



Welcome to Platform Studio Help!

Getting Started

Here is where you learn how to make games with Platform Studio. Just follow the step-by-step [tutorials](#) and you will be on your way! Or, you can learn about the Platform Studio's [windows and editors](#) individually.

Quick Reference

Use the contents, index, and search tabs on the left part of this window. If it is not visible, click the "Show" button on the toolbar.

Scripting Help

If you do not already know the syntax of VB.NET and are not familiar with the .NET Framework, and you want to try scripting, we recommend you visit [Microsoft MSDN's website](#). This is also a good place to look up certain functions, keywords, etc. This site can also be accessed quickly through the Help menu in the Script Editor.

Once you learn VB.NET, you should read the [Platform Studio Scripting Documentation](#) to get the most out of scripting.

Support

Our website is located at <http://soft.firstproductions.com/pstudio>. This is also the home of our community forums. We encourage you to drop by and become a member, so our community can grow and get questions answered quicker for you, and the rest of the community. This is also the place to give us feedback, so we can see how we're doing.

Or, you can email us at pstudio@firstproductions.com. We will gladly answer any questions you might have.



Copyright 2005, First Productions, Inc.



Tutorials up

Extended Tutorials

[Getting started](#)
[Walkthrough of creating a platform game](#)
[Creating a tileset](#)
[Creating a level](#)

"How-To" Tutorials

[Creating a checkpoint](#)
[Creating "secret" areas](#)

[Create a High Scores list](#)
[Create a shooting enemy](#)
[Speed up your game](#)



Copyright 2005, First Productions, Inc.



Tutorial: Getting Started up

This tutorial will guide you through the process of making a new game using Platform Studio. Follow the instructions in order below:

To create a blank game...

1. Click **File**, and select **New** or click the New button on the toolbar.
2. That's it!

Use the New Game Wizard to customize basic options and themes...

1. Click **File**, and select **New Game Wizard**.
2. Click **Next**.
3. Type in the general information about your game. You can always come back to this screen later (with additional fields you can set) by clicking the File menu and selecting Game Properties. After you have finished, click **Next**.
4. If you already know how many levels you want to have, use the **Add** button to add new levels to your game. You can also specify a custom level name, or you can use the default. If you know what size levels you want to use, specify it under **Default level size**. Under that, you can also choose which default background you want to use, and also customize it by selecting (Custom). When you have finished, click **Next**.
5. Select your resolution, or choose Windowed mode, and specify your window options. Click **Next**.
6. Choose your user interface's color scheme. You can customize it by clicking the ... buttons and changing properties in the Property Grid on the right portion of the window. (If you want to change your scheme later, you can do so in the Window Editor by clicking the Tools menu and selecting Window Scheme Wizard.) Click **Next**.
7. Choose and customize the behavior for the "About" and "Help" buttons on your menu. If you want a popup message showing your about information, select **Message dialog** under **About style**, or if you want your about information shown in a customizable screen, select **Window**. You can also choose if you want a button that links to your website (that you specified earlier), or a button that opens the user's email program to send an email to your support email address (that you specified earlier). Note that if you change your website or support email after the wizard, you will have to open the Window Editor and change the buttons' behaviors to what it should be.
Under **Help style**, you can choose between a window, or open a help file that you have made in a different program. It can load .chm, .hlp, .htm, .html, .txt, .rtf, and .doc. If you want to load another format, you can do so by customizing the Help button's action in the Window Editor. Click **Next**.
8. Click **Finish** to create the new game.

Copyright 2005, First Productions, Inc.



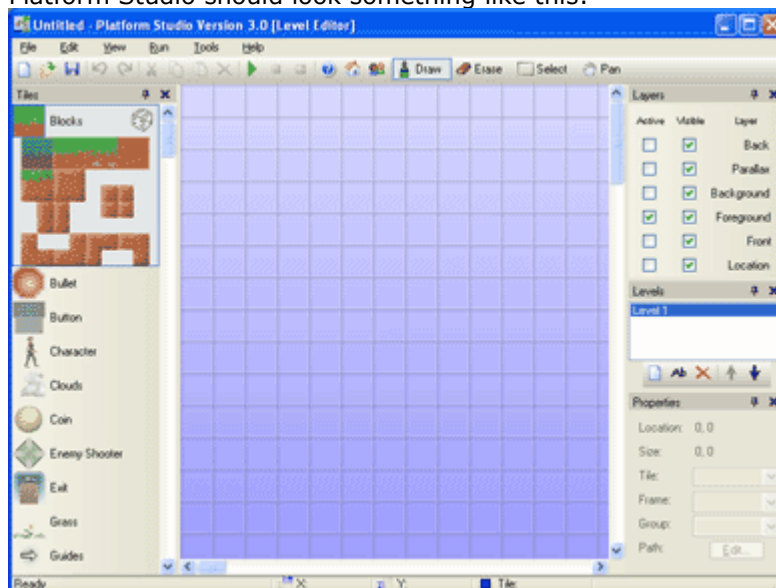
Walkthrough of Creating a Platform Game up

This tutorial will guide you through the process of making a simple platform game using Platform Studio. Follow the instructions in order below:

I. Creating a new game

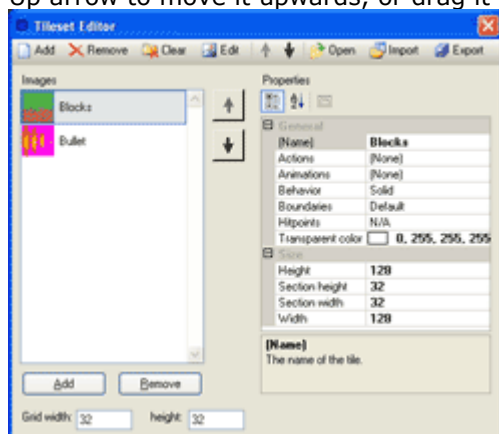
1. Click **File**, and select **New Game Wizard**.
2. Click **Next**.
3. Under **Name of game**, type **Platform Game Tutorial**.
4. Under **Credits** where it says "Created by...", replace MyName with your name.
You are allowed to delete the "Powered by Platform Studio" if you want.
5. Under **Company**, type in the name of your company or leave it blank.
6. Under **Website**, type in the name of your website, including the "http://".
7. Under **Support email**, type in your support email address, or your personal email address.
8. Click **Next**, and then click **Next** again.
9. Make sure the **Windowed** option is **checked**, and make sure the window size is set to **450x300**. Under **Window title**, type **Platform Game Tutorial**.
10. Click **Next**.
11. Choose a color scheme (optional), and click **Next**.
12. Click **Next**, and then click **Finish**.

Platform Studio should look something like this:

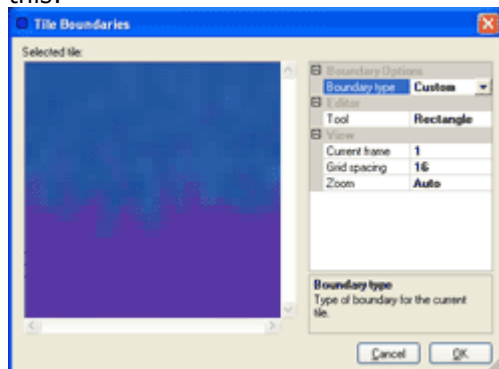


II. Setting up a tileset

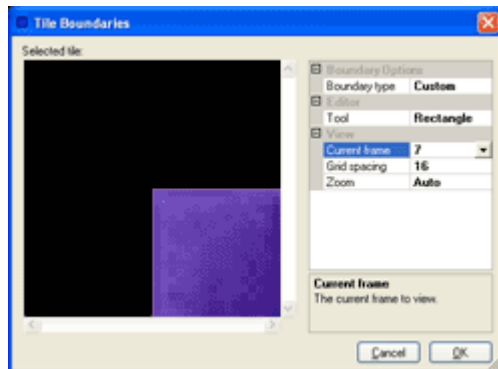
1. Click the **Tools** menu, and select **Tileset Editor**.
2. Click **Clear**. When it asks if you are sure, click **Yes**.
3. We want our grid to be 32x32 pixels, so make sure the **Grid with** and **height** both contain values of **32** (located at the bottom of the dialog).
4. Now we are going to add the tiles to the tileset. Click **Add**. Navigate to the installation folder (normally C:\Program Files\Platform Studio). Browse to the Graphics folder, and then the Tileset folder. In this tutorial, we are just going to set up the first two tiles manually so you can see how it's done. Select **blocks.png** and **bullet.png** (hold down Ctrl and click each file). Click **Open**. If Bullet comes before Blocks in the list of tiles, select the Blocks tile and click the Up arrow to move it upwards, or drag it to the top of the list.



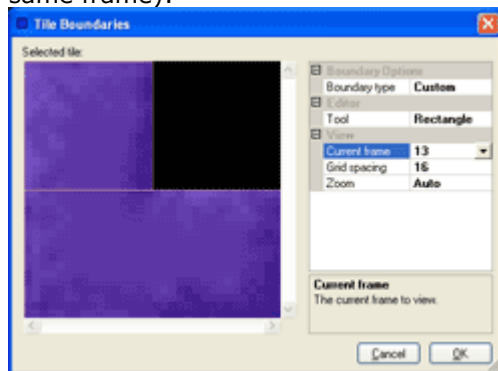
5. Now, we must configure the tiles in the tileset (their sizes, animations, behavior, etc.) First we are going to configure the transparent color. Since both tiles have magenta backgrounds, you can select both tiles at the same time to change both of their transparent colors. Click **Blocks** and then hold down **Ctrl** and click **Bullet**. Click the **Transparent color** property, the **down arrow**, and then select **Magenta** (the color in the third row on the far right).
6. Now, we need to configure the boundaries for the Blocks tile. This step is optional but it is highly recommended as it will lead to much greater performance in game. Click **Blocks**, and under **Boundaries**, click (...). Change the **Boundary type** to **Custom**. Now, by clicking with the mouse and dragging on the tile preview, draw a box that fills the solid areas of the tile. If you make a mistake, click what you have drawn and press Delete. You should now have something that looks like this:



Now, go to the next frame by double-clicking **Current frame**. Do the same thing, and repeat for each frame. For example, frame 7 should look like this:

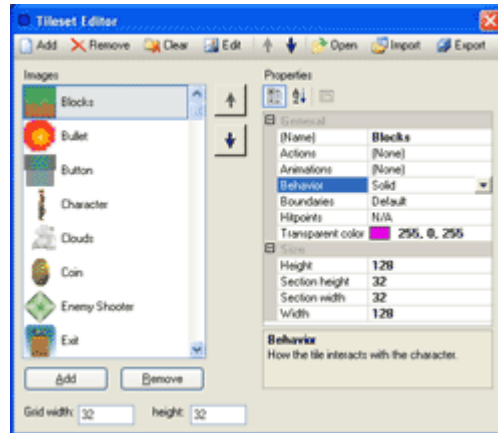


and frame 13 should look like this (yes, you can draw multiple shapes on the same frame):



Now click **OK**.

7. Since the Blocks tile does not need any more configuring, select the **Bullet** tile. Notice the strange preview next for this tile. That is because the section size is not properly configured. So, under **Section width**, type (or select). The preview should look much better now.
8. Now we are going to add an animation to the Bullet tile. Under **Animations**, click (...). Under **Animation**, select **Disappear**. Under **Start frame**, type **2**, and under **Stop frame**, type **4**. Under **Frame interval**, type **0.04**. Click **OK**. Notice how the "Animations" property now has a value of "Disappear", since we have added a disappear animation.
9. Since this tile is so small, we can just give it a Rectangular boundary. So, under **Boundaries**, click (...). Under **Boundary type**, select **Rectangular**. Or, if you wanted, you could try a circular boundary. Instead of selecting Rectangular, select **OneCustom**. Under **Tool**, select **Circle**, and under **Grid spacing**, type **8**. Now click and drag from the top-left corner to the bottom-right corner. Click **OK**.
10. The last thing we need to do for this tile is set up a different behavior. Under **Behavior**, click the **down arrow** and select (**Custom...**). In the "Customize Behavior" dialog, scroll up and select **Bullet**. The default values are fine so click **OK**.
11. Now that you've seen how to edit properties in the Tilesset Editor, lets add the rest of the tiles that we need quickly. Click **Import** (on the toolbar) and double-click **Tilesset.pst**. Now an "Import Tilesset" dialog should appear. We already have "Blocks" and "Bullet" so select all of the tiles from "Button" to "Water" by clicking **Button** and dragging the mouse downward until the rest of the tiles have been selected. Now click **OK**. The Tilesset Editor should now look something like this:



12. Finally, we have finished editing the tileset. Click the **X** button in the upper-right corner of the dialog. Your changes will be saved automatically.

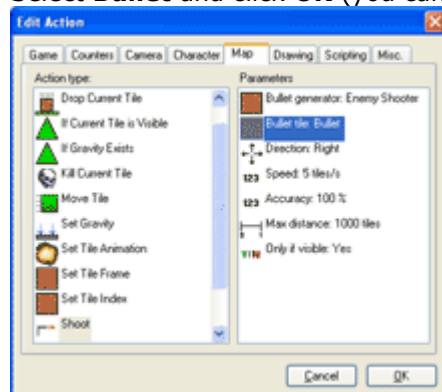
III. Action Editor

1. Click **Tools** and select **Action Editor**.
2. We are going to make a new timer for the shooting enemy. Click the **Timers** button on the toolbar (the button that looks like a clock). Click **Add**. Under **Name**, type **Shoot Timer**. Under **Interval**, type **1**.

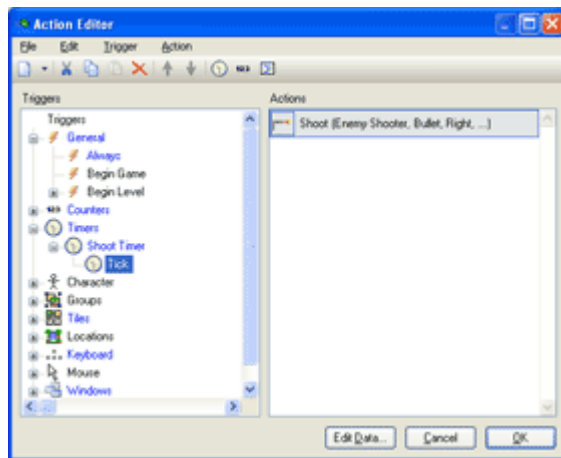


Click **OK**.

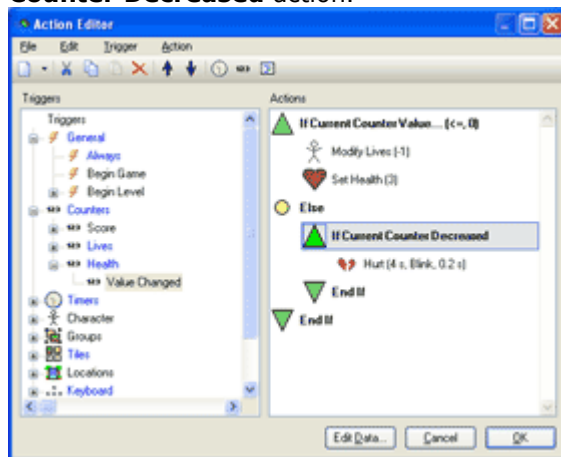
3. Now we are going to make a new action for the Shoot Timer. Under the **Triggers** portion of the Action Editor, double-click **Timers**, double-click **Shoot Timer**, and click **Tick**. On the Actions list to the right, right-click, select **New Action**, select **Map**, and then click **Shoot**. The Edit Action dialog should appear. Now look under **Parameters**. Double-click the first item (Bullet generator) and select **Enemy Shooter** in the dropdown box. Click **OK**. Double-click the next item (Bullet tile). Select **Bullet** and click **OK** (you can leave Frame blank).



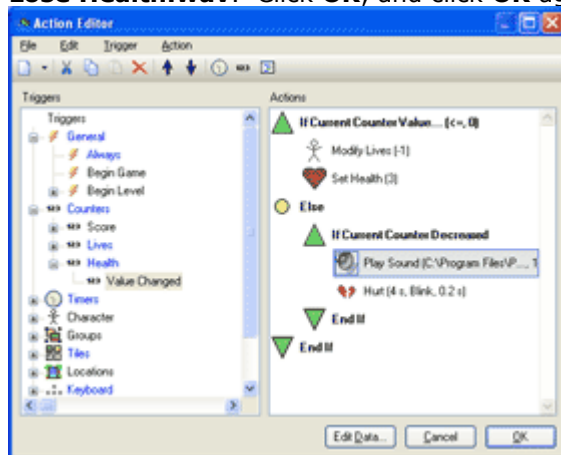
The other default values are fine so click **OK**.



4. Our next action will be to add the decrease health sound effect. Under the **Triggers** portion of the Action Editor, double-click **Counters**, double-click **Health**, and click **Value Changed**. On the actions list, select the **If Current Counter Decreased** action.



Right-click the selected action. Select **New Action**, **Miscellaneous**, and click **Play Sound**. The Edit Action dialog should appear. Under **Parameters**, double-click **Sound** and click (...). Navigate to the Sounds folder from the installation folder (default path is C:\Program Files\Platform Studio\Sounds), and double-click **Lose Health.wav**. Click **OK**, and click **OK** again.

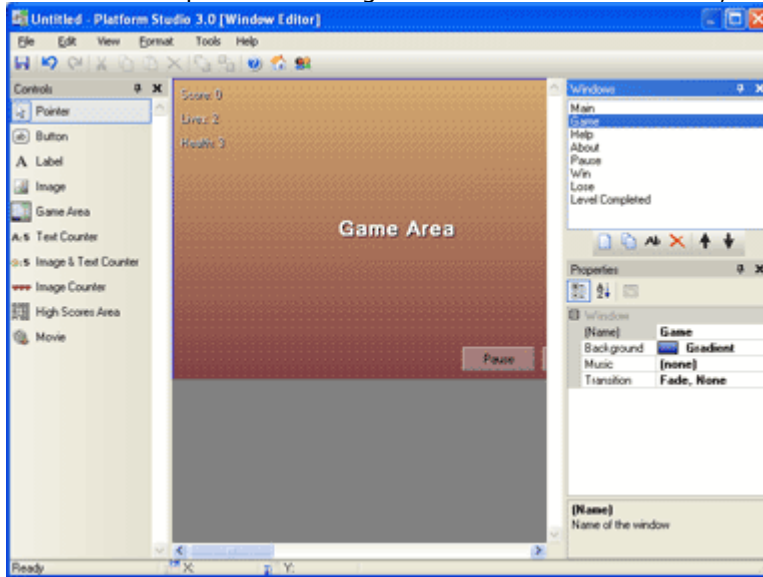


5. We are done with the Action Editor, so click **OK**.

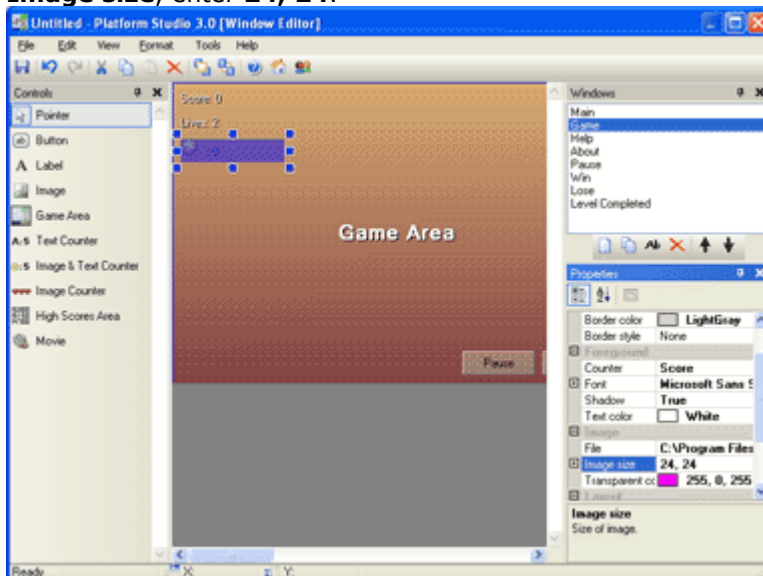
IV. Window Editor

1. Click **Tools** and select **Window Editor**.
2. We are going to create some different types of counters in the game window. In

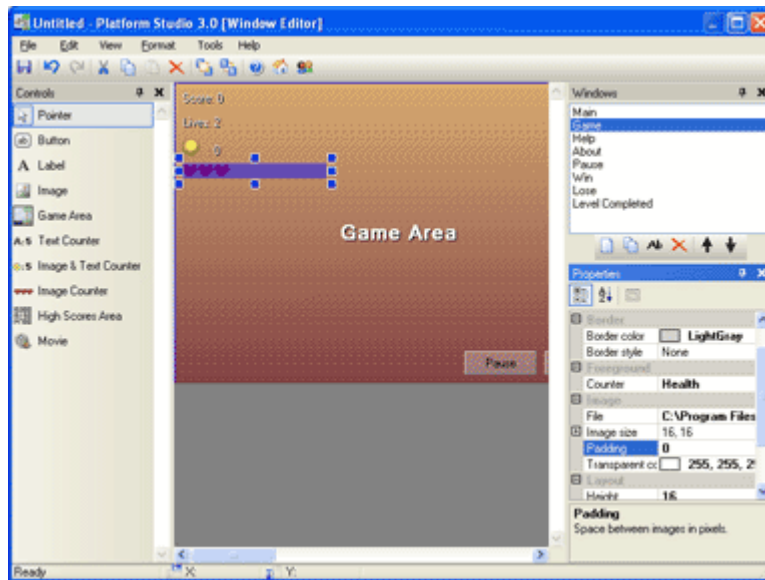
the **Windows** panel on the right side of the Window Editor, select **Game**.



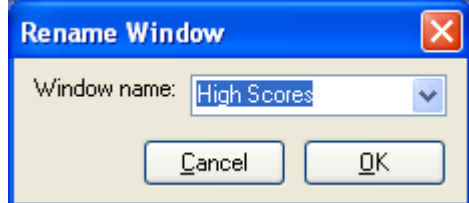
Delete the health counter (3rd control from the top). To delete a control, select it by clicking it with the mouse and press the delete key on your keyboard. In the **Controls** panel on the left side of the Window Editor, select **Image & Text Counter**. Where the health counter used to be, drag a new Image & Text Counter in its place. Select the new control. In the Properties panel, find the **Counter** property, and select **Coins**. Under **Shadow**, select **True**. Check **Shadow**. Under **File**, click the (...) button. Navigate to the installation folder (default is C:\Program Files\Platform Studio), browse to the Graphics folder, then Counter Icons folder, and double-click **coins.bmp**. Under **Transparent Color**, click the **down arrow** and select **Magenta** (third row on the far right). Under **Image size**, enter **24, 24**.



Now we are going to create the health counter. Select the **Image Counter** control, and place a new one under the coin counter. Look at the Properties panel. Under **Counter**, select **Health**. Under **File**, click (...), and browse to the installation folder, then go to Graphics, and then Counter Icons. Double-click **hearts.bmp**. Under **Transparent color**, click the **down arrow** and select **White** (the top-left box). Under **Padding**, enter **0**.

