# Level Creation

In this document a step-by-step guide is given on how to add levels to the various games in Ruler of the Plane. There exists currently three games that make use of fixed levels, namely Kings Taxes, The Divide, and Art Gallery. For each game there is a tutorial on how to add a level to the game. Since the Voronoi game does not make use of levels, it will not appear in these tutorials.

In general, the games make use of two methods to create a level: using the drawing editor IPE and manual insertion of values.

When using IPE, we first create a file and populate it with our new level data. The .ipe file can then be imported into Unity after which an associated level object is automatically extracted. Each game contains a levels folder which is used to store its level objects. Finally, each game will have a main game controller that holds an ordered collection of levels. The new level needs to be added to this list in order for it to show up in the actual game.

Levels can also be manually created inside Unity. This is not recommended for levels of decent size, since it requires much manual input. However, since each level is completely editable in the Unity Inspector, one can simply create a new level object similar to how you create new assets in Unity. Just like before, the new level needs to be added to this levels list in the main game controller in order for it to show up in the actual game.

## Prerequisites

* The Ruler of the Plane unity project.
* The drawing editor IPE (download at [www.ipe.otfried.org](http://www.ipe.otfried.org))

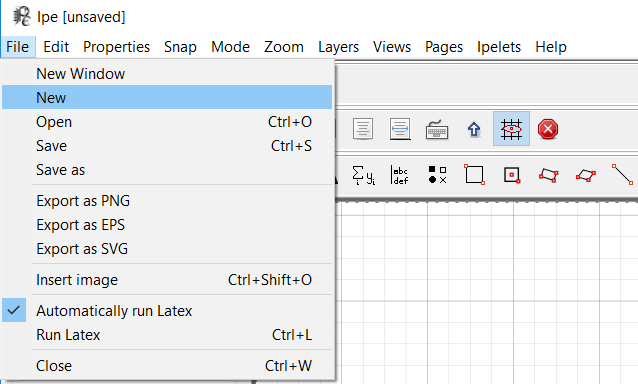
## Kings Taxes

A kings taxes level consists of a collection of points in the plane. In the three mini-games (MST, TSP, Spanner) these points will need to be connected according to different goals and constraints. There are two different types of points, villages and castles, but at the moment there is only a small size difference between the two. The *t*-spanner game will also have the additional parameter *t*.

**Using IPE:**

1. **Open the IPE drawing editor and create a new file**

After creating a new file, you should see an empty white background (by default the underlying grid is shown). This is where we will draw our new level.



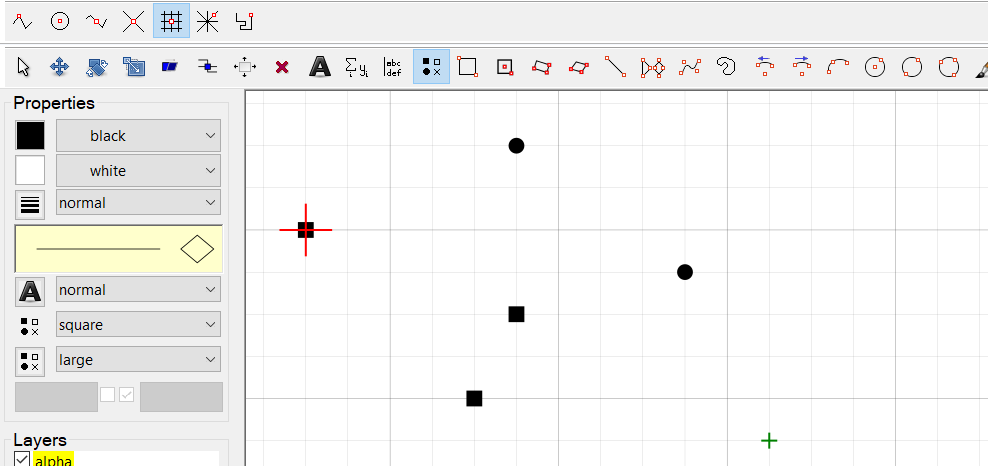
1. **Place a collection of marker points on the canvas.**

Select ‘Marks’ by pressing the ‘M’ key or clicking the -icon at the top of the screen.

