# A Maze Kit v1.1.2

Equilibre Games

## 1.Introduction

A Maze Kit is a Level Design Unity3D package that allows you to draw and generate mazes and other paths from pre-made rooms. It includes prefabs so you can start right away to draw mazes and use them in your games, but it's up to you to improve or replace them so you've got a personal touch on the final result.
Both drawing and generation are available in 3D, but you can constraint on a 2D Plane if you want to. The only limit is your imagination (and the memory of your hardware) !

This document contains the change log and some help to :
- install the package
- update the package : read carefully if you update from 1.0 !
- create a randomly generated maze
- draw a maze manually
- do a quick TPS test session with your newly created maze
- quick tour of the 3 themes included

## 2.Change Log

### 1.1.2 MOD is for modification, NEW for a new feature, FIX for a bug fix

- NEW : Theme name can be modified so the maze uses different prefab easily
- NEW : 2 added themes so you have a game right away : Cube Shooter and Sneaker
- MOD : The button "Reset to default prefabs" is now "Set prefabs from theme"
- NEW : Inspector now highlight the "Set prefabs from theme" when prefabs are not from the entered theme directory
- NEW : Algorithms settings are now stored in a local asset file and are the algorithm entry point for the editor
- NEW : Click on the 'prefs' button on the right of the algorithm to select the preferences file in the project folder
- NEW : New Openness parameter for Depth First Search algorithm, allowing for multiple paths
- NEW : Room prefabs are now set in the MazeData to be available from Runtime
- NEW : New class MazeRoomUtility for : GetMaxBounds of RoomPrefabs
- NEW : New class GameObjectUtility for : RemoveGameObjectChildren
- NEW : DFS algorithm takes now a new parameter : Openness that allows to have more paths through the maze
- NEW : Frame the newly created maze (use an undocumented API :( ))
- NEW : OnMazeDone event sent when all the maze is built, with the Maze GameObject as parameter
- MOD : OnMazeUpdate event is sent after the room object has been assigned so it can be used
- MOD : More classes are made public and available for Runtime scripts to enable creating new mazes from them: MazeRoomPrefabs, Algorithms (DFS, IMazeGenerator), MazeDataCreate

- MOD : IMazeGenerator interface is now generic and can be used for future new algorithms
- MOD : Generation split in 3 methods to allow runtime generation : Initialize, Step, End
- MOD : Unselect the maze game object when you click outside the green box
- FIX : All Classes are now in EquilibreGames.AMazeKit namespace (some of them had no namespace)
- FIX : The prefab section does not take too much space anymore (was using a scroll view to display only one item)

## 3.How to install A Maze Kit

If you read this, it means that you already have downloaded the package from one of Unity store : Unity Asset Store or Unity Magic. You've got now at least two ways to install it :

- install it from the Asset Store window
- install it manually from your local hard drive if you bought it elsewhere

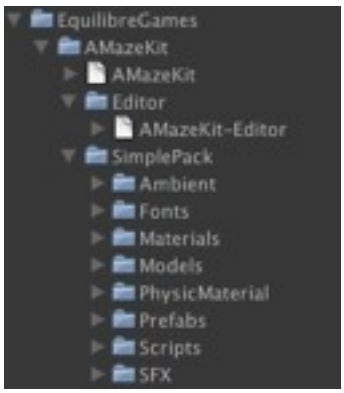
### Install from the Asset store


Asset Store windows

Go to the bottom right of the window, click on «My stuff» and if the kit is not on the list that appears, click on «See complete list».
Search for «A Maze Kit» in the list, then click on the «Import» button on the right.

The button can be named «Download» if you reinstalled your system or cleaned the packages on your local hard drive.

Once the package is imported, you should have a folder named «EquilibreGames» in your project view, and a folder «AMazeKit» in it (and no error in the console view that relates to this particular package).




You're ready to go !

## Install manually

Locate the folder where the unity package is.

### Locate Asset Store folder

If you bought it from the unity asset store, the package is on a shared folder :

Windows : `C:\Users\<account name>\AppData\Roaming\Unity\Asset Store`
Mac : `~/Library/Unity/Asset Store`

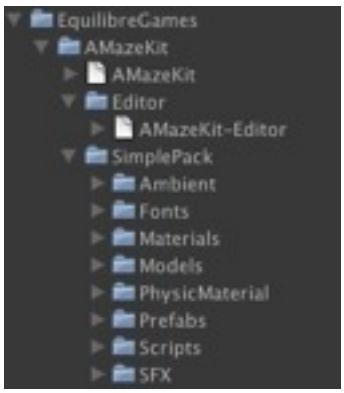
Those folders are hidden by default, so you could have to search how to display them on the Internet !

Navigate to this local folder then you should have a folder named «Equilibre Games». Open this folder, then search for the package aMazeKit : it should be in a folder name the «Editor ExtensionsDesign».

### Import the package

Double click on it to import it automatically in Unity (you can have Unity opened or not). If you get an error at this point, you surely have multiple install of Unity. Open Unity, do a right-click in the "Project" view and select "Import package / Custom package..." and navigate to the folder where the package is.

Once the package is imported, you should have a folder named «EquilibreGames» in your project view, and a folder «AMazeKit» in it (and no error in the console view that relates to this particular package).




You're ready to go !

## 4.Update the package

Some classes have moved from the 1.0.0 release, that can break some links in your maze data :

- for every MazeDescription asset, you have to select the new MazeDescription script from aMazeKit folder
- the settings for Depth First Search Algorithm are now stored in an asset in the project Assets/Preferences folder. Your previous global settings are lost, sorry.

