



---

## Introduction

What is the Penguins Editor? The Penguins Editor was used to create all the levels as well as the UI in the game. With the editor you can create vast and very complex levels for the Penguins game or even redesign the entire game's UI. The editor features some pretty nice functionality when it comes to creating levels but the most powerful tool of the editor is its logic groups and timelines. With logic groups and timelines you can create very dynamic and complex levels. The following site was created to help you figure out how to create levels and how to pass them around to friends to share.

## Get the Game

To get the editor to work you must download the Penguins! game. You can find it here:

### [Step 1. GET PENGUINS!](#)

The editor was built upon .NET, which means you need the .NET Framework for it to work. You can download that here:

### [Step 2. GET .NET FROM MICROSOFT](#)

If for some reason that link is broken, simply go to Microsoft.com and do a search for the .NET Framework.

Once you have both of these installed you're ready to install the editor. Just download it from here:

### [Step 3. GET THE LEVEL EDITOR](#)

Copy the contents of the .zip into the location where the Penguins! Game was installed. If you don't know where the game was installed, right clicking on the Penguins! icon and selecting properties will bring up a menu with the path to the game. If you're using Windows XP, you can also select Find Target.

**NOTE:** The editor is a work-in-progress. It is a non-supported tool and is only designed to make additional levels for the Penguins! game.

## Using the Level Editor

If this is your first time using the Penguins! Level editor, try watching the editor demonstration video at [www.playpenguins.com/leveleditor.html](http://www.playpenguins.com/leveleditor.html)

### Keyboard Shortcuts

To use the shortcuts, make sure at least one object is selected. Then hold the key down and left click while dragging your mouse.

<b>C</b>	Copies object.
<b>O</b>	Change opacity.
<b>S</b>	Change sizes (snaps to grid)
<b>R</b>	Rotation (snaps to grid)
<b>W</b>	Change Width (snaps to grid)
<b>H</b>	Change Height (snaps to grid)

### Templates

In the editor.zip file downloaded from the site are six template files. You should use these level templates to create more advanced levels. During the creation of Penguins! we used templates like these to save the team hours of work.

To find the templates, load the editor and look in the following folder:  
Resources->BonusMaps->TestFolder

A note about the TestFolder; if you want to test your level, this is a great place to put your new levels. You can then play them via the “more levels” button in the main menu of the Penguins! game.

### Creating Objects:

To create objects, you can either double click on them in the “Object Types Window” or drag each object you wish to make into the objects window. For example, to create a “Gadget” click the plus sign to expand the “Penguins” tree in the top of the objects window. Look for the word “Gadget” and double-click it. You can drag created objects onto most other objects. When you do this, it creates a parent child relationship. The object containing the others is called the parent and the objects inside of the parent are the children. This allows you to use objects like folders to keep your levels organized. If you move the parent, the children will move along with the parent.

### Setup:

Make sure snap is turned on, the box at the top of the editor. There should be a check in the box next to the word “Enabled”. Also make sure the grid is at 8.

### **Step 1.**

Choose the “Maya” decals for this tutorial. Visit the link on the left “Tiles and Decal Maps” to see how to load the tiles. Be sure to load the Universal Tiles too.

### **Step 2.**

Create an editor level object. You can find the object under the “Logic” group inside the “Object Types” window. When you do this, you should see a large yellow area appear in the Editor Window. This will be the size of the level. Make sure it is no smaller than 1024x768. If it’s smaller, it could crash the game. The window can be as big as you like. We recommend 1420x1060.

Selecting objects:

When you want to select something, you can either select it in the Objects window or by left clicking on it in the render window. **SELECTING OBJECTS IN THE RENDER VIEW USES THE LEFT MOUSE BUTTON.** You may also drag while holding the left mouse button and it will select multiple objects. If you want to deselect an object(s), hold ctrl down while selecting.

### **Step 3.**

Select the newly created “EditorLevel” object. On the right side you will see the properties of the Level. First, make the depth -100. Set the Level Name to “My First Level”. In the “Times” field in the property window, you will see a property called “Gold”. To get a gold trophy in the game, the player must beat a specified time. We suggest 120 seconds as an appropriate time setting. The last property is the “Penguins Count”. This represents the number of how many Penguins the player must save to reach the next level. Our suggestion is 25 Penguins.

### **Moving Objects:**

To move objects around, select them via the Objects Window or the Render View. Then right click and drag into the render window.

### **Step 4.**

Create a “LevelPieceStart” and “LevelPieceEnd”. You can find these two objects in the “Penguins” group in the Object Types window. You should see the start and exit pipes appear in the level.

**Note 1:** If you do not see the pipes but do see the objects in the Objects Window, this means your “EditorLevel” object’s depth has not been set to -100 and the pipes are hidden underneath the big yellow square. Go back to step three and set the depth to -100.

**Note 2:** You must have ONLY one LevelPiece “start” and ONLY one LevelPiece “end.”

### **Copying objects:**

To copy objects you must select them. Then hold down “C” and drag the object you wish to copy. You should now see two replicas of the object.

