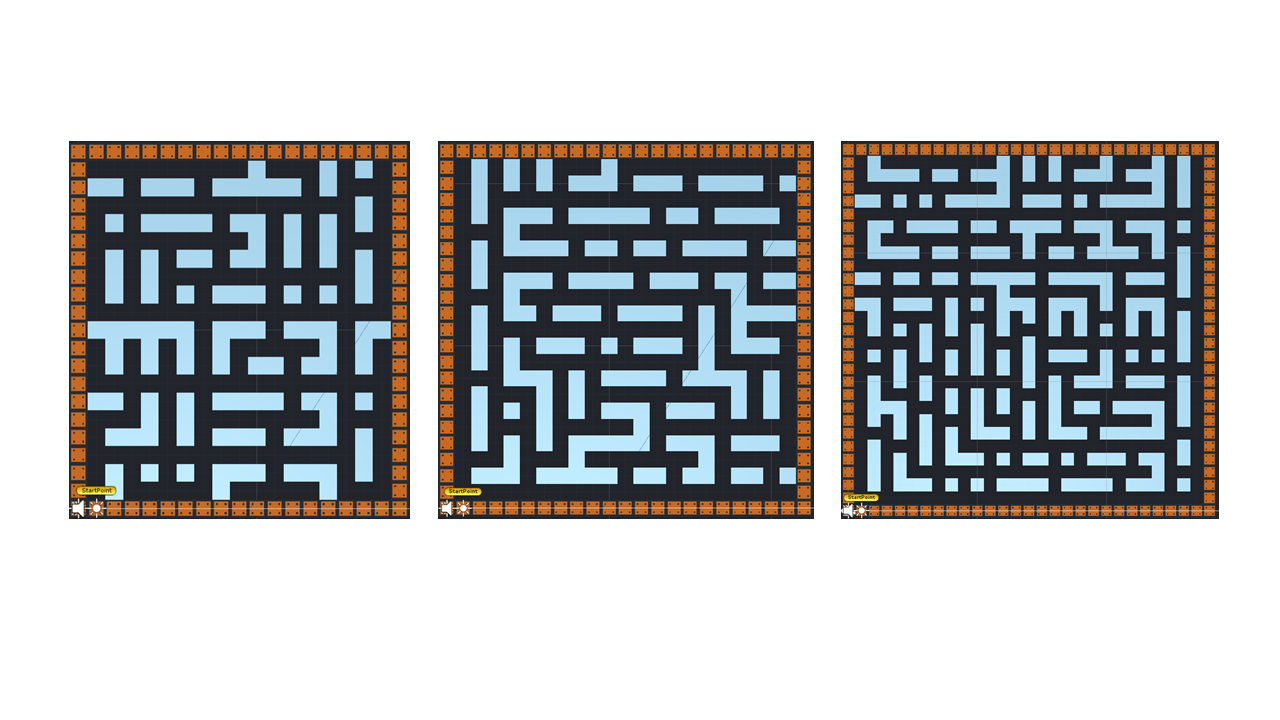
* 1. **Realisation**

In the implement stage, most game objects meet the requirements of design and all expect functions are realised in the final product. Some modifications have also been made to the ‘Item’ game object to make it fit requirements of this game better. Some extra components including animation system and particle system are also added to provide better gameplay experience. This section will discuss the implement for each component in detail.