To update the package, open a blank new scene, delete the AMazeKit folder entirely, then follow the normal installation process.

## 5.How to create a randomly generated maze

### Create the maze object


A Maze

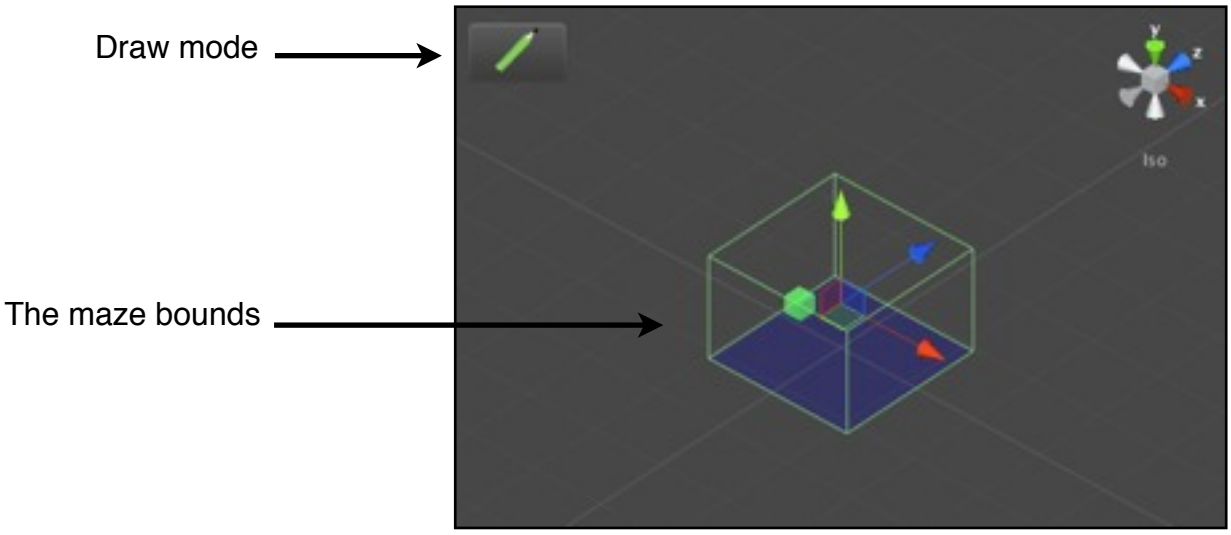
Now that you have a fully functional package, let's create a random maze so you see its simple power in action !

Click on **Create** then **A Maze** in the Hierarchy view.

The newly created maze GameObject named **aMaze** is automatically selected and you can see a green cube on the scene view.

The description of this maze is stored in the folder **Mazes** in the Assets folder.

***Tips** : Focus the whole object with the «F» shortcut if you don't see the same as below.*




In the Scene View, you can see the bounds of the maze. This is where the maze will be and where you can interact with it. They are updated when room are added or removed. They do not shrink though unless you remove floors (force the bounds to be entirely recalculated).

## Configure the algorithm

The Depth First Search is the first algorithm to be included in the package. It has three parameters :

- size : choose how many rooms there will be for each axis X, Y and Z
- next choice : decide how will be chosen the next available room to be search during the algorithm.
- openness : open more ways for a percentage of rooms (new in 1.1)

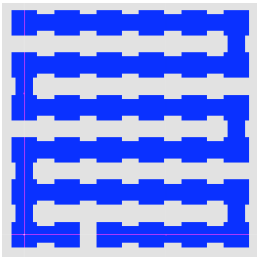
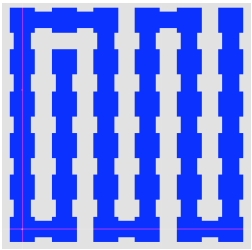
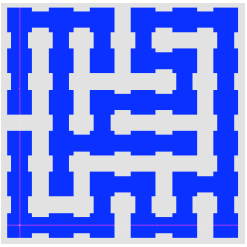
### Size

If you want to have a 2D maze, just put 1 on the Y axis, there will be only one floor, that's the simplest maze to solve.
If, on the contrary, you want a 3D maze, put anything you want ! Remember though it uses memory, and the generation uses **every** coordinate. That means a 10x20x5 maze will have 1000 rooms ! That's a lot to navigate !

### Next choice

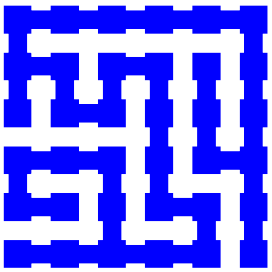
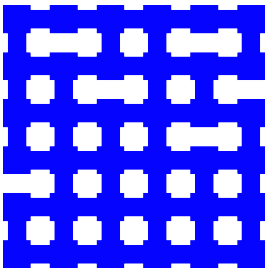
This choice is used when the algorithm searched for a room where to dig a hole : it always pick the choice you decide : the first, the last one, a random one, or the one in the middle of all choices it got.

Here's a sample on a 2D maze so you can see the effects.

| First | Middle | Last | Random |
|---|---|---|---|
|  |  |  |  |
| Mostly horizontal maze | Mostly diagonal maze | Mostly vertical maze | Complete random maze |

## Openness

This percentage of rooms tells the algorithms to create other issues to created rooms. The greater it is, the more the maze will be an «opened» maze : multiple paths will be created.

| 0 % | 100 % |
|---|---|
|  |  |
| Classic maze | At least two opened doors for every room |

## Launch the algorithm

Click on the **Clear & Generate** button to generate your new maze.


