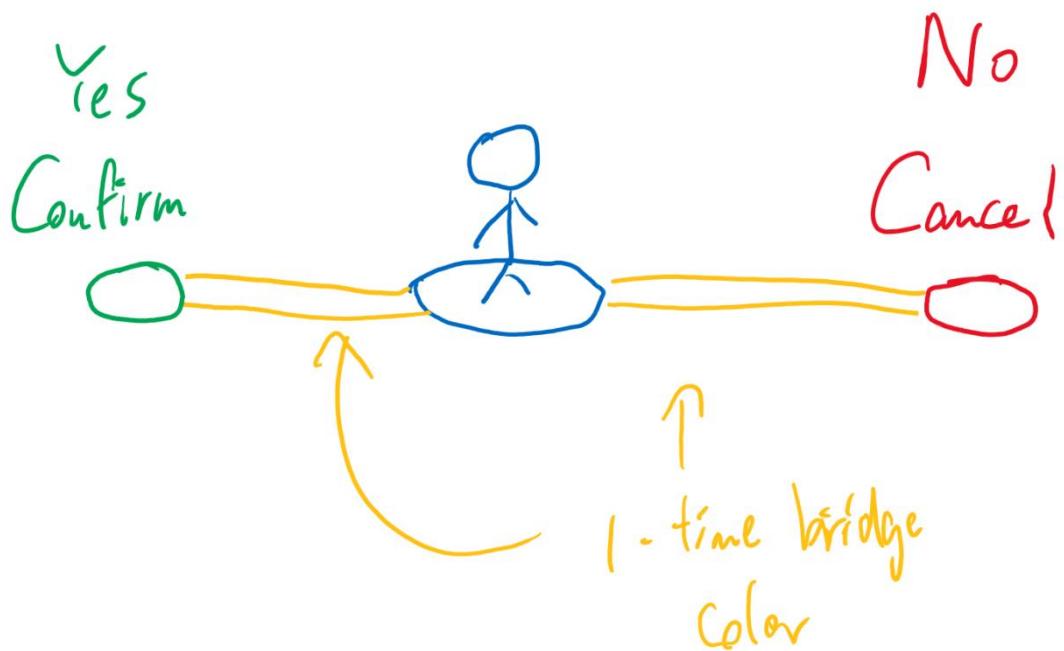
Endless mode (2D gameplay)



Re-Generate

## Appendix C: Game design document

Select Y/N



This is to replace the common confirm button, which makes it more immersive.



## World 1.

The world one is a tutorial world that, go through what the player has to do to win.

They will spawn at the Start position and check win condition when they stand on End position.

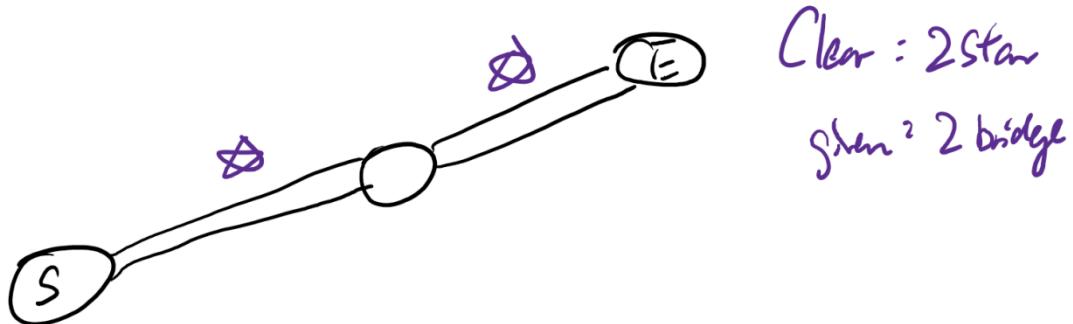
They can walk through edges once only and they will have to collect the stars along their path.

Several dialogues will also pop up in different stages to guide them how to play the game.

World one will be focusing on walk but not other game mechanics. They will be introduced in later stages to reduce complexity and have a progressive learning curve.

### Stage 1-1

#teach how to move

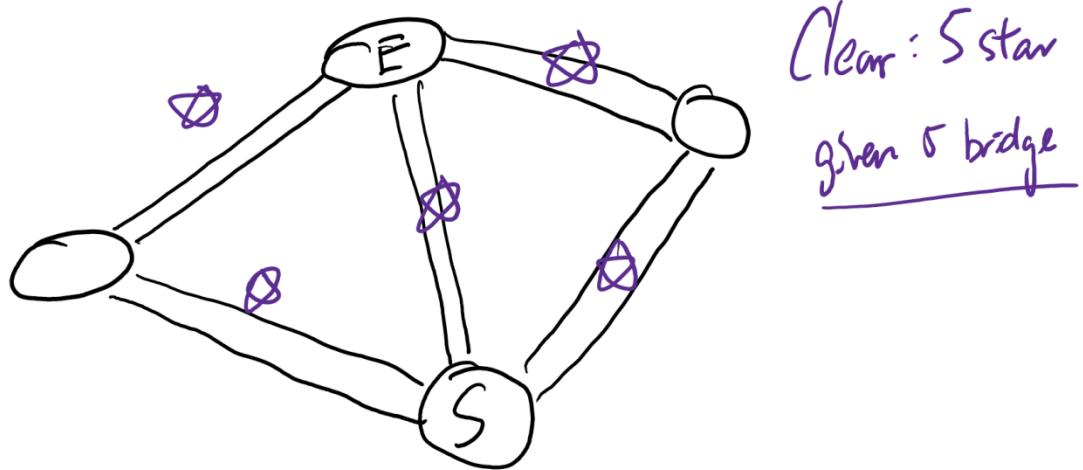


The player will spawn at point S, collecting 2 stars along his path and go to point E. At point E we will check if the player fulfills the clear condition (2/2 stars).

This is a gameplay demo with stage 1-1 and stage 1-2

[https://hkustconnect-my.sharepoint.com/:v/g/personal/nhto\\_connect\\_ust\\_hk/EXmGXh2JnLBNojNLqHOOzsBrz29-57MP9ZSP7WaqTyIWw?e=UKsVvO](https://hkustconnect-my.sharepoint.com/:v/g/personal/nhto_connect_ust_hk/EXmGXh2JnLBNojNLqHOOzsBrz29-57MP9ZSP7WaqTyIWw?e=UKsVvO)

## Stage 1-2



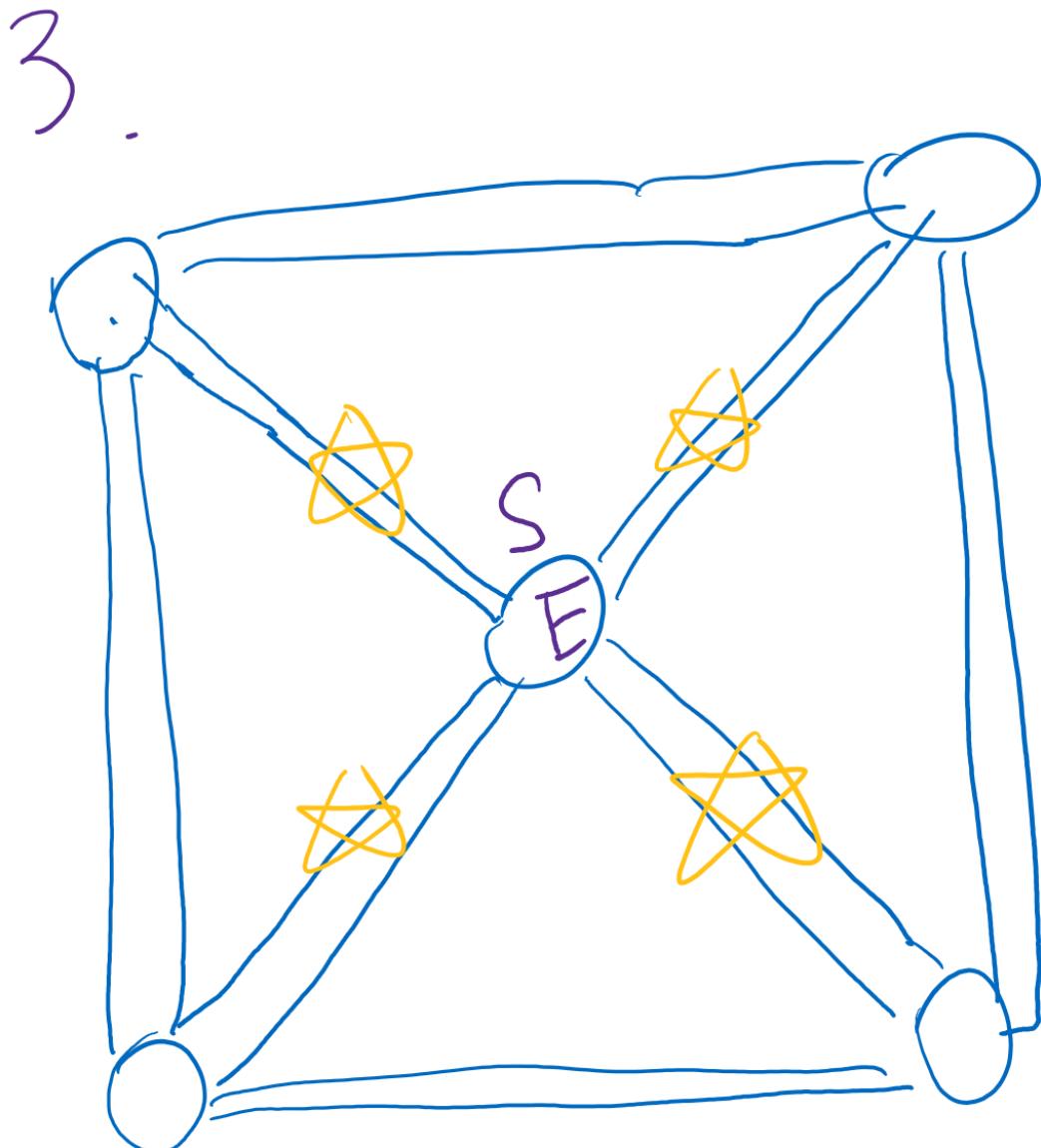
Similar to Stage 1-1, just construct a eulerian path from S to E so that the player can collect all the stars during their path.