Select either the disk or square marker in the left menu for placing villages or castles respectively. Currently, there is only a slight size difference between the two.

Next, press at different locations on the canvas to place marker points, one point for each village/castle in the game.

The level will be scaled to fit the screen, so only relative positioning is important.

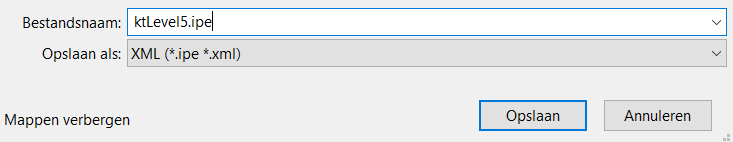


Note: By default the cursor will snap to the grid points, but this can be disabled by pressing the F7 key or clicking the -icon at the top of the screen.

1. **Save the file with the naming format ‘ktLevel?.ipe’ or ‘ktLevel?\_??.ipe’.**

When importing IPE files, the importer script will use the file name to determine the type of level, therefore each Kings Taxes level should start with ‘ktLevel’.

For the *t*-spanner game we can also use the file name to store the parameter *t*, by giving the *t* value after a ‘\_’ (underscore) character (e.g. “ktLevel5\_1.8.ipe”). If not specified, a value of 1.5 is used.

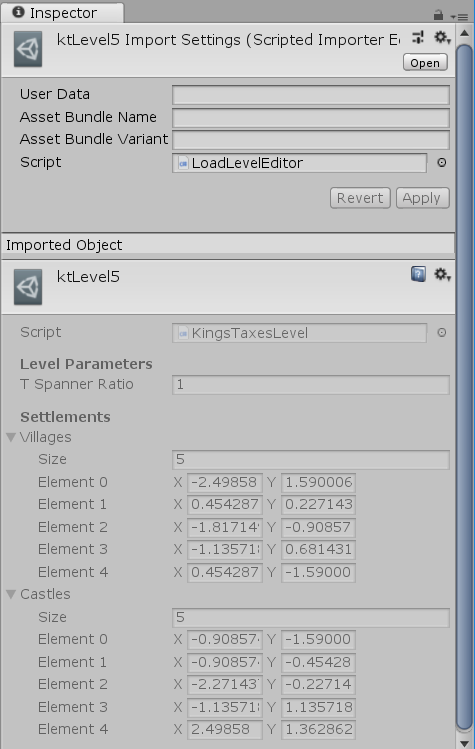


1. **Inside the Unity project view, import the IPE file into the appropriate folder inside ‘Assets/KingsTaxes/Levels’.**

There should be three subfolders inside ‘Assets/KingsTaxes/Levels’, one for each of the mini-games (MST, TSP, Spanner) respectively. Choose the appropriate one.

We can import the IPE file by simply dragging and dropping the file inside the Unity project view.

Alternatively, one can right click inside Unity Project view, click on “Import New Asset…” and select the .ipe file from your local files.



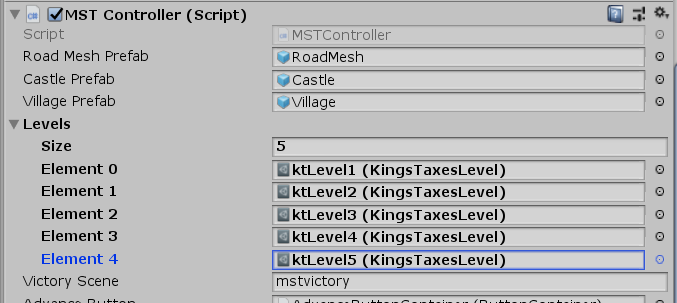
1. **Add the level to the ‘Levels’ list in the game controller of the appropriate game.**

Until now, we have created a level and imported it into Unity, but we still need to add it to the playable levels inside the Kings Taxes game controller.