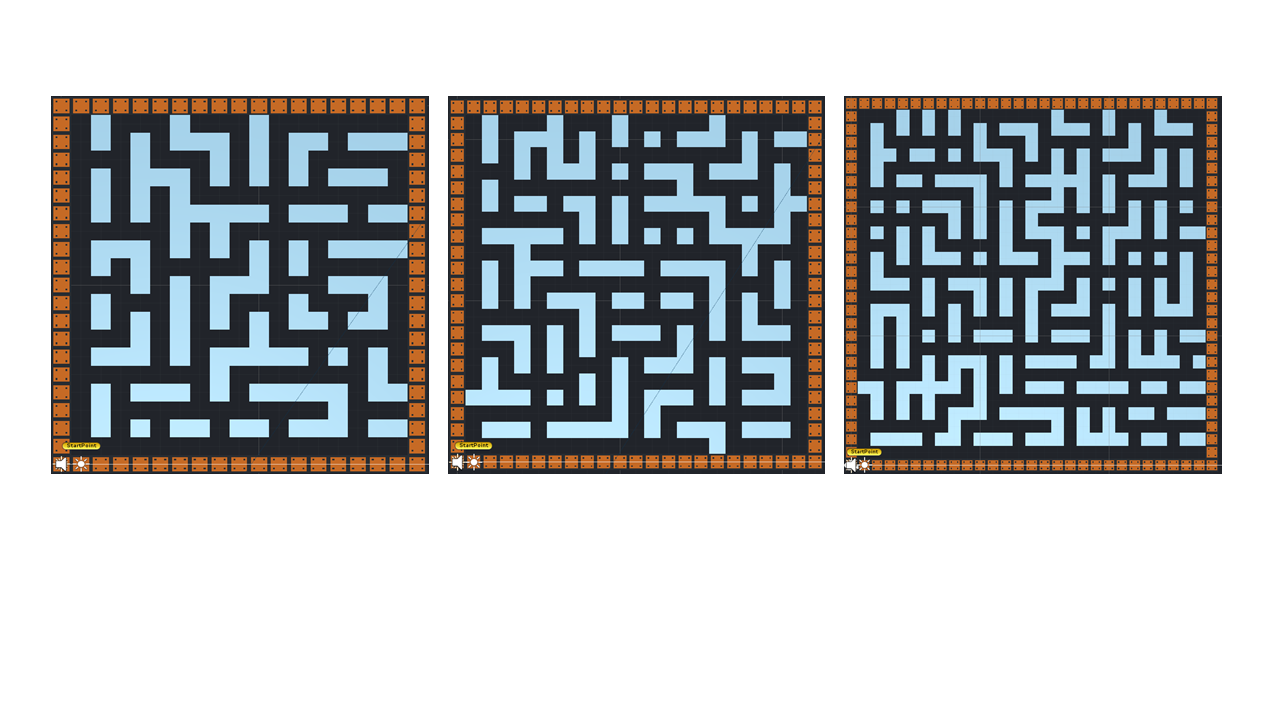
* **Maze generation**

Since the maze generation system is the basic and key component for the game, it should be completed first. All three algorithms mentioned in the ‘Design’ section are implemented in the Unity environment. An abstract super class “MazeGenerator” is created as the parent of all maze generators. The MazeGenerator class also contains necessary API for the generation of mazes. Then, three C# scripts are created to implement the logic of each maze generation algorithm. Finally, an empty GameObject is created in the game scene and three MazeGenerator scripts are attached to that object. When *MazeGenerator.GenerateMaze(width, height)* function is called, mazes with different types and sizes will be generated. Figure 5.7.1, 5.7.2 and 5.7.3 shows mazes generated by different algorithms.



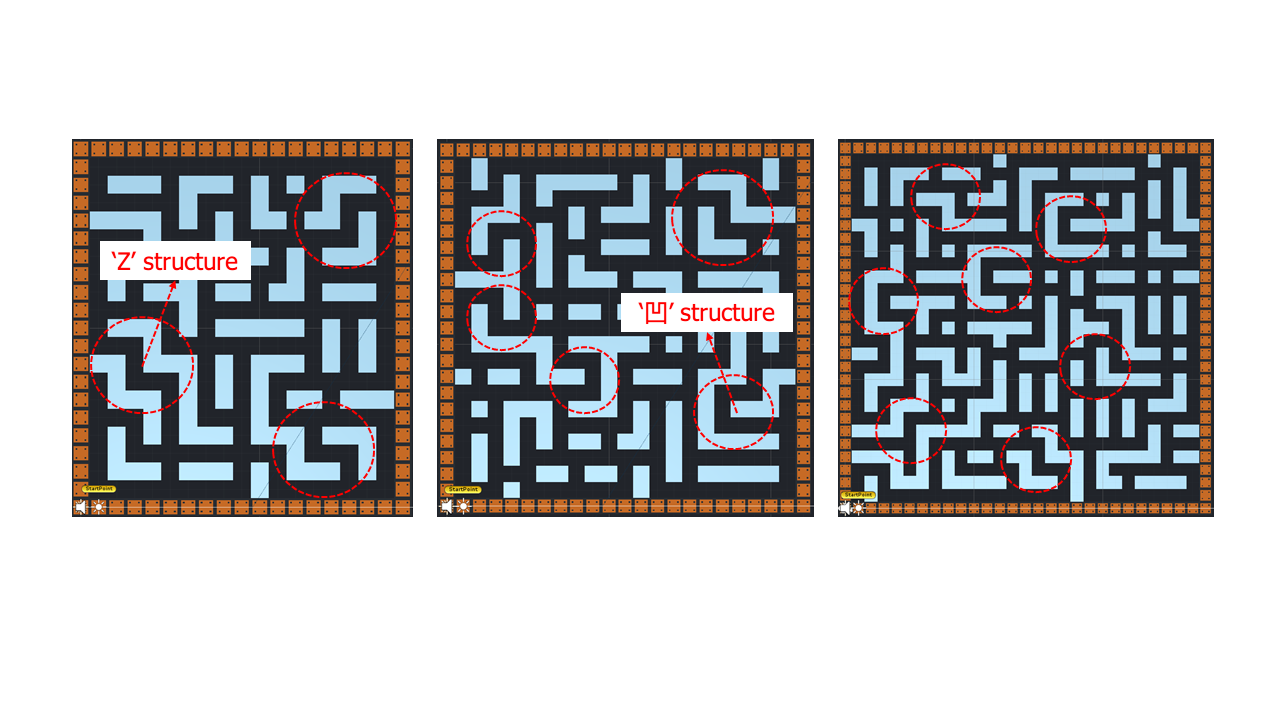
*Figure 5.7.1 Mazes generated by recursive division*

Mazes in Figure 5.7.1 are generated by recursive division. Mazes generated by this kind of algorithm usually consist of many short vertical and horizontal walls. There are also many long, straight roads, which makes this maze easy to solve. Therefore, this kind of maze generator are used to create the simplest maze in the game.



*Figure 5.7.2 Mazes generated by randomized Prim’s algorithm*

These mazes shown in Figure 5.7.2 make use of the randomized Prim’s algorithm. Compared with recursive division, this algorithm can generate mazes which have more corners. Players need to adjust PacMan’s direction more frequently and try to find different way to solve the maze. This maze generator is used to provide mazes with medium difficulty.



*Figure 5.7.3 Mazes generated by recursive backtracker algorithm*

Mazes generated by recursive backtracker algorithm contain many ‘Z’ and ‘凹’ structure (Figure 5.7.3). Both ‘Z’ and ‘凹’ structure have a feature that if a PacMan enter this structure, there will be only one exit. In this game, ghosts will be placed in the maze later to hunt PacMan. Therefore, such kind of structure can be dangerous because PacMan will have no other way to run. With this feature, recursive backtracker algorithm is used to generate those most difficult mazes.

* **Player**
* **Items**
* **Ghosts**
* **User interface**
* **Game loop**