Depending on the size and your hardware, it can be instantaneous of take some time ! There's some progress bar to help you figure out what it does :

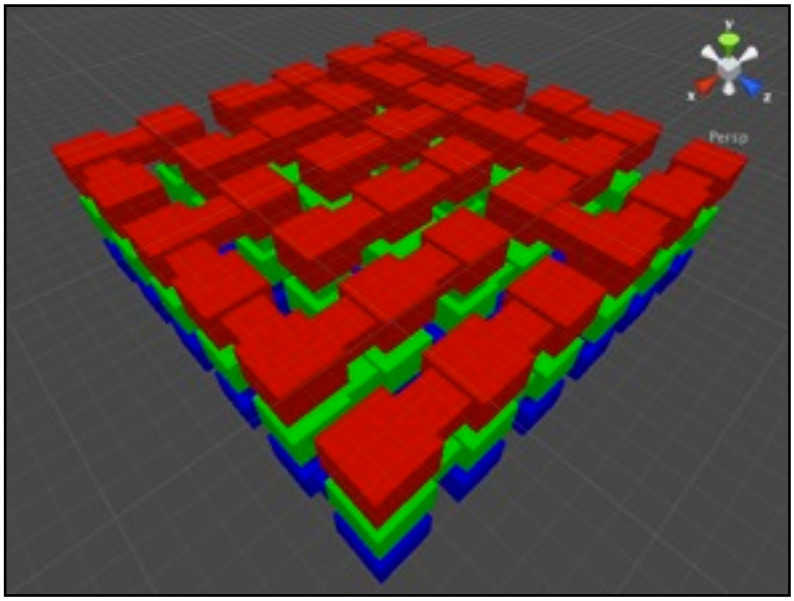
- remove the old objects (if needed)
- generate the maze from the algorithm

Don't hesitate to drop me a comment or a suggestion : FredericRP@equilibregames.com

- create the visual cubes for visual feed back

***Remark*** *:With a large maze, Unity may take a few seconds after the script has ended to display the cubes and let you handle it again.*
***Hint*** *: The little «prefs» buttons on the first line select the settings asset for the algorithm in the project view.*

Here's a sample generated maze with a size of (**6, 3, 6**) and **random** chosen as next choice, with an openness of 0%.

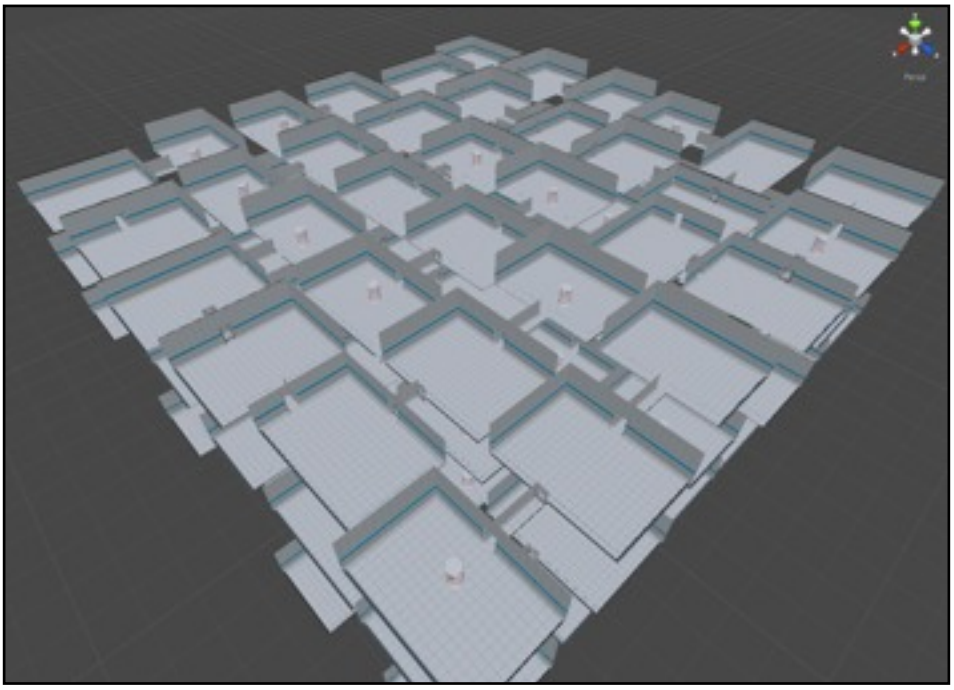


You can now edit the maze, generate a new one of build the maze with Prefabs.

# Instantiate Prefabs

Click on **Instantiate Game Objects**

The maze is now fully generated and available to play !
Here's a sample of a generated maze with the default "SimplePack" theme.



# 6.How to draw manually a maze

We'll begin the same way as the previous chapter to create a maze.

## Create the maze object


Now that you have a fully functional package, let's create a random maze so you see its simple power in action !

Click on **Create** then **A Maze** in the Hierarchy view.

The newly created maze GameObject named **aMaze** is automatically selected and you can see a green cube on the scene view.