Navigate to the ‘Assets/KingsTaxes/Scenes’ folder, choose the appropriate subfolder according to the mini-game, and open the scene ending in ‘level’. In the Scene hierarchy view, find the game controller object (the root object in the scene).

In the game controller object, scroll down to find the Kings Taxes controller script. This has a field called ‘Levels’ where we want to add our new level object.

Simply increase the size of the array, which should add a new element. Then drag our level object into the final element. This should add it to the collection.



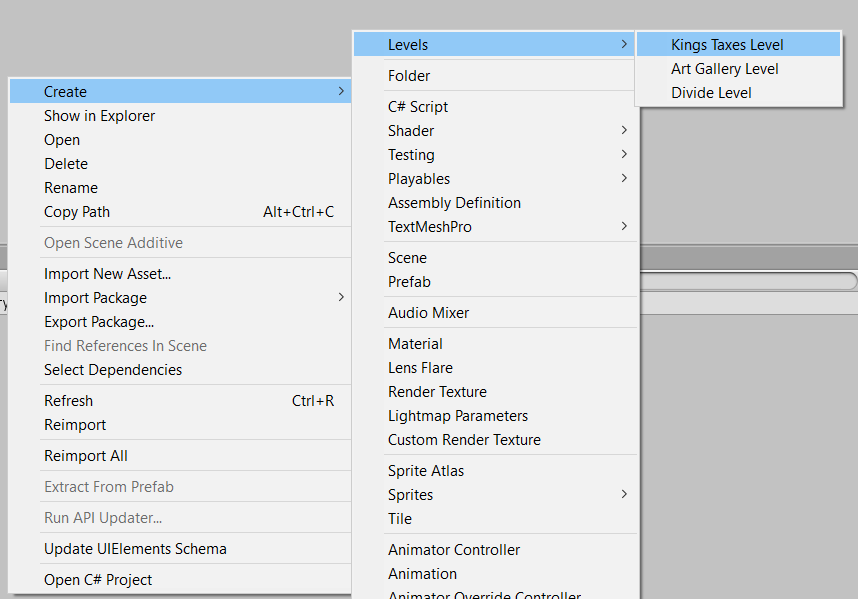
**Manual (not recommended):**

1. **In the Unity project view, open the appropriate levels folder inside ‘Assets/KingsTaxes/Levels’.**

There should be three subfolders inside ‘Assets/KingsTaxes/Levels’, one for each of the mini-games (MST, TSP, Spanner) respectively. Choose the appropriate one.

1. **Right click and select ‘Create/Levels/Kings Taxes Level’.**

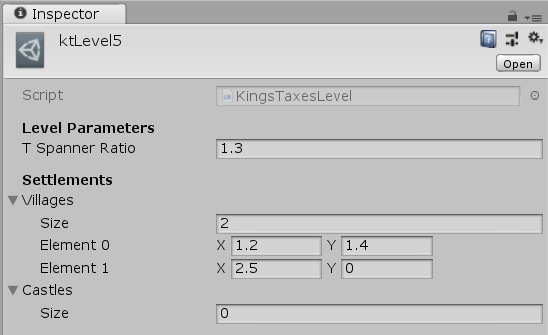
This action should generate a new empty Kings Taxes level



1. **Click on the created level object and insert the necessary values manually.**

After selecting the created Kings Taxes level, you should see the variables Villages, Castles and the *t*-spanner ratio (the latter is only relevant for the Spanner mini-game).

Insert the settlement points as (x, y) floating point coordinates until satisfied with the result.



Note: To fill in arrays in Unity, you first need to enter the size.

1. **(Optional) Give the level a better name.**

By default, the level will have a placeholder name, so you can replace this with something more fitting. It is not necessary to have the level start with ‘ktLevel’, but this might still be good standards.