### **Step 5.**

When you're making the ground for the penguins to walk on, it is important to remember that Penguins are always at depth 0 in the game.

To do this, create a "ParalaxingFolder" which can be found in the Penguins group in the Object Type Window. Change the depth property of the ParalaxingFolder to -40. Rename the ParalaxingFolder to "Ground Folder". To change the name of an object, select it once in the Objects Window, wait a second and click it again. Giving at least one second of wait time will allow you to edit the name of the folder.

Now to create the ground piece, create a "LevelPiece" object found in the Penguins group in the Object Types window. Drag and place the newly created LevelPiece into the GroundFolder you created. This made the LevelPiece a child of the folder (and the GroundFolder is the parent.). Any properties of the folder will be inherited if shared by the LevelPiece. What does this mean? It means when you set your Ground Folder depth to -40 and added the LevelPiece, the LevelPiece depth is now -39. Why -39 and not -40? If you select the LevelPiece and look at the properties of the object, you should see that it's depth is 1. So the folder's depth is at -40 and LevelPiece's depth is 1, add them together and you get -39.

Select the LevelPiece you created. Now in its properties you should see a section called Images with a property named Normal. This is the normal image the LevelPiece will show in the game. If you select the property "Normal", a pull-down should appear with all the images you imported. If "none" is the only option, you didn't load any images. Look at step 1 again. If none is NOT the only option, select the image "TILE\_GROUND1".

### **Setting image size:**

When any object gets created it gets created at a default size, unless copying an object. Since most images aren't the default size, you need to resize the level piece every time you change its image. To do this, select the LevelPiece or pieces, find the property named "Use Image Size", and set it to true. As a shortcut, you can just double click this property to change it to true.

Select "TILE\_GROUND1" and make the tile its default size. Next, move it halfway down and to the left of the screen. Then you must copy that ground piece. Select the piece, hold down the "C" button on your keyboard. With the key held down, right-click and drag the LevelPiece. You should see a copy of the object. Place it right next to the original LevelPiece you created. In the Objects Window, under your GroundFolder you should see two LevelPieces now. Continue copying the pieces until you have ground reaching from the left all the way to the right of the screen. You can do this faster if you select and copy more than once piece at a time. Hold the CTRL key and click additional pieces to select them. If you use left click and drag to select more pieces, you must also select the yellow EditorLevel piece that is behind the blocks. Avoid that for now.

### **NOTE:**

At this point it's a good idea to make your EditorLevel object invisible. Because the game must only have 1 EditorLevel object you must be careful not to create more than

one, however it's very easy to create multiple copies of it if you are not careful. If you select the object and set its visible property to false it you won't have to worry about accidentally moving or copying it.

#### **Step 6.**

Once you have ground reaching all the way across the screen, place the start and end pieces on the ground.

#### **Step 7.**

Now lets place some dirt below our ground. Select any one piece of the LevelPieces you have created. Copy the piece and drag it below the ground line you created. With it selected, change the "Images/Normal" property on it to "TILE\_STRUT". Now make a line of the newly created TILE\_STRUT LevelPieces right below your ground line pieces. Meaning a line of TILE\_STRUT LevelPieces should stretch from the left of the screen to the right of the screen. NOTE – Remember you can select multiple objects and copy them!

#### **Step 8.**

Now select the entire line of tile-strut LevelPieces you created. Copy and drag the entire line as many times as you have to for it to reach the bottom of the screen.

#### **Step 9.**

The last step to handle for the level is collision. Do not play the level before collision is added or the Penguins will fall right through the ground. To add a collision piece, create a new folder named "CollisionFolder". Once you have your folder, drag a LevelPiece into the newly created CollisionFolder. The LevelPiece you created has a property called collision types. Select "Box" from the collision types pull-down menu. This piece now has collision in the shape of a box. Penguins who run into the piece from any direction will stop. Note that nothing has changed on the screen. To have a visual representation of what the collision looks like, in the "Images/Normal" Property of the LevelPiece, select the image "COL\_BOX". Here are the rules to collision:

1. Do not make all level pieces have collision. Make as few collision pieces as you can.
2. Gadgets cannot be dragged into solid collision.
3. Collision turns invisible in game (making the red "col\_box" disappear.)
4. Collision ignores the rotation property.
5. 45-degree collision must use the invertx and inverty properties to change orientation.
6. There are numerous images to use for representing your collision. (col\_box, col\_45, col\_45top, etc.)
7. Collision types of top, meaning "box top", "45 top" means Penguins can come up through the collision and are only stopped if they going down and hit the top of the box.

#### **Resizing objects:**

If you want to change an objects size easily, select the object and hold “S” down and drag with the right mouse button.

Position your collision box so it appears over the top of the entire ground and strut’s you made in steps 5-8. Once you’ve set up the collision, you may want to make it invisible. If you’re not working on the collision, it can get in the way.

### **Step 10.**

The final step is to add the level background piece. Create a Background Controller Object. In it’s properties of “Level”; select Monkey. You won’t see the background in the editor, but it will be there when the level is loaded in-game.