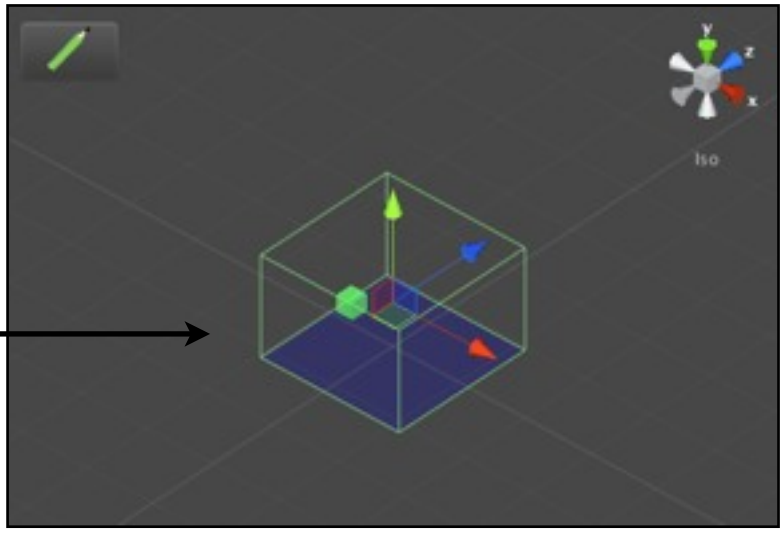
The description of this maze is stored in the folder **Mazes** in the Assets folder.

***Tips** : Focus the whole object with the «F» shortcut if you don't see the same as below.*

A Maze

Draw mode

The maze bounds



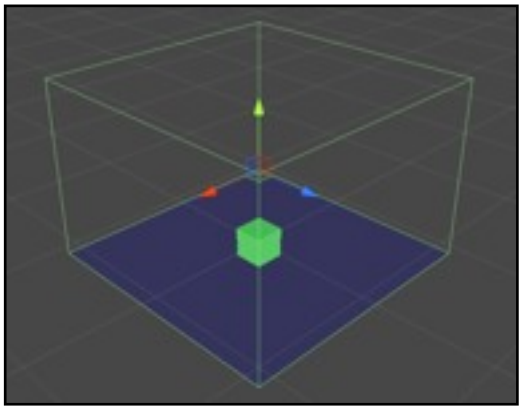

In the Scene View, you can see the minimum bounds of the maze. This is where the maze will be and where you can interact with it.

## Know how to use the 3D view

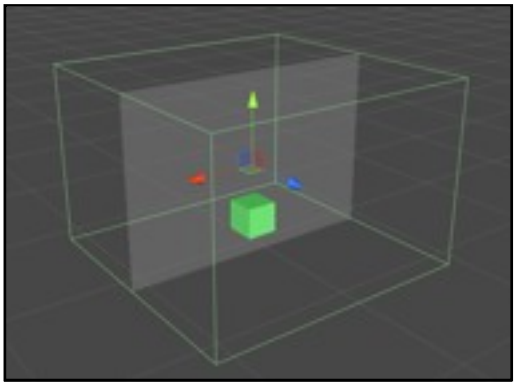
Before drawing stuff, you have to know how the tool uses the 3D view.

From your point of view, it will decide wether you want to draw on the XZ plane or the XY plane. You'll see that from a 2D colored plane inside the green wireframe box.

With the default point of view, you'll see a blue plane, on the bottom side of the green wireframe box. This is the ground floor, and the plane is the XZ plane representing the surface where you can draw.



If you watch from a little below (Alt+Left mouse button down then go up, release the button), you'll see the blue horizontal plane changing to a grey vertical plane. You're now on the vertical mode.



Depending of where you are from the maze, it will choose between those two draw planes.

***Hints** : you can draw in 3D switching between vertical and horizontal mode by moving your point of view of a few degrees. In a next release, the YZ plane will also be available to draw on.*

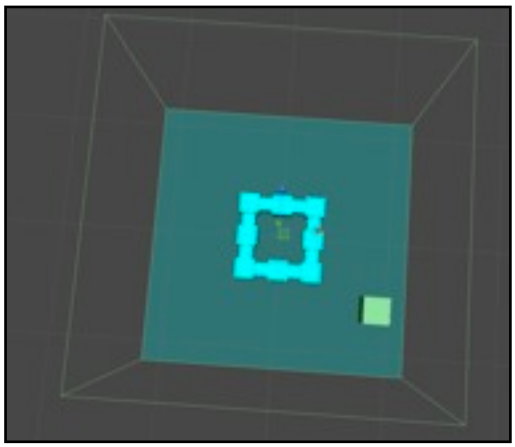
***Hints** : you can unselect the maze object by clicking outside left the colored plane.*

## Edit the maze

The green wireframe box is the 3D space where you can edit the maze : it will grow when you draw on the maze and shrink on special actions (remove the last floor).

Let's say you want to create a simple square path to start with. Click with the left mouse button and maintain the button down when drawing on the plane.

I like it when I watch the maze from a top view, so the screenshot will be taken from that point of view.



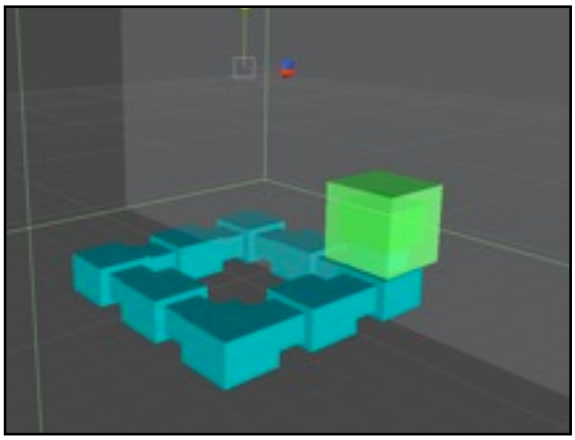
Then we could go up and draw some more rooms. But before that, maybe you've done some mistakes or want to change something in your maze ? You can remove rooms by two means :

- change the current mode to : eraser by clicking on the **pen** button on the scene view (switch between draw and erase modes)
- Ctrl/Command-Click to remove rooms : the Ctrl/Command modifier works for single click, drags and lines (see below).