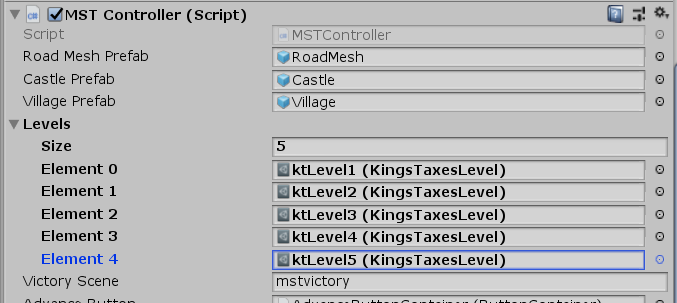
1. **Add the level to the ‘Levels’ list in the game controller of the appropriate game.**

Until now, we have created a level and imported it into Unity, but we still need to add it to the playable levels inside the Kings Taxes game controller.

Navigate to the ‘Assets/KingsTaxes/Scenes’ folder, choose the appropriate subfolder according to the mini-game, and open the scene ending in ‘level’. In the Scene hierarchy view, find the game controller object (the root object in the scene).

In the game controller object, scroll down to find the Kings Taxes controller script. This has a field called ‘Levels’ where we want to add our new level object.

Simply increase the size of the array, which should add a new element. Then drag our level object into the final element. This should add it to the collection.



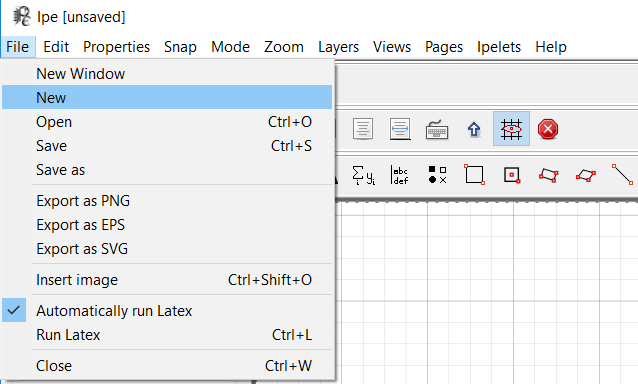
## The Divide

A Divide level consists of a collection of three types of points, that correspond to spearmen, archers, and mages, plus a parameter *k* for the maximum number of swaps. Each set should consist of an even number of points, such that they can be cut evenly in two.

**Using IPE:**

1. **Open the IPE drawing editor and create a new file**

After creating a new file, you should see an empty white background (by default the underlying grid is shown). This is where we will draw our new level.



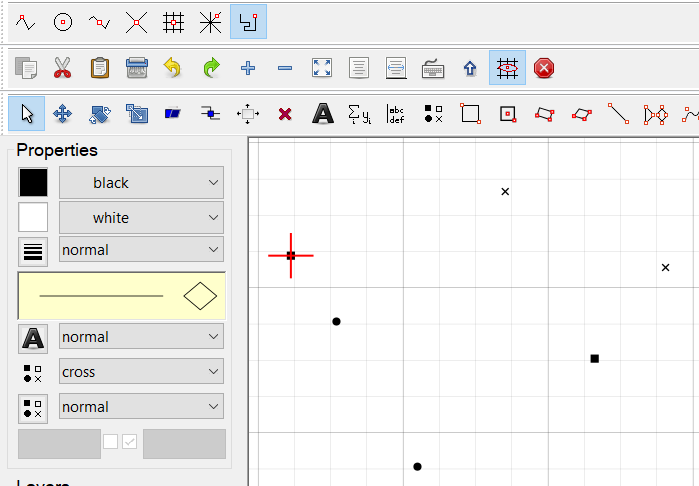
1. **Place a collection of marker points on the canvas.**

Select ‘Marks’ by pressing the ‘M’ key or clicking the -icon at the top of the screen.