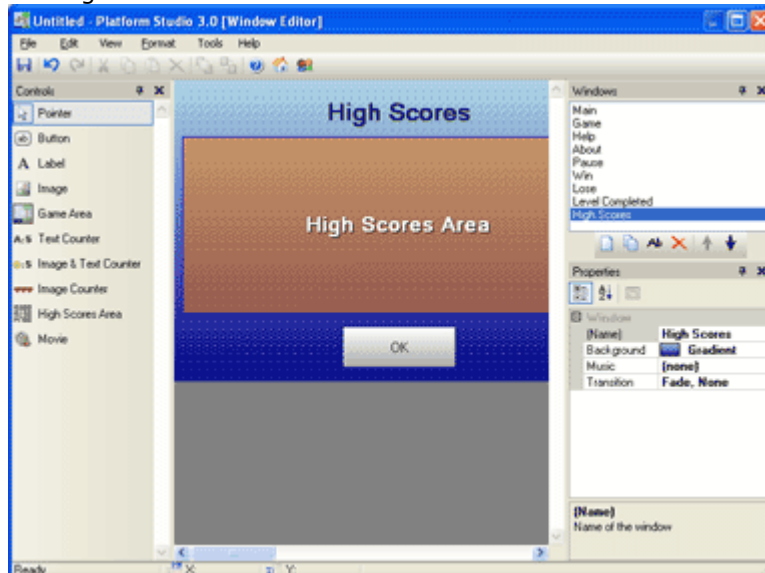


- The last thing we are going to do in the window editor is to create a High Score window. In the **Windows** panel on the right side of the Window Editor, select the **Help** window so we can use it as a template. On the toolbar below the list of windows, click the **Copy Window** button (second button from the left). Then click the **Rename Window** button (third button from the left). Under **Window Name**, click the dropdown arrow and select **High Scores**.



Click **OK**.

Click the **Help** label and change the text property to **High Scores**. Delete the second label (in the middle of the window). In its place, add a **High Scores Area** control. Move the top label upwards, move the OK button downwards, and resize the High Scores Area control so that there is more room to display high scores.



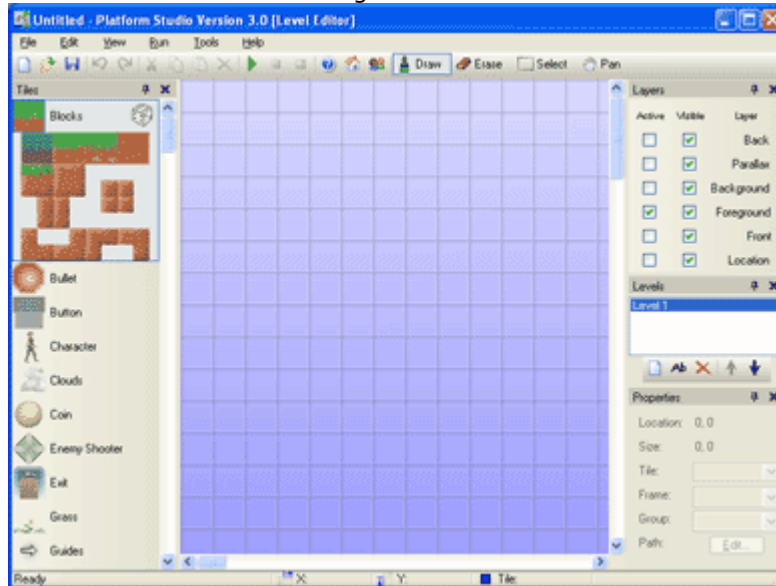
- Now we need to tell the OK button to bring us back to the main menu. Select the **OK** button, under the **Action** property, click (...). Right-click the action list, select **New Action**, select **Miscellaneous**, and click the **Show Window** near the top of the list. In the Edit Action dialog double-click **Window name** and select **Main**.

Then click **OK**, and then **OK** again.

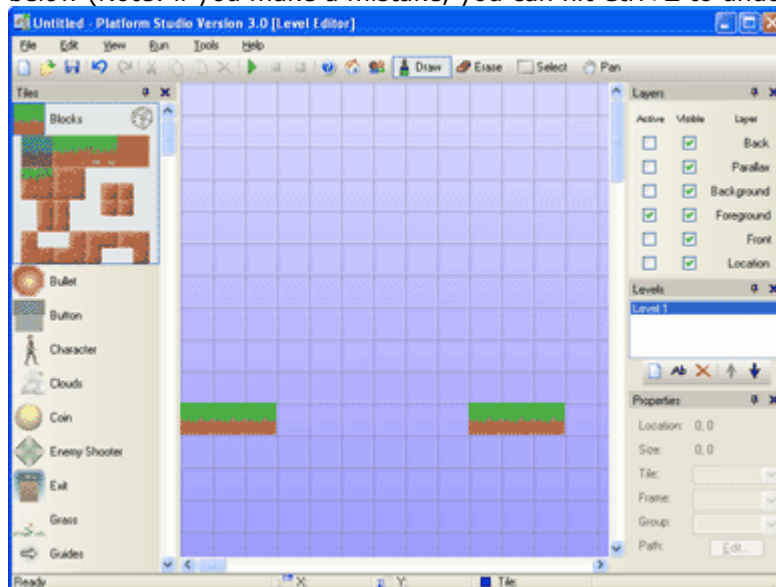
5. We are done with the window editor, so close it by clicking the **X** button on the top-right corner. Your changes will be saved automatically. (Note that this is only true for the Window Editor and the Tileset Editor).

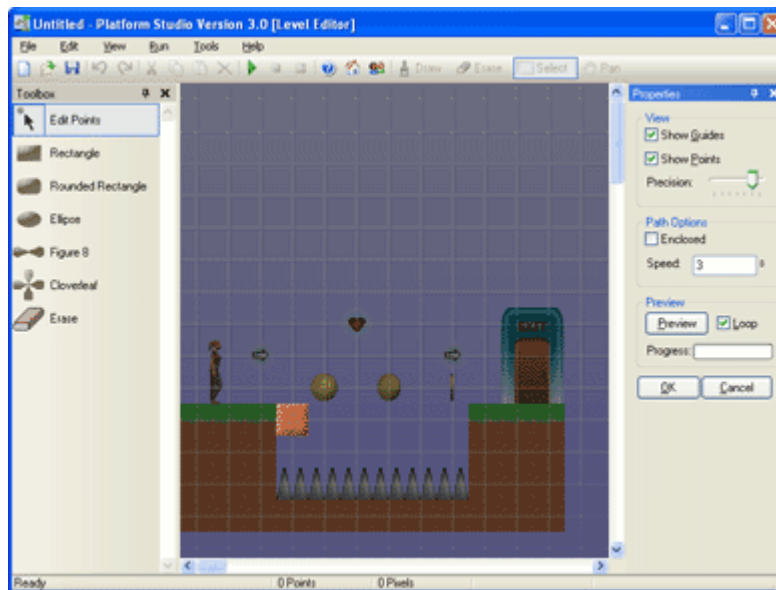
V. Level Editor

1. We should be in the level editor, which is always the top-level window. Your screen should look something like this:

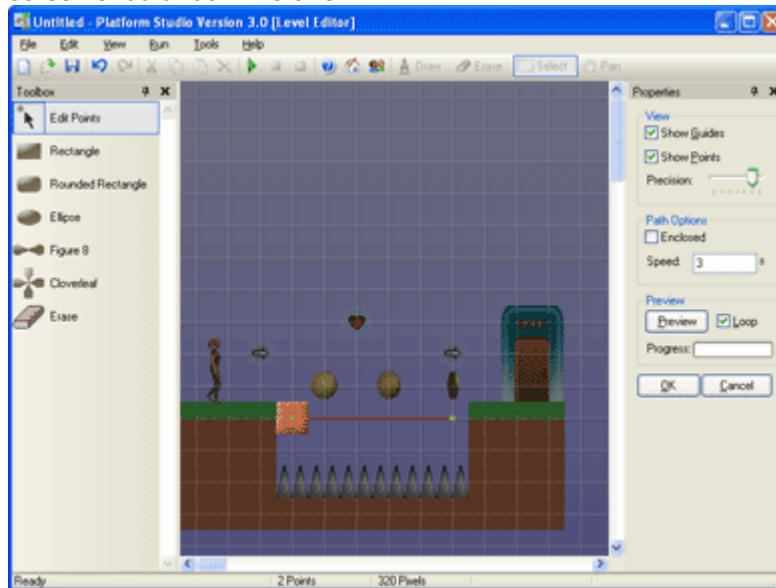


2. Notice that we are currently working in the Foreground layer by looking in the **Layers** panel on the right side of the Level Editor. To draw on a different layer, select the "Active" checkbox for the layer you want to draw on. You can change the state of the "Visible" checkboxes to show or hide layers. For now, leave the **Foreground** layer active.
3. To use the Tiles panel on the left side of the Level Editor, select the tile you want to draw with, and, if a subtitle selector is shown below the selected tile, click the subtitle you want to draw with. When you select a tile, you switch to Draw mode (shown at the toolbar on the top of the Level Editor). Right now, select the **Blocks** tile and the top-left subtitle. This is shown in the screenshot in step 1.
4. Now, begin drawing with the selected tile. Make it look like the screen shown below (Note: if you make a mistake, you can hit Ctrl+Z to undo it):



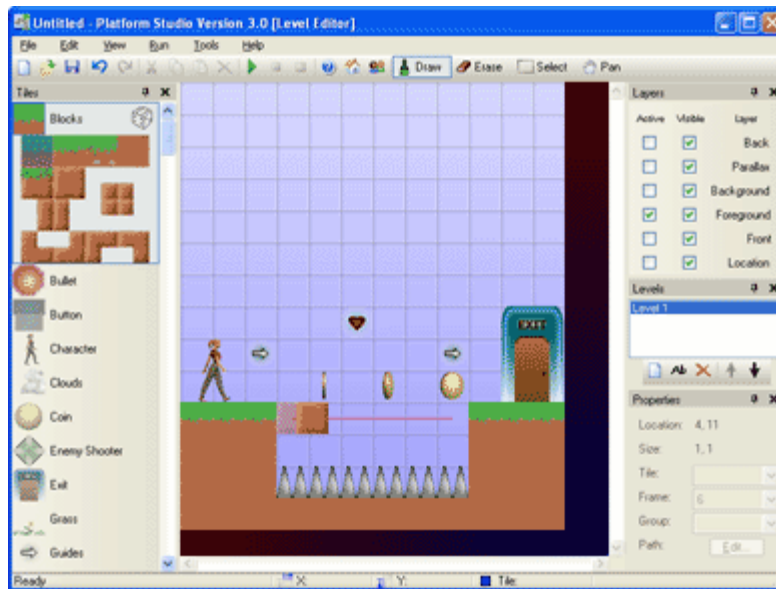


Notice how the interface changes. The Tiles panel is replaced with the Toolbox, the Layers and Levels panels are now gone, and the Properties panel is changed to show information about the current path. Also, the background is dimmed and the grid is moved to the centers of the tiles instead of their borders. Create a new point for the path by clicking in the center of the selected tile. Then, make another point by clicking in the spot a tile under the rightmost coin tile. Your screen should look like this:



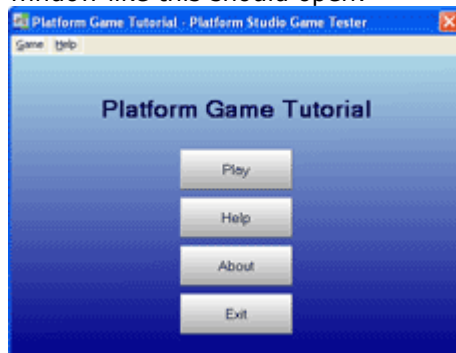
For this path, we do not need to modify the speed; the 3-second speed will work just fine. Click **Preview** to preview how your tile will move before you accept the changes. Then click **OK**.

8. The last thing we need to do with this test level is to resize it. To do this, hover your mouse over the bottom-right tile that you drew, and look at the coordinates on the status bar (at the bottom of the editor). The X coordinates should be 12 and the Y coordinate should be 14. Click **Edit** and select **Resize Level**. Under **Width**, type **12**, and under **Height**, type **14**. Click **OK**.



VI. Testing

1. First, save your project if you wish to. Click **File**, and choose **Save As**. Type in a name for your file and click **Save**.
2. To test your game, click **Run** and choose **Start**, or, click the green "play"-shaped button on the toolbar. It may take several seconds to compile. When it's done, a window like this should open:



3. Click **Play**.



4. Congratulations, you have just created a platform game!

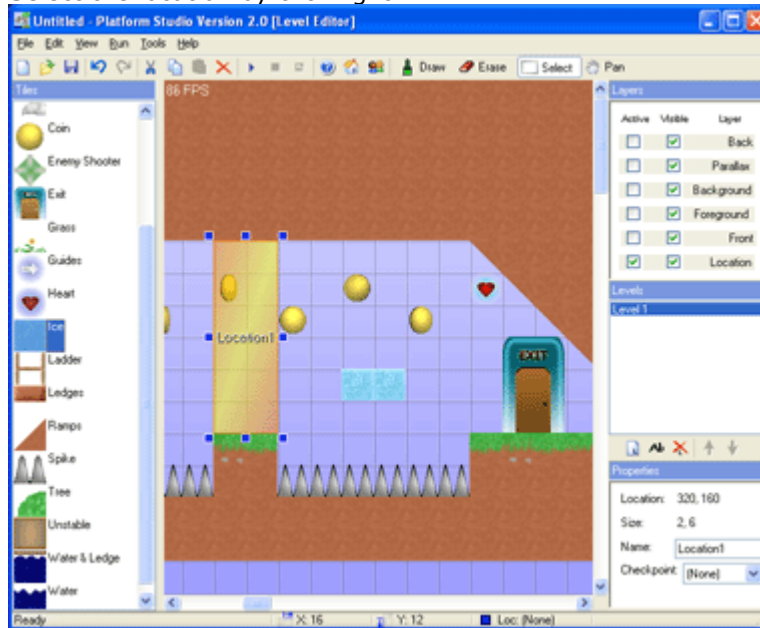


Copyright 2005, First Productions, Inc.

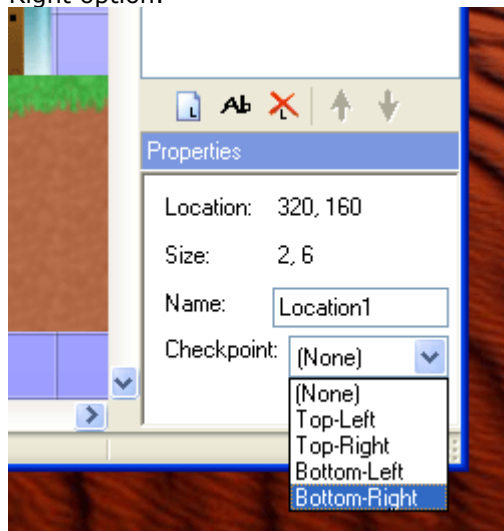
Platform Studio Help

How to Create a Checkpoint up

1. In the level editor, switch to the location layer.
2. Drag a new location to show where the player can be to activate the checkpoint.
3. Select the location by clicking it.



4. In the Properties panel on the bottom-right corner of the Level Editor, choose an option for **Checkpoint**. In this case (shown above), we would use the Bottom-Right option.



5. A preview of the checkpoint is shown on the location.

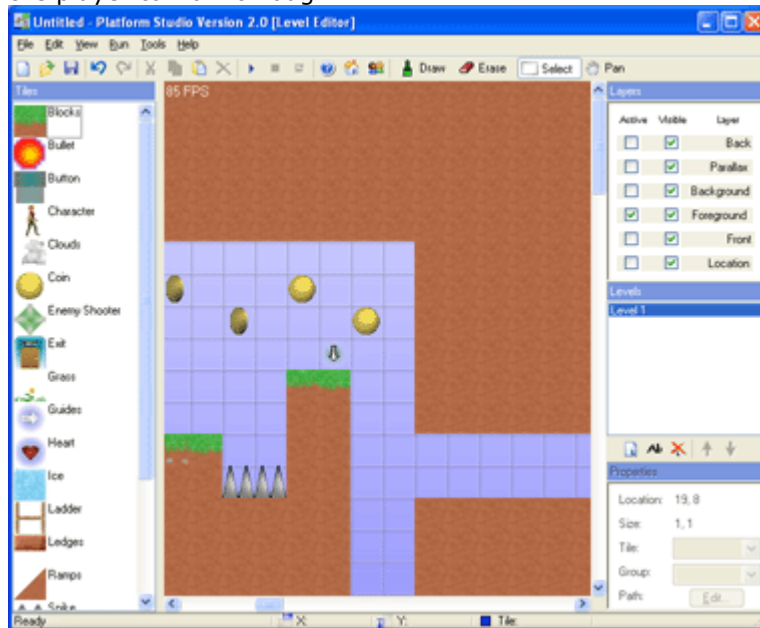


Copyright 2005, First Productions, Inc.

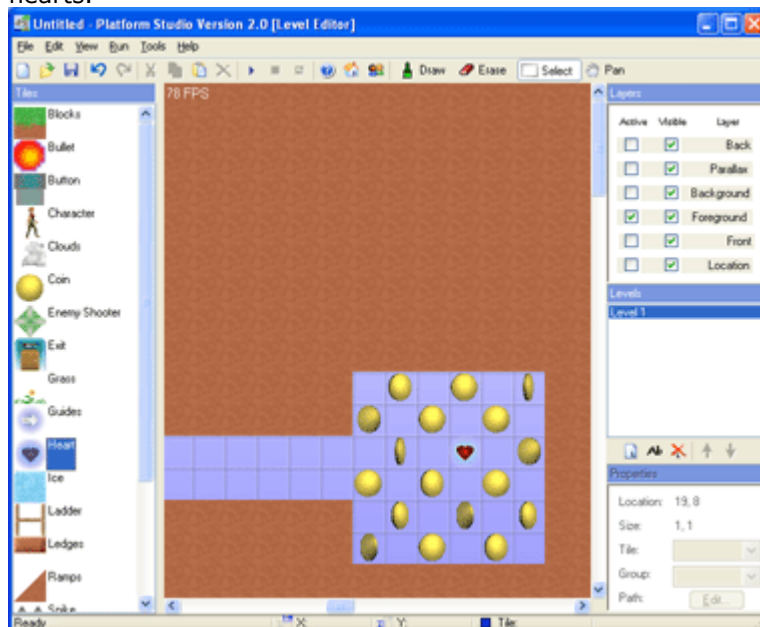
Platform Studio Help

How to Create a Secret Area up

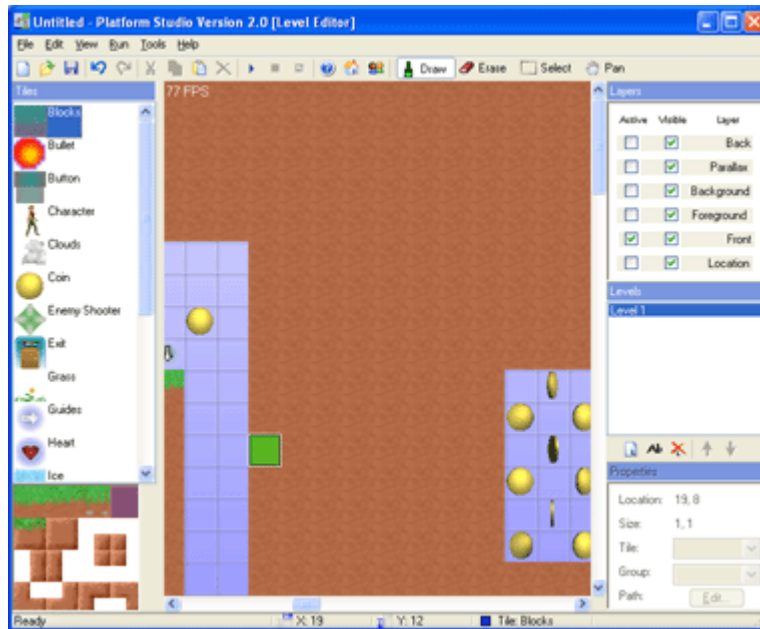
1. Using the foreground layer, create a wall with an opening the middle of it to allow the player to walk through.



2. Off of the path, create an open area filled with coins and, if you want, one or two hearts.



3. Fill up the path connecting the secret area with the rest of the level by drawing tiles in the Front layer.



4. Use a similar strategy to create other kinds of secret areas.

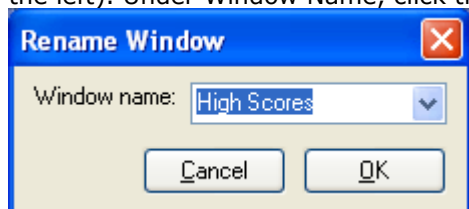


Copyright 2005, First Productions, Inc.

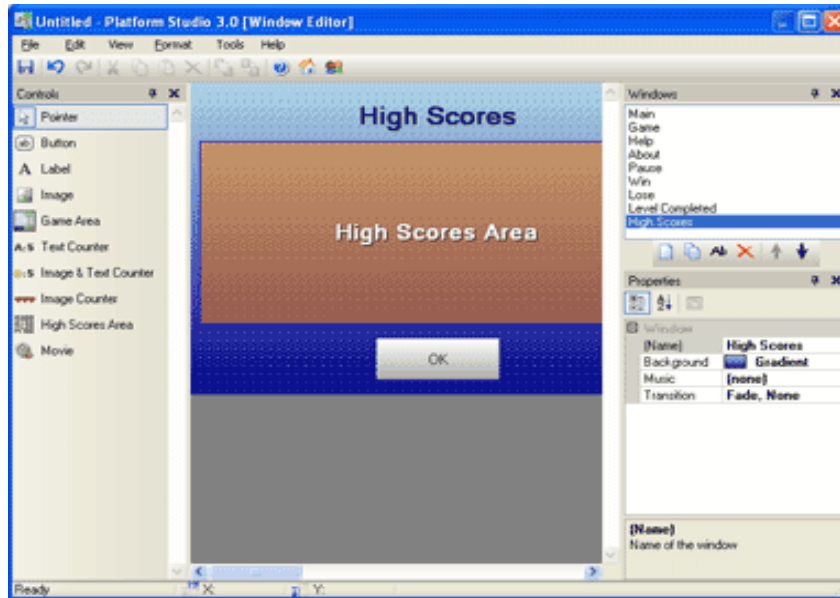


How to Create a High Scores List up

1. In the Windows panel of the Window Editor, select the Help window so we can use it as a template.
2. On the toolbar below the list of windows, click the Copy Window button (second button from the left). Then click the Rename Window button (third button from the left). Under Window Name, click the dropdown arrow and select High Scores.