Link to gameplay demo: (this is the 3D-gamemode, players can freely choose whether they want to play the game in 2D, just like the previous video, or 3D like this one)

<https://drive.google.com/file/d/1W6hHD4Bt5dAaM2gT-SahaiKF3gLwn2aj/view?usp=sharing>

**Stage 1-3**

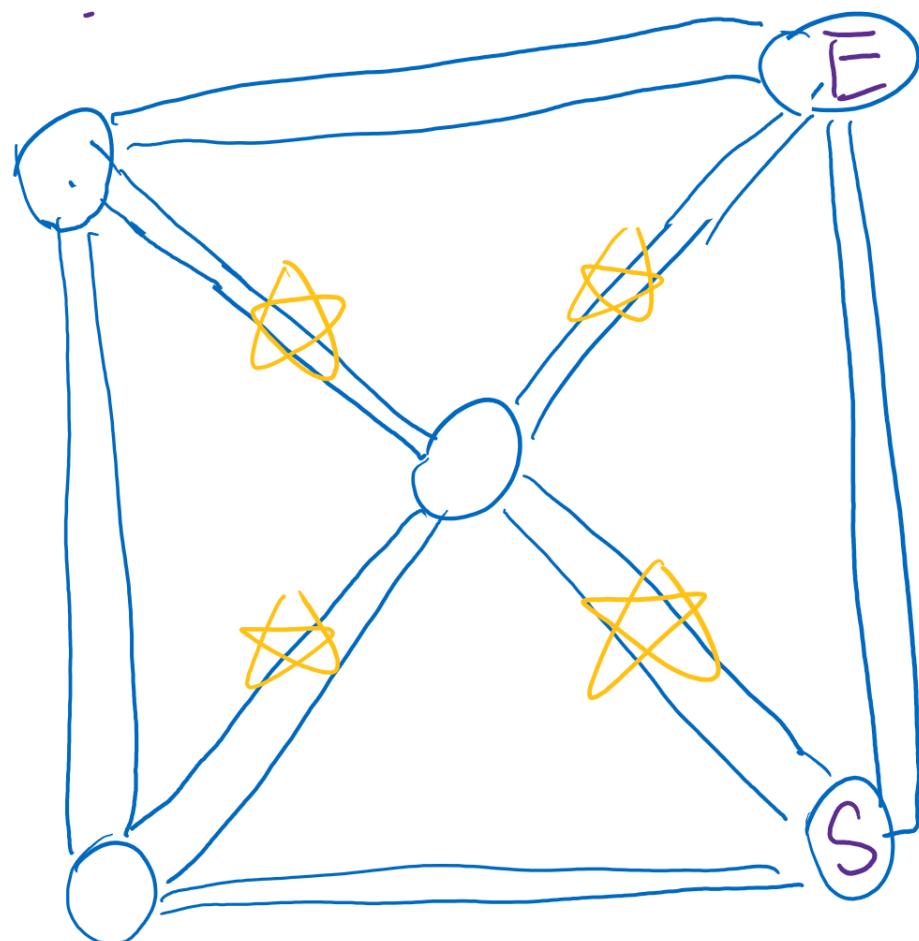
(Best = 6)



**Stage 1-4**

(Best = 5)

4



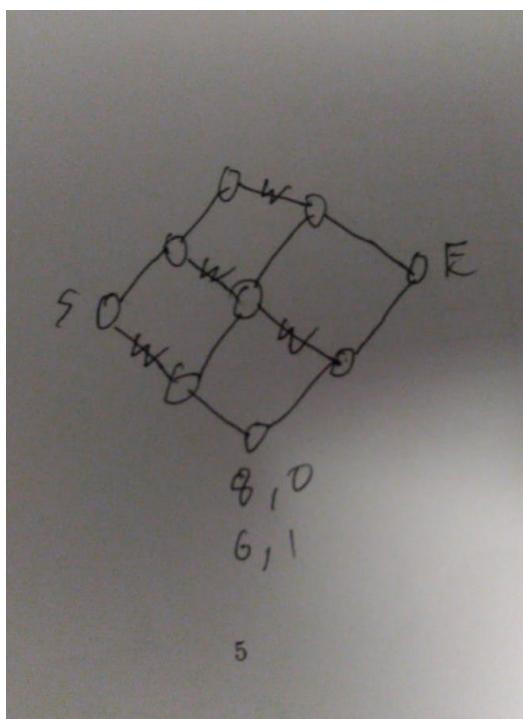
## World 2

#introduce pull which allow the player to pull the objects without walk the path  
Since pull is a powerful tool to collect stars from edges nearby, the number of pull available must be limited.

### Stage 2.1

also a tutorial for pull

Pull = 1, bridge = 6 #unconfirmed if best



### Stage 2.2

Pull = 1, bridge = 16 #unconfirmed if best

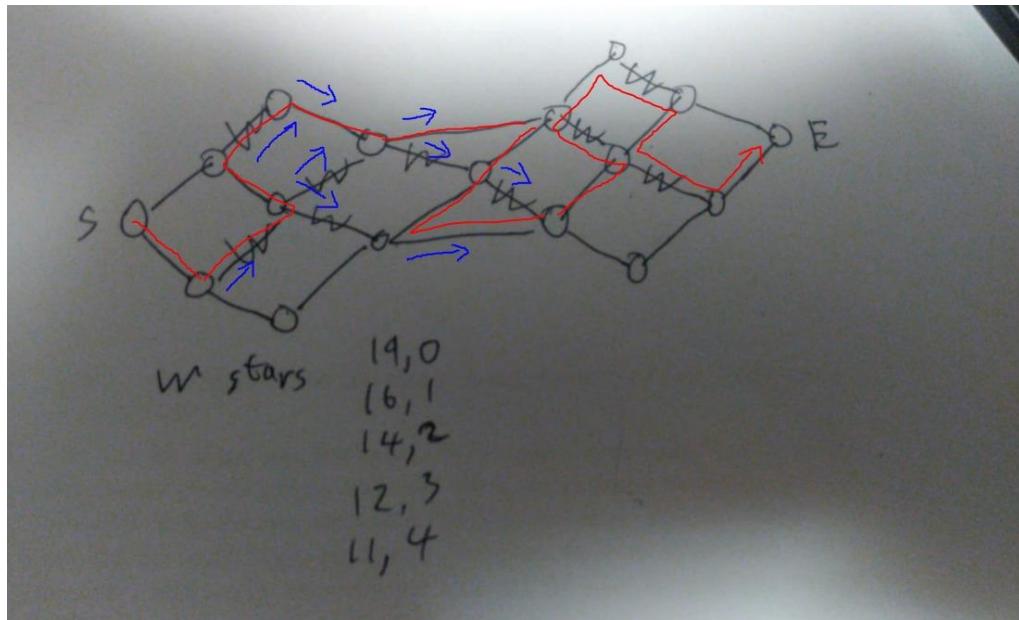
### Stage 2.3

Pull = 3, bridge = 12 #unconfirmed if best

### Stage 2.4

Pull = 4, bridge = 10 #unconfirmed if best

all these 3 level will use the same map while the solution will be reasonably different and we hope that reversing the idea of improving the old solution but to create a new solution for new difficulties will be interesting



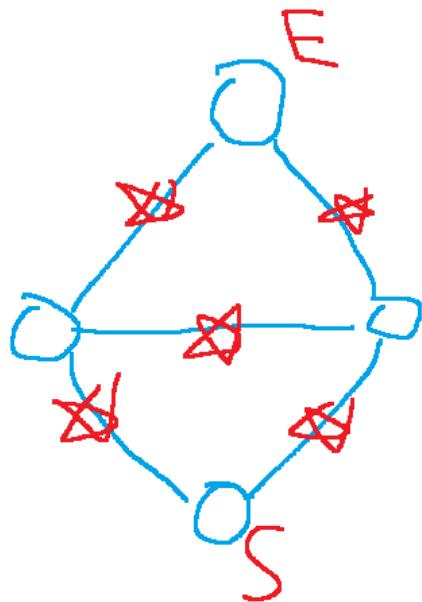
(p.s. I ctrl-CV wrong pic, please ignore the arrows and lines)

