



# DS Game Maker

The next step in DS homebrew

Manual Revision 1.6 by James Garner

2010

---

# CREATING YOUR OWN GAMES

It is fair to say that most enjoy playing computer games either alone or with their friends, but what is actually more fun, is making your *own* games and letting others play them. Unfortunately, the DS is not an easy platform for which to develop and teams of 50 professionals can take more than a year to finish a game which you would buy at a store. These people are highly experienced: designers, sound technicians, researchers and programmers, etc.

Does this mean creating games for the Nintendo DS is impossible? Luckily, No! You should not be expecting to create Super Mario Bros. or Metroid Prime Hunters within a few weeks. Simpler games like Pacman and Space Invaders are also fun to play and much easier to create, but they still require good programming skills and understanding of the DS' hardware to handle graphics and sound, for example.

So here comes DS Game Maker which was created to make game creation like the above much easier. There is no need to have knowledge of programming or how the inside of the DS works. In fact, DS Game Maker simplifies game creation so it is possible, with commitment, to make great games in little time. You can import your own resources, including images and sounds. DS Game Maker is room-orientated (don't worry!). Every room includes backgrounds and sprites (moving images) and a list of *actions* to make your games come alive.

If you want full control of homebrew development for the DS, there is extensive support for the use of your own code at many places within the program. You may change the code for every action that comes with the software. We made the software simple to use but did not limit its use!

DS Game Maker focuses really on 2D games and it is not in any way designed to create 3D games. However, don't let this put you down. *Street Fighter* has proved there is still a market for 2D games and many games such as *Command & Conquer* are 2 dimensional games even though they look to be 3 dimensional. Also, designing 2D games is much easier and faster in development for any console.

There are two editions of DS Game Maker, the Free Edition and the Pro Edition. The Free Edition can be used free of charge but it shows a DS Game Maker advert when the game is run and it is also limited in functionality. You are free to distribute any games you make with the software, whether you use the Free or Pro version, we don't mind. You don't even have to mention you used DS Game Maker in the credits. With this in mind, you are strongly encouraged to upgrade to the Pro Edition of DS Game Maker. It extends the functionality of the software and will remove the advertising splash screen before the game launches. It will also support the further development of DS Game Maker.

This help file will tell you all that you need to know about DS Game Maker and how to create your own games with it. Please do realize however, that even with a program such as DS Game Maker, designing games is not completely effortless. Many aspects are important in creating a successful game, including planning, graphics, user interaction and music. These are elements that you should think about when creating your game to impress your audience. Start with easy examples and build up your knowledge of the program and skills with it. You will soon realize that creating your own games is great fun.

---

# INSTALLATION

You have probably already installed DS Game Maker but if not, here is how to install the program.

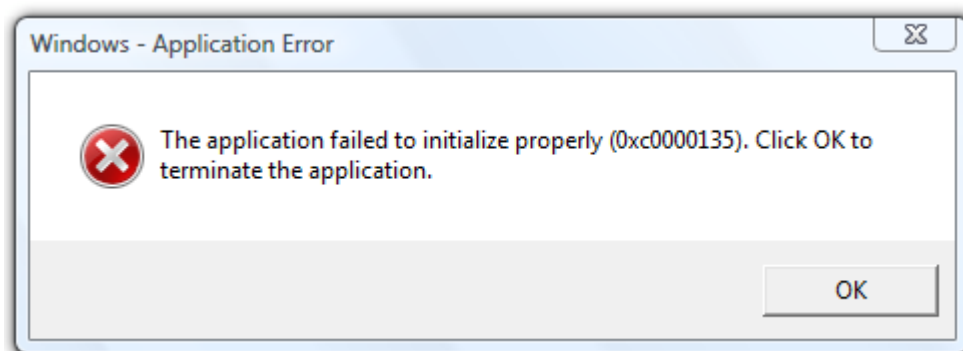
To begin you must download the installer from the internet. This is available from:

**<http://download.dsgamemaker.com>**

Note that the link above will always give you the latest version of the software. Other websites may host their own mirrors of the DS Game Maker installer. The issue with this is that other websites do not update their copies as updates are released (and hence as bugs are fixed, things are added etc). Also by downloading the installer from our website, you can be sure it is 100% virus/spyware free.