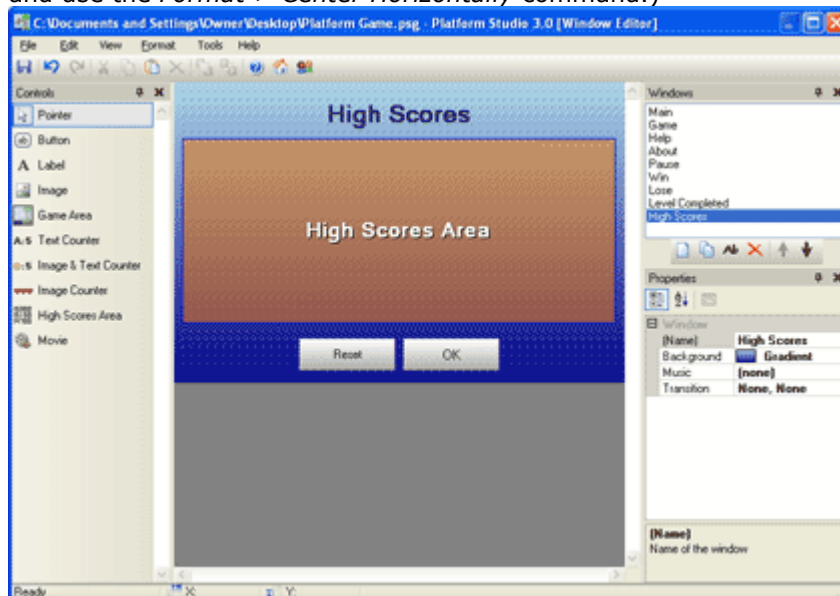
3. Click OK.
4. Select the Help label and change the *Text* property to "High Scores".
5. Delete the second label down (in the middle of the window). In its place, add a High Scores Area control.
6. Move the top label upwards, move the OK button downwards, and resize the High Scores Area control so that there is more room to display high scores.



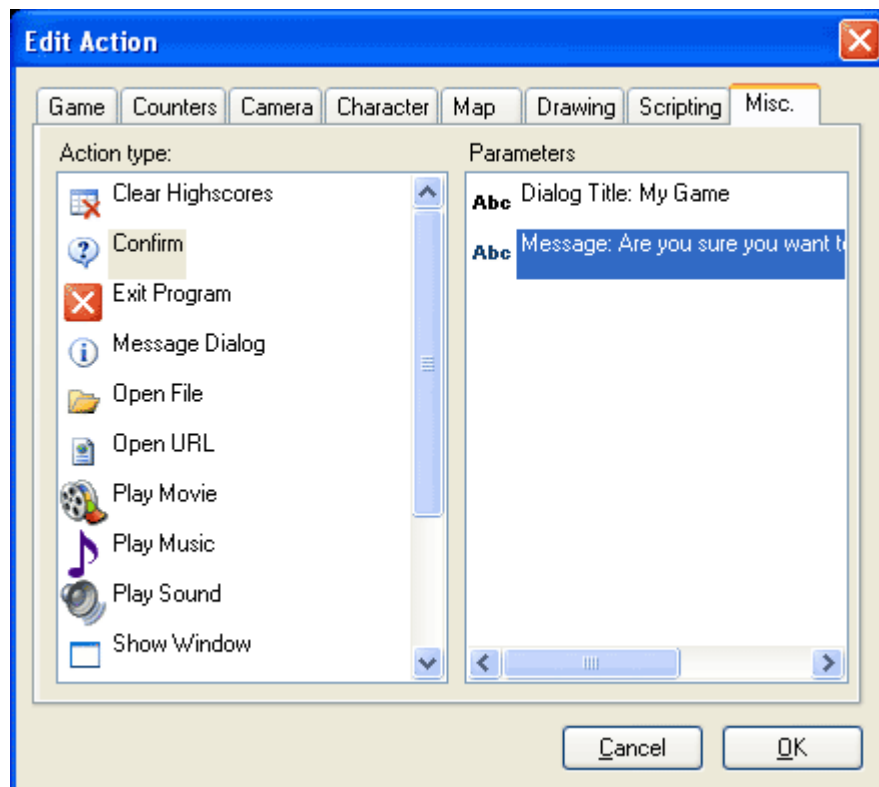
- Now, when you test your game and you get a score that should belong on the High Scores list, it will prompt you for your name after you finish the game and will advance to the window containing the high scores list.

To create a "Reset Scores" button:

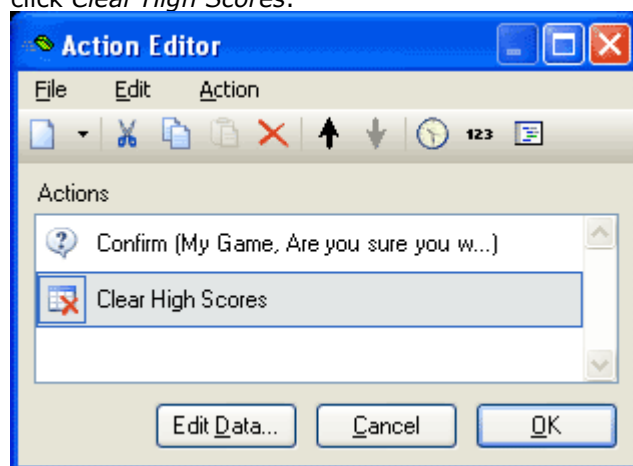
- Create a button, change the *Text* property to "Reset", and position it next to the OK button as shown below: (Hint: To center the two buttons, select them both and use the *Format > Center Horizontally* command.)



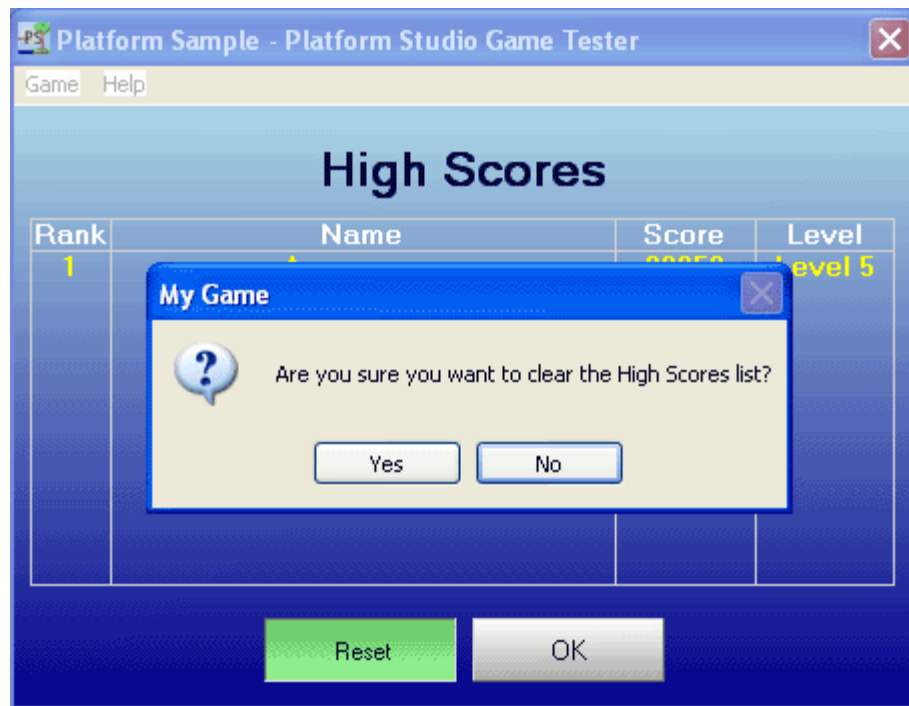
- Click the Reset button, click the *Action* property, and then click the (...) button. The Action Editor should pop up.
- Right-click the *Actions* list, highlight *New Action*, then *Miscellaneous*, and then click *Confirm*.
- Double-click *Dialog Title*. Enter the name of your game and press Enter.
- Double-click *Message*. Enter "Are you sure you want to clear the High Scores list?" and press Enter.



6. Click OK.
7. Right-click the *Actions* list, highlight *New Action*, then *Miscellaneous*, and then click *Clear High Scores*.



8. Click OK.
9. When you click the Reset button in your game, you should see the following:



Copyright 2005, First Productions, Inc.



How to Create a Shooting Enemy up

1. In the Tileset Editor, select the tile you want to turn into a shooting enemy. Under Behavior, click (...). Scroll down in the list to Enemy, to create a default enemy tile (jump on it to kill it; hit it to lose health). Customize any options if you want, and click OK.
2. If you do not have a bullet yet, select the tile you want to turn into a bullet. Under Behavior, click (...). Scroll down in the list to Bullet, customize any of the options if you want, and click OK.
3. Click Done in the Tileset Editor.
4. Click Tools > Action Editor.
5. Click Edit > Timers.
6. Click Add. Set the name to Shoot Timer and the interval to how long in seconds you want the shooter to wait before shooting. A good number for this is 1.5 or 2.
7. Click OK.
8. In the list of trigger on the left, select Timers > Shoot Timer > Tick.
9. Click Action > New Action > Map > Shoot. The Edit Action dialog will appear.
10. Under the Parameters area, double-click Bullet generator and select the shooting enemy tile. Under Bullet tile, select the tile you want to use as the bullet. Customize the other parameters if you want.
11. Click OK.
12. Close the Action Editor by clicking OK.

Copyright 2005, First Productions, Inc.



How to Speed Up Your Game up

Here are some tips on how to increase the performance of your game:

1. Whenever possible, use *Custom* or *OneCustom* boundaries for tiles. Using *Rectangular* boundaries will speed up your game the most.
 - Go to the *Tools* menu and select *Tileset Editor*. Under *Boundaries*, click (...).
2. Do not use large tiles. If you are using a tile that is much larger than the grid size, for example, a 256x256 tile on a 32x32 grid, Platform Studio has to do more work to ensure that tiles of this size are drawn when they are on the edge of the viewport. If possible, try to break these large tiles into sections or subtiles to increase speed.
 - Go to the *Tools* menu and select *Tileset Editor*. Look at the *Section Width* and *Section Height* properties.
3. Contrary to #2, if all of your tiles are large, increase the grid size instead of reducing the tile size. The larger the grid, the less tiles that will have to be drawn or examined, and the faster your game.
 - Go to the *Tools* menu and select *Tileset Editor*. Look at the *Grid size* option at the bottom of the window.
4. Using a high resolution, color depth, or window size can reduce performance.
 - To adjust these settings, go to the *File* menu and select *Game Properties*. Then click the *Windows* tab.
5. MP3s are usually faster to load the MIDI files.
 - Go to the *Edit* menu and select *Background Music*.
6. Very large images may take a long time to load, or may not load at all on older video cards. Generally the maximum image size you should use is 2048x2048 for compatibility.

These will only have a **negligible** effect on the overall speed of your game:

1. Large levels vs. small levels (except when loading).
2. Amount of tiles in your tileset.
3. Amount of actions in your game (except for large scripts or highly involved actions such as Shoot, but remember that playability greatly outweighs any small speed decreases).



Copyright 2005, First Productions, Inc.



Scripting Documentation up

Scripting Scope

Click [here](#) to learn about the two scopes of Platform Studio scripting. This includes procedure-level and class-level.

Scripting Objects

This is the reference for the different scripting objects and their members. Here is the list:

[Game](#)
[curCounter](#)
[curTimer](#)
[curGroup](#)

VB.NET Reference

Visit [Microsoft MSDN's website](#) for a thorough documentation of all VB.NET and .NET-related topics. This is a good place to go if you want to look up a specific method, keyword, object, or property.

Copyright 2005, First Productions, Inc.



Scripting Scope up

There are two types of scripting scopes, procedure-level and class-level. These can be accessed using the Procedure View and Class View tabs, respectively, in the Scripting Editor.

Procedure-level Scope

This is the most commonly used scope, because you can enter a different script for each action. The code you are entering is equivalent to the body of a function. For example, if you had a Tile Hit action for certain tile, the code would be as follows:

```
Function [FunctionName]()
    Procedure-level code goes here
End Function
```

Since Platform Studio automatically creates the Function...End Function code for you, you can see that the procedure-level is equivalent to the body of a sub or function.

Now, let's say you made a script in the score counter's "Value Changed" trigger. Then, the code for your function would be,

```
Function [FunctionName](ByVal curCounter As _
    PlatformStudio.CurrentCounter)
    'Procedure-level code goes here
End Function
```

The code you would type in this function would be the the same as the previous function, since you are only typing the body of the function. However, in this function, you can also make use of a parameter, which is curCounter. Even though you are not declaring curCounter, you can still make use of it because Platform Studio declares it for

you.

One final note about procedure-level: you cannot execute other triggers' functions because they are independent of each other. If you want to share code, put it in Class View.

Class-level Scope

The other type of scope is called Class-level. In class-level, you can write global code that can be used in any procedure in your game. All of this code is located in a class (whose definition is automatically created by Platform Studio). Therefore, if you type this,

```
Dim Game As PlatformStudio.psGame
```

Platform Studio will create this (which you will not see in Class View),

```
Class ScriptObject
    Dim Game As PlatformStudio.psGame
End Class
```

The Game object can also be used throughout your procedures in Procedure View.

Another way to take advantage of class-level scripting is to create global functions. For example, if you create a function that draws the number of points earned when the user collects a coin or heart, you can access it in both the Coin.Collect trigger and the Heart.Collect trigger.

Putting It All Together

The final output Platform Studio creates is a DLL, which combines all of your scripts, actions, and global code and puts it in a single class, to be compiled to a DLL file. If there is an error compiling the DLL, you will see the source code that Platform Studio generates from your actions and code. To familiarize yourself with it, your DLL's code might look something like this:

```
Imports System
Imports System.Drawing
Imports System.Windows.Forms
Imports Microsoft.VisualBasic

Namespace Script
    Class ScriptObject
        'This code imports the current game object
        'so it can be used through scripting.
        'The Game object is set by the game player.
        Public Game As PlatformStudio.IGame

        'Trigger actions
        '-----
        Function cval2(ByVal curCounter As PlatformStudio.CurrentCounter)
            If curCounter.Value < 0 Then
                Game.QuitGame()
            End If
        End Function

        Function cval3(ByVal curCounter As PlatformStudio.CurrentCounter)
            If curCounter.Value <= 0 Then
                Game.ModifyLives(-1)
                Game.SetHealth(3)
            Else
                If curCounter.Decreased Then
                    Game.Hurt(4, "Blink", 0.2)
                End If
            End If
        End Function
    End Class
End Namespace
```



```

        End If
    End If
End Function

Function khol37()
    Game.MoveCharacter("Left", 4, True)
End Function

Function khol39()
    Game.MoveCharacter("Right", 4, True)
End Function

Function khol38()
    Game.Jump(0.6, 0.25, 1)
End Function

Function krel38()
    Game.StopJumping()
End Function

Function kpre27()
    Game.QuitGame()
End Function
End Class
End Namespace

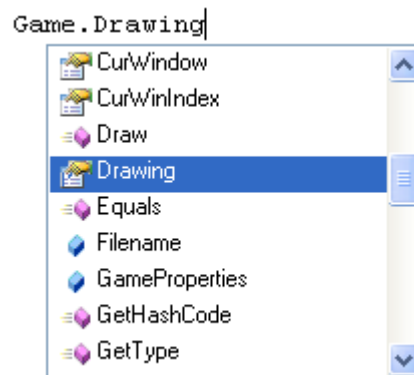
```

Copyright 2005, First Productions, Inc.



Scripting Object: Game [up](#)

The Game object's members can be accessed by simply typing `Game.` in the scripting editor. When you type `Game.`, the Scripting Editor pops up an AutoComplete list which shows all accessible PlatformStudio.psGame members. Here is an example:



The Game object provides access to both high-level and lower-level aspects of the game. It lets you access the DirectX 9 drawing system, properties for a control or a window, as well as high-level functions for a group of tiles, and much more.

Here are some documented methods and properties of the Game object:

Game.AutoUpdateCamera

Specify whether you want the camera to automatically update each frame. If this is turned off, you must move the camera manually using Game.Camera. This is useful for creating non-platform games such as scrolling shoot-'em-up games.

Game.Camera

This gets the current viewport, which is set automatically by Platform Studio if Game.AutoUpdateCamera is set to True. If Game.AutoUpdateCamera is set to False, you can use this property to change the camera location.

Game.CharacterX

Gets the X coordinate of the character in pixels.

Game.CharacterY

Gets the Y coordinate of the character in pixels.

Game.CharacterW

Gets the width of the character in pixels.

Note: The character width and height are not including any extra transparent pixels around the character, so it may differ from the character tile size.

Game.CharacterH

Gets the height of the character in pixels.

Game.CharAffectedByGravity

Returns True if the character is affected by gravity, or False if the character is not affected by gravity.

Game.CharFacingRight

Returns True if the character is facing right, or False if the character is facing left.

Game.CharFalling

Returns True if the character is falling, or False if the character is not falling.

Game.CharJumping

Returns True if the character is jumping, or False if the character is not jumping.

Game.ClearHighscores

Clears all high scores in the high scores list.

Game.Compatibility

Returns an object that contains methods that were removed or changed in newer versions of Platform Studio. This should only be used when converting games created with older versions of Platform Studio to newer versions.

Game.Confirm

Prompts the user with a question. Returns True if the user clicks Yes and False if the user clicks No. This is preferred over the use of MsgBox or MessageBox.Show because it allows the game player to temporarily pause the game while the dialog is being shown.

Game.ConvPixelsToTilesX

Converts a pixel X coordinate to a tile X coordinate.

Game.ConvPixelsToTilesY

Converts a pixel Ycoordinate to a tile Ycoordinate.

Game.ConvTilesToPixelsX

Converts a tile X coordinate to a pixel X coordinate.

Game.ConvTilesToPixelsY

Converts a tile Y coordinate to a pixel Y coordinate.

Game.DisintegrateCurrentTile

Disintegrates the current tile (only usable in a Tile or Group action) after a specified period of time. The disintegrate action works with the Disappear animation.

Game.DrawAnimationAtCurrentTile

Draws an animation at the current tile's location. This is useful if you want a custom explosion for a particular tile.

Game.DrawHealthBarForCurrentTile

Draws a health bar next to the current tile. In conjunction with the Mouse Hover trigger, this is useful to let the player roll over his or her mouse over a tile to view its health.

Game.Drawing

Provides access to the Drawing class. The Drawing class contains methods for loading images and fonts, and drawing points, lines, rectangles, sprites, and text using DirectX 9.

Game.DropCurrentTile

Similar to Game.DisintegrateCurrentTile, except the tile drops before it disintegrates.

Game.EnsureCharacterIsVisible

Ensures that the character tile is completely in view. In auto-scrolling games, this is useful to prevent the player from moving his character too far forward (or backward) to leave the action.

Game.ExecuteTrigger

Executes another trigger. To access the list of triggers, type "." after "Game.ExecuteTrigger". For more information on triggers, see [this page](#).

Game.ExecuteTriggerString

Executes another trigger, such as a tile hit trigger, specified as an action code. For more information on action codes, see [this page](#).

Game.ExitProgram

Exits the application containing the game. Has a similar result as clicking the X button at the top-right corner of the game window.

Game.FPS

Returns the current frames per second as a decimal value.

Game.GetControl

Provides access to a specified control's properties.

Game.GetCounter

Gets the value of the specified counter.

Game.GetHealth

Gets the health of the character.

Game.GetLevel

Gets the current level number.

Game.GetLevelName

Gets the current level name.

Game.GetLives

Gets the number of lives the player has left.

Game.GetScore

Gets the player's score.

Game.GetTileIndex

Retrieves the index of a specified tile on the foreground layer of the current level.

Game.GetTileName

Retrieves the name of a specified tile on the foreground layer of the current level.

Game.GetWindow

Provides access to a specified window's properties.

Game.GravityExists

Returns True if gravity exists, or False if gravity doesn't exist. (Same as Game.CharAffectedByGravity).

Game.Hurt

Provides invincibility to the character for a specified period of time, and shows a visual effect during the period of invincibility. This method is usually called after the health counter decreases.

Game.Jump

Makes the character jump.

Game.KillCurrentTile

Destroys the current tile. This calls the Tile Destroyed and Group Destroyed triggers.

Game.KillTile

Destroys a specified group. This calls the Tile Destroyed and Group Destroyed triggers.

Game.Lose

Ends the game as a loss, and shows the Lose window.

Game.Math

Provides access to a small math library that contains collision functions and other miscellaneous functions.

Game.ModifyCounter

Modifies a specified counter by a specified value.

Game.ModifyHealth

Modifies the player's health by a specified value (positive numbers increase health, and negative numbers decrease health). If the player is currently invincible, this method will have no effect if the value is less than zero.

Game.ModifyLevel

Modifies the current level by a specified value. This method is usually used with the value 1 when the player reaches the end of a level.

Game.ModifyLives

Modifies the player's lives by a specified value.

Game.ModifyScore

Modify the player's score.

Game.MouseLButton

Return True if the left mouse button is pressed, or False if the left mouse button is not pressed.

Game.MouseMButton

Return True if the middle mouse button is pressed, or False if the middle mouse button is not pressed.

Game.MouseRButton

Return True if the right mouse button is pressed, or False if the right mouse button is not pressed.

Game.MouseX

Returns the X coordinate of the cursor in relation to the game window. The upper-left corner of the game window is (0, 0), and the bottom-right corner is (window width - 1, window height - 1).

Game.MouseY

Returns the Y coordinate of the cursor in relation to the game window.

Game.MoveCameraToDefaultPosition

When AutoUpdateCamera is set to False, this moves the camera to the position it would be at had AutoUpdateCamera been turned on. This is useful at the beginning of the level or when the player loses a life, as it resets the character position.

Game.MoveCharacter

Moves the character in the specified direction. Use the strings "Left", "Right", "Up", "Down", or "Current Direction" as the direction parameter.

Game.MoveTile

Moves a tile or a group of tiles in the specified direction. Use the strings "Left", "Right", "Up", or "Down" as the direction parameter. Note: If the tile is currently on a path, the path will be deleted and the tile will start moving from its default position.

Game.MoveToCheckpoint

Moves the character to the checkpoint that was last set. If no checkpoint was set, the character will move to its starting location for the current level. If necessary, the game will pause for the camera to scroll to its new position.

Game.OpenFile

Opens a specified file with its default associated application.

Game.OpenURL

Opens a URL in the user's default web browser. If a "mailto:" URL is given, opens the default email application with the specified address in the To field.

Game.PauseGame

Pauses the game, and shows the Pause window.

Game.PlayMovie

Plays a movie at a certain point during the game. This movie has the same behavior as a movie control in the Window Editor.

Game.PlayMusic

Plays a specified music file. At the end of the song, Platform Studio will loop back to the

beginning.

Game.PlaySound

Plays a specified sound file. Use 0 (mute) to 100 (loudest) as values for Volume, -100 (left speaker) to 100 (right speaker) as values for Pan, and 11 (low pitch) to 44 (high pitch) as values for frequency. The default values are 100, 0, and 22 respectively.

Game.Properties

Retrieves the general properties for the game, such as its name, its version, its credits, etc.

Game.QuitGame

Quits the game and shows the window that was shown before starting the game (usually the menu window). If the player got a high score, it will first prompt for a name (if necessary), and then show the high scores window, before returning to the menu.

Game.Scroll

Scrolls the camera in a specified direction and speed. Use this only if AutoUpdateCamera is turned off.

Game.SetCheckpoint

Sets a checkpoint either at the current location of the character (`SetCheckpoint()`), or a specified (X, Y) tile coordinate (`SetCheckpoint(X, Y)`).