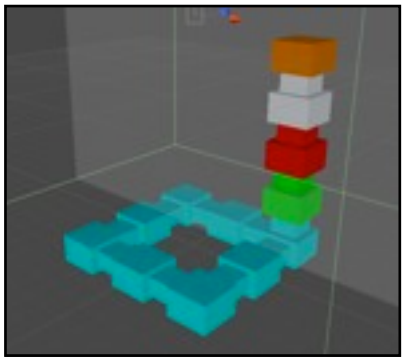
To add a floor, you can :

- rotate the view to switch to vertical mode then draw up : floors are automatically added
- click on the **add floor** button on the floors section on the inspector GUI

Let's try going up with the vertical mode, then draw some rooms back in the horizontal mode :



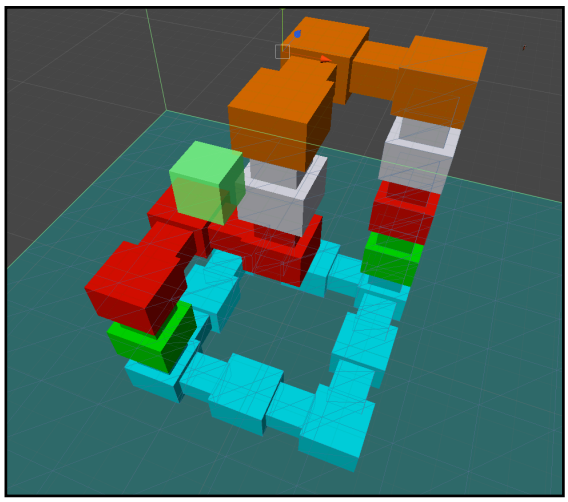

I got to the fifth level ! That's ok, but I just want to go back to the first room now !



Then you can draw a line between those two rooms : just hold Shift while you click on the next room :

- Select the ground floor on the inspector GUI
- move your point of view so you're in the horizontal mode
- Shift+Left click on a room

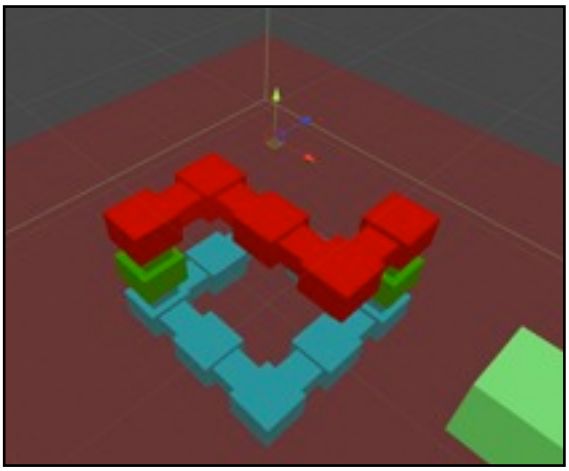
The tool allows you to draw lines from the last room clicked to the next click : remember that because it could help you a lot !
Here's the created line from the top right orange room to the bottom left cyan room. It's of course generated with a room path, not a straight pixelated line !



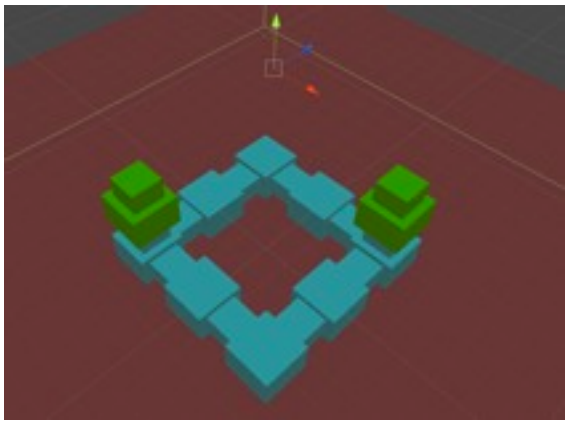
Let's try to reduce it a little : remove the last two floors, and connect the red rooms together !

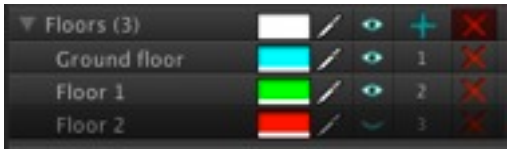
By clicking on the red cross on the right of every floor you can achieve to :

- remove the floor if it's the last one
- clear all rooms from the floor otherwise



The existing links between the rooms are updated so there's no way entering the void. You can check up links by switching the visibility of a floor : click on the eye on the floor line.






You are now ready to use A Maze Kit, congratulations !

If you want to test your new maze, go read the **Try it with a little FPS session** of the previous chapter !
For a more evolved usage of A Maze Kit, open the game scene in the CubeShooterDemo folder, and start from it !

---

## 7.Try it with a little FPS session

Let's try this new maze with a little FPS session !

So you can work with the maze, we'll use demo scripts included in the package, and do another few things :

- import and add the Unity standard first person controller
- tag the player prefab as «Player» (default tag)
- add the «HandleTeleportation» script to the player prefab
- change the scene default light

### First Person Controller

Import a standard package : click right on the Project view, select **Import Package**, then **Character Controller**.

Once it's imported, find the **First Person Controller** prefab, then drag into a room on the Scene View.
Remove the Main Camera from the scene as the First Person Controller prefab already includes one.