## MICROSOFT .NET FRAMEWORK

In order to run DS Game Maker, you require Microsoft .NET Framework 3.5 to be installed on your computer. Microsoft Windows 7 and Windows Vista include this (so you can run the application straight away and ignore this step). Most Windows XP users should have it installed too (from Windows Update), but this is not always the case. So, if you do not have the framework you will receive the following error:



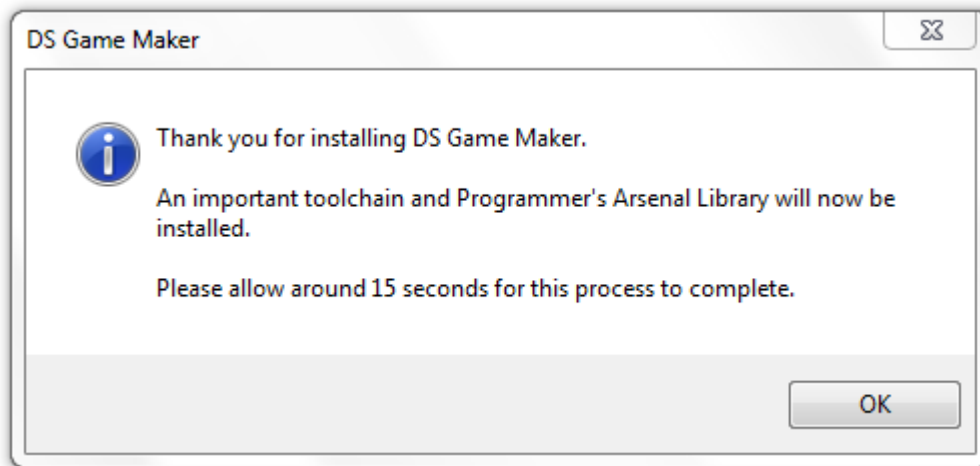
To install the framework simply download the installer from the Microsoft website:

**<http://bit.ly/MicrosoftDotNet>**

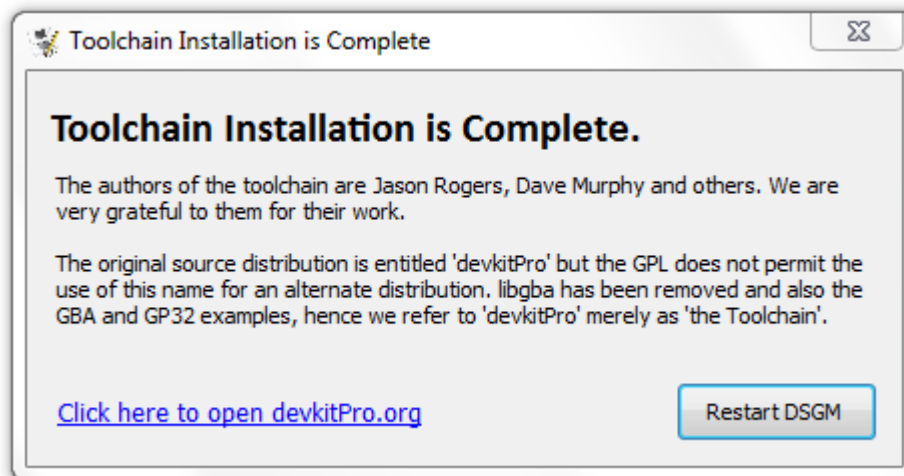
Now you may run DS Game Maker.

## INSTALLING THE DEVELOPMENT TOOLCHAIN

The first time you run DS Game Maker you are prompted to install the development Toolchain.