Game.SetClimbingState

Sets whether the character can be climbing or not.

Game.SetCounter

Sets a specified counter to a specified value. If a Counter action is triggered, the `curCounter.Increased`, `curCounter.Decreased`, and `curCounter.ChangedBy` values will be set to False, False, and 0, respectively.

Game.SetGravity

Sets whether gravity should exist or not, and the gravity intensity (ScaleSpeed).

Game.SetHealth

Sets the health of the character. to a specified value

Game.SetLevel

Sets the current level number.

Game.SetLives

Sets the amount of lives the player has left.

Game.SetPosition

Sets the position of the character in tiles. If necessary, the game will pause for the camera to scroll to its new position.

Game.SetScore

Sets the player's score.

Game.SetTileAnimation

Sets the current animation for a specified tile or group of tiles. To specify the current group (in this method or in any method that calls for a group), use "(Current Group)". You can either specify a number (0 to 5), or a string, ("Normal", "Disappear", "Custom 1", "Custom 2", "Custom 3", or "Custom 4").

Game.SetTileAtLocation

Sets the tile index, frame index, and/or animation index at a specific x-y location (in tiles). Another version of this method exists that accepts a layer as a numerical value (Parallax-0, Background-1, Foreground-2, Front-3).

Game.SetTileFrame

Sets the frame for a specified tile or group of tiles.

Game.SetTileIndex

Sets the tile index for a specified tile or group of tiles. The tile name is the name of a tile in the tileset. You can also specify the tile's frame, animation index, or any combination of them. For example, you can specify both a tile's frame and animation index by typing

```
Game.SetTileIndex("(Current Group)", , 7, "Custom 1")
```

Game.Shoot

Shoots a tile from a specified tile name. Use "Left", "Right", "Up", "Down", "Up Left", "Up Right", "Down Left", "Down Right", "Towards Character", or a number specifying an angle in degrees as values for Direction.

Game.ShootAnythingFromGroup

This action is almost the same as Game.Shoot except that it also can damage tiles with hitpoints. Hence its name, "shoot anything".

Game.ShootFromCharacter

Shoots a tile from the character. For the direction value, you can specify the 8 directions listed in Game.Shoot, as well as "Towards Cursor" and "Current Direction". Towards Cursor makes the character aim towards the mouse pointer, and Current Direction makes the character shoot in its current direction (left or right). You can also specify an angle in degrees, with 0 being right, 90 being up, 180 being left, and 270 being down. Also, the offset coordinates are relative to the center of the character. This is useful for allowing the character to fire two bullets at the same time. Finally, the "Fire Rate" parameter is the amount of seconds between shots. If this method is called before the next shot is allowed, this method will return False and do nothing; otherwise, it will return True.

Game.ShootFromGroup

Shoots a tile from a specified tile group.

Game.ShootFromPoint

Shoots a tile from a specified point. You must specify the bullet's starting left edge and top edge in pixels.

Game.ShowMessageBox

Shows a message dialog with a specified title, message, and icon. The icon can contain a value of "Error", "Information", "Question", or "Exclamation". This is preferred over the use of MsgBox or MessageBox.Show because it allows the game player to temporarily pause the game while the dialog is being shown.

Game.ShowWindow

Shows a specified window.

Game.StartTimer

Resets and starts a specified timer.

Game.StopAllSounds

Stops all sounds that are currently playing.

Game.StopJumping

Stops the character from continuing to jump. However, the character must jump for the minimum specified time before it can stop.

Game.StopTimer

Stops a specified timer.

Game.TimeElapsed

Returns the time elapsed between the last two frames in seconds.

Game.Win

Ends the game as a win, and shows the Win window.

Copyright 2005, First Productions, Inc.



Trigger Codes [up](#)

List of Trigger Codes

Explanation: A trigger code is an abbreviated form of a trigger's name and its parameters. Naming triggers always follow the rule:

category code (1 character) + trigger name code (3 characters) + trigger parameters (any combination of letters and numbers)

Here is a table that will help you create a trigger code:

Trigger Name	Code	Parameters
General > Always	aalw*	(none)
General > Begin Game	nbgm	(none)
General > Begin Level (for any level)	nblv	(none)
General > Begin Level (for a specific level)	nblv	Level (5 digits) + _
Counters > Value Changed	cval	Counter index (number)
Timers > Tick	itic	Timer index (number)
Character > Jump	hjum	(none)
Character > Hit Head while Jumping	hstj ¹	(none)
Character > Fall	hfal	(none)
Character > Land	hlan	(none)
Groups > Hit	ghit	Level (5 digits) + Group name
Groups > Hit Left Edge	glef	Level (5 digits) + Group name
Groups > Hit Right Edge	grig	Level (5 digits) + Group name
Groups > Hit Top Edge	gtop	Level (5 digits) + Group name
Groups > Hit Bottom Edge	gbot	Level (5 digits) + Group name
Groups > Exclusively on Top	gexc	Level (5 digits) + Group name
Groups > Collect	gcol	Level (5 digits) + Group name
Groups > Shot	gsho	Level (5 digits) + Group name
Groups > Destroyed	gdes	Level (5 digits) + Group name

Groups > Change Directions	gcha	Level (5 digits) + Group name
Groups > Ch. Dir. > To Forward	gctf	Level (5 digits) + Group name
Groups > Ch. Dir. > To Backward	gctb	Level (5 digits) + Group name
Groups > Hover	ghov	Level (5 digits) + Group name
Groups > Click	gcli	Level (5 digits) + Group name
Tiles > Hit	thit	Tile index
Tiles > Hit Left Edge	tlef	Tile index
Tiles > Hit Right Edge	trig	Tile index
Tiles > Hit Top Edge	ttop	Tile index
Tiles > Hit Bottom Edge	tbot	Tile index
Tiles > Exclusively on Top	texc	Tile index
Tiles > Collect	tcol	Tile index
Tiles > Shot	tsho	Tile index
Tiles > Destroyed	tdes	Tile index
Tiles > Change Directions	tcha	Tile index
Tiles > Ch. Dir. > To Forward	tctf	Tile index
Tiles > Ch. Dir. > To Backward	tctb	Tile index
Tiles > Hover	thov	Tile index
Tiles > Click	tcli	Tile index
Locations > Enter	lent	Level (5 digits) + Location name
Locations > Exit	lexi	Level (5 digits) + Location name
Locations > Inside	lins	Level (5 digits) + Location name
Keyboard > Press Key	kpre	Key index (number)
Keyboard > Release Key	krel	Key index (number)
Keyboard > Hold Key	khol	Key index (number)
Mouse > Left Button Press	mlpr	(none)
Mouse > Left Button Release	mlre	(none)
Mouse > Left Button Hold	mlho	(none)
Mouse > Right Button Press	mrpr	(none)
Mouse > Right Button Release	mrre	(none)
Mouse > Right Button Hold	mrho	(none)
Mouse > Middle Button Press	mmpr	(none)
Mouse > Middle Button Release	mmre	(none)
Mouse > Middle Button Hold	mmho	(none)
Mouse > Move Mouse	mmov	(none)

Notes

*The "a" prefix on the Always action code remains for compatibility purposes only.

¹stj stands for **stop** jumping.

Here are some examples:

Code	Description
aalw	Always
cval1	Score counter changed
itic3	Timer (3) tick
ghit00001Button	Hit "Button" group on level 1
thit5	Hit tile index 5
lent00012Chkpt1	Enter the "Chkpt1" location on level 12
krel38	Release the up arrow key
mmov	Move the mouse
nblv00007_	Level 7 began

For more information on key codes:

Key Code	Key Name
8	Backspace
13	Enter
16	Shift
17	Ctrl
18	Alt
19	Print Screen
27	Escape
32	Space
33	Page Up
34	Page Down
35	End
36	Home
37	Left Arrow
38	Up Arrow
39	Right Arrow
40	Down Arrow
45	Insert
46	Delete
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y

90	Z
96	Numpad 0
97	Numpad 1
98	Numpad 2
99	Numpad 3
100	Numpad 4
101	Numpad 5
102	Numpad 6
103	Numpad 7
104	Numpad 8
105	Numpad 9
106	Numpad *
107	Numpad +
109	Numpad -
110	Numpad .
111	Numpad /
112	F1
113	F2
114	F3
115	F4
116	F5
117	F6
118	F7
119	F8
120	F9
121	F10
122	F11
123	F12
186	;
187	=
188	,
189	-
190	.
191	/
192	`
219	[
220	\
221]
222	'

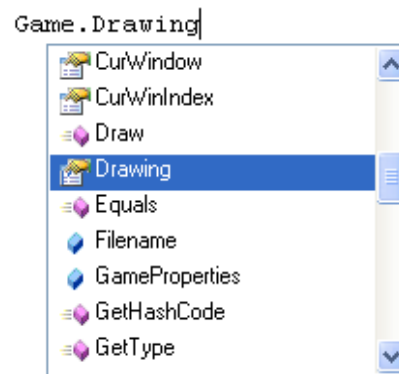
Copyright 2005, First Productions, Inc.



Scripting Object: `curCounter` up

The current counter object's members can be accessed by simply typing `curCounter.` in the scripting editor. When you type `curCounter.`, the Scripting Editor pops up an

AutoComplete list which shows all accessible PlatformStudio.CurrentCounter members.
Here is an example (using `Game.`):



In addition to its inherited members, here are the members of `curCounter`:

ChangedBy
Decreased
Increased
Name
Value

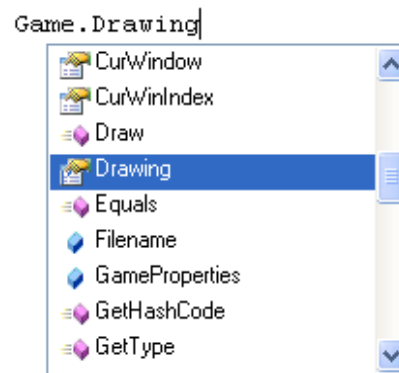
Note: The Increased and Decreased properties are only set to True if the counter was modified (`[Counter].Modify(Value)`), not set to a certain value (`[Counter].Value = Value`). Also, keep in mind that using Not Increased doesn't necessarily mean Decreased, and Not Decreased doesn't necessarily mean Increased.

Copyright 2005, First Productions, Inc.



Scripting Object: `curTimer` [up](#)

The current timer object's members can be accessed by simply typing `curTimer.` in the scripting editor. When you type `curTimer.`, the Scripting Editor pops up an AutoComplete list which shows all accessible PlatformStudio.CurrentTimer members.
Here is an example (using `Game.`):



In addition to its inherited members, here are the members of `curTimer`:

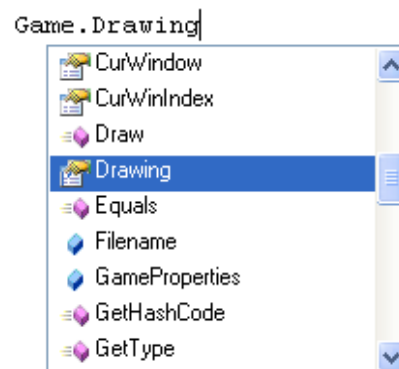
Enabled
Interval
Name

Copyright 2005, First Productions, Inc.



Scripting Object: `curGroup` up

The current group object's members can be accessed by simply typing `curGroup.` in the scripting editor. When you type `curGroup.`, the Scripting Editor pops up an AutoComplete list which shows all accessible `PlatformStudio.CurrentGroup` members. Here is an example (using `Game.`):



In addition to its inherited members from `System.ValueType`, here are the members of `curGroup`:

Name (retrieves the name of the group)
`curTileX` (retrieves the x coordinate of the current tile)
`curTileY` (retrieves the y coordinate of the current tile)

curTileW (retrieves the width of the current tile)
curTileH (retrieves the height of the current tile)
CurTileHP (retrieves the hitpoints of the current tile)
CurTileVisible (returns if the current tile is visible or not)

Copyright 2005, First Productions, Inc.



Windows ^{up}

Editors

These are the five main editors in Platform Studio:

[Level Editor](#)
[Path Editor](#)
[Tileset Editor](#)
[Action Editor](#)
[Window Editor](#)

Dialogs

These are individual dialogs found throughout the program:

Level Editor

[New Game Wizard](#)
[Game Properties](#)
[Find/Replace Tiles](#)
[Resize Level](#)
[Level Background](#)
[Background Music](#)
[Options](#)

Tileset Editor

[Customize Behavior](#)
[Animation Editor](#)
[Tile Boundaries Editor](#)

Action Editor

[Edit Action](#)
[Script Editor](#)
[Edit Timers](#)
[Edit Counters](#)

Window Editor

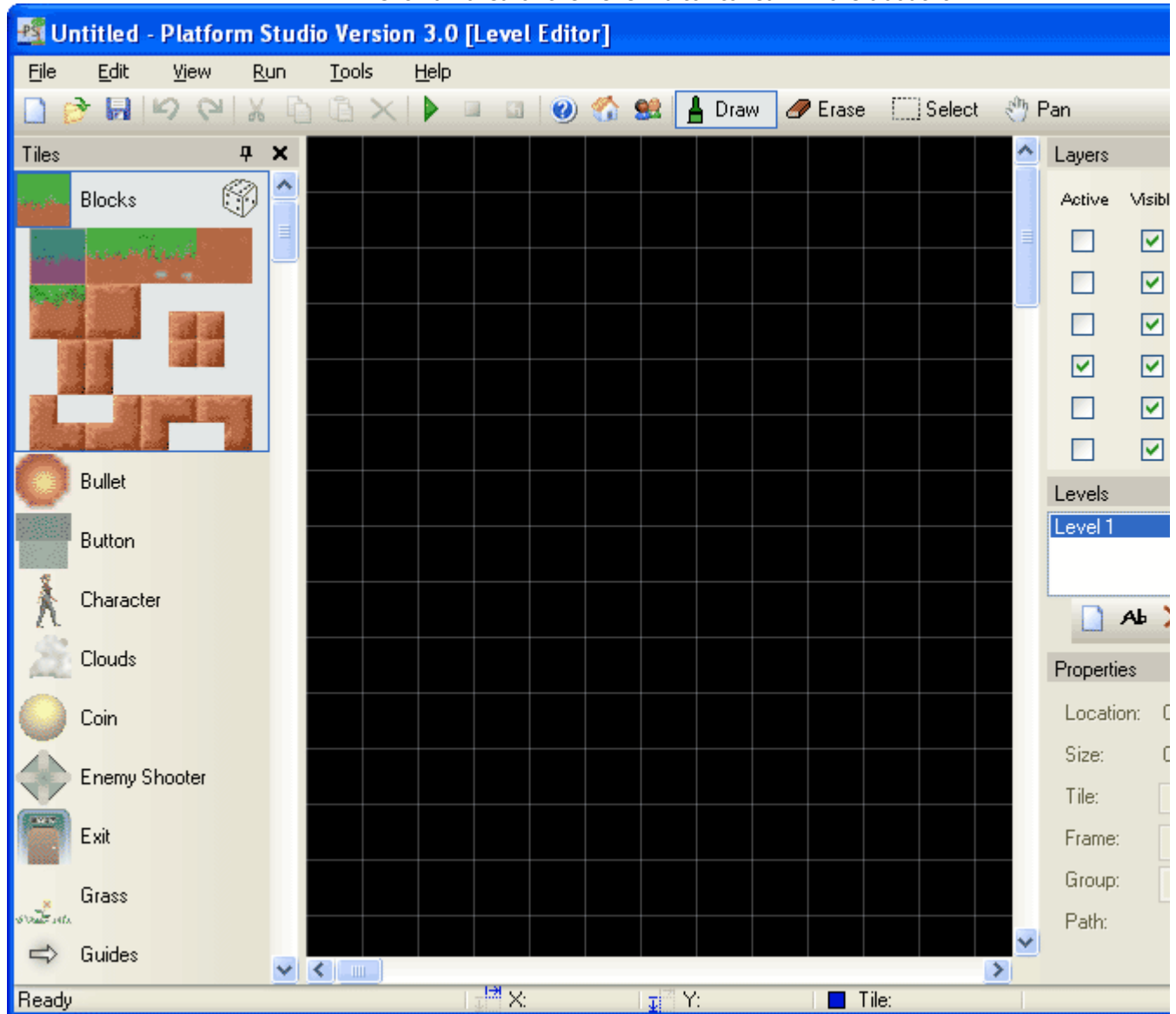
[Window Background](#)
[Window Music](#)
[Window Transition](#)
[Window Scheme Wizard](#)
[Options](#)

Copyright 2005, First Productions, Inc.

Platform Studio Help

Level Editor up

Click an area of the Level Editor to learn more about it.



Menu Bar

This contains functions for handling files, performing edits, changing views, debugging, accessing tools, and game execution. Here is the list of menus:

- File
 - New: Creates a new game.
 - New Game Wizard: Creates a new game using the [Platform Studio New Game Wizard](#).
 - Open: Opens an existing game.
 - Save: Saves the current game. If this is the first time saving, Platform Studio will prompt you for

- Save As: Saves the current game after prompting you for a new filename.
 - Game Properties: Shows the [Game Properties](#) dialog.
 - Compile: Compiles your game so it can be distributed.
 - [Recent Games]: Contains a list of recently used games. Clicking one will open it.
 - Exit: Exits Platform Studio.
- Edit
 - Undo: Undoes the last action.
 - Redo: Redoes the last undid action.
 - Cut: Copies the selected controls, and then deletes them.
 - Copy: Copies the selected controls.
 - Paste: Copies the contents of the clipboard to the editor.
 - Delete: Deletes the selected controls.
 - Select All: Selects all the controls in the current window.
 - Deselect: Deselects all the selected controls.
 - Find/Replace Tiles: Shows the [Find/Replace Tiles dialog](#), allowing you to find or replace tiles throu game.
 - Resize Level: Shows the [Resize Level dialog](#), allowing you to resize the current level.
 - Level Background: Shows the [Background Editor dialog](#), allowing you to edit the background ima current level.
 - Background Music: Shows the [Background Music dialog](#), allowing you to change or clear the back for the current level.
- View
 - Grid: Shows/hides the grid.
 - Toolbar: Shows/hides the [toolbar](#).
 - Status Bar: Shows/hides the [status bar](#).
 - Tile Selector: Shows/hides the [tile selector](#).
 - Layers Panel: Shows/hides the [Layers panel](#).
 - Levels Panel: Shows/hides the [Levels panel](#).
 - Properties Panel: Shows/hides the [Properties panel](#).
- Run
 - Start: Runs the game in debug mode.
 - Start in Separate Process: Runs the game in debug mode as a separate process (using psTest.ex significantly slower than *Start* and should only be used when a serious error is anticipated (such : an endless loop or an error that would otherwise crash Platform Studio or render it unusable until you are not using scripts or advanced actions, such as Begin Loop and End Loop, ignore this com
 - Stop: Closes the debugger.
 - Restart: Closes the debugger, and then runs the game again.
 - Debug Options
 - Skip Windows: When this is checked, the debugger will begin at the window containing the
 - Make Invincible: When this is checked, you will not be able to lose health or lives.
 - Start at Level: When this is checked, the debugger will start at a specified level.
 - Show FPS: When this is checked, the debugger will show the framerate of your game.
- Tools
 - Tileset Editor: Opens the [Tileset Editor](#).
 - Action Editor: Opens the [Action Editor](#).
 - Window Editor: Opens the [Window Editor](#).
 - Options: Shows the [Options dialog](#).
- Help
 - Show Help: Shows Platform Studio Help.
 - On the Web: Opens the Platform Studio website in a new internet browser.
 - Forums: Opens the Platform Studio forums in a new internet browser.
 - Support: Opens your email program and creates a new email message addressed to our support (psstudio@firstproductions.com).
 - Check for Updates: Checks for a new version of Platform Studio or any addons that might be ava Professional Edition, this dialog will prompt you for your username and password to access restric
 - About: Shows the About dialog.

Toolbar

This contains commonly-used functions, such as saving, undoing, and changing the current edit mode. Here a in order from left to right:

- New: Creates a new game.
- Open: Opens an existing game.
- Save: Saves the current game. If this is the first time saving, it will prompt you for a filename.
- Undo: Undoes the last action.
- Redo: Redoes the last undid action.
- Cut: Copies the selected items to the clipboard, and then deletes them.
- Copy: Copies the selected items to the clipboard.
- Paste: Copies the contents of the clipboard to the editor.
- Delete: Deletes the selected items.
- Run: Runs the game in debug mode.
- Stop: Closes the debugger.
- Restart: Closes the debugger, and then runs the game again.
- Help: Shows the Platform Studio Help.
- Website: Opens the Platform Studio website in a new internet browser.
- Forums: Opens the Platform Studio forums in a new internet browser.
- Draw: Changes the current mode to draw. This allows you to draw tiles on the map, or draw locations (layer), by clicking and dragging with the mouse.
- Erase: Changes the current mode to erase. This allows you to erase tiles or locations by clicking them with the mouse.
- Select: Changes the current mode to select. This allows you to select tiles or locations by clicking them with the mouse. You can select multiple tiles or locations by dragging a box through them, or holding down Shift and clicking individually clicking them. You can deselect individual tiles or locations by holding down Shift and clicking a tile or location. To deselect everything that is selected, click outside of any tiles or locations.
- Pan: Changes the current mode to pan. This allows you to quickly browse through the level by clicking anywhere with the mouse.

Tiles Panel

This allows you to select a tile and/or subtile to draw with it on the map. In the list of tiles, click a tile to select it, then click the Draw mode. Below the selected tile might be an area where you can select a subtile (section) of the selected tile. For example, if you loaded a Blocks image that has 4 sections across, and 4 sections down (set in the Tilesheet Editor), you can select an individual section to draw with (if it is not animated). You can also select multiple subtiles at once by dragging with the mouse in the subtile selector. The "Die" icon next to the selected tile, when selected, tells Platform Studio to choose a random subtile that is selected to draw with, rather than all of them at the same time (so multiple subtiles should be selected under this option).

Edit Area

This area is where you can Draw, Erase, Select, or Pan. If you right-click, a context menu pops up, which contains items as in the Edit menu.

Layers Panel

Here you can change the active layer (by clicking a checkbox under Active), or hide and show layers (by clicking a checkbox under Visible). Here is a description of the different layers:

- Back: Changes the level background.
- Parallax: This layer acts like a normal layer, but it scrolls at 50% of the normal scroll speed. Tiles in this layer are treated as background, and cannot interact with the character.
- Background: This is a normal layer for drawing tiles. All tiles in this layer except collectable tiles are treated as background, and cannot interact with the character.
- Foreground: This is a normal layer for drawing tiles. All tiles in this layer can interact with the character.
- Front: This layer acts the same as the Background layer, except that it is drawn on top of the Foreground instead of behind it.
- Location: This layer is used for drawing and editing locations, which are invisible areas that can perform actions when the player enters, exits, or is inside them. Locations can also be used to house checkpoints. For more information on creating checkpoints, click [here](#).

Note: all layers are drawn in the order that they were listed (i.e. Back is drawn first, and Location is drawn last).