### World 3.

#introduce push

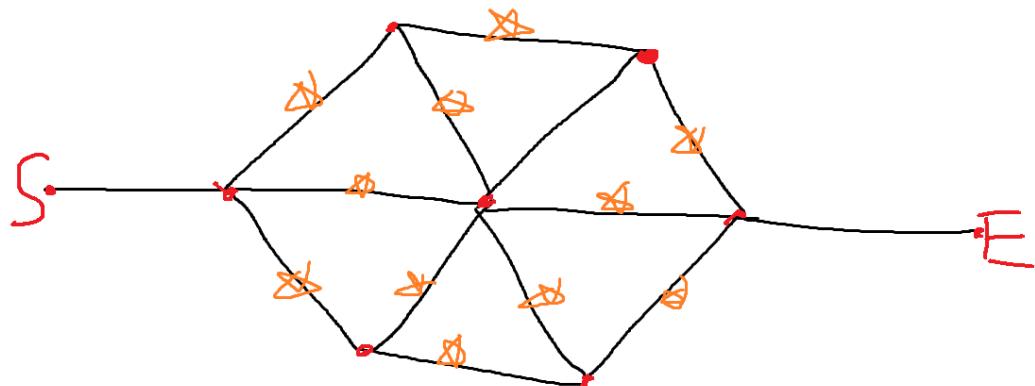
#### Stage 3-1

Push = 2, bridge = 3



#### Stage 3-2

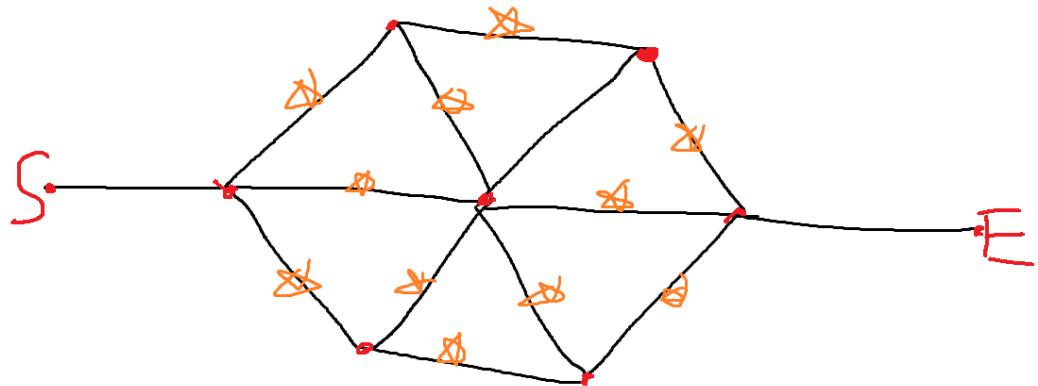
Push = 2, bridge = 12



### Stage 3-3

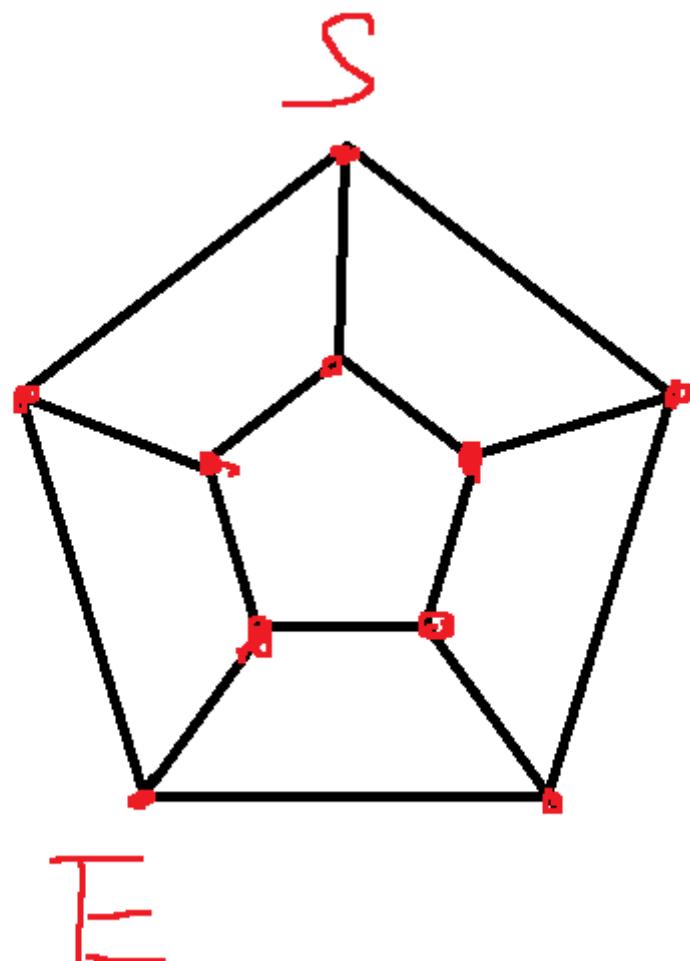
Push = 5, bridge = 9

#Favor text: oh no we have not enough material, use the push ability to get all the stars!



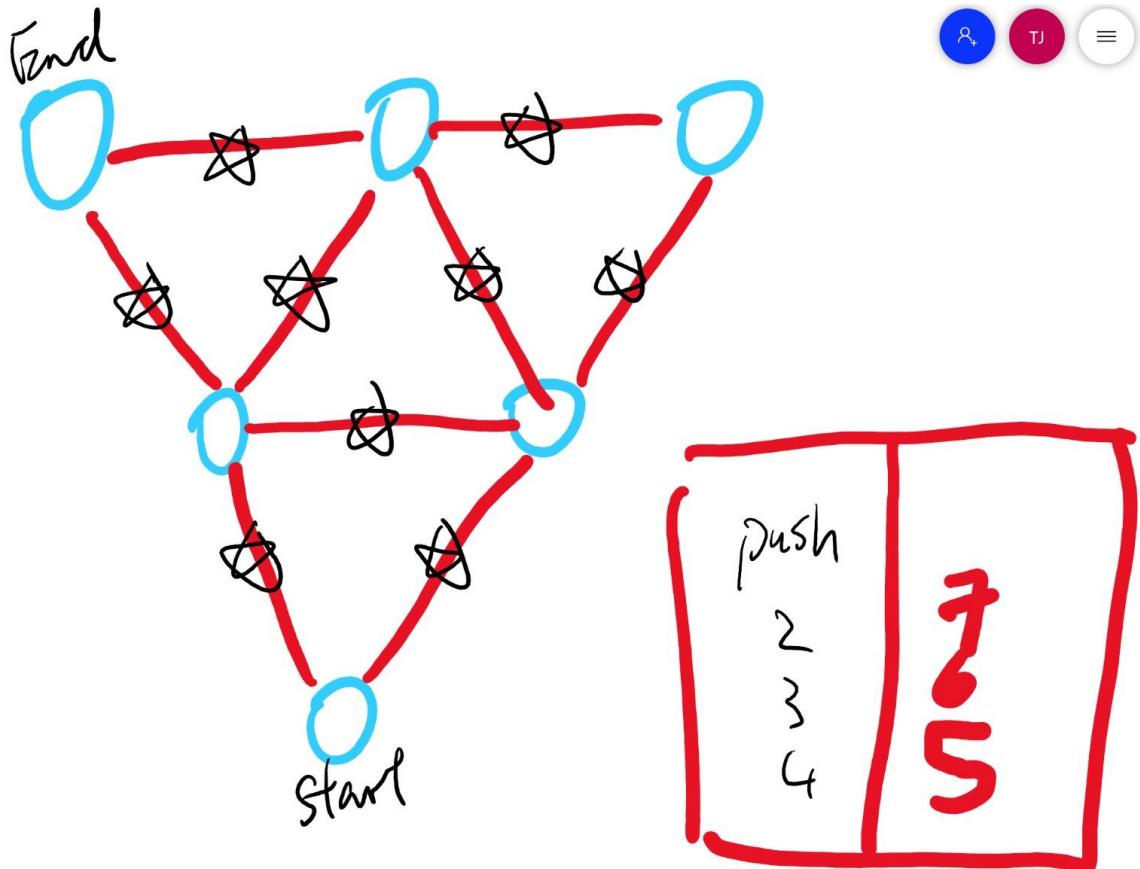
**Stage 3-4**

Push = 6, bridge = 9



### Stage 3-5

Push = 2, bridge = 7



### Stage 3-6

Push = 4, bridge = 5

(same map as above)

## World 4.

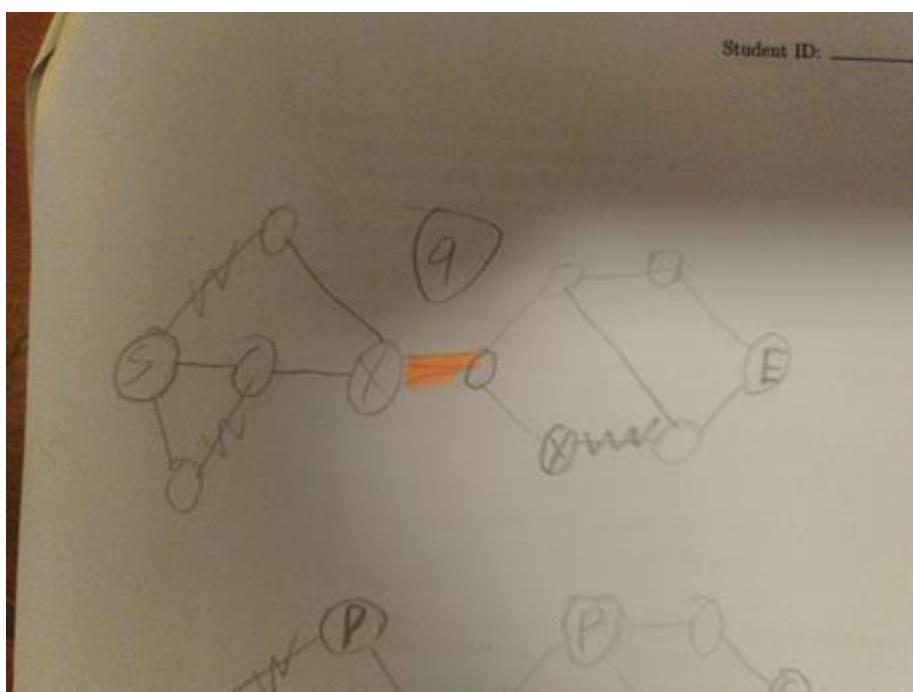
#introduce portal

We introduce portal which play can portal to the same group (indicated as X,P,Y,Q) with no cost. And since portal will need the map to be big while it may only increase the complexity in a tedious way, so I add the link that you can push but can't build and walk on(indicated as Orange line which will also have star on in)