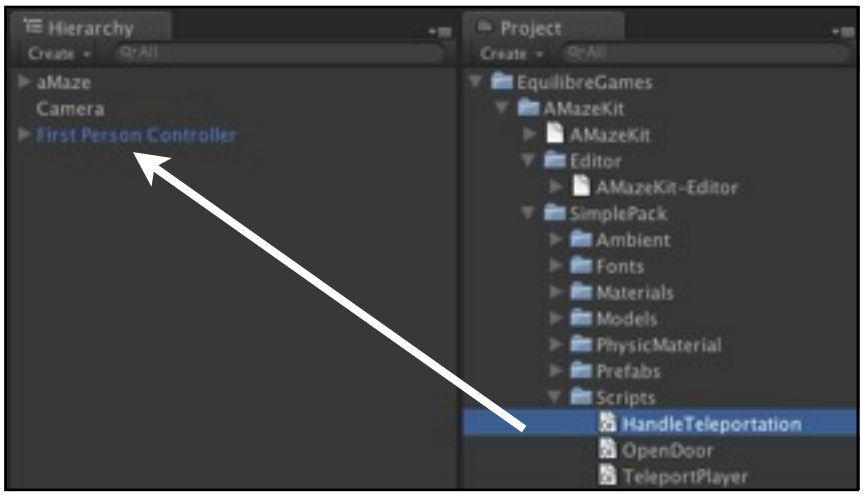
***Hints** : dragging the prefab into a room allows Unity to place it on this room ground automatically. If you drag the prefab in the scene view but outside any other object, it will only be placed in a arbitrary 3D position and you'll have to manually put it in a right location.*

## Tag the player prefab

The First Person Controller is not tagged as Player by default. To do so, select it in the hierarchy, then select **Player** from the tag drop down menu.

## Handle teleportation

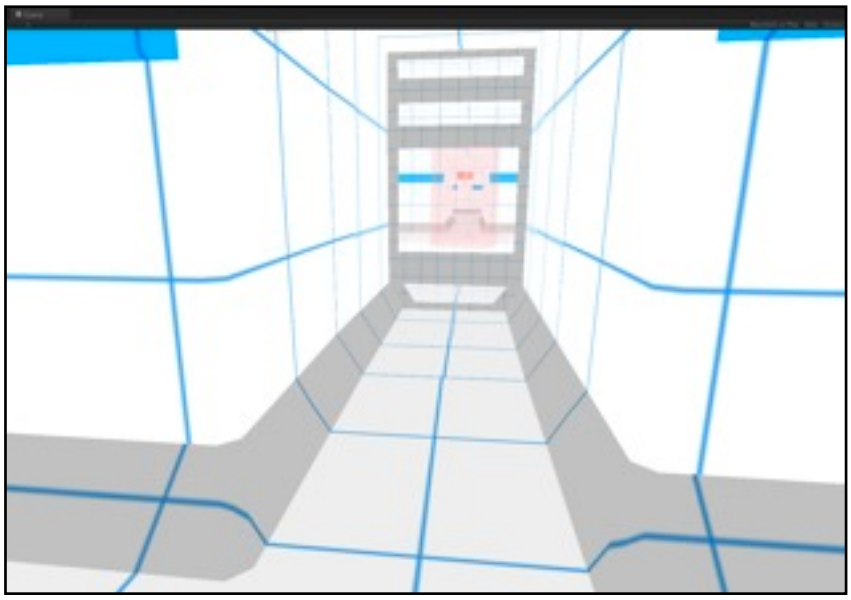
That's way easier in Unity that in real life to handle teleportation : drag the **HandleTeleportation** script from A Maze Kit **SimplePack/Scripts** folder on the player.




## Change the scene default light

To have a more coherent ambient light, update the ambient light to full white :
Select **Edit / Render Settings** then Ambient Light and change it to a full white.

Click **Play** !



You're ready to add anything you want in your mazes !

### Third Person Controller

You can also test it with the 3rd Person Controller, it's a little different from the FPS :

- you have to maintain the Main Camera in the scene (if you tried the FPS but want to try now, just add a Camera and tag it **MainCamera**)
- the default controller is a little too small for this demo, scale the player to **1,5** in each axis so it feels nicer.
- In order to handle teleportations well, we suggest that you change the default **Height Smooth Lag** in the **Third Person Camera** to **0**, otherwise it can cause hick ups when teleporting
- Don't forget to add the HandleTeleportation script on the player before playing !

## 8.Quick Tour of the themes

For a more evolved usage of A Maze Kit, themes have been added so you can see most features of the package:

- CubeShooter : a demo of a TPS where you stand as a cube and you have to shoot other cubes. You can reach the level 5 of experience with more bullets and faster bullets, a shield and a grater life bar, but beware: your enemies evolve too !
- Sneaker : a side scrolling sneaker where you have to sneak into buildings. Find the green desk by hiding in closets and desks and avoiding guards and their lights.