Levels Panel

Here you can manipulate your game's levels, as well as change the active level. To change the active level, click the level you want to be active.

the list of levels. The toolbar below the list of levels is used for manipulating the levels. Here are the buttons in left to right:

- New Level: Creates a new level. Platform Studio will prompt you for a name, and a size (in tiles), that y new level to be.
- Rename Level: Allows you to rename the active level.
- Delete Level: Deletes the active level.
- Move Level Up: Moves the active level up (so that it will be played before the level before it).
- Move Level Down: Moves the active level down (so that it will be played after the level after it).

Properties Panel

Displays and lets you edit the properties for the selected tile or location. This panel will be enabled only if there is one tile or location selected, and will change depending on if it is a tile or location.

Tile Properties

- Location: Displays the location of the selected tile (x, y), in tiles.
- Size: Displays the size of the selected tile (width, height), in tiles.
- Tile: Displays and lets you change the tile index of the selected tile.
- Frame: If the tile is not animated, allows you to change the subtile of the selected tile. This is displayed so 1 would be the top-left, 2 would be the next one to the right, and so on. If there are, for example, 4 rows, then subtile 5 would be at row 2, column 1, and 16 would be the bottom-right subtile.
- Group: Displays and lets you change the group of the selected tile. The dropdown arrow allows you to view existing groups.
- Path: Allows you to edit the path of the selected tile.

Location Properties

- Displays the location of the selected location (x, y), in pixels.
- Size: Displays the size of the selected location (width, height), in tiles.
- Name: Displays and lets you change the name of the selected location.
- Checkpoint: Displays and lets you change the location of the checkpoint in the selected location. Choose to clear the checkpoint.

Status Bar

Displays information about what your mouse is hovering over, as well as the status of Platform Studio. In the status bar, it also displays if Platform Studio is ready for your input, or if it is currently performing an operation that requires your input. The second and third panels display the X and Y coordinates, respectively, of your mouse (in tiles). The fourth panel displays the name of the tile or location your mouse is hovering over.

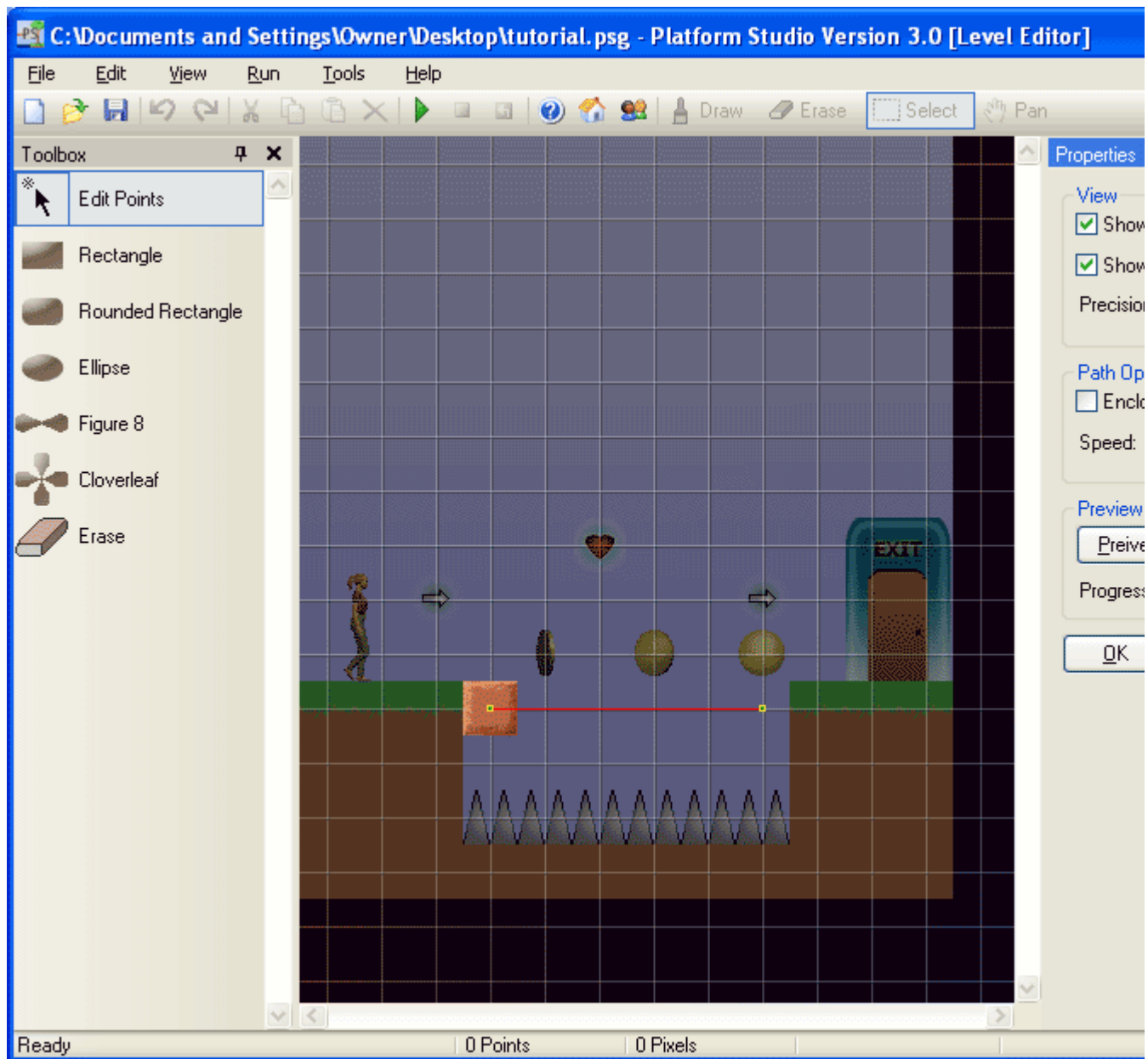


Copyright 2005, First Productions, Inc.



Path Editor 

Click an area of the Path Editor to learn more about it.



Menu Bar

This contains functions for handling files, performing edits, changing views, debugging, accessing tools, and game testing. The following is the list of menus:

File | Edit | View | Run | Tools | Help

Context Menu

The context menu (which pops up when you right-click the edit area) contains 4 items. They are:

- Undo: Undoes the last action.
- Redo: Redoes the last undid action.
- Copy Point: Allows you to copy a point that was right-clicked. This is useful if you want to insert a point at the same location as other points.
- Delete Point: Allows you to delete a point that was right-clicked.

Toolbar

This contains commonly-used functions, such as saving and undoing. Here are the buttons in order from left to right (grayed items are disabled in the Path Editor):

- New: Creates a new game.
- Open: Opens an existing game.
- Save: Saves the current game. If this is the first time saving, it will prompt you for a filename.
- Undo: Undoes the last action.
- Redo: Redoes the last undid action.
- *Cut*
- *Copy*
- *Paste*
- *Delete*
- Run: Runs the game in debug mode.
- Stop: Closes the debugger.
- Restart: Closes the debugger, and then runs the game again.
- Help: Shows the Platform Studio Help.
- Website: Opens the Platform Studio website in a new internet browser.
- Forums: Opens the Platform Studio forums in a new internet browser.
- *Draw*
- *Erase*
- *Select*
- *Pan*

Toolbox

This allows you to draw a preset path, clear the current path, or draw points with the mouse. Here are the different tools:

- Edit Points: Allows you draw and edit points with the mouse. To draw a point, click where you want the point. If you want that point to be curved, you can drag the mouse while you click it to change the curve. If you do not drag the mouse, a straight point will be added instead. To modify a point, right-click it. You can click a point to move it, or click and drag a guide to adjust the curve after the point is drawn. Points are yellow. Guides are green. You can hold down Shift and drag with the mouse to move the whole path.
- Rectangle: Click and drag in the edit area to create a rectangle. The center of the rectangle will be drawn first, then click the mouse. Also, note that whenever you create a preset shape, the current path will be cleared. This is a mistake you can always undo it.
- Rounded Rectangle: Creates a rounded rectangle.
- Ellipse: Creates an ellipse.
- Figure 8: Creates a figure 8.
- Clover Leaf: Creates a clover leaf.
- Clear: Clears the current path.

Edit Area

This area is where you can edit the current path. For more information on how to create a path, see [Toolbox](#). If you right-click in the edit area, a context menu pops up. To learn more about the context menu, click [here](#).

Properties Panel

This is the "control panel" for the Path Editor. It contains view options, path options, preview options, and buttons to save and discard your changes.

- Show Guides: Choose whether you want to view the guides (blue guide lines that connect a point guide to its point).
- Show Points: Choose whether you want to view the points (yellow points and green point guides).
- Precision: Choose how much detail you want to draw. The higher the detail, the slower it will draw.
- Enclosed: Choose whether you want Platform Studio to enclose your path (draw 3 sides of a square with Platform Studio, will connect the first and the last points together to make the last side). If this is unchecked, when the tile reaches the end of the path, it will reverse direction and go back to the beginning of the path.
- Speed: Choose how many seconds you want the tile to take to complete one cycle of the path.
- Preview button: Click it to preview the tile moving along your path. Click it again to stop the preview.
- Loop: Choose whether you want the preview to keep moving through the path infinitely, or just complete one cycle and then stop.

- Progress: Displays how much of the path the tile has gone through.
- OK button: Accepts your changes to the path and closes the path editor.
- Cancel button: Discards your changes to the path and closes the path editor.

Status Bar

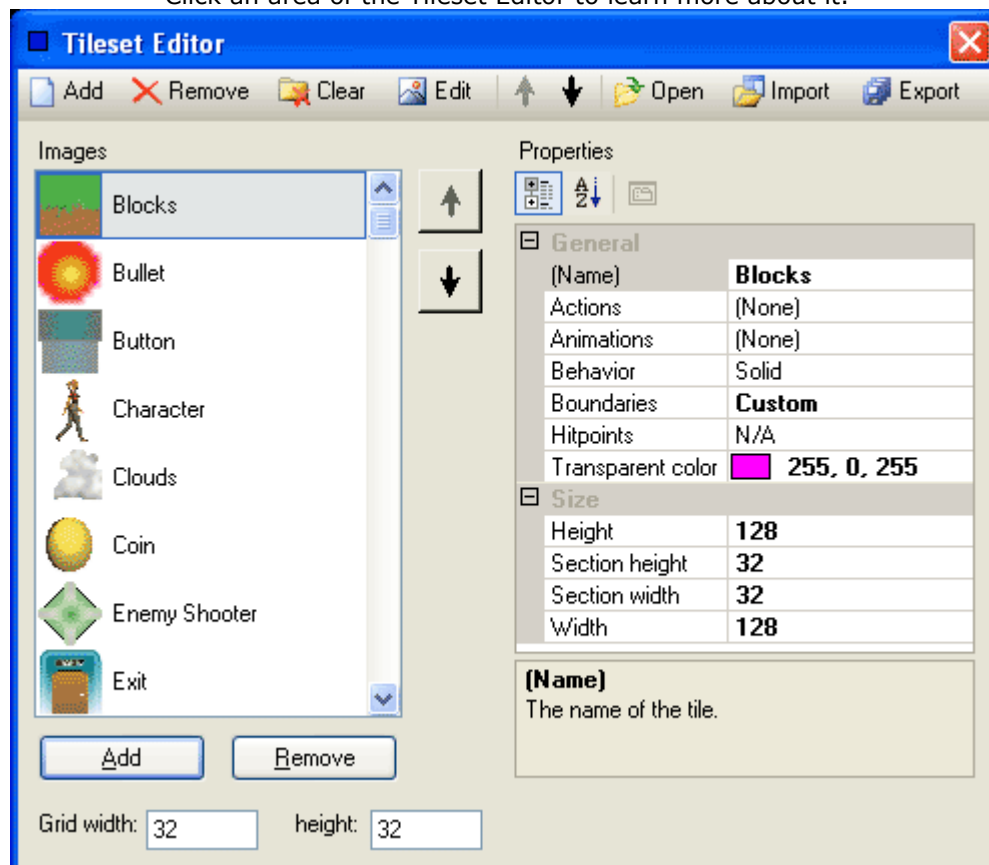
Displays context information to help you draw your path, and also displays how many points are in your path, pixels your path covers.

Copyright 2005, First Productions, Inc.



Tileset Editor [up](#)

Click an area of the Tileset Editor to learn more about it.



Toolbar

- Add: Adds a tile to the tileset.
- Remove: Removes the selected tile from the tileset.
- Clear: Removes all tiles from the tileset.
- Edit: Edits the selected tile in the specified image editing program. You can

change this in the [Options dialog](#).

- Up Arrow: Move the selected tile up in the list of tiles.
- Down Arrow: Move the selected tile down in the list of tiles.
- Open: Deletes the current tileset and opens a saved one in its place.
- Import: Adds tiles from a saved tileset into the current tileset.
- Export: Saves the tileset as a *.pst file, so it can later be opened in another game or imported from. The tileset is automatically saved in the current game file, so this button is not necessary unless you need to access this tileset in another game.

Tile List

Contains the list of tiles. Click a tile to edit its properties. You can select multiple tiles by using the Ctrl key (toggle selection) and Shift key (select range) while clicking on a tile. You can easily move a tile through the list of tiles by clicking it and dragging it to where you want it to be located.

Add Button

Click this to add one or more images to the tileset.

Remove Button

Click this to remove the selected tile from the tileset, and delete all of its actions.

Up Button

Click this to move the selected tile up in the list of tiles.

Down Button

Click this to move the selected tile down in the list of tiles.

Tile Name

The name of the selected tile.

Actions

Opens the Action Editor with the selected tile's triggers.

Hitpoints

The amount of damage a tile can take (the amount of bullets that can be fired at a tile) before it is destroyed. If the tile is invincible, specify N/A.

Boundaries

Opens the [Tile Boundaries Editor](#), which allows the customization of a tile's boundaries.

Tile Width

The width, in pixels, of the selected tile.

Tile Height

The height, in pixels, of the selected tile.

Section W

The section width, in pixels, of the selected tile. This number must be a factor of the tile width. If the section width is smaller than the tile width, then the tile can be divided into subtiles. For example, if the tile width was 64 and the section width was 32, then there would be two subtiles (assuming that both the tile height and section height were the same). Subtiles can either be used as different tiles when drawing in the level editor, or they can be used in a tile animation.

Section H

The section height, in pixels of the selected tile. This acts just like the [section width](#) but

it is vertical, instead of horizontal.

Transparent Color

The color to make transparent in the tile's image. In PNG and GIF images with transparency, the transparent color will still be counted as transparent in addition to the specified transparent color.

Behavior

The general behavior of the tile can be edited with the dropdown box. These are the different behaviors:

- Solid (Default): The character can not go through this tile when placed in the Foreground layer.
- Background: The character can not interact with this tile, no matter what layer it is on.
- Collectable: This tile can be collected by the character when placed on either the Background, Foreground, or Front layers of the level.
- Ledge: The character can walk through this tile unless it is on top of it.
- Character: This is the tile that will be used to draw the character.
- No Gravity: This tile is a combination of Solid and Ledge. It is partially solid in the fact that all hit actions will be processed when the character either hits or is on top of the tile. It is partially a ledge because the player can walk through it from the the right, left, or bottom, but can not from the top. Unlike a ledge, if the player is overlapping the tile in any way, the player will not fall. This tile is often used as a ladder or water, which creates an action that turns off the gravity when the character overlaps it, hence the name, No Gravity.

If you click the (...) button, you can further customize the selected tile's behavior with the [Customize Behavior dialog](#).

Animations

Edits the animations of the selected tile using the [Animation Editor](#).

Grid Width

This is the overall tile width, which most tiles' widths should be a multiple of. The grid's spacing will change depending on this number, any units that contain the word "tiles" will change depending on this number, and any new tiles that are added will automatically be given a section width based on this number, if possible.

Grid Height

This is the overall tile height, which most tiles' heights should be a multiple of. The grid's spacing will change depending on this number, any units that contain the word "tiles" will change depending on this number, and any new tiles that are added will automatically be given a section height based on this number, if possible.

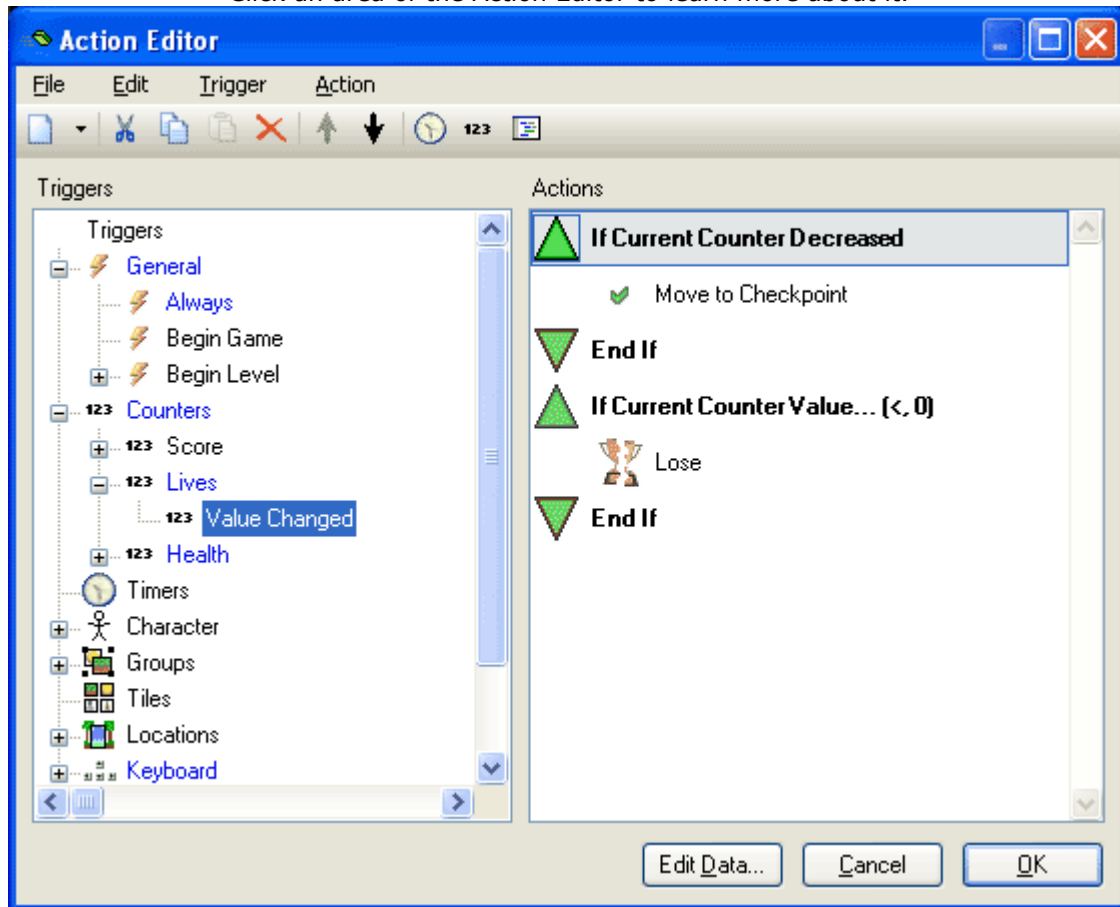


Copyright 2005, First Productions, Inc.



Action Editor up

Click an area of the Action Editor to learn more about it.



Triggers

Contains the list of triggers. Select a trigger to add an action (Note: Keep in mind that not everything in the list is a trigger. Counters > Score is not a trigger, but Counters > Score > Value Changed is). Anything in **blue** either contains actions, or a trigger inside it contains actions. Anything in black contains no actions, and none of the triggers inside it contains actions.

Actions

This is the list of actions for the selected trigger. Double-click an action to [edit](#) it. To easily move an action through the list, click it and then drag it to where you want it to be located.

Menus

- File
 - Close and save changes: see [OK Button](#)
 - Close and discard changes: see [Cancel Button](#)
- Edit
 - Timers: Shows the [Edit Timers](#) dialog, allowing you to edit the timers.
 - Counters: Shows the [Edit Counters](#) dialog, allowing you to edit the counters.
 - Global script: Shows the [Script Editor](#) dialog, allowing you to edit the global script for your game.
- Trigger
 - Expand: Expands the currently selected trigger category to show the related triggers or categories.
 - Collapse: Collapses the currently selected trigger category to hide the related triggers or categories.
 - Expand All: Expands the currently selected trigger category as well as all of its

- subcategories.
 - Copy Child Actions: Copies all actions for the selected trigger. If a category is selected, then all actions for all triggers of the selected category will be copied.
 - Paste Child Actions: Pastes all copied actions into the selected trigger or triggers of a selected category. This action can only be done if the hierarchy of the copied triggers or category matches the hierarchy of the selected triggers or category.
 - Paste Link to Child Actions: Does the same thing as Paste Child Actions, but in place of the actual actions, it creates an "Execute Script" action which then links to the copied trigger. Thus, if you then change the original action that was copied, the triggers that are linked to it will automatically reflect this change.
 - Delete Child Actions: Clicking this will delete all actions for the selected trigger. If a category is selected, then all actions of all triggers in this category will be deleted.
- Action
 - New Action
 - New Action: This will prompt you with the *Edit Action* dialog to specify an action to create, as well as its parameters.
 - (Action categories and related actions): An action of the selected type will be created. If parameters are required, the *Edit Action* dialog will appear, asking you to specify values for these parameters.
 - Execute Script: This will create a new *Execute Script* action under the selected action. The *Script Editor* will then open, allowing you to edit the script.
 - Cut: Copies the selected action to the clipboard and then deletes it.
 - Copy: Copies the selected action to the clipboard.
 - Paste: Adds the copied action to the list of actions, below the selected item.
 - Delete: Deletes the selected action.
 - Move Up: Moves the selected action upwards, so it is executed before the action that precedes it.
 - Move Down: Moves the selected action downwards, so it is executed after the action that is after it.

Toolbar

The toolbar contains buttons for adding and modifying actions, as well as buttons for editing timers and counters. Here is a list of its buttons in order from left to right:

- New Action: Creates a new action below the selected item. If nothing is selected, it creates it at the end of the list of actions.
- Cut: Copies the selected action and then deletes it.
- Copy: Copies the selected action.
- Paste: Adds the copied action to the list of actions, below the selected item.
- Delete: Deletes the selected action.
- Move Action Up: Moves the selected action upwards, so it is executed before the action that precedes it.
- Move Action Down: Moves the selected action downwards, so it is executed after the action that is after it.
- Edit Timers: Shows the [Edit Timers](#) dialog, allowing you to edit the timers.
- Edit Counters: Shows the [Edit Counters](#) dialog, allowing you to edit the counters.
- Edit Global Script: Shows the [Script Editor](#) dialog, allowing you to edit the global script for your game.

Edit Data Button

Edits the action data. You can create new action presets or modify them.

Cancel Button

Discards all of your changes in the Action Editor, and then closes it.