Select either a disk, a square, or a cross marker in the left menu for placing spearmen, archers, and mages respectively.

Next, press at different locations on the canvas to place marker points, one point for each spearmen/archer/mage.

The level will be scaled to fit the screen, so only relative positioning is important.

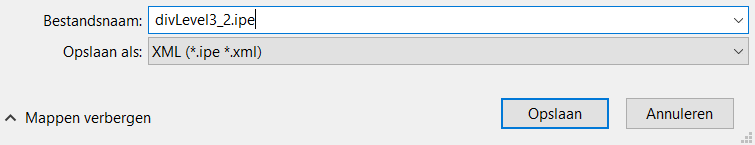


Note: By default the cursor will snap to the grid points, but this can be disabled by pressing the F7 key or clicking the -icon at the top of the screen.

1. **Save the file with the naming format ‘divLevel?.ipe’ or ‘divLevel?\_??.ipe’.**

When importing IPE files, the importer script will use the file name to determine the type of level, therefore each Divide level should start with ‘divLevel’.

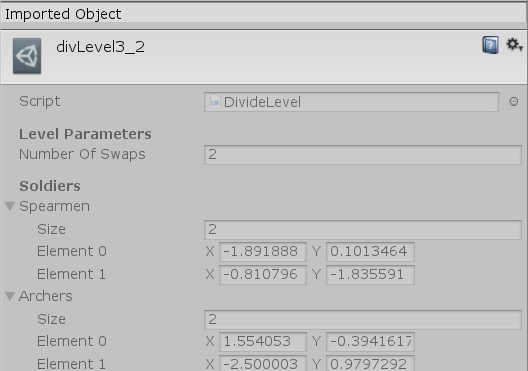
The number of swaps parameter can also be stored in the filename, by giving the value after a ‘\_’ (underscore) character (e.g. “divLevel5\_2.ipe”). If not specified, no swaps are allowed.



1. **Inside the Unity project view, import the IPE file into the folder ‘Assets/Divide/Levels’.**

We can import the IPE file by simply dragging and dropping the file inside the Unity project view.

Alternatively, one can right click inside Unity Project view, click on “Import New Asset…” and select the .ipe file from your local files.



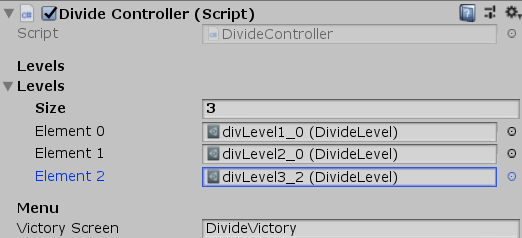
1. **Add the level to the ‘Levels’ list in the game controller of game.**

Until now, we have created a level and imported it into Unity, but we still need to add it to the playable levels inside the Divide game controller.

Navigate to the ‘Assets/Divide/Scenes’ folder and open the scene named ‘DivideGame’. In the Scene hierarchy view, find the game controller object (one of the root objects in the scene).

In the game controller object, scroll down to find the DivideController script. This has a field called ‘Levels’ where we want to add our new level object.

Simply increase the size of the array, which should add a new element. Then drag our level object into the final element. This should add it to the collection.



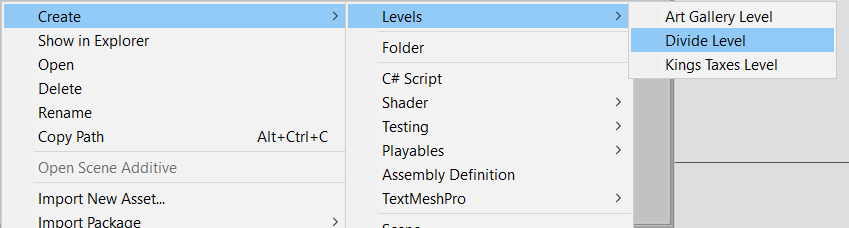
**Manual (not recommended):**

1. **In the Unity project view, open the levels folder ‘Assets/Divide/Levels’.**

This folder is used for storing all level objects.