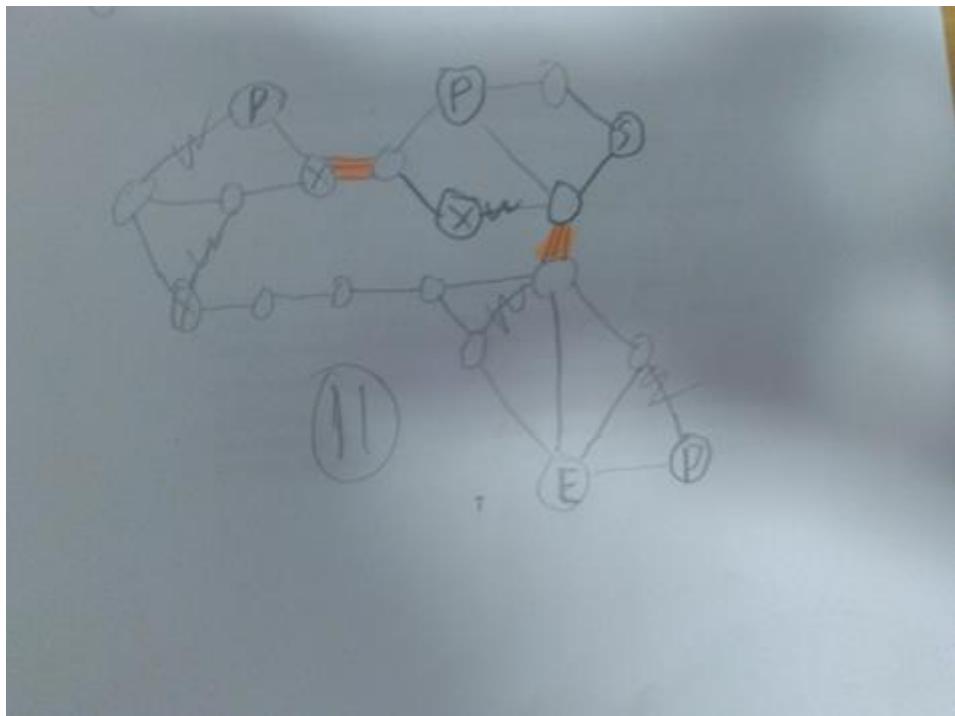
And unlimited push for these 4 levels we have now.

This stage were mainly about abusing portal (portal in, push, portal out will be 0 move) and how much start and end in different location can change the cost of the path  
Also the map will grow so it can serve for increasing difficulties and maybe story purpose.

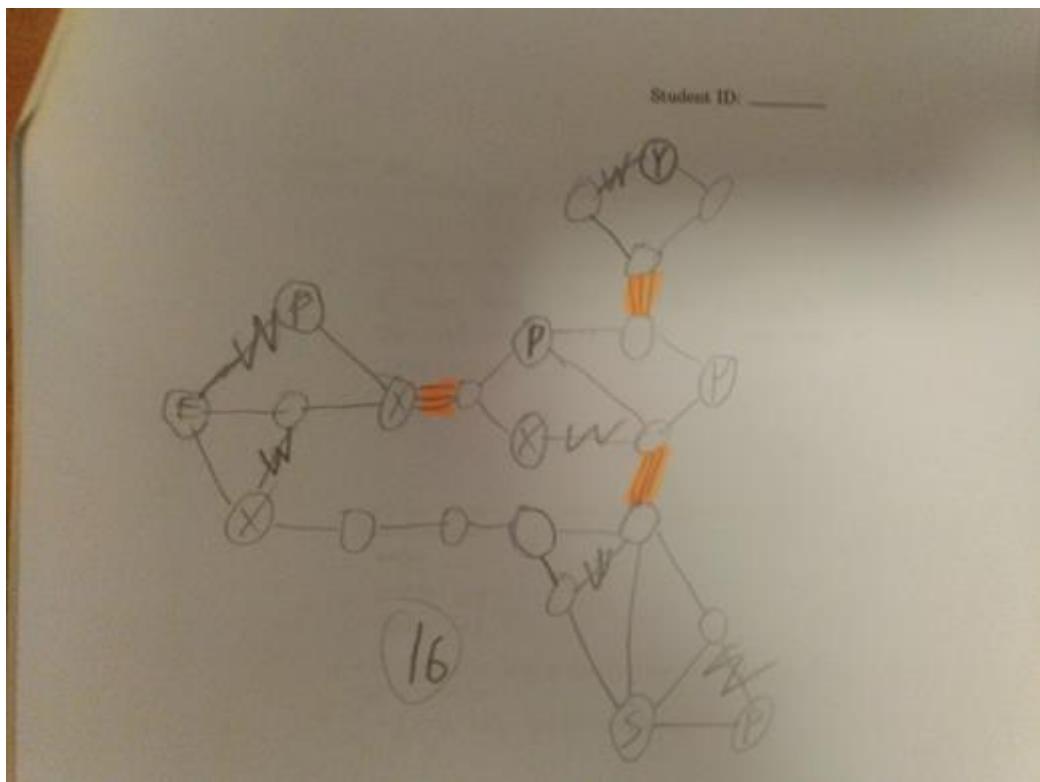
### Stage 4.1



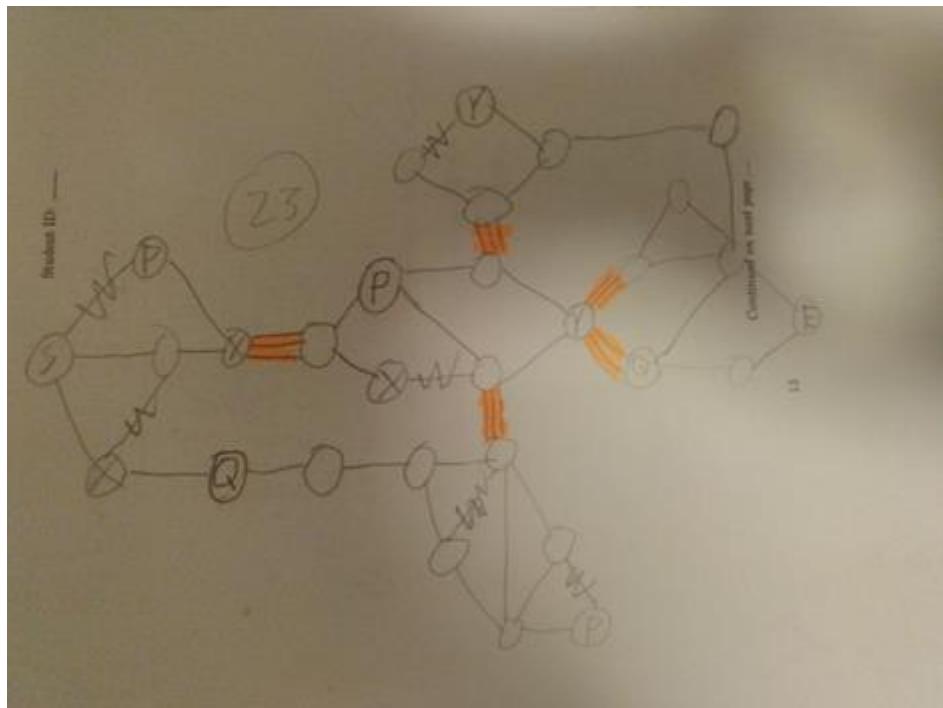
**Stage 4.2**



**Stage 4.3**



### Stage 4.4



## World 5

In this world,

1. player will have a mode (red/ blue ) which the player can only interact with the object with same colour (walk pass without collecting the object with different colour)
2. the player will be able to change their mode to the other color (limited times)