OK Button

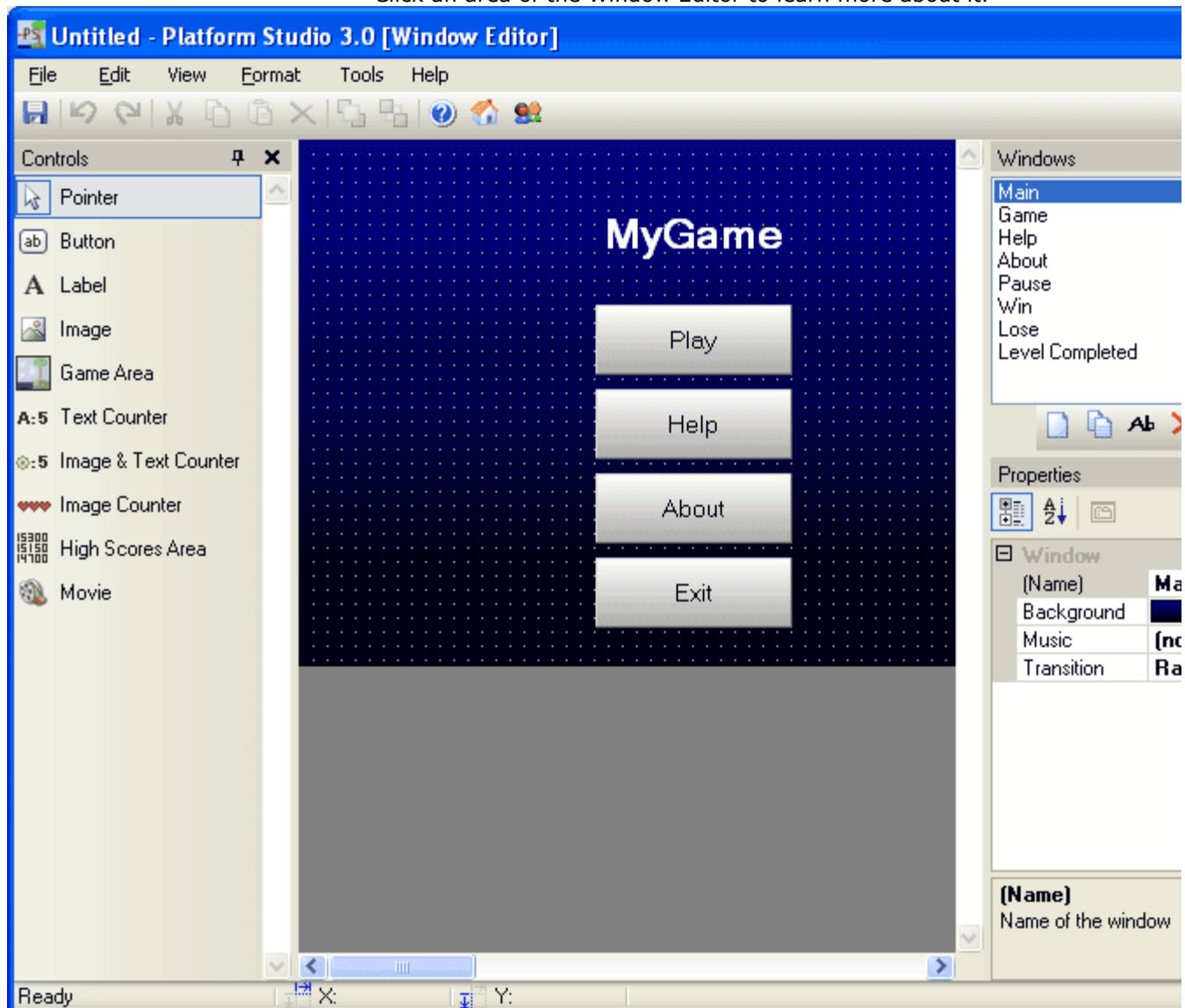
Saves all of your changes in the Action Editor, and then closes it.



Copyright 2005, First Productions, Inc.

**Window Editor up**

Click an area of the Window Editor to learn more about it.



Menu Bar

This contains functions for handling files, performing edits, changing views, formatting controls, and getting the list of menus:

- File
 - Save: Saves the current game. If this is the first time saving, Platform Studio will prompt you for a filename.
 - Save As: Saves the current game after prompting you for a new filename.
 - Close: Closes the window editor.
- Edit
 - Undo: Undoes the last action.
 - Redo: Redoes the last undid action.
 - Cut: Copies the selected controls, and then deletes them.
 - Copy: Copies the selected controls.
 - Paste: Copies the contents of the clipboard to the editor.
 - Delete: Deletes the selected controls.
 - Select All: Selects all the controls in the current window.
 - Deselect: Deselects all the selected controls.
 - Window Properties: Shows the Windows tab of the [Game Properties dialog](#).
 - Window Background: Shows the [Background Editor dialog](#), allowing you to edit the background of the current window.
 - Window Music: Shows the [Background Music dialog](#), allowing you to change or clear the background music of the current window.
 - Window Transition: Allows you to edit transition information for the current window.
- View
 - Grid: Shows/hides the grid.
 - Toolbar: Shows/hides the toolbar.
 - Status Bar: Shows/hides the status bar.
 - Control Selector: Shows/hides the control selector.
 - Right Sidebar: Shows/hides the right sidebar.
- Format
 - Center Horizontally: Centers the selected controls horizontally in the current window.
 - Center Vertically: Centers the selected controls vertically in the current window.
 - Align to Grid: Aligns the selected controls to the grid.
- Tools
 - Window Scheme Wizard: Opens the Window Scheme Wizard, which allows you to choose basic window schemes including a color scheme. Note: This will delete all current window information. You may undo this action.
 - Level Editor: Closes the Window Editor and shows the [Level Editor](#).
 - Options: Shows the [Options dialog](#).
- Help
 - Show Help: Shows Platform Studio Help.
 - About: Shows the About dialog.

Toolbar

This contains buttons for commonly-used functions, such as saving and undoing. Here are the buttons in order from left to right:

- Save: Saves the current game. If this is the first time saving, it will prompt you for a filename.
- Undo: Undoes the last action.
- Redo: Redoes the last undid action.
- Cut: Copies the selected controls to the clipboard, and then deletes them.
- Copy: Copies the selected controls to the clipboard.
- Paste: Copies the contents of the clipboard to the editor.
- Delete: Deletes the selected controls.
- Help: Shows the Platform Studio Help.
- Website: Opens the Platform Studio website in a new internet browser.
- Forums: Opens the Platform Studio forums in a new internet browser.

Controls Panel

This allows you to select a control to draw on the current window. If you select the Pointer, you can modify the

controls in the current window. (Note: You may only have one Game Area in your game.) Select a control or click it.

- **Pointer:** Lets you modify the existing controls in the current window. Left-click a control to select it, or click multiple controls by holding down Shift and/or dragging a box around the controls you want to select. You can click individual controls by holding down Shift and clicking a selected control, or you can deselect everything outside of any control. Double-click a control to edit its properties.
- **Controls:** Lets you add a control to the current window. Click and drag with the mouse where you want to be, and a new control is created. The Pointer then becomes active.

Windows Panel

This panel lets you select and manipulate the windows in your game. Select a window in the list of windows to become active. Use the toolbar to add, edit, or delete windows. Here are its buttons in order from left to right:

- **New Window:** Creates a new window. It prompts you for a window name first.
- **Copy Window:** Copies the active window. The copy is placed at the end of the list of windows.
- **Rename Window:** Renames the active window.
- **Delete Window:** Deletes the active window.
- **Move Window Up:** Moves the window up in the list of windows. Note: The first window is shown first when you start your game.
- **Move Window Down:** Moves the window down in the list of windows.

Properties Panel

This is a list of properties that apply to the control(s) or the active window (if no controls are selected).

Edit Area

Here is where you can add or edit the controls in the current window. For more information on how to add and edit, see [Controls Panel](#).

Status Bar

Displays the status of Platform Studio as well as the coordinates of the cursor. In the first panel, it displays if Platform Studio is ready for your input, or if it is currently performing an operation that requires you to wait. The second and third panels display the X and Y coordinates, respectively, of your mouse (in pixels).



Copyright 2005, First Productions, Inc.



New Game Wizard [up](#)

Step 1



- Check "Create a new blank game" to skip the rest of the wizard, and just use the default settings.

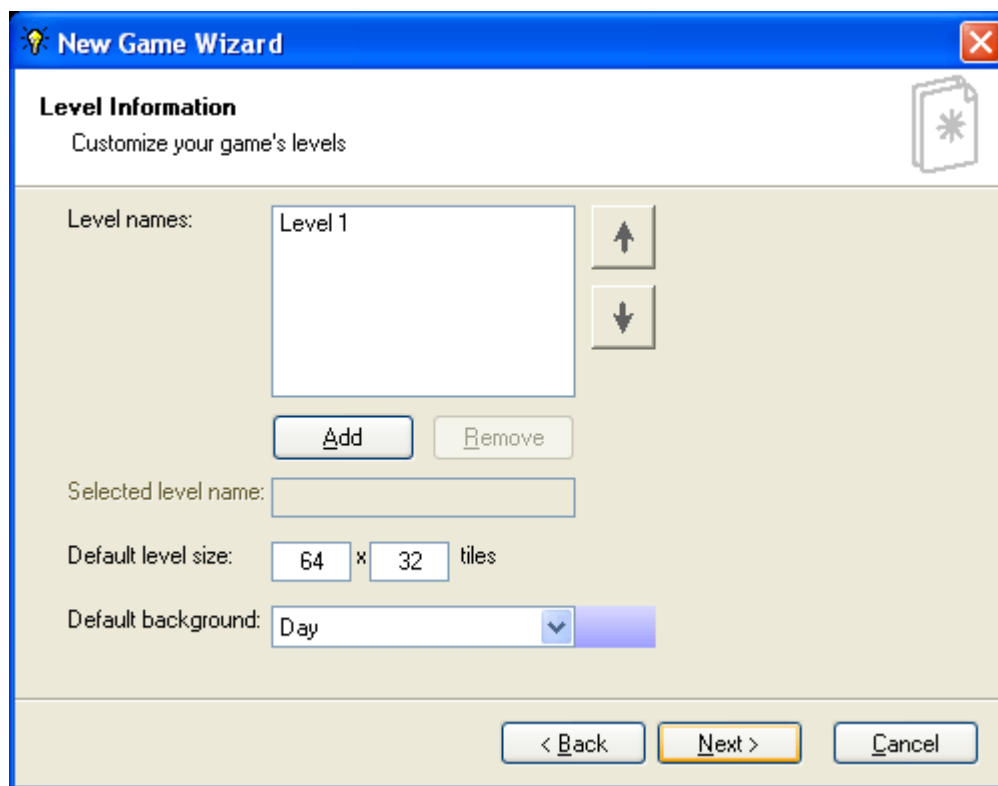
Step 2

The screenshot shows the 'New Game Wizard' dialog box, Step 2: Game Information. The title bar is blue with a close button. The main area has a light beige background. At the top, it says 'Game Information' and 'Please specify some basic information about your game'. Below this, there are several input fields: 'Name of game:' with the text 'MyGame'; 'Credits:' with a text area containing 'Created by MyName' and 'Powered by Platform Studio'; 'Company:' with the text 'MyCompany, Inc.'; 'Website:' with the text 'http://www.mywebsite.com'; and 'Support email:' with the text 'support@mywebsite.com'. At the bottom, there is a note: '[Note: You can edit this information later by clicking File -> Game Properties in the main editor]'. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

- Name of game: Specify the name of your game. This will be used in various places, such as for filenames during compilation, or game information displayed in your user interface.
- Credits: Specify the credits of your game. This will be shown in your About window.
- Company: Specify your company, if you have one. This will be shown in your About window.
- Website: Specify your website, if you have one. This will be shown in the About window, and the Help menu (if you have one).
- Support email: Specify your support email address, if you have one. This will be shown the About window, and the Help menu (if you have one).

Note: You can edit this information later by clicking File and selecting Game Properties in the level editor.

Step 3

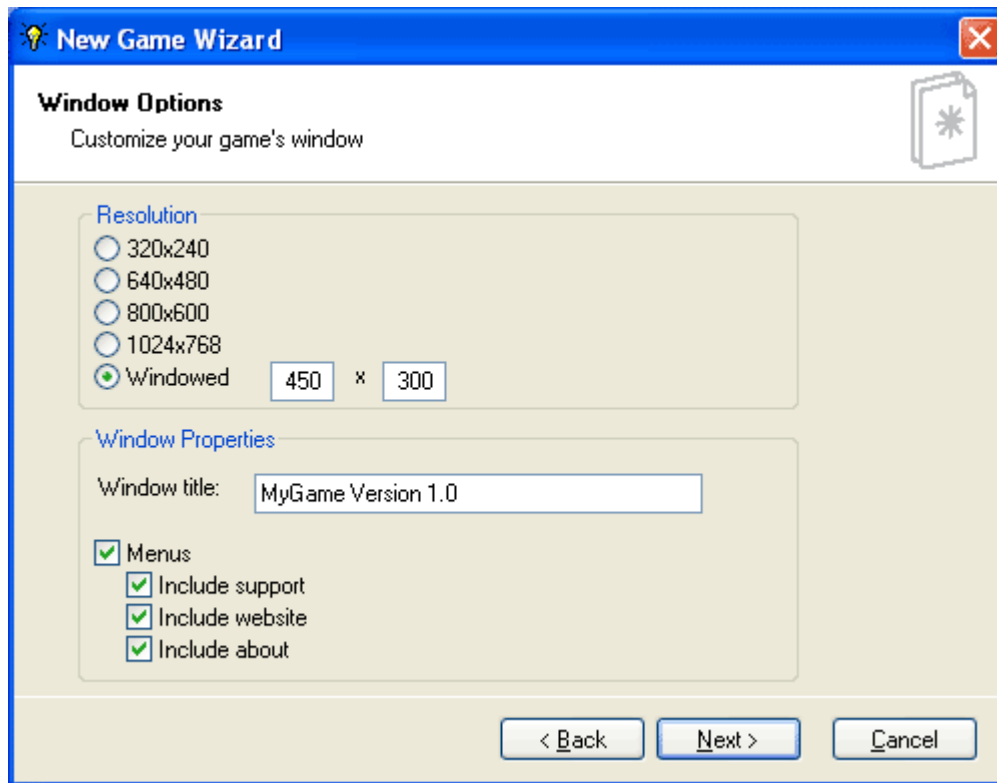


The screenshot shows the 'New Game Wizard' dialog box with the 'Level Information' tab selected. The dialog has a blue title bar with a lightbulb icon and the text 'New Game Wizard'. Below the title bar is a tabbed interface with 'Level Information' as the active tab. The main area is light beige and contains the following controls:

- Level names:** A list box containing 'Level 1'. To its right are two buttons: an upward arrow (Move Up) and a downward arrow (Move Down).
- Add** and **Remove** buttons are located below the list box.
- Selected level name:** A text input field.
- Default level size:** Two input fields containing '64' and '32', followed by the text 'tiles'.
- Default background:** A dropdown menu currently showing 'Day'.
- At the bottom are three buttons: '< Back', 'Next >' (highlighted with a yellow border), and 'Cancel'.

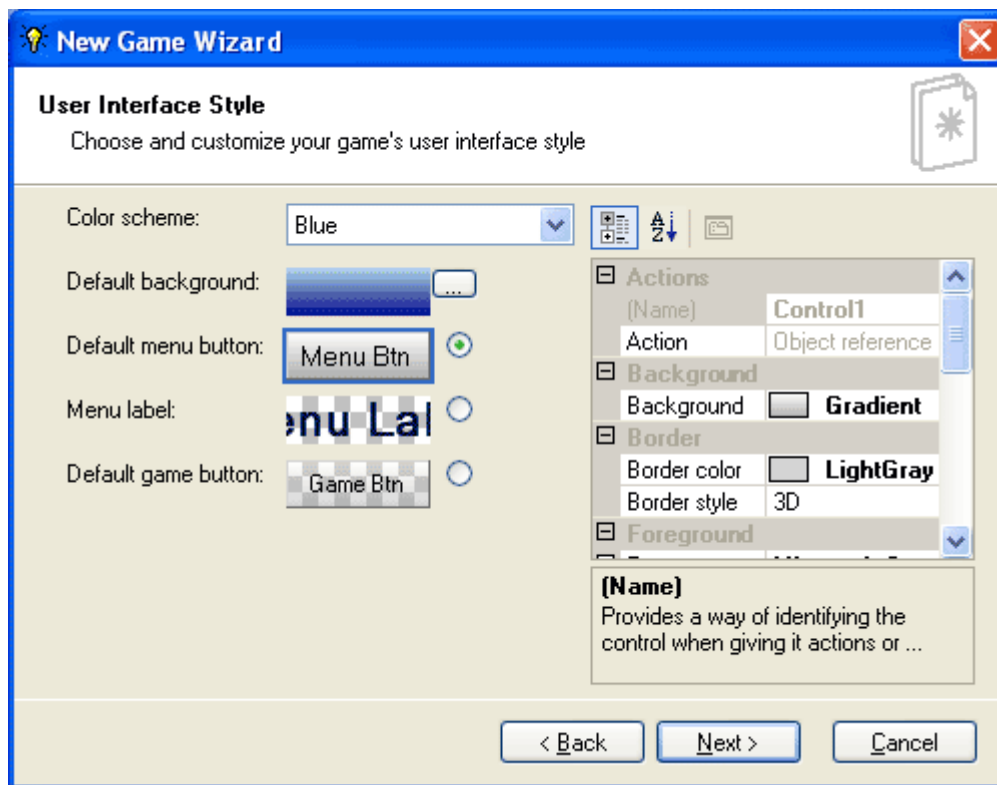
- Level names: The names of the levels the wizard will generate.
- Add button: Adds a new level.
- Remove button: Removes the selected level.
- Move Up button: Moves the selected level up in the list.
- Move Down button: Moves the selected level down in the list.
- Selected level name: Changes the selected level's name.
- Default level size: Specifies the size to make all the generated levels.
- Default background: Changes the background for all the generated levels.

Step 4



- Resolution: Specifies your game's screen resolution. The smaller the resolution, the faster your game will be, but you will have less space to work with. Choose Windowed if you want your game to run in a window, instead of fullscreen.
- Window title: Change the title of your window.
- Menus: Check if you want your window to contain menus.
- Include support: Check if you want your Help menu to include a Support item.
- Include website: Check if you want your Help menu to include a Website item.
- Include about: Check if you want your Help menu to include an About item.

Step 5



- Color scheme: Changes the color scheme for your use interface.
- Default background: Click (...) to change the default background of your windows.
- Default menu button: Click the button and use the property grid to the right to change the default menu button and other non-game buttons.
- Menu label: Click the label and use the property grid to the right to change the default menu label (as well as other labels throughout your game).
- Default game button: Click the button and use the property grid to the right to change the default game button style.

Step 6



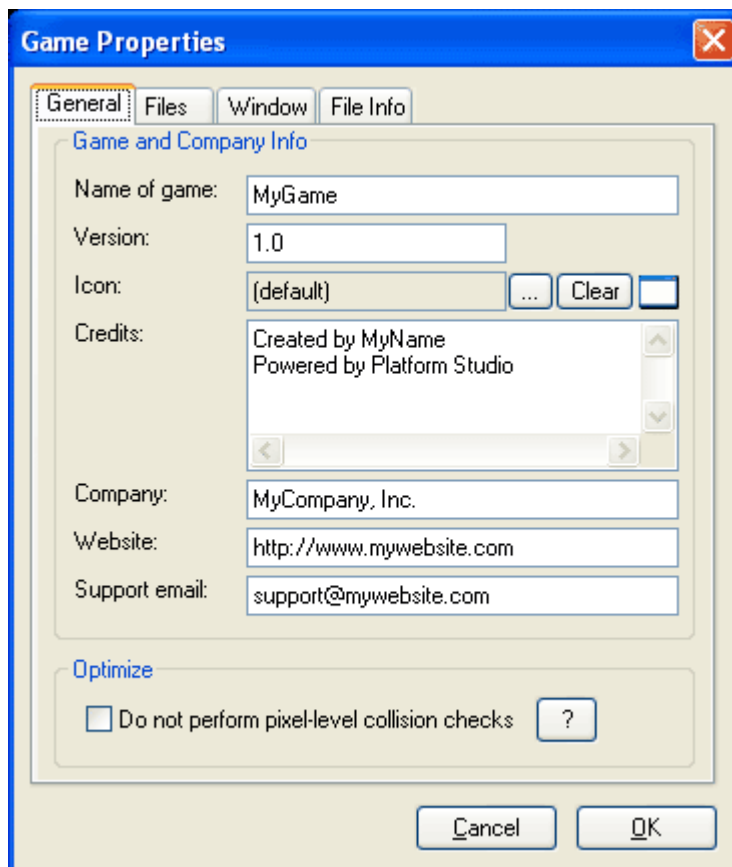
- About style: Message dialog: Shows the about information in a message box.
- About style: Window: Shows the about information in a window.
- Help style: Window: Shows the help information in a window.
- Help style: Window: Edit Help Text: Edits the help text to be shown in the help window.
- Help style: External file: Opens a help file when the user clicks the Help button.
- Help style: External file: (...): Prompts you for a file.

Copyright 2005, First Productions, Inc.