You’re DONE! Now save your level. On the link to the left entitled “How to Add a level to the game” will give a quick how-to on getting your level into the game.

## Share your Levels

If you want to pass around your levels, create the following:

1. Create a jpg image that is 64x64 pixels.
2. Create a description.txt file. It can be named anything you like, but must end with “.txt”. In the description file you will add three lines.
  - a. The first line is the name of the level. This could be called anything from "Joe’s Levels" to something like "The most difficult levels in the world!" It can be named whatever you like.
  - b. The second line is a small description of the levels.
  - c. The third line is a longer description about anything you like. We recommend telling a bit about yourself.

The end result should be to create a level that can be shared with others. If that is the case, you should have three files.

1. Mylevel.xml
2. MyImage.jpg
3. MyDescription.txt

## Add your new Level to the Game

- Step 1. Create a new folder under the “BonusMaps” folder, which can be found under Resources\BonusMaps\.
- Step 2. Copy your new level file into the newly created folder.
- Step 3. Go into Penguins! and select “”More Levels”. Your newly created level should appear in the list. You can now put as many levels as you like in your newly created folder and they should appear in the game.

# Appendix A: Logic Group

*This page is being updated and is a work in progress.*

## **OBJECTS PROPERTIES**

**ALL PROPERTIES HAVE HELP IN THE EDITOR. CLICK ON A PROPERTY AND AT THE BOTTOM OF THE PROPERTIES WINDOW IS THE HELP.**

## **LOGIC GROUP**

*NOTE: If you need more info on an input or output of an object the link to the left "Ask for Help". NOTE: Support will be very limited for logic objects.*

<b>BroadcastSequence</b>	<p>This object is made for the LogicGroup object. It broadcasts in a given sequence whatever is input into it.</p> <p><b>inputs</b></p> <ul style="list-style-type: none"><li>• Broadcast</li></ul> <p><b>outputs</b></p> <ul style="list-style-type: none"><li>• On1...On10</li></ul>
<b>EditorLevel</b>	<p>You must create an Editor Level for any level to work. You must only have 1 Editor level per level.</p> <p><b>inputs</b></p> <ul style="list-style-type: none"><li>• RequestSkipButton - not supported</li></ul> <p><b>outputs</b></p> <ul style="list-style-type: none"><li>• LevelStart - called only once a level starts</li><li>• LevelReset - called every time a level starts</li></ul>
<b>LogicAdder</b>	<p>Made for the LogicGroup object. When an object outputs an event you can link it to the add. Once X number of inputs has occurred, then a broadcast event will happen. A good example of this is if you wanted to add switches to open a door. When a switch gets pulled it adds to the adder. When the adder equals two (two switches pulled) then it will broadcast the "OnEquals" event.</p> <p>NOTE: the adder gets reset every time a level is started or reset.</p> <p><b>inputs</b></p> <ul style="list-style-type: none"><li>• Add - adds one</li><li>• Subtract - subtract one</li></ul> <p><b>outputs</b></p> <ul style="list-style-type: none"><li>• OnEquals - When the value is equal to the "Number" property of the LogicAdder</li></ul>

	<ul style="list-style-type: none"> <li>• OnNotEqual - called whenever an input occurs (add or subtract) and the LogicAdder is not equal to the "Number" Property of the Logic Adder</li> </ul>
<b>LogicGroup</b>	<p>The easiest way to think about a Logicgroup is that of a flow-chart. You drag created objects into the logic group. Each object has inputs and outputs. Linking inputs and outputs of objects causes certain other outputs to occur or events to occur in the game. Logic groups ARE NOT independent groupings; just because you have an object in one Logic group does not make it a different object in another Logic group.</p> <p><b>inputs</b></p> <ul style="list-style-type: none"> <li>• Add - adds one</li> <li>• Subtract - subtracts one</li> </ul> <p><b>outputs</b></p> <ul style="list-style-type: none"> <li>• OnEquals - When the value is equal to the "Number" property of the LogicAdder</li> <li>• OnNotEqual - called whenever an input occurs (add or subtract) and the LogicAdder is not equal to the "Number" Property of the Logic Adder</li> </ul>

## Appendix B: Tiles and Decals

The name of the categories below match folders that can be found under the resources\tiles inside the Penguins! game. Use the images->load sheets in the editor's pull-down menus to load the desired tiles or decals. Some of the decals below can be found under the "universal" folder (found under resources\tiles\universal\)

### **Penguins\_ex**

- Tiles
- Decal sheet 1
- Decal sheet 2

### **Maya**

- Tiles
- Decals

### **Nocturnal**

- Tiles
- Decals

### **Asia**

- Tiles
- Decal sheet 1
- Decal sheet 2



- Decal sheet 3

### **Atlantis**

- Tile sheet 1
- Tile sheet 2
- Decal sheet 1

### **Bayou**

- Tiles
- Decal sheet 1
- Decal sheet 2
- Decal sheet 3

### **Baghdad**

- Tiles
- Decals

### **Savannah**

- Tiles
- Decals