so each map will be 2 level start with different colour

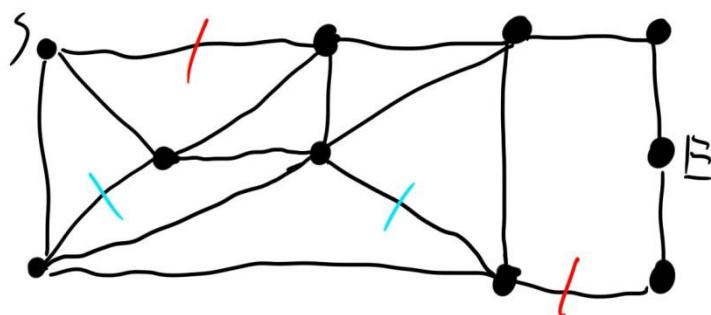
the first 2 will be a simpler level

### Stage 5.1

start with mode **blue** with 11 bridge, 1 change

### Stage 5.2

start with mode **red** with 11 bridge, 1 change



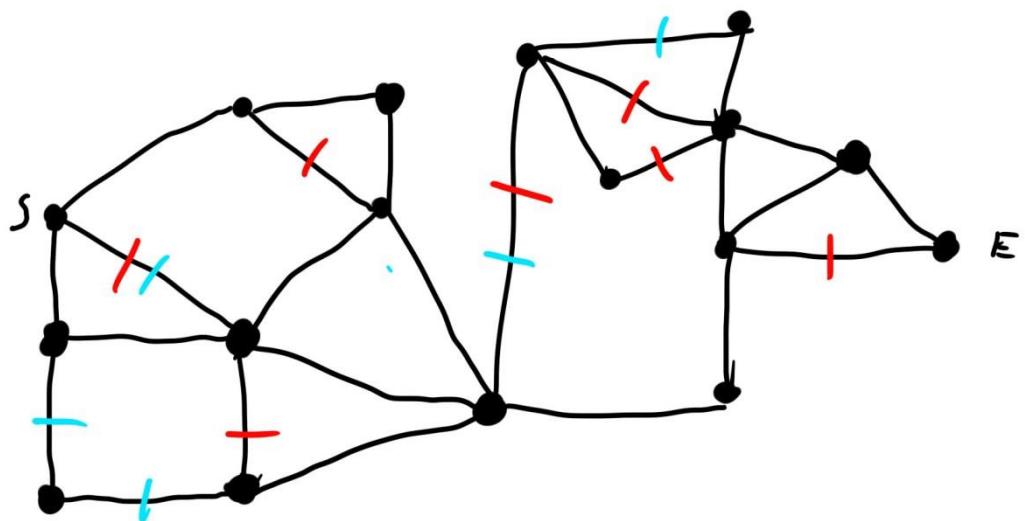
the next 2 will try to use the idea of having both type of object on same link.  
so the player cannot collect both if they only walk that path and very likely have to also  
change form ion one of those note.

### Stage 5.3

start with mode blue with 16 bridge, 1 push, 3 pull, 2 change

### Stage 5.4

start with mode red with 14 bridge, 1 push, 3 pull, 2 change



## World 6

this world is similar to world 4 but instead of 2 type of object, now have 2 type of bridges.

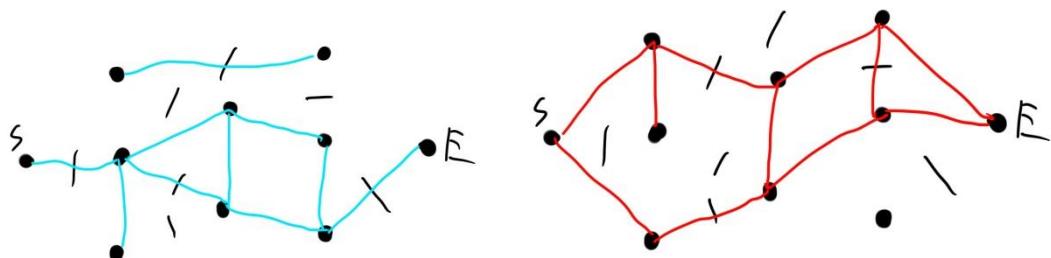
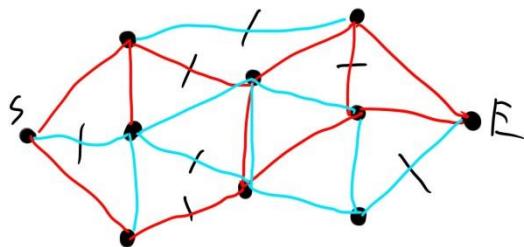
so you can only interact(walk push pull) with bridges that are in your world (we may call this two worlds to identify it from world 4)

### Stage 6.1

start in world **blue** with 9 bridge, 1 push, 4 change(to change world)

### Stage 6.2

start in world **red** with 12 bridge, 1 push, 4 change



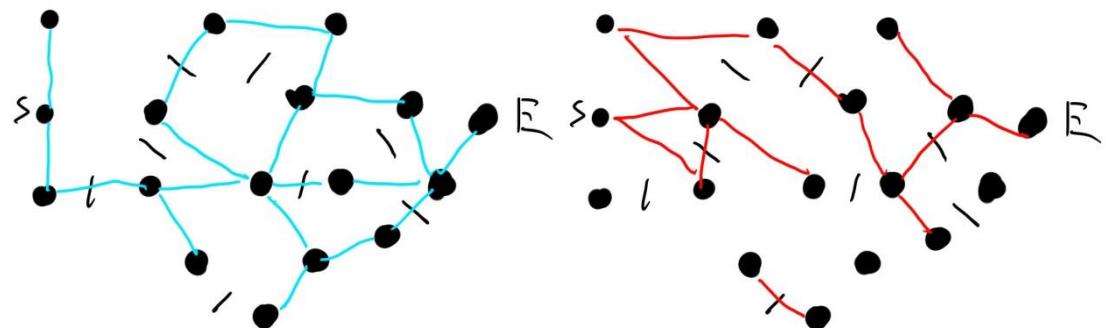
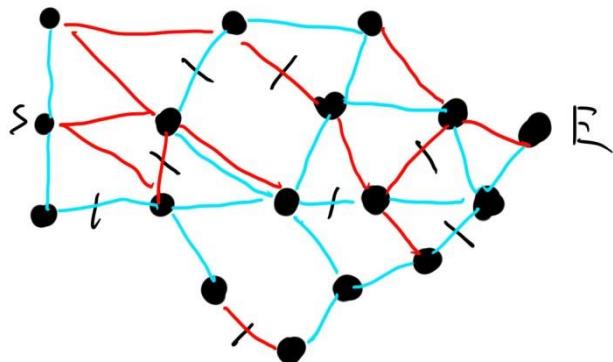
And this is the hard level, world 4 and 5 inherently have the idea of using same map for different solution.

### Stage 6.3

start in world **blue** with 16 bridge, 5 change

### Stage 6.4

start in world **red** with 24 bridge, 5 change

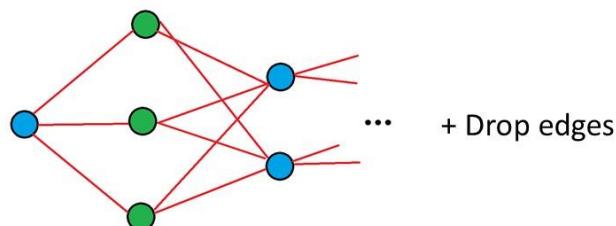


## Appendix D: Endless mode PowerPoint

PowerPoint slides:

### Meeting With Professor (Jan)

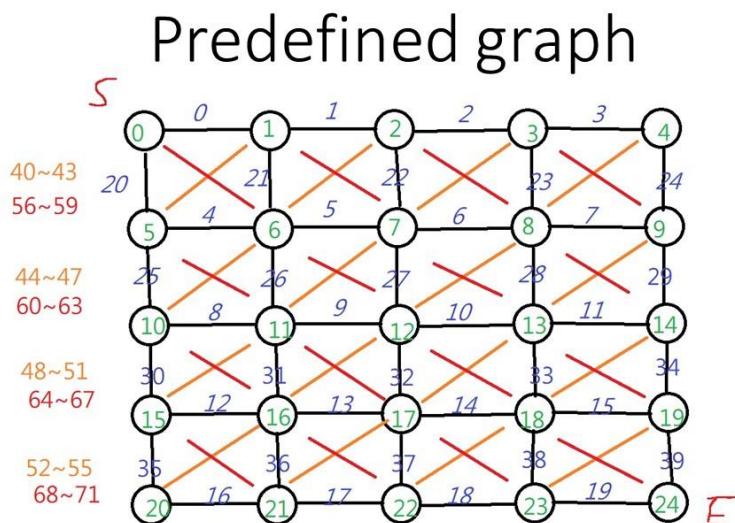
#### Some Previous Approaches



- Characteristic: Between each layers, it has distance of 1
- Benefit: Easy to calculate/control whether the map exist sol. or not
- Problem: It will be quadrilaterals everywhere
- Investigation: We should not make overlap edges. Also, triangles and squares everywhere (which is actually good!)

## Random Stage Generation Flow

- Generate from predefined graph
- Generate Cross(Vertical/horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End
- random star on path ( $p=0.3?$ )
- BFS best solution

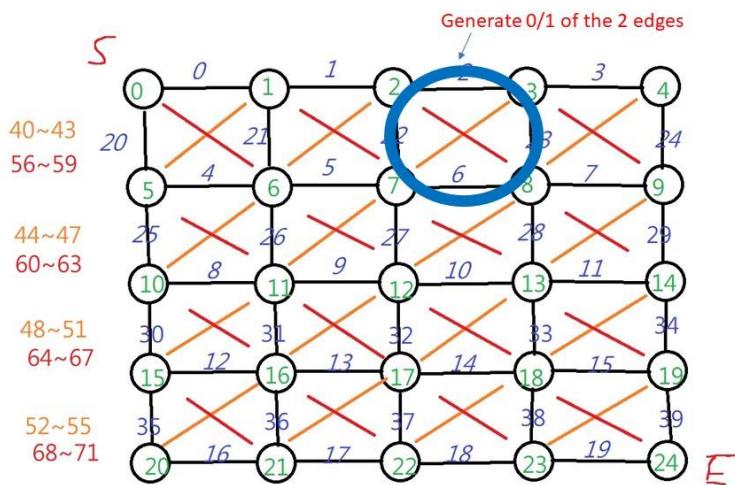


- Generate from predefined graph
- Generate Cross (Vertical/horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End
- random star on path ( $p=0.3?$ )
- BFS best solution

## TODO:

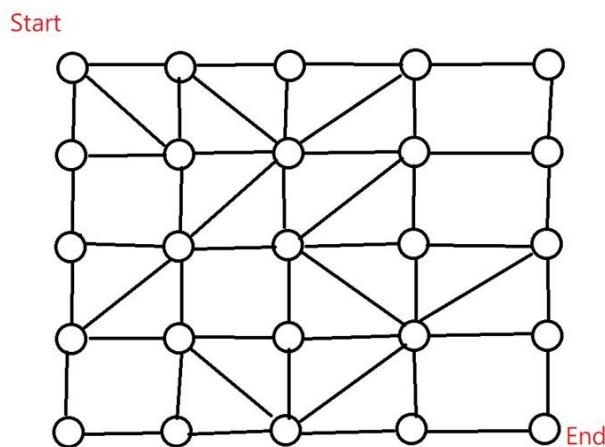
- After testing the code we will map it to a Unity Game Scene.
- Passing Arrays/Lists to control whether the game object exists or not.