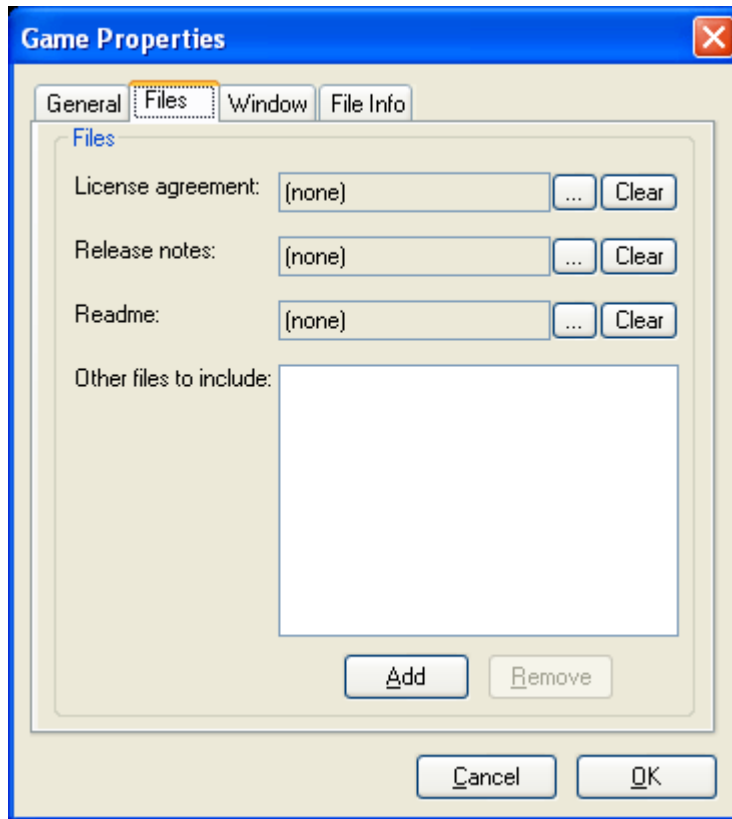
Game Properties [up](#)

General



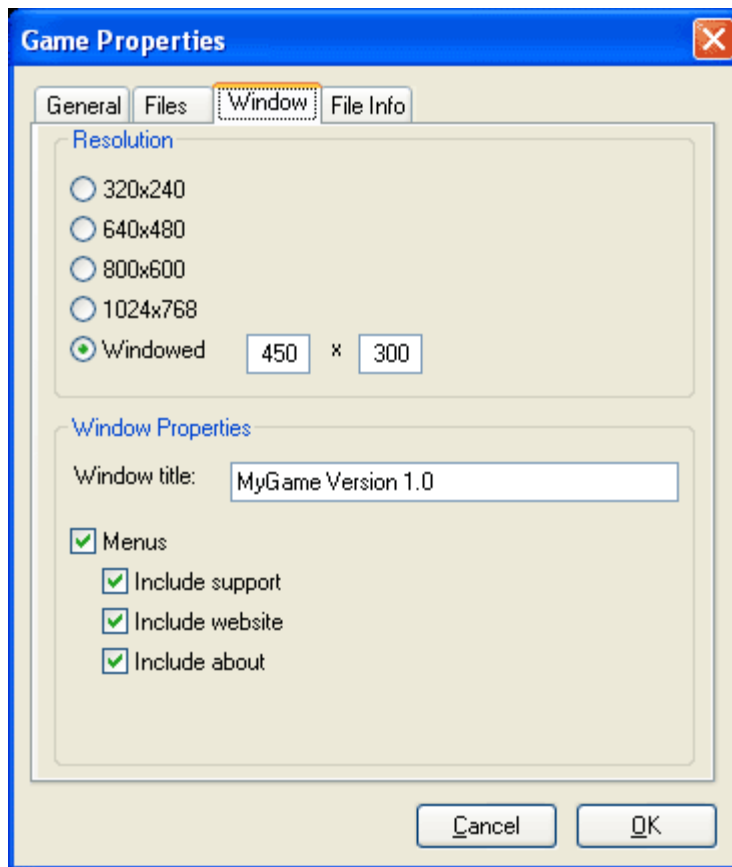
- Name of game: Specify the name of your game. This will be used in various places, such as for filenames during compilation, or game information displayed in your user interface.
- Version: Specify the version of your game.
- Icon: Specify the icon that you want to use in your compiled game. Only relevant to the Professional Edition.
- Credits: Specify the credits of your game. This will be shown in your About window.
- Company: Specify your company, if you have one. This will be shown in your About window.
- Website: Specify your website, if you have one. This will be shown in the About window, and the Help menu (if you have one).
- Support email: Specify your support email address, if you have one. This will be shown the About window, and the Help menu (if you have one).
- Do not perform pixel-level collision checks. Check this if you do not want Platform Studio to perform pixel-level collision checks. This will increase speed, but decrease accuracy. Click the (?) button for more information.

Files



- License agreement: (...): Specify a license agreement file.
- License agreement: Clear: Click to not use a license agreement file.
- Release notes: (...): Specify a release notes file.
- Release notes: Clear: Click to not use a release notes file.
- Readme: (...): Specify a readme file.
- Readme: Clear: Click to not use a readme file.
- Other files to include: A list of other files to include with your game during compilation.
- Add button: Add a file to the list of other files to include.
- Remove button: Removes the selected file from the list.

Window



- Resolution: Specifies your game's screen resolution. The smaller the resolution, the faster your game will be, but you will have less space to work with. Choose Windowed if you want your game to run in a window, instead of fullscreen.
- Window title: Change the title of your window.
- Menus: Check if you want your window to contain menus.
- Include support: Check if you want your Help menu to include a Support item.
- Include website: Check if you want your Help menu to include a Website item.
- Include about: Check if you want your Help menu to include an About item.

File Info

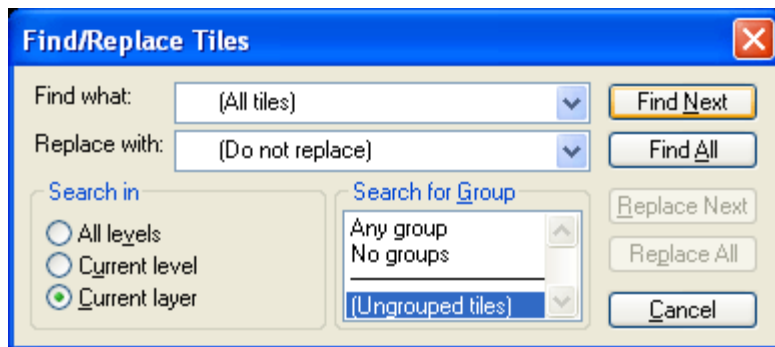
Displays statistics about your game file.



Copyright 2005, First Productions, Inc.



Find/Replace Tiles ^{up}



- Find what: Choose the tile that you want to search for. Choose (All tiles) to search for all tiles.
- Replace with: Choose the tile that you want the tiles found to be replaced with. Choose (Do not replace) to not replace any tiles, but just find them.
- Search in: Choose where you want to look
 - All levels: Search in all the levels in your game
 - Current level: Search in only the current level
 - Current layer: Search in only the current layer
- Search for group: Choose the groups that you want to search in. Click (Any group) to search in any group, (No groups) to search in no groups, and (Ungrouped tiles) to search in all tiles that have no specified group.
- Find Next button: Finds the next tile and selects it.
- Find All button: Finds all the tiles that match your search terms and selects them.
- Replace Next button: Finds the next tile and replaces it with the specified tile.
- Replace All button: Finds all tiles that match your search terms and replaces them with the specified tile.
- Cancel: Closes the dialog.

Copyright 2005, First Productions, Inc.



Resize Level up

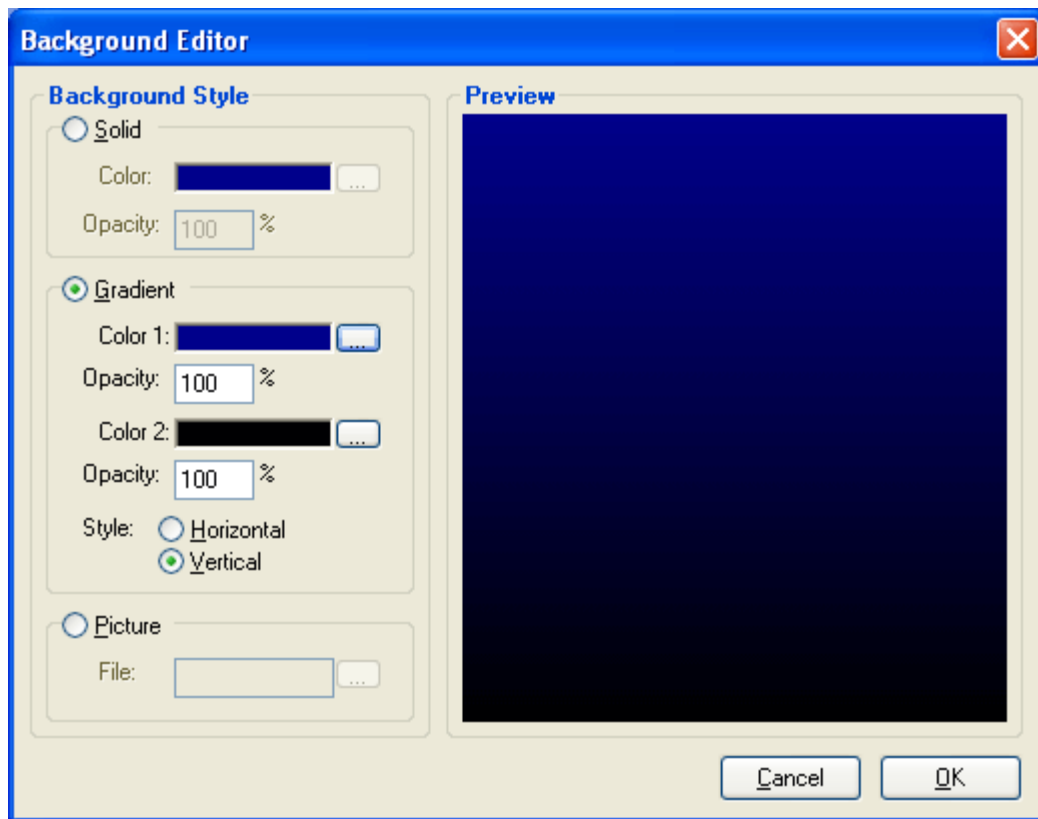


- **Resize:** Resizes the level to a certain size.
 - **Width:** Specify the new width in tiles of the current level.
 - **Height:** Specify the new height in tiles of the current level.
- **Translate:** Shifts all tiles in a level in a specified direction.
 - **Horizontally:** The distance in tiles to shift the level to the right. A negative value indicates a shift to the left.
 - **Vertically:** The distance in tiles to shift the level downward. A negative value indicates an upward shift.
- **Revert button:** Reverts the width and height settings to the current values.
- **Cancel:** Closes the dialog and cancels.
- **OK:** Closes the dialog and resizes the level.

Copyright 2005, First Productions, Inc.



Background Editor [up](#)

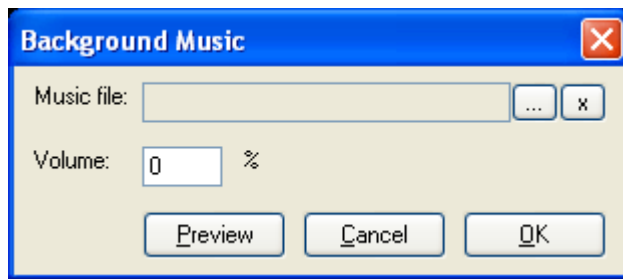


- Background Style
 - Solid
 - Color: Specify the color for the solid background.
 - Opacity: Specify the opacity (100% = solid, 0% = transparent) for the color.
 - Gradient
 - Color 1: Specify the first color for the gradient background.
 - Opacity: Specify the opacity for the first color.
 - Color 2: Specify the second color for the gradient background.
 - Opacity: Specify the opacity for the second color.
 - Style
 - Horizontal: Horizontal gradient.
 - Vertical: Vertical gradient.
 - Picture
 - File: Click (...) to choose an image file for the background.
- Preview: Shows the preview of the background.
- Cancel: Closes the dialog and cancels.
- OK: Closes the dialog and saves your changes.

Copyright 2005, First Productions, Inc.



Background Music up



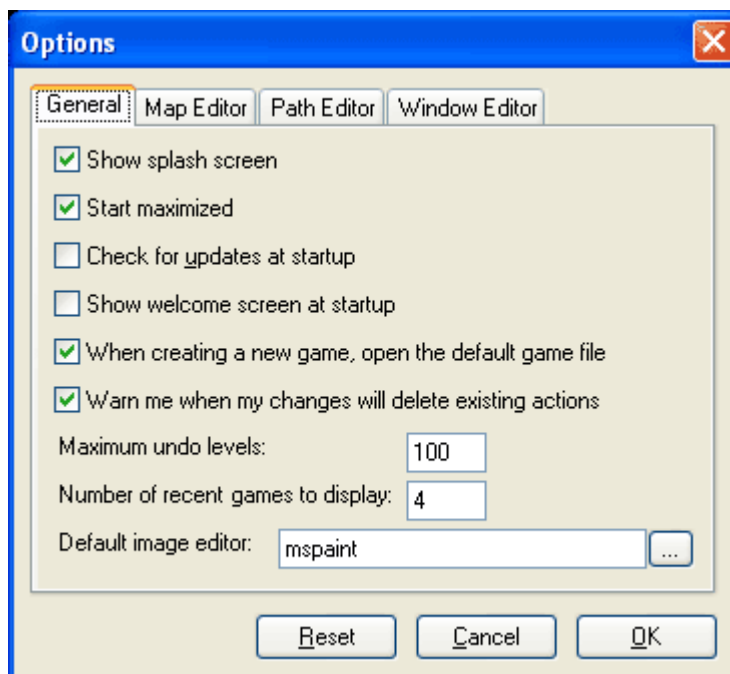
- Music file: Click (...) to specify the music file. Click (x) to choose no file.
- Volume: Specify the volume of the music. 0% is muted, and 100% is full volume.
- Preview: Plays the music with the current settings.
- Cancel: Closes the dialog and cancels.
- OK: Closes the dialog and saves your changes.

Copyright 2005, First Productions, Inc.



Options up

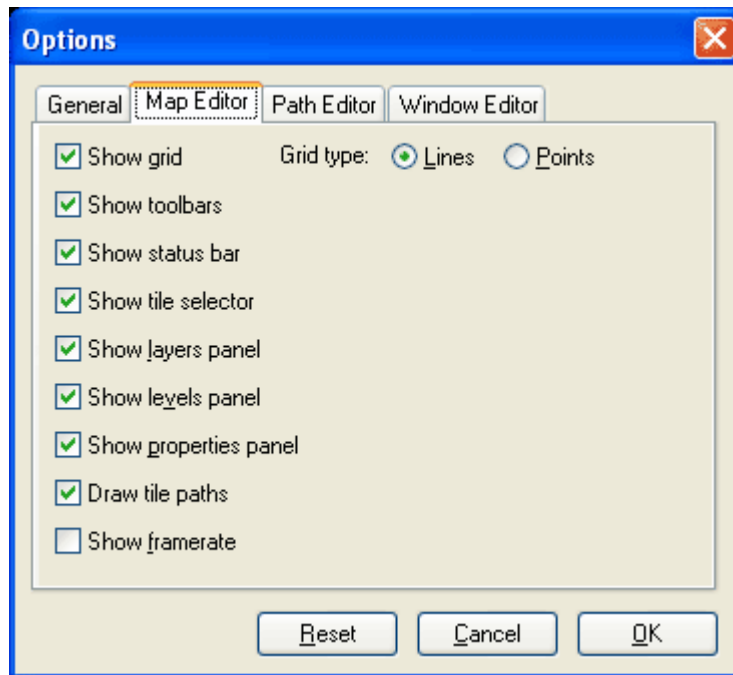
General



- Show splash screen: Shows the splash screen at startup if checked.
- Start maximized: Platform Studio will start maximized if checked.

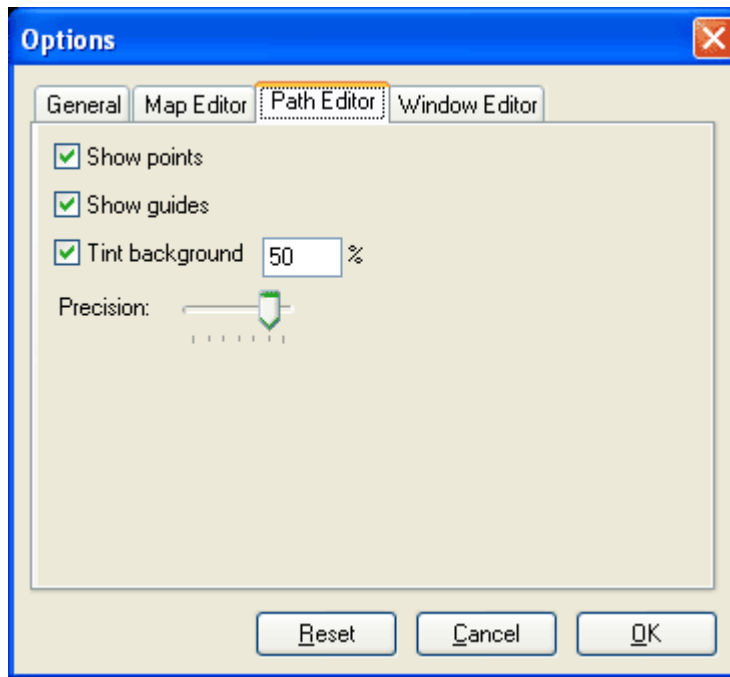
- Check for updates at startup: Platform Studio will check for updates at startup if this option is checked. It is recommended that you enable this option only if you have a constant internet connection (such as Cable or DSL), since an internet connection is required to check for updates.
- Show welcome screen at startup: Platform Studio will show the "Welcome to Platform Studio" screen with buttons to quickly create or open games, access help, receive support. The Welcome screen also has a small "Tip of the Day" area.
- When creating a new game, open the default game file: When checked, Platform Studio will load the default tileset and actions when creating a new game.
- Warn me when my changes will delete existing actions
- Maximum undo level: Changes the maximum amount of undo levels stored. The higher the number, the more memory it will take.
- Number of recent games to display: Specifies the amount of recent games to display in the File menu and in the Welcome screen.
- Default image editor: Specifies the image editor you want Platform Studio to open when editing a tile's image in the Tileset Editor.

Map Editor



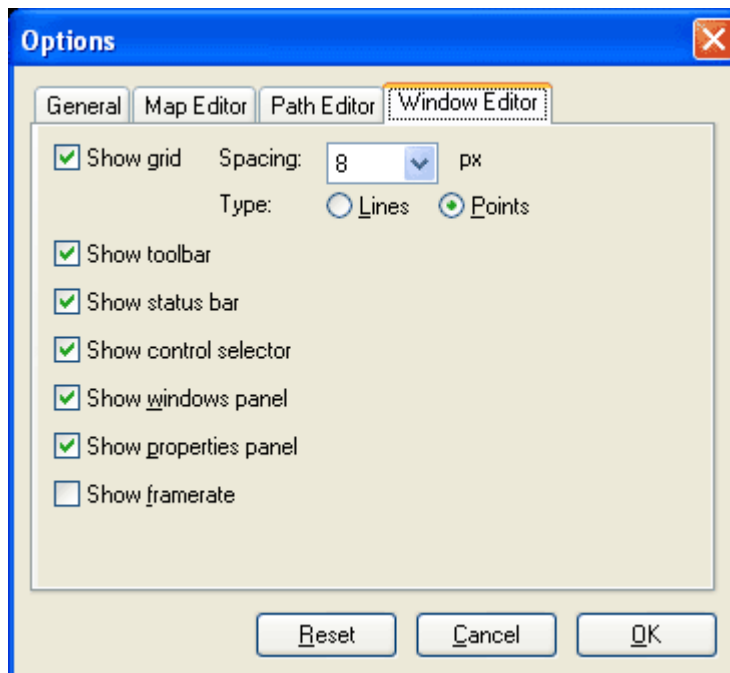
- Show grid: Shows the grid if checked.
- Grid type: Choose between a line grid and a point grid.
- Show toolbars: Shows the toolbars if checked.
- Show status bar: Shows the status bar if checked.
- Show tile selector: Shows the tile selector if checked.
- Show layers panel: Shows the layers panel if checked.
- Show levels panel: Shows the levels panel if checked.
- Show properties panel: Shows the properties panel if checked.
- Draw tile paths: Draws the path line for all tiles that contain paths.
- Show framerate: Shows the framerate (frames per second) at the top-left corner of the Level Editor's edit area when selected. This option is useful to get a rough estimate of how fast your game will run at different points in your level, but will only be useful if you resize the Level Editor to the size of your game window, since it takes more work to draw more tiles, and less work to draw less.

Path Editor



- Show points: Shows the points of the path if checked.
- Show guides: Shows the blue guide lines of the path if checked.
- Tint background: Tints the background to make it easier to see the path if checked. 0% is not tinted, and 100% is solid black.
- Precision: Controls the precision of which to draw the path line. The higher the number, the slower it will draw.

Window Editor



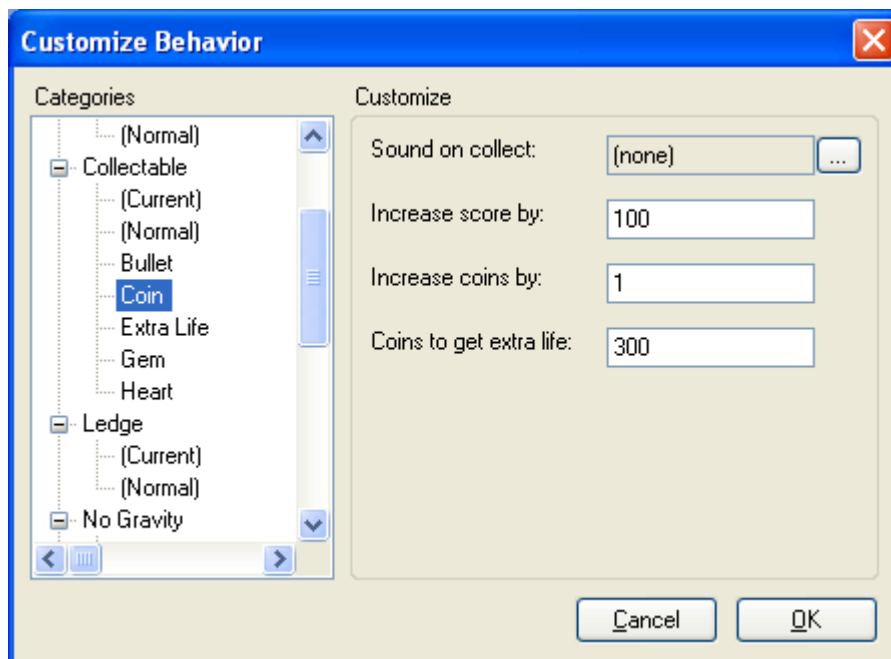
- Show grid: Shows the grid if checked.
- Spacing: Controls the grid spacing in pixels. The smaller the grid, the finer control you have.

- Type: Choose between a line grid and a point grid.
- Show toolbar: Shows the toolbar if checked.
- Show status bar: Shows the status bar if checked.
- Show control selector: Shows control tile selector if checked.
- Show properties panel: Shows the properties panel if checked.
- Show framerate: Shows the framerate (frames per second) if checked at the top-left corner of the edit area.

Copyright 2005, First Productions, Inc.



Customize Behavior up

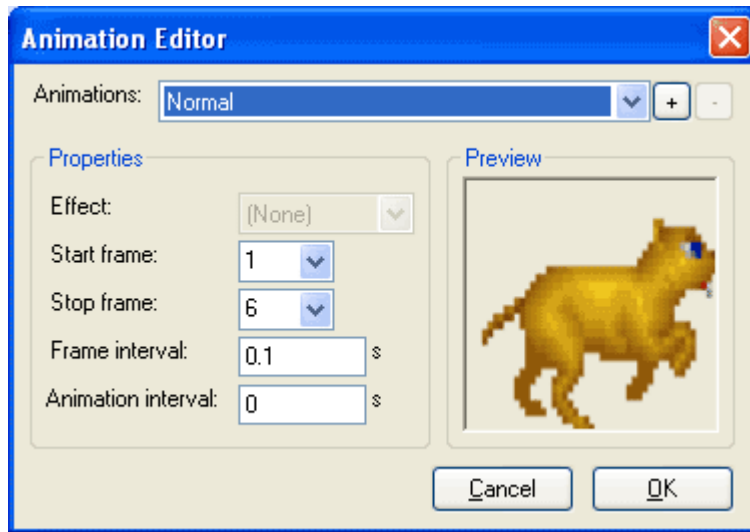


- Categories: Select a category to customize it. Note: If you select something like Bullet, you do not have to use it for just bullets. You can also use it for things such as poison.
- Customize: Sound on Collect: Click (...) to customize the sound.
- Cancel button: Closes the dialog and cancels.
- OK button: Closes this dialog and saves your changes. All actions for the current tile will be deleted and replaced with new ones based on the parameters you give it.

Copyright 2005, First Productions, Inc.