1. **Right click and select ‘Create/Levels/Divide Level’.**

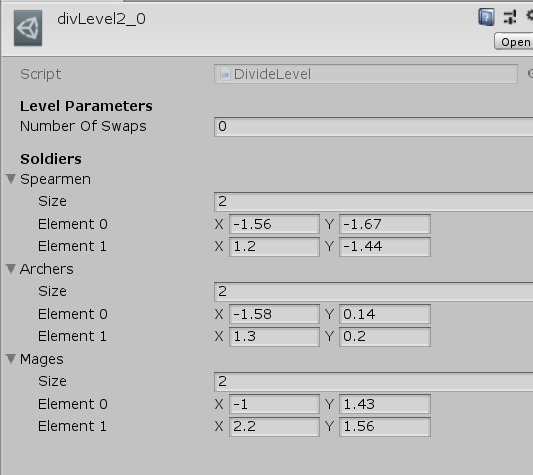
This action should generate a new empty Divide level



1. **Click on the created level object and insert the necessary values manually.**

After selecting the created Divide level, you should see the variables Spearmen, Archers, Mages and the number of allowed swaps.

Insert the soldier points as (x, y) floating point coordinates until satisfied with the result. Optionally change the number of allowed swaps.



Note: To fill in arrays in Unity, you first need to enter the size.

1. **(Optional) Give the level a better name.**

By default, the level will have a placeholder name, so you can replace this with something more fitting. It is not necessary to have the level start with ‘divLevel, but this might still be good standards.

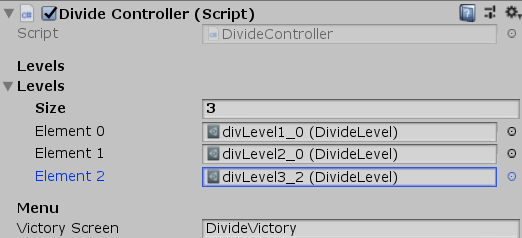
1. **Add the level to the ‘Levels’ list in the game controller of the appropriate game.**

Until now, we have created a level and imported it into Unity, but we still need to add it to the playable levels inside the Divide game controller.

Navigate to the ‘Assets/Divide/Scenes’ folder and open the scene named ‘DivideGame’. In the Scene hierarchy view, find the game controller object (one of the root objects in the scene).

In the game controller object, scroll down to find the DivideController script. This has a field called ‘Levels’ where we want to add our new level object.

Simply increase the size of the array, which should add a new element. Then drag our level object into the final element. This should add it to the collection.



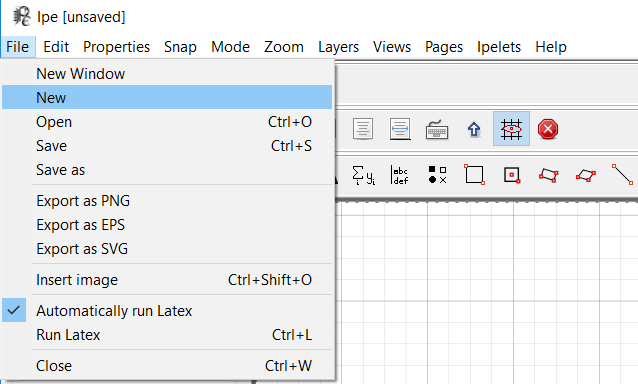
## Art Gallery

An art gallery level consists of a simple polygon without holes and a parameter *k* for the maximum number of lighthouses. The number of light houses should be set such that there exists a solution to the art gallery problem for placing *k* guards.

**Using IPE:**

1. **Open the IPE drawing editor and create a new file**

After creating a new file, you should see an empty white background (by default the underlying grid is shown). This is where we will draw our new level.