## Generate Cross

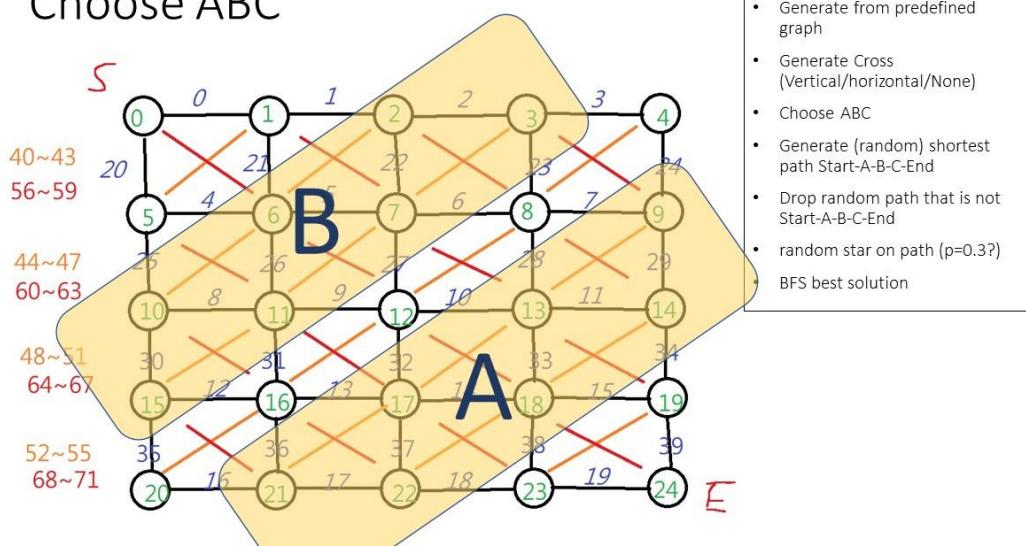


- Generate from predefined graph
- Generate Cross (Vertical/Horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End
- random star on path ( $p=0.3?$ )
- BFS best solution

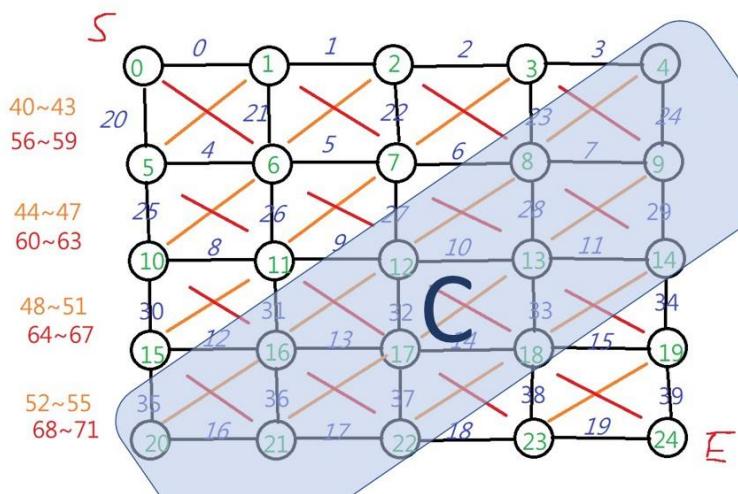
## Example



## Choose ABC

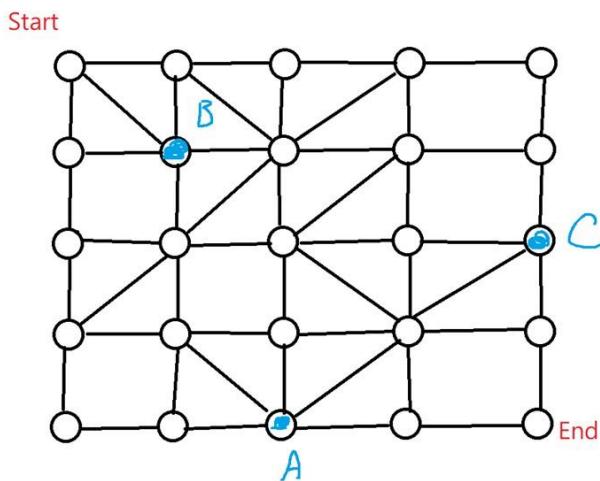


## Choose ABC

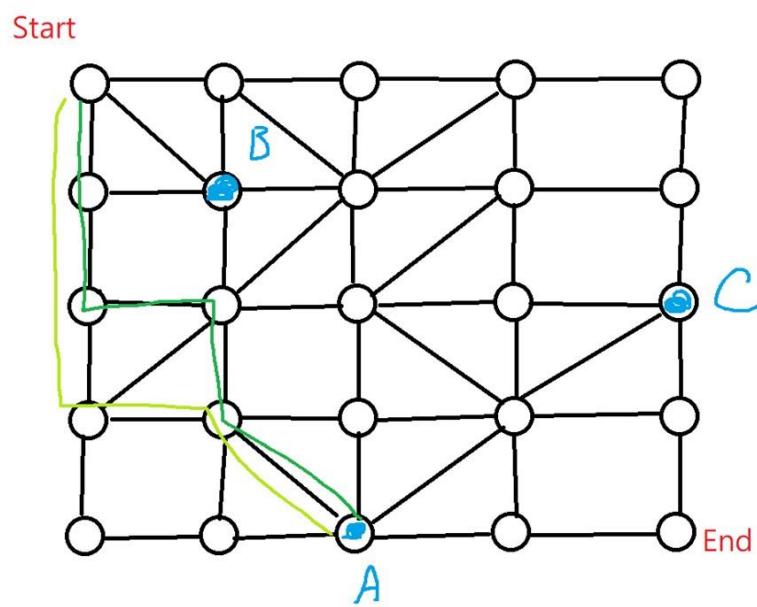
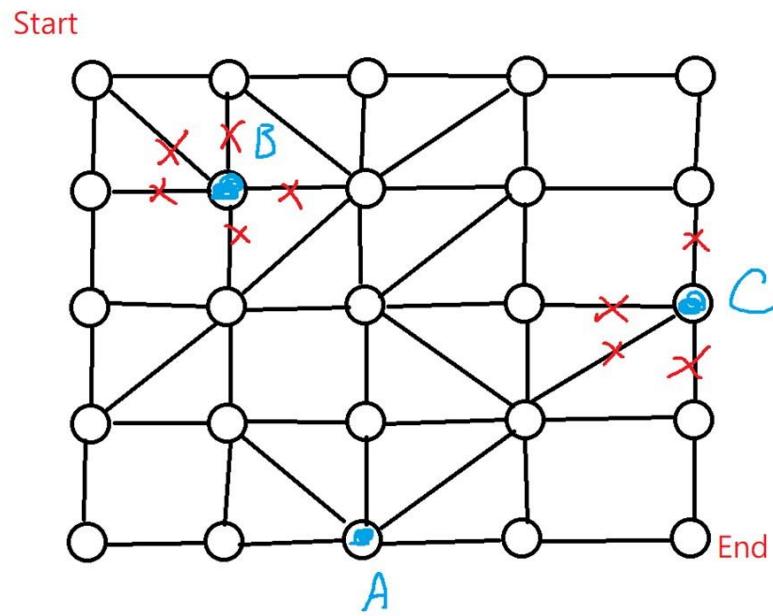


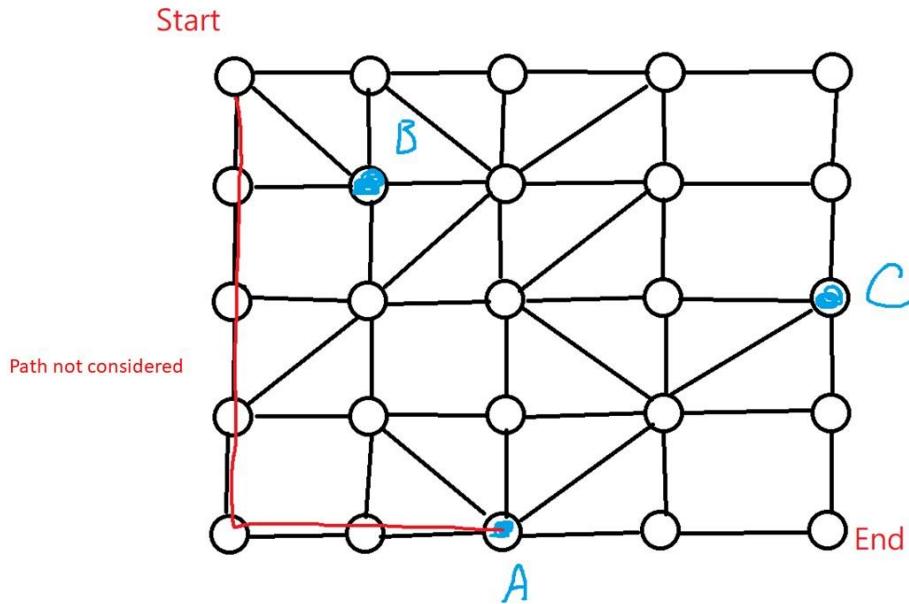
- Generate from predefined graph
- Generate Cross (Vertical/horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End
- random star on path ( $p=0.3?$ )
- BFS best solution