Animation Editor up

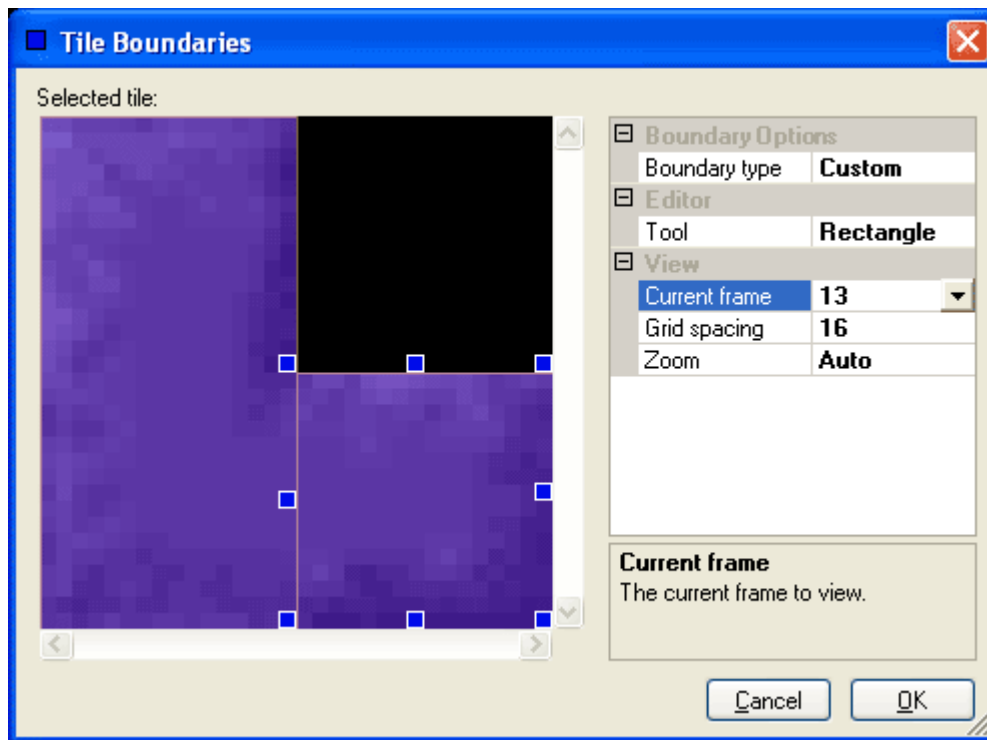


- Animations: Select the current animation you want to work with.
- (+) and (-) buttons: Add or remove animations. To play custom animations, use the *Set Tile Animation* action in the Action Editor.
- Effect: Only for Disappear animations. Choose None to show a normal animation. To show a dynamically generated effect, choose one of the options, such as fade or fall. If you choose an effect, the start Start frame and Stop frame options are disabled, and Platform Studio will use the current frame of the Normal animation. Also, the Frame interval box changes to an Animation length box. Specify how long you want the animation to play for, in seconds.
- Start frame: The first frame of the animation.
- Stop frame: The last frame of the animation. Note: You can make the animation play backwards by making the stop frame less than the start frame. You can create frames by changing the section width and section height of the current tile in the [Tileset Editor](#). To determine the frame order, Platform Studio goes first from the left section to the right section, and then from top to bottom.
- Frame interval: The interval (in seconds) between two frames. A value of 0.1 is equivalent to 10 FPS, and a value of 0.05 is equivalent to 20 FPS.
- Animation interval: The time (in seconds) to wait between two cycles of the animation. This value is usually set to 0.
- Cancel button: Closes the dialog and cancels.
- OK button: Closes the dialog and saves the changes.

Copyright 2005, First Productions, Inc.



Tile Boundaries Editor up



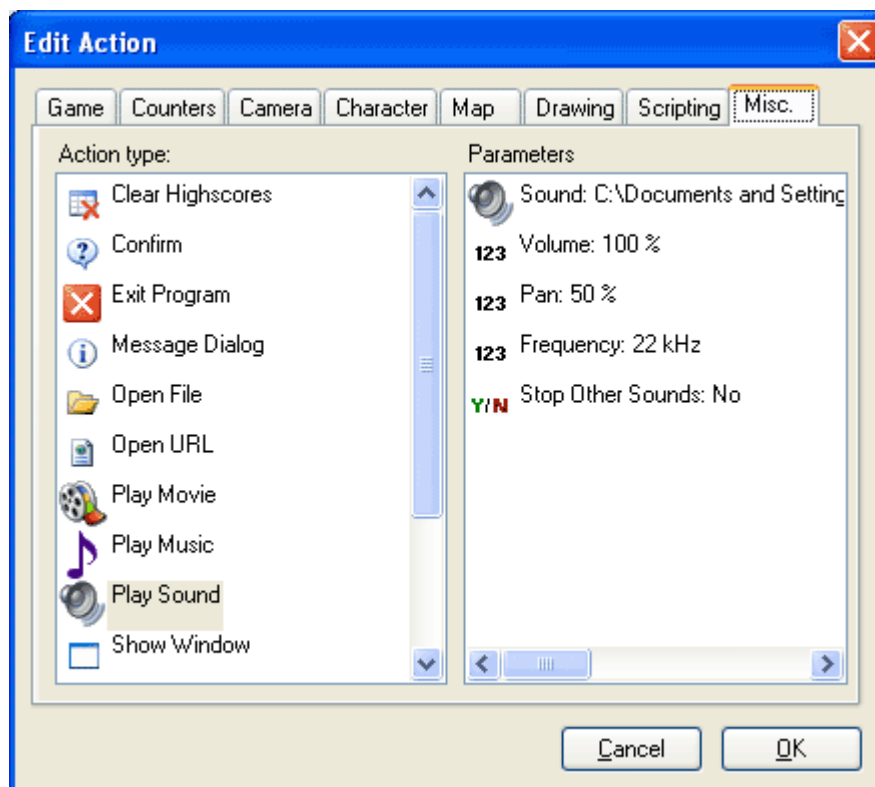
- **Selected tile:** The preview of the current frame of the selected tile (to change the frame, change the *Current frame* property). You may do these actions only if the Boundary type is set to OneCustom or Custom: To select a shape, click it. You can move it by dragging it, resize it by dragging the handles, or delete it by pressing the delete key. To draw with the current shape, click and drag with the mouse in an empty location.
- **Boundary type:**
 - **Default:** The only option available in previous versions of Platform Studio. This can slow down your game by magnitudes if a tile has a significant amount of empty space around it, since Platform Studio will be forced to detect in real-time where the boundaries of a tile lie. Although this is the easiest option to configure, since Platform Studio will do all the work, but it is recommended that you choose one of the other options.
 - **Rectangular:** This option is the fastest but the least accurate, except in rectangular tiles with no empty space around them. Platform Studio will detect a collision with this tile if an object touches its bounding rectangle, even if this includes empty space. This option is recommended for tiles with no empty space around them.
 - **OneCustom:** This option is roughly as fast as Rectangular, but allows greater customization. You may draw rectangles, ellipses, and triangles to specify the tile's boundaries. However, this applies to ALL frames of the tile.
 - **Custom:** This option is roughly as fast as Rectangular, and allows the most customization out of all the boundary types. You may draw shapes to specify the boundaries for each frame of the tile. However, if a tile has a lot of frames, this can be tedious. This option may not be used for animated tiles.
- **Tool:** Choose between Rectangle, Circle, Triangle 1, 2, 3, and 4 (all right triangles - the hypotenuse is in the numbered quadrant of the xy-plane), and Erase. You can also click a shape and press the delete key to erase it.
- **Current frame:** The currently active frame.
- **Grid spacing:** The amount in pixels to snap the shape to when drawing.

- Zoom: The amount to zoom in. The default value is Auto.
- Cancel button: Closes the editor and discards any changes that have been made.
- OK button: Closes the editor and saves the changes that have been made.

Copyright 2005, First Productions, Inc.



Edit Action Up

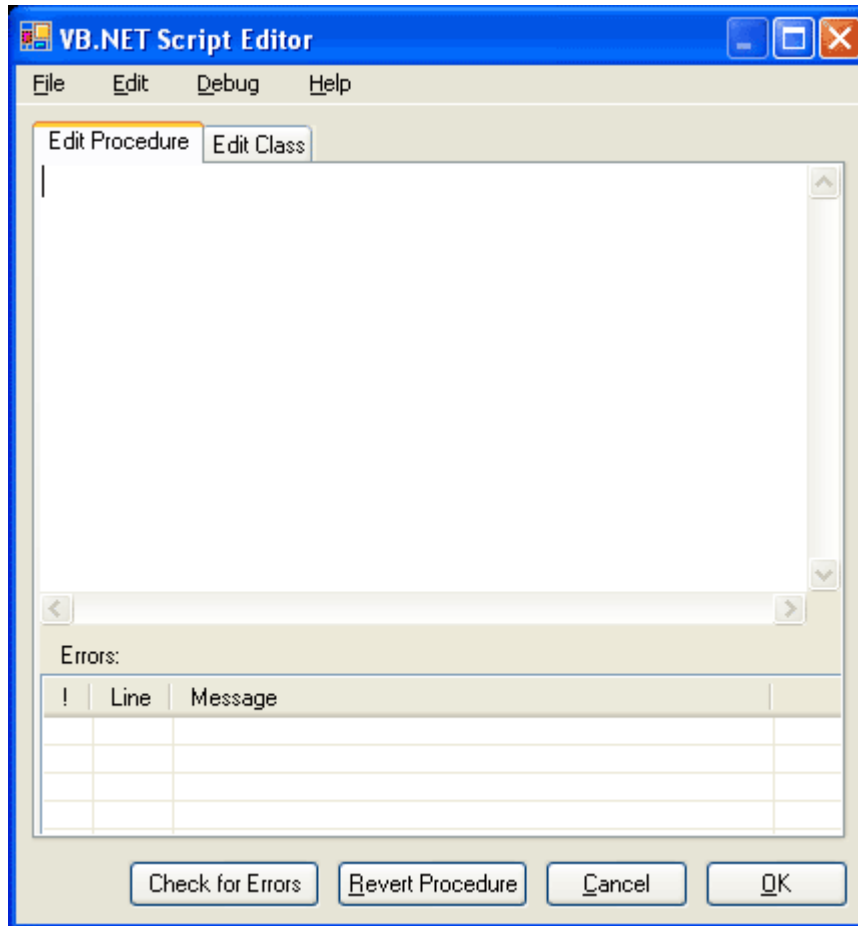


- Tabs at top: Choose the category of the action.
- Action type: Choose the action you want to perform.
- Parameters: The list of parameters you can set for the selected action. When you select an action, the parameters are set to the defaults. Double-click a parameter to edit the value.
- Cancel button: Closes the dialog and cancels.
- OK button: Closes the dialog and saves your changes.

Copyright 2005, First Productions, Inc.



Script Editor up



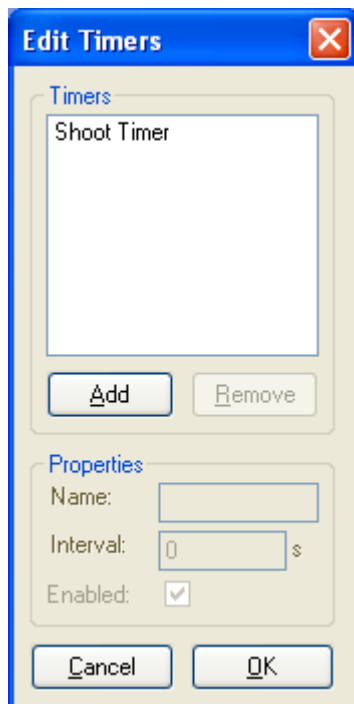
- Menus
 - File
 - Import: Imports a text file into the current view. All code in the current view will be lost.
 - Export
 - Plain Text: Exports a plain text file containing the text in the current view.
 - Rich Text: Exports a rich text file containing the text in the current view.
 - Revert: Reverts the code in the current view to what it used to be before the script editor was opened.
 - Close
 - Save changes: Closes the script editor, saving changes.
 - Discard changes: Closes the script editor, discarding changes.
 - Edit
 - Undo: Undoes the last action.
 - Redo: Redoes the last undid action.
 - Cut: Copies the selected text to the clipboard, and then deletes the selected text.

- Copy: Copies the selected text to the clipboard.
- Paste: Copies the contents of the clipboard to the current view.
- Delete: Deletes the selected text.
- Select All: Selects all the text in the current view.
- Known Objects: Inserts the name of a known object, followed by a dot (.), and then pops up a list of methods, properties, and fields that belong to that object.
- Debug
 - Lists all the errors in the script in the error list.
- Help
 - VB.NET Help: Opens the [Visual Basic .NET section of the MSDN website](#) in a new internet browser.
- Edit Procedure: This tab lets contains the code for the current action.
- Edit Class: This tab contains the global code. For more information about scope, click [here](#).
- Errors: Contains the list of errors in your code for the current view. Click an error to highlight the line containing it in your script.
- Check for Errors button: Lists all the errors in the script in the error list.
- Revert Procedure/Revert Class button: Reverts the code in the current view to what it used to be before the script editor was opened.
- Cancel button: Closes the script editor, discarding changes.
- OK button: Closes the script editor, saving changes.

Copyright 2005, First Productions, Inc.



Edit Timers up

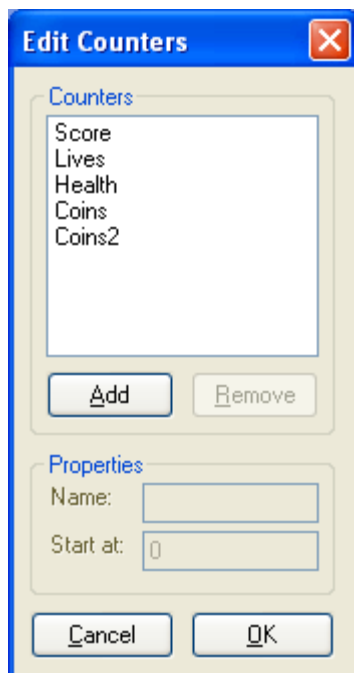


- Timers list: Contains a list of the timers in your game. Click an item to select it.
- Add button: Adds a new timer to your game.
- Remove button: Deletes the selected timer and its associated actions.
- Name: Specify the name of the selected timer.
- Interval: Specify the interval of the selected timer (how often it triggers the Tick trigger).
- Enabled: Performs the Tick trigger if checked; otherwise, it does not.
- Cancel button: Closes the dialog, discarding changes.
- OK button: Closes the dialog, saving changes.

Copyright 2005, First Productions, Inc.



Edit Counters [up](#)

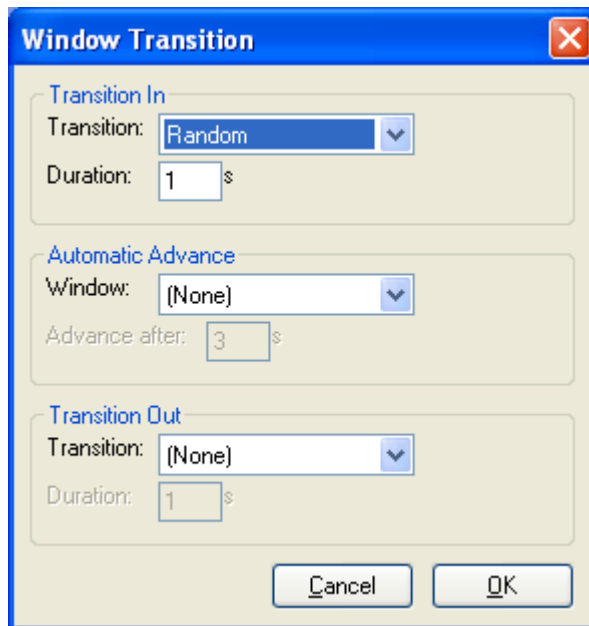


- Counters list: Contains a list of the counters in your game. Click an item to select it.
- Add button: Adds a new counter to your game.
- Remove button: Deletes the selected counter and its associated actions.
- Name: Specify the name of the selected counter.
- Start at: Specify the value to set the selected counter to when starting a new game.
- Cancel button: Closes the dialog, discarding changes.
- OK button: Closes the dialog, saving changes.

Copyright 2005, First Productions, Inc.



Window Transition up



- Transition In
 - Transition: Choose the transition you want Platform Studio to play when your window is shown. Choose (None) to have no transition.
 - Duration: Specify how long you want the transition to last for (in seconds).
- Automatic Advance
 - Window: Choose what window you want Platform Studio to advance to after a certain amount of time. Choose (None) to not advance to another window.
 - Advance after: Specify how long to wait until Platform Studio advance's to the specified window.
- Transition Out
 - Transition: Choose the transition you want Platform Studio to play when the user leaves the current window. Choose (None) to have no transition.
 - Duration: Specify how long you want the transition to last for (in seconds).
- Cancel button: Closes the dialog, discarding changes.
- OK button: Closes the dialog, saving changes.

Copyright 2005, First Productions, Inc.



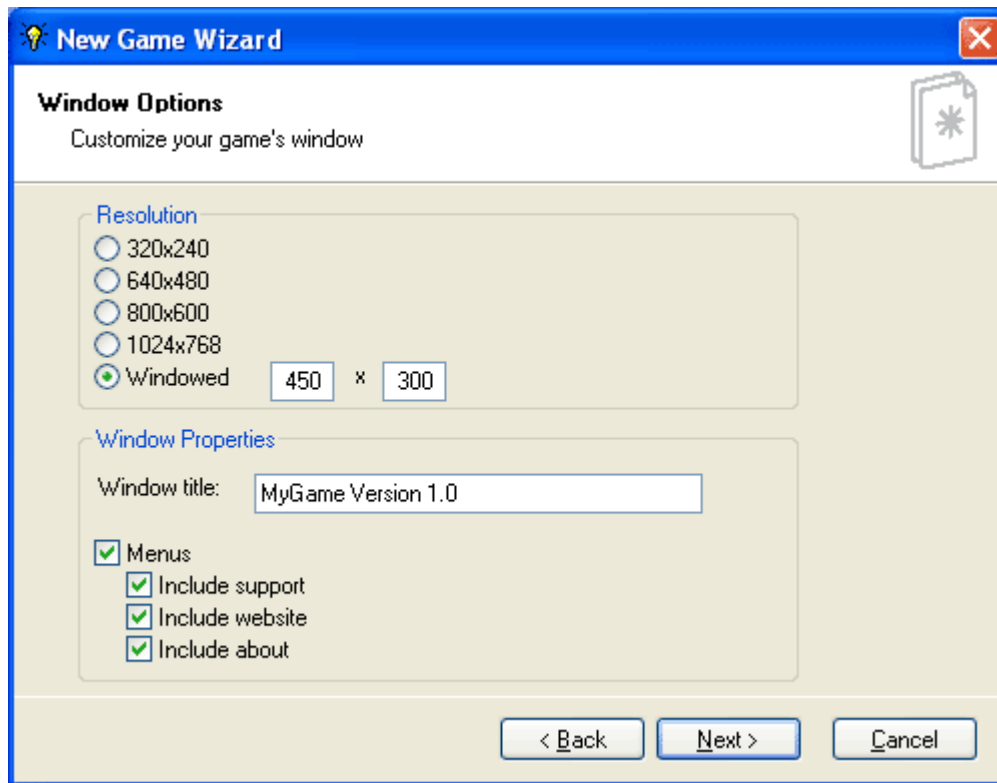
Window Scheme Wizard up

Step 1



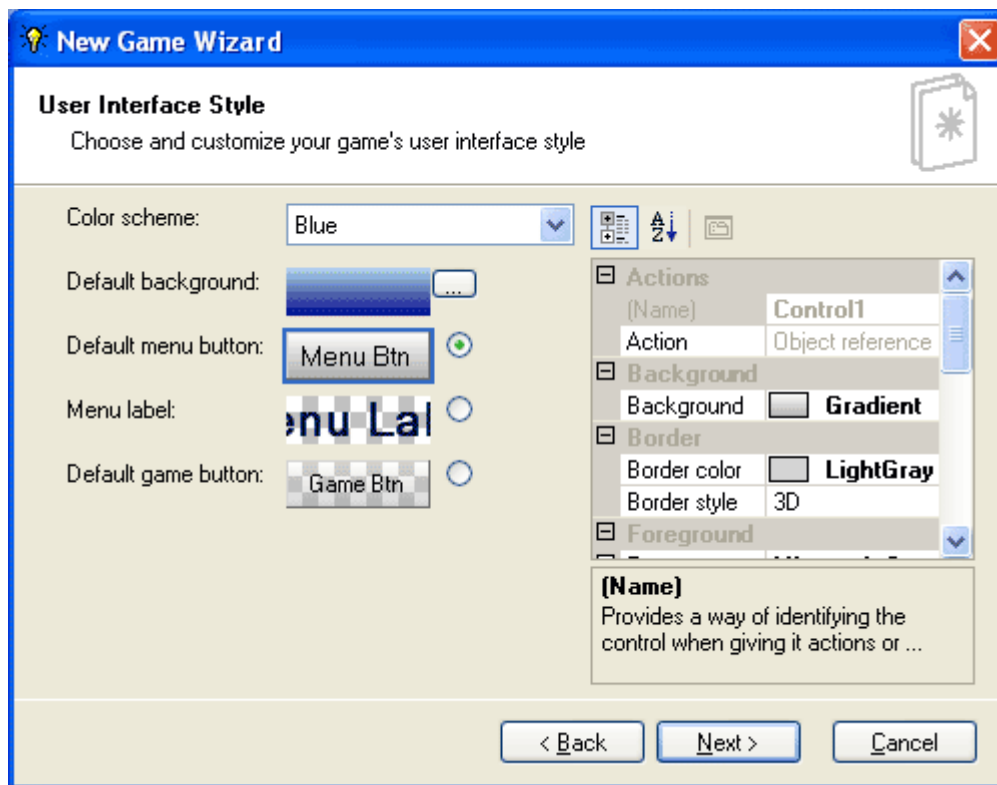
- Click Next to continue.

Step 2



- Resolution: Specifies your game's screen resolution. The smaller the resolution, the faster your game will be, but you will have less space to work with. Choose Windowed if you want your game to run in a window, instead of fullscreen.
- Window title: Change the title of your window.
- Menus: Check if you want your window to contain menus.
- Include support: Check if you want your Help menu to include a Support item.
- Include website: Check if you want your Help menu to include a Website item.
- Include about: Check if you want your Help menu to include an About item.

Step 3



- Color scheme: Changes the color scheme for your use interface.
- Default background: Click (...) to change the default background of your windows.
- Default menu button: Click the button and use the property grid to the right to change the default menu button and other non-game buttons.
- Menu label: Click the label and use the property grid to the right to change the default menu label (as well as other labels throughout your game).
- Default game button: Click the button and use the property grid to the right to change the default game button style.

Step 4



- About style: Message dialog: Shows the about information in a message box.
- About style: Window: Shows the about information in a window.
- Help style: Window: Shows the help information in a window.
- Help style: Window: Edit Help Text: Edits the help text to be shown in the help window.
- Help style: External file: Opens a help file when the user clicks the Help button.
- Help style: External file: (...): Prompts you for a file.

Copyright 2005, First Productions, Inc.