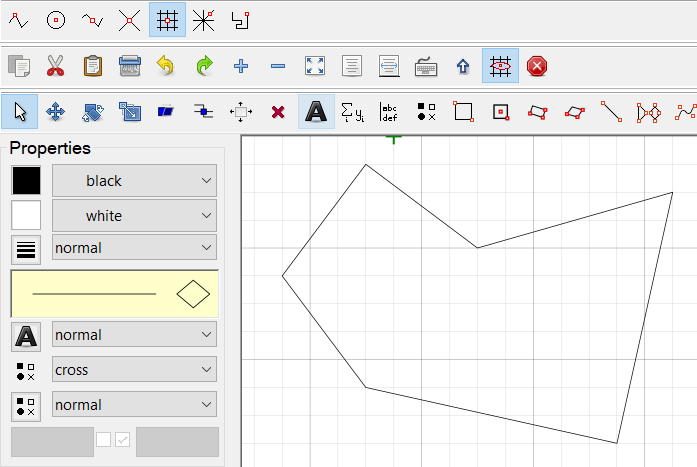


1. **Create a polygon in the canvas.**

Select the polygon tool by pressing Shift + P or by clicking on the -icon.

Next, press the left mouse button to place the vertices of the polygon. Press the right mouse button to place the last vertex and finish the polygon. The Polygon can be either drawn clockwise or counter-clockwise. The polygon should not intersect itself.

The level will be scaled to fit the screen, so only relative positioning is important.

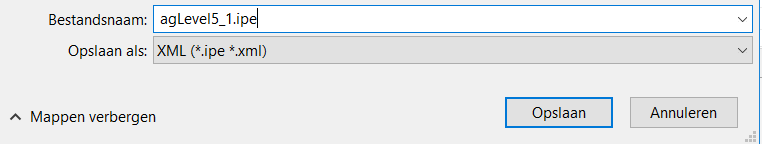


Note: By default the cursor will snap to the grid points, but this can be disabled by pressing the F7 key or clicking the -icon at the top of the screen.

1. **Save the file with the naming format ‘agLevel?.ipe’ or ‘agLevel?\_??.ipe’.**

When importing IPE files, the importer script will use the file name to determine the type of level, therefore each Art Gallery level should start with ‘agLevel’.

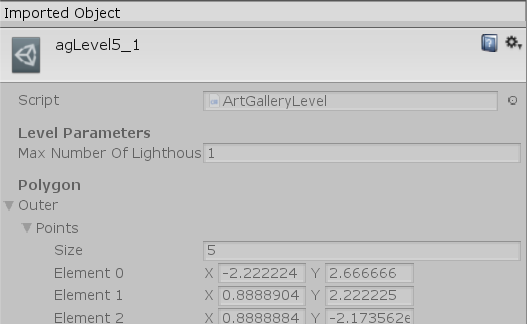
The number of lighthouses that can be placed can also be stored in the filename, by giving the value after a ‘\_’ (underscore) character (e.g. “agLevel5\_1.ipe”). If not specified, 1 lighthouse is allowed to be placed.



1. **Inside the Unity project view, import the IPE file into the folder ‘Assets/ArtGallery/Levels’.**

We can import the IPE file by simply dragging and dropping the file inside the Unity project view.

Alternatively, one can right click inside Unity Project view, click on “Import New Asset…” and select the .ipe file from your local files.



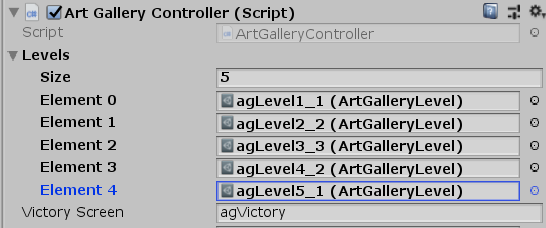
1. **Add the level to the ‘Levels’ list in the game controller of game.**

Until now, we have created a level and imported it into Unity, but we still need to add it to the playable levels inside the Art Gallery game controller.