## Generate path

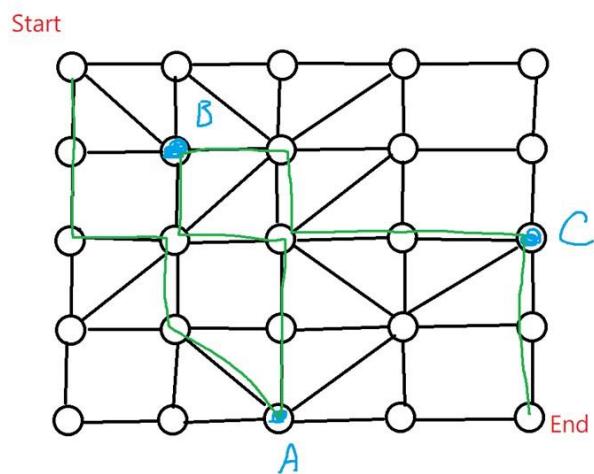


- Generate from predefined graph
- Generate Cross (Vertical/horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End
- random star on path ( $p=0.3?$ )
- BFS best solution





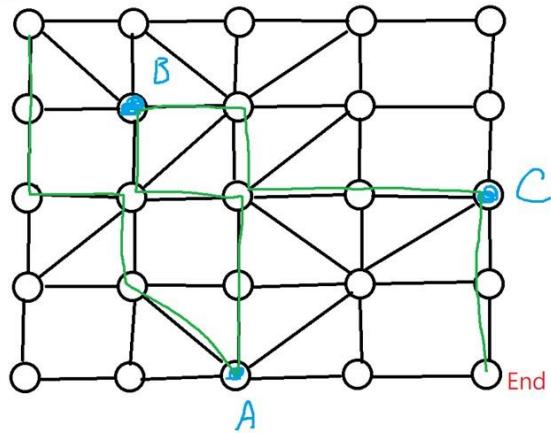
## Finish Generating a random path



- Generate from predefined graph
- Generate Cross (Vertical/horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End
- random star on path ( $p=0.3?$ )
- BFS best solution

## Drop random path (Not Green)

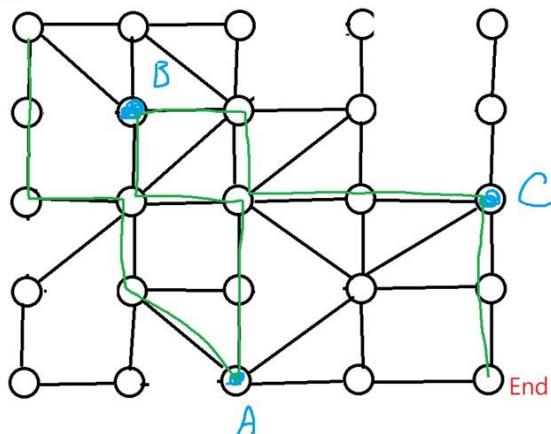
Start



- Generate from predefined graph
- Generate Cross (Vertical/horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End
- random star on path ( $p=0.3?$ )
- BFS best solution

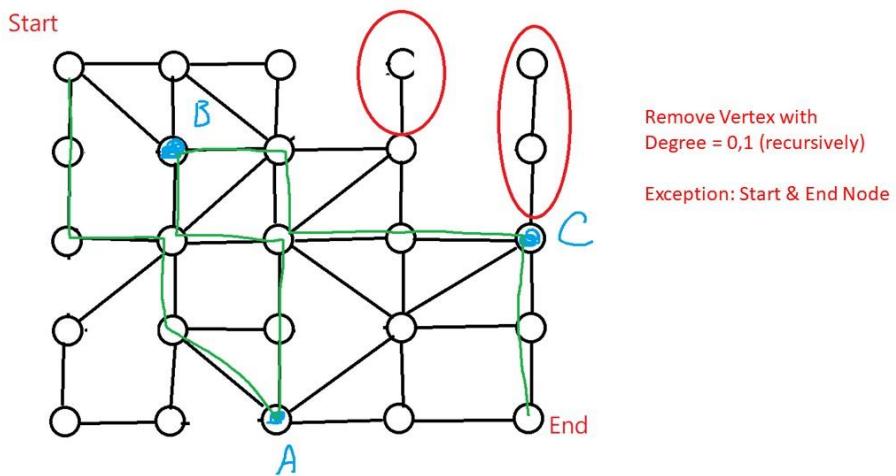
## Drop random path (Not Green)

Start

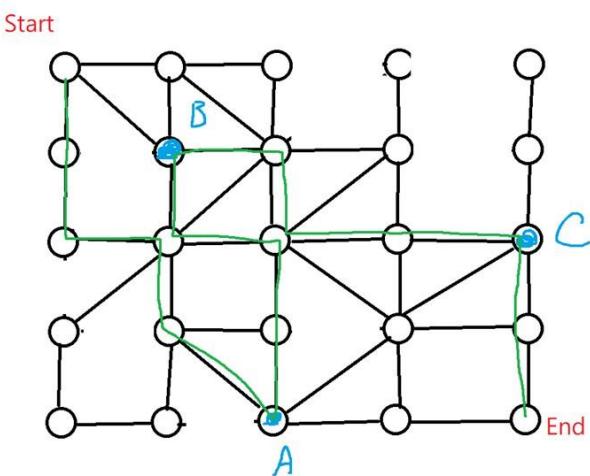


- Generate from predefined graph
- Generate Cross (Vertical/horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End
- random star on path ( $p=0.3?$ )
- BFS best solution

## Clean up



## Drop random path (Not Green)



- Generate from predefined graph
- Generate Cross (Vertical/horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End
- random star on path ( $p=0.3?$ )
- BFS best solution

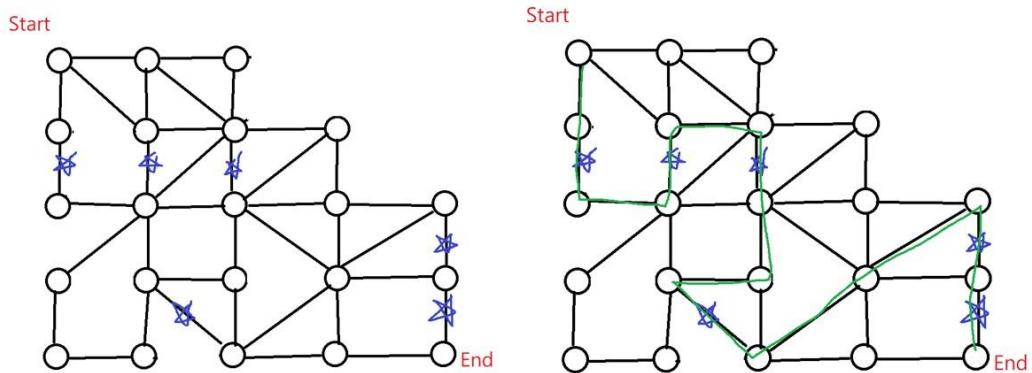
## Sprinkle stars over the green path

- Only protected (green) path will add stars
- To ensure that there exist solution
- (not guarantee best)
- Pseudorandom to prevent extreme cases
- (less to no stars // star at each edge)

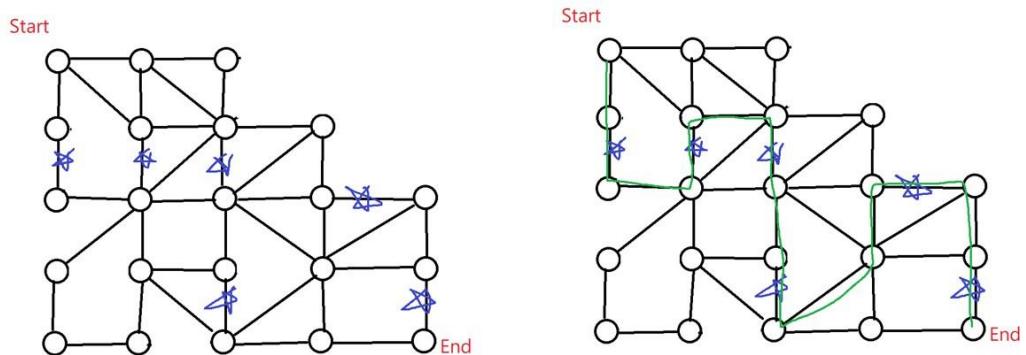
- Generate from predefined graph
- Generate Cross (Vertical/Horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End
- random star on path ( $p=0.3?$ )
- BFS best solution

and then.....BFS best solution

## Example 1

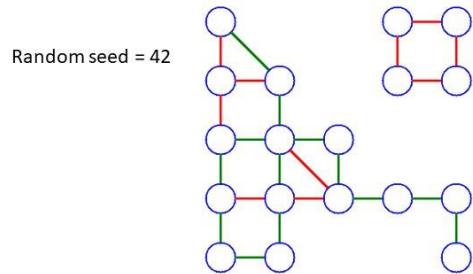


## Example 2

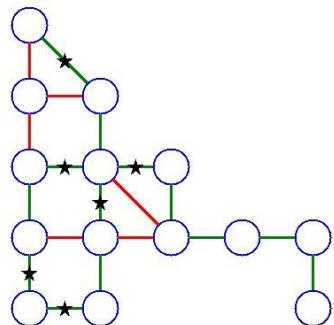


## BUG:

- Can be solved by finding its distance from start (which should be inf)
- Should be performed when cleanup