***Hints** : to select a theme, you have to write down its name for now. Enter the name in the "Theme" field of the "Room Prefabs" part of the maze inspector. Then click on the green button "Set prefabs for theme" and build the maze.*

### Cube Shooter

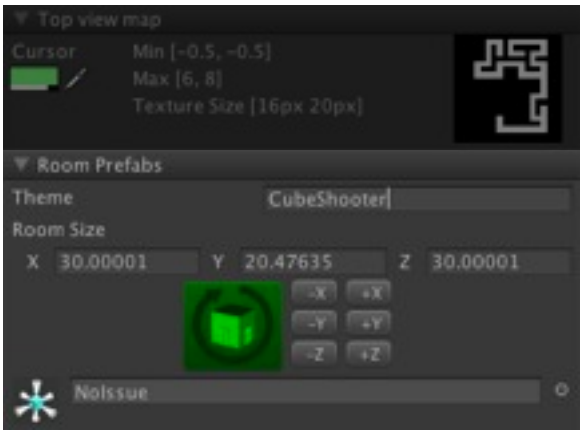
A sample of a game produced with the CubeShooter theme is available from its root. Open the scene named **game** in the CubeShooter folder :

Don't hesitate to drop me a comment or a suggestion : FredericRP@equilibregames.com

`EquilibreGames/AMazeKit/CubeShooter/game`

## Select the theme

Set the theme as **CubeShooter** (no space, use capital letters when needed, depending on your system) in the maze inspector, once selected.




Once you've done that, draw, generate a maze and click on the **Instantiate prefabs** button.

## Add the player

A player prefab has been included in the package, here's the path, inside the maze kit folder :

`CubeShooter/Prefabs/_Other/player`

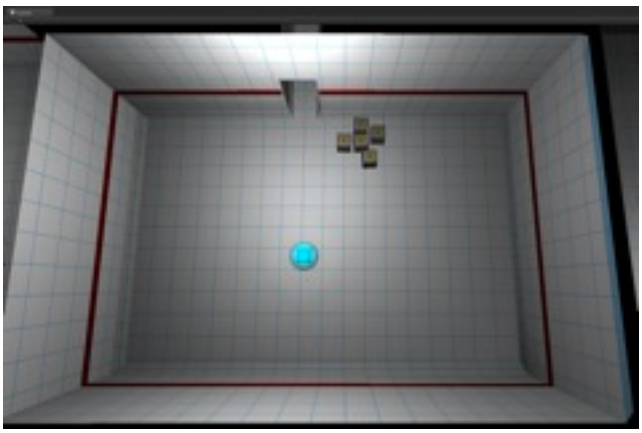
Drag and drop the player prefab inside a room, once the maze has been instantiate.

***Hint** : there should be an empty room (one and only one) where you can safely locate the player without being harmed as soon as the game starts.*

***Hint** : dragging the prefab inside an instantiated room ensures the it's put on a room floor. Otherwise, the prefab can be instantiated anywhere in the 3D world. If you've created the prefab elsewhere, just move its transform so it is located in a room of the maze.*

## Camera settings

Move your main camera above the previous prefab, and rotates it until you see the blue cube (the player) in the middle of it.



The Player script automatically searches for the camera tagged **MainCamera** at the beginning of a play session, so you just have to locate it right before you hit the "Play" button.

### Play !

Hit the "Play" button : you're in !

Keys :

- up/left/down/right (or ZQSD or WASD) to move the player
- mouse to change shooting direction
- mouse left click to shoot

Try to eliminate all monsters and reach level 5 !

For more challenge, put the **ReplayGui** on a GameObject (MainCamera or a new empty GameObject for instance).

---

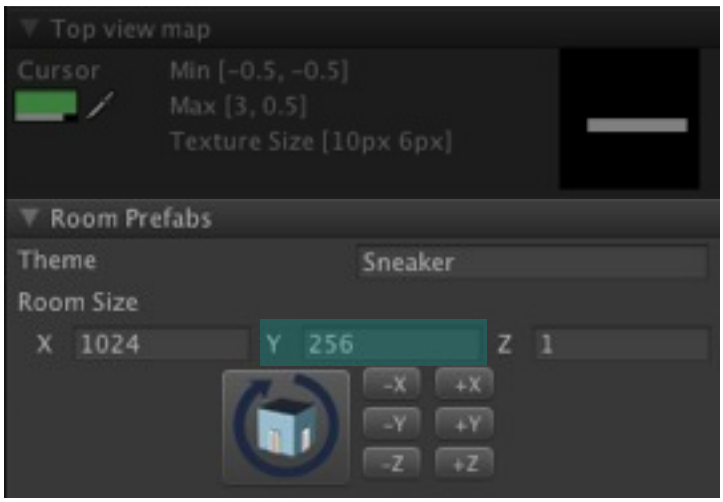
## Sneaker

A sample of a game produced with the Sneaker theme is available from its root. Open the scene named **game** in the Sneaker folder :

`EquilibreGames/AMazeKit/Sneaker/game`

### Select the theme

Set the theme as **Sneaker** (no space, use capital letters when needed).
Here, the package detects a wrong room size as we have some rooms that display things outside of their "should be" bounds. After setting the new theme's prefabs, change the Y size so it's 256 as below :




Once it's done, generate a new maze in the X and Y dimensions. Look at your generated maze globally by selecting it in the hierarchy, press " to focus the whole, then click on the Scene View gizmo so the X axis is pointing right.