Navigate to the ‘Assets/ArtGallery/Scenes’ folder and open the scene named ‘agLevel’. In the Scene hierarchy view, find the game controller object (one of the root objects in the scene).

In the game controller object, scroll down to find the ArtGalleryController script. This has a field called ‘Levels’ where we want to add our new level object.

Simply increase the size of the array, which should add a new element. Then drag our level object into the final element. This should add it to the collection.



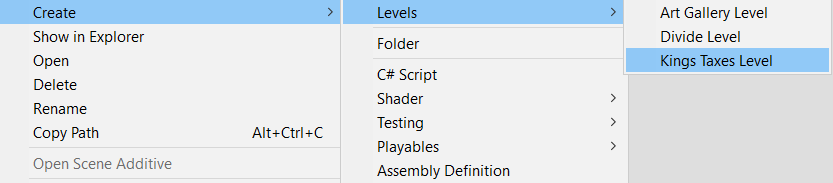
**Manual (not recommended):**

1. **In the Unity project view, open the levels folder ‘Assets/ArtGallery/Levels’.**

This folder is used for storing all level objects.

1. **Right click and select ‘Create/Levels/Art Gallery Level’.**

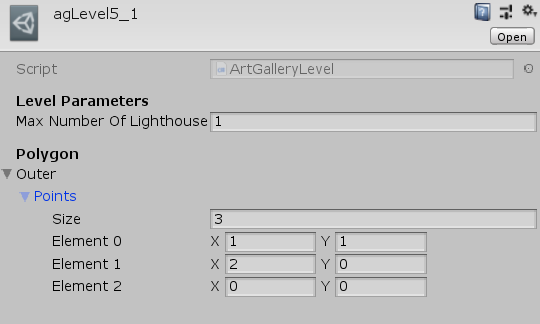
This action should generate a new empty Art Gallery level



1. **Click on the created level object and insert the necessary values manually.**

After selecting the created Divide level, you should see the variables Outer polygon and the maximum number of lighthouses.

Insert each polygon vertex of the outer boundary as (x, y) floating point coordinates until satisfied with the result. Optionally change the maximum number of lighthouses.



Note: To fill in arrays in Unity, you first need to enter the size.

1. **(Optional) Give the level a better name.**

By default, the level will have a placeholder name, so you can replace this with something more fitting. It is not necessary to have the level start with ‘agLevel, but this might still be good standards.

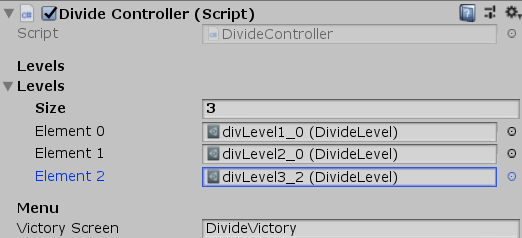
1. **Add the level to the ‘Levels’ list in the game controller of the appropriate game.**

Until now, we have created a level and imported it into Unity, but we still need to add it to the playable levels inside the Art Gallery game controller.

Navigate to the ‘Assets/ArtGallery/Scenes’ folder and open the scene named ‘agLevel’. In the Scene hierarchy view, find the game controller object (one of the root objects in the scene).

In the game controller object, scroll down to find the ArtGalleryController script. This has a field called ‘Levels’ where we want to add our new level object.

Simply increase the size of the array, which should add a new element. Then drag our level object into the final element. This should add it to the collection.



## Tips

* When using IPE, the last created object will remain selected until you create a new object or deselect it. This means that if you change a property (e.g. marker size or shape), you might accidently change the property of the last created object. To avoid this, first deselect the object by pressing on the background.
* If changes need to be made to an imported level, double click the generated level object. If you have IPE as your default program for files with extension .ipe, this will open the IPE editor for the given file. You can edit the the file and save it, after which Unity will trigger a re-import.