Just fixed this morning 😊



## Current Random Stage Generation Flow

- Generate from predefined graph
- Generate Cross(Vertical/horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End ( $p=0.45?$ )
- random star on path ( $p=0.3?$ )
- **Clean up**
- BFS best solution

## Some of the TODO:

### Game

- Improve minimap in 3D scene of current game
- Continue game stages implement according to document
- Start confirming arts

### Random stage

- Unity Stage Implementation
- Code optimization (if slow in lower spec computer)

## Some of the TODO:

- Regular meeting

## Meeting With Professor (Feb)

## (Jan) Random Stage Generation Flow

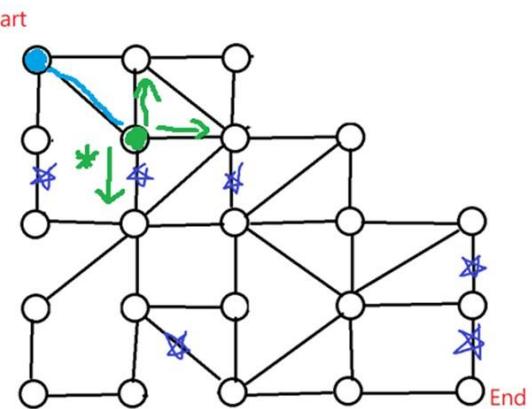
- Generate from predefined graph
- Generate Cross(Vertical/horizontal/None)
- Choose ABC
- Generate (random) shortest path Start-A-B-C-End
- Drop random path that is not Start-A-B-C-End ( $p=0.45?$ )
- random star on path ( $p=0.3?$ )
- Clean up
- **BFS best solution**

## BFS best solution

- Previously can only handle “Walk only”.
- (also known as finding shortest path from Start to End that reaches all required edges – edges with stars)
- Now can handle Push/Pull

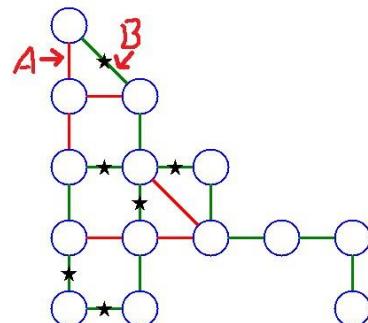
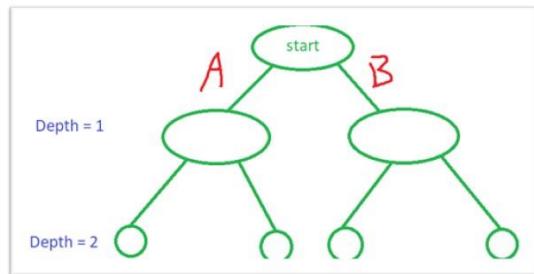
## Design 1 (old).

- Try push/pull in every step

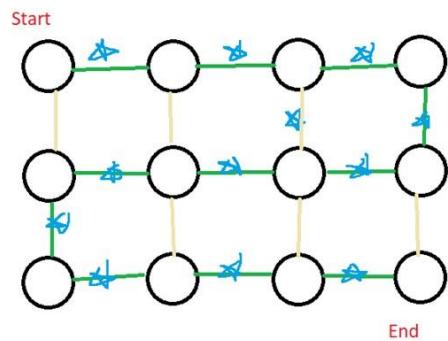


## Current design:

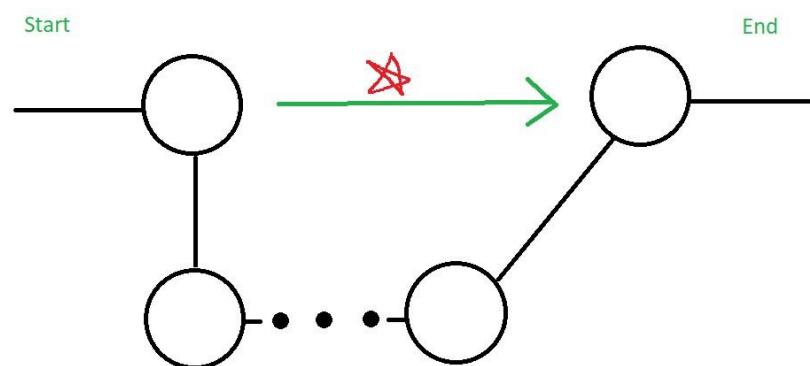
- Slight explain BFS

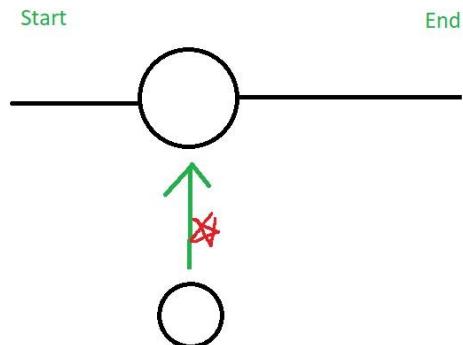


How and when Push/Pull?



How and when Push/Pull?





- If a puzzle can be solved with  $N$  move + 1 push, then it must can be solved with  $N$  move + 1 pull.
- $N$  push  $\rightarrow$   $(N-1)$  push + 1 pull  $\rightarrow$  ....

```

1 BFS:
2 <List> VerticesWalkOrder ... //can repeat
3 <List> EdgeWalkOrder ..... //cannot repeat
4 .
5 .
6 //try from depth 1,
7 .
8 .
9 .
10 //Solution finder part
11 If last of VerticesWalkOrder is END:
12 ... Check if EdgeWalkOrder consist of all the 'Starred Edges' ..... // Walk-only solution
13 ...
14 ... Check if EdgeWalkOrder consist of (all - 1) 'Starred Edges' && **OnePushPath** ... // 1 Push solution
15 ... Check if EdgeWalkOrder consist of (all - 1) 'Starred Edges' && **OnePullPath** ... // 1 Pull solution
16 ...
17 ... Check if EdgeWalkOrder consist of (all - 2) 'Starred Edges' && 2x **OnePushPath** ... // 2 Push solution
18 ... Check if EdgeWalkOrder consist of (all - 2) 'Starred Edges'
19 ... && **OnePullPath** && **OnePushPath** (which is not the same) ..... // 1 Pull + 1 push solution
20 ... Check if EdgeWalkOrder consist of (all - 2) 'Starred Edges' && 2x **OnePullPath** ... // 2 Pull solution

```

- For the 1 Push solution, we can check if we walked pass the two Vertices of the missing starred Edge. If that is true then we can push it when we reached the first one and push it to the opposite vertex. Then we can collect it afterwards.

```

Check if EdgeWalkOrder consist of (all - 1) 'Starred Edges' && **OnePushPath** ... // 1 Push solution
Check if EdgeWalkOrder consist of (all - 1) 'Starred Edges' && **OnePullPath** ... // 1 Pull solution

Check if EdgeWalkOrder consist of (all - 2) 'Starred Edges' && 2x **OnePushPath** ... // 2 Push solution
Check if EdgeWalkOrder consist of (all - 2) 'Starred Edges'
&& **OnePullPath** && **OnePushPath** (which is not the same) ..... // 1 Pull + 1 push solution
Check if EdgeWalkOrder consist of (all - 2) 'Starred Edges' && 2x **OnePullPath** ... // 2 Pull solution

```

- For the 1 Pull solution, it has a less demanding requirement.
- We only have to check if we had walked pass one of the 2 Vertices connected to that Edge.

```
Check if EdgeWalkOrder consist of (all - 1) 'Starred Edges' && **OnePushPath** // 1 Push solution
Check if EdgeWalkOrder consist of (all - 1) 'Starred Edges' && **OnePullPath** // 1 Pull solution

Check if EdgeWalkOrder consist of (all - 2) 'Starred Edges' && 2x **OnePushPath** // 2 Push solution
Check if EdgeWalkOrder consist of (all - 2) 'Starred Edges'
&& **OnePullPath** && **OnePushPath** (which is not the same) // 1 Pull + 1 push solution
Check if EdgeWalkOrder consist of (all - 2) 'Starred Edges' && 2x **OnePullPath** // 2 Pull solution
```

## Selecting the “good” solution

- After the BFS, we may then have a lot of solutions (0push+0pull | 1push+0pull | 0push+1pull ...)
- Right now, we allow at most 3 for total push + pull
- And we try to come up “a cost ” for both move so we can determine how effective the move is and pick the most efficient solution

## TODO (in the infinite mode):

- Finetune the parameters of generating maps.
- it is rare to have map that use more than one moves
- Possible fix
  - A: Slightly change the generation procedure to accommodate push/pull abilities if needed.
  - B: Decrease the cost with each extra move
  - C: Reconstruct the way we put objects on
  - D: Try to add new object on the solution and check if it works