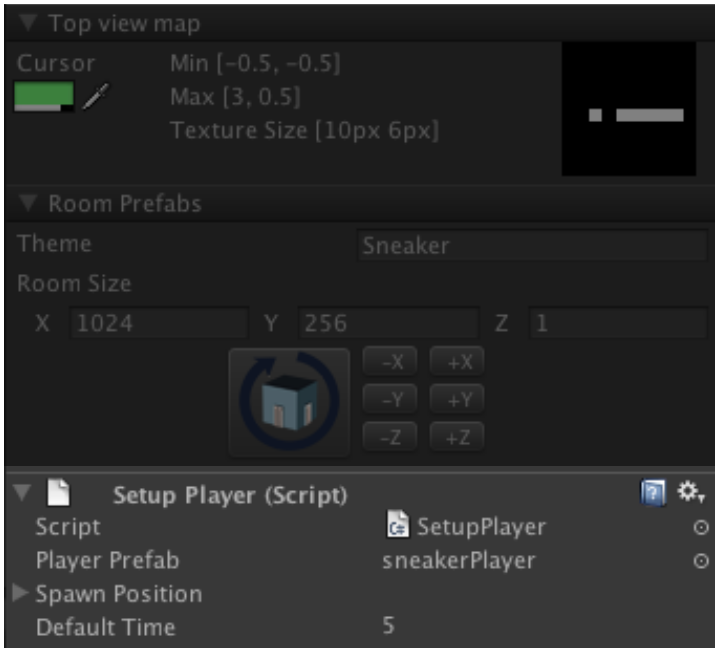
***Warning** : this theme is side-2D only, do not use the Z axis to create rooms or you will have weird behavior (actually, missing rooms).*

## Add the player

The player prefab can be added like in the CubeShooter demo or automatically. As there's other scripts needed, we'll use the **SetupPlayer** script here so it does all, located in :

`EquilibreGames/AMazeKit/Sneaker/scripts/LevelDesign`

Add the **SetupPlayer** component to the maze GameObject in your scene, then select the player prefab from the `Sneaker/Prefabs/_Other` folder in the corresponding field in the inspector.



The default Spawn Position should be ok with the default maze parameters.

Click on **Instantiate Prefabs** to instantiate the prefabs : the new script instantiate also the player and add two scripts to the MainCamera :
- FollowPlayer : make the camera follow the player transform, with a smooth move and a configurable forward position
- Time Slider : an easy GUI to modify the timescale, setting the default TimeScale to 5 when starting.

## Camera settings

This is a side scrolling 2D sneaker, so set the Camera with the following settings :
- Projection : orthographic
- Size : 512
- Clipping planes : near = 1 and far = 100
- Background : black color suits well.

No rotation should be applied to this camera.

## Play

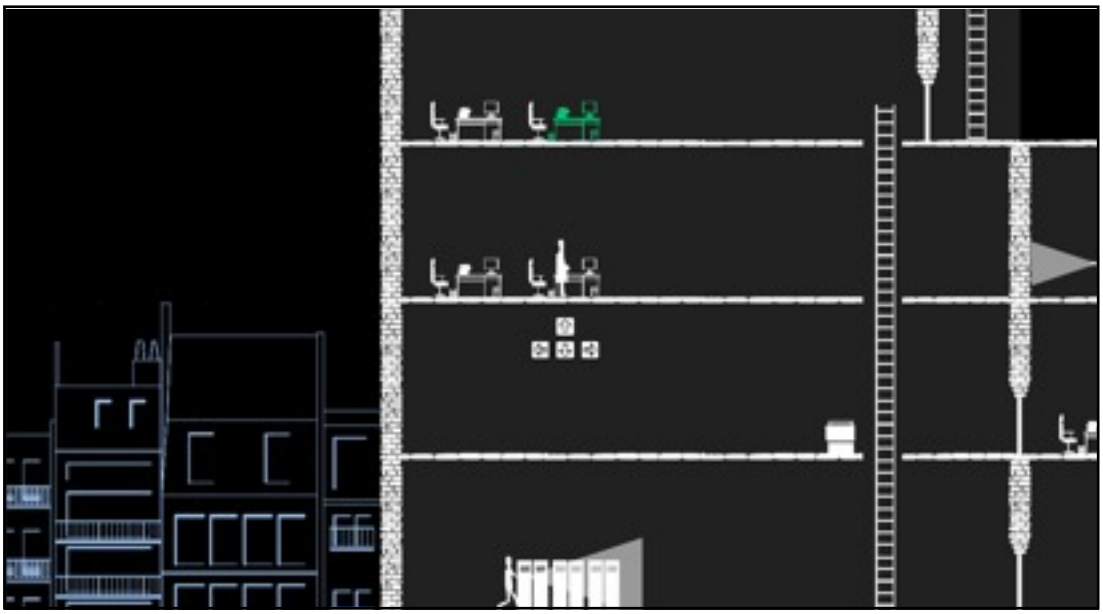
Hit the "Play" button : you're in !

Keys :
- up/left/down/right (or ZQSD or WASD) to move the player and do contextual actions

Context actions :
- near a desk : down to hide under
- near a closet : up to hide inside
- near a door: up to open/close it
- near a copier: up to jump on it (it does not hide you from guards !)
- near a ladder : up/down to climb up/down

Try to find the green desk and avoid all guards !

There's a building prefab allowing you to add a complete look to your game, just drag it on the scene.



***2D with Unity*** *: prefabs from this theme used an included simplified plane (only two triangles) and can be used without external plugin. But if you consider building a theme like this or using it in a real game, beware of performance issues : you should use a specific 2D package like NGUI, 2D Toolkit, ex2D or any other tool that fits your needs.*

## TimeScale and Physics

Why does this theme set the timescale to 5 by default ?
This specific timescale allows the game to run well while having a more arcade-style physic behavior (not still very arcadish though) and being able to use the physic engine.

Try setting the timescale to 1 and you'll see how slow the character falls from the ladder (the guard and player horizontal speeds are script parameters so you change change them with whatever value you want).
You could reduce the scale of every prefab (rooms, guards and player) but you'll encounter soon enough physic issues as a player that falls through the floor.

Different solutions exist to work with those behaviors, but this is not what this package is about, we let you decide of what fits your needs for that !