The installation of the development Toolchain and Programmer's Arsenal Library is quick and painless. After 15 seconds it is complete:



Simple click the "Restart DSGM" button to restart DS Game Maker, and you are ready to start creating DS games.

---

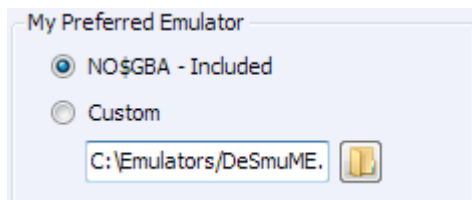
# HOW DO I PLAY/TEST MY GAMES?

There are 2 ways in which you can test and enjoy your finished game, demo or WIP. The first is running the game on the Nintendo DS and the second is emulating it on your PC. Emulation is much quicker and preferred for testing, particularly when fixing bugs or with small updates.

## PHYSICAL CONSOLE (DS) WITH FLASHCART

- A MicroSD memory card that holds the games (and other media such as *music and films*). Note that 2GB is enough for 99% of people.
  - [Kingston MicroSD 2GB \(Japan\)](#)
  - [Kingston microSDHC 4GB](#)
  - [SanDisk microSDHC 8GB \(Class 4\)](#)
- A Flash Cart, which allows the console to read files from the above MicroSD memory card. All of these flashcarts are compatible with the original DS, DS Lite and DSi, including DSi Firmware version 1.4:
  - [R4i-SDHC](#)
  - [Acekard 2i](#)

## EMULATOR WITH NO\$GBA OR OTHER



In order to use an Emulator and test your games without a physical Nintendo DS console, open up the Options dialog in DS Game Maker by going to 'Tools' on the menu and selecting 'Options...'. You may also access this dialog with the keyboard shortcut Control + P.

Ensure that the first option is selected under the heading 'My Preferred Emulator', as shown above. If you wish to use another emulator, for example DeSmuME, you may select the second option and type the path to the executable in the text box.

# UPGRADING TO THE PRO EDITION

The **Free Edition** is really meant for those taking their first steps on the path of developing games. It can be used freely but is limited in its functionality and shows a DS Game Maker splash screen on games exported from the software. When you are using the software regularly, you are strongly encouraged to upgrade to the Pro Edition.

The **Pro Edition** contains considerably more functionality and does not display the splash screen:

- The limit of maximum 3 rooms in your game is removed
- You may add sound including sound effects and background music
- You may add saving and loading to your games.

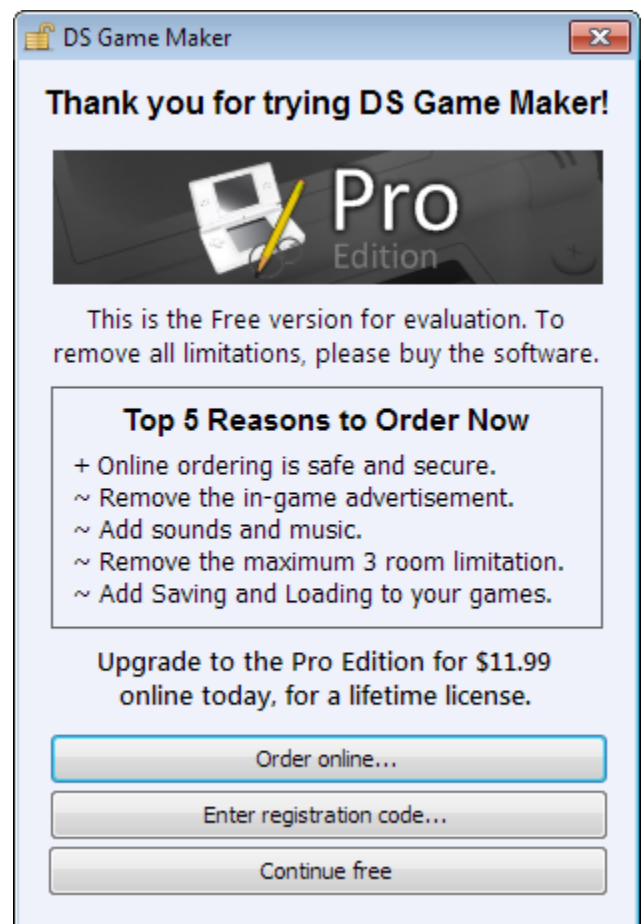
More details, as always are available on the website:

<http://order.dsgamemaker.com/>

Upgrading to the Pro Edition costs just \$11.99. In other currencies that is 8.00 Euros or 7.31 British pounds. This is a onetime fee that is valid for all current and future versions of DS Game Maker, hence a lifetime license.

**When you are running the Free Edition and you start DS Game Maker, the window to the right is shown.**

You can also get this window by clicking the "Upgrade to Pro" button in DS Game Maker at any time:



The easiest way to purchase the software is to pay online. You can pay by Paypal or Google Checkout. Both these methods of paying for DS Game Maker are incredibly secure, we never see any of your financial details. After payment you will receive your Registration code instantly if you pay by PayPal, and within 24 hours if you choose Google Checkout.

You may also too pay by mail, but it takes time for post to arrive especially if you do not live in the United Kingdom. If you pay this way, you must accept that there will be a delay.

---

# LIMITS OF DS DEVELOPMENT

It would be great to live in a world where any kind of development was not limited in some way by the software or hardware of a games console. Unfortunately, the DS is actually very low spec and when designing games, it has to be taken into consideration.

When creating games, you may have to think about your use of high quality graphics and background music, etc. The following limits are in place as results of the DS' hardware, PALib and DS Game Maker functionality:

- Maximum of 32 256-color sprites in the project
- Maximum of 128 instances of the above sprites per screen at once
- 4mb cap on all resources except MP3s (DS RAM is 4mb)
- MP3s require a low bitrate

As a new user, I would really not worry about these limits. You can cross this bridge when you come to it. It is just useful and important in my opinion, to inform users about this.

---

## PIECING IT TOGETHER

Before diving into the possibilities of DS Game Maker it is important that you understand the key concepts behind games created using the software. Games created in DS Game Maker take place in one or more *rooms* (they are flat, not 3D, but may contain 3D looking graphics). In these rooms you place *sprites*, for example walls, moving balls, monsters or the main character, and you also set backgrounds for the room like wallpaper. Some sprites will move around and react to player input, for example from the stylus or DS' buttons. Some sprites trigger logic when they collide with other sprites, for example when the play meets a monster he might die.

Once you have added sprites it is time to define the *rooms* in which they are placed. Rooms can be used as levels in your game or as menu screens, or for anything. Rooms, have 2 *backgrounds* (one for each screen). Next you can place instances of your sprites in the room. You can place multiple instances of the same sprite in the room (you only need to add 1 wall sprite but you can use it in many places, like to make a maze).

The most important elements in your games are *actions*. These are put simply "things that happen" and some examples include Play Sound, Switch Room, Output text, Delete Sprite. These actions interact with the sprites you have plotted, other rooms and other actions with the list.

Each room also has "View" settings (similar to those found in *Game Maker*). With this feature, you are able to add sprites off the screen and scroll them into view as the player navigates the room. See it as an overhead camera that is already programmed for you!

### **So in summary:**

- Rooms: The program is room orientated. Wherever you are in the game, you are in a room.
- Sprites: Small images that are either still or animated. You place them in rooms and the room's actions (logic) moves them about, sometimes according to player input (e.g. stylus) and sometimes automatically.
- Backgrounds: Large images usually the size of the DS screen applied to a room. They can be scrolled but commonly they are there simply to make the room look nice.
- Sounds: Sections of audio, either background music (MP3) or sound effects (RAW).
- Fonts: Different fonts that can be used so that text appears differently.

---

## VERSATILITY

Many people compare DS Game Maker to Game Maker by Mark Overmars. In reality they are different products by different people (one is a university professor; one is a 16 year old). DS development is different from computer development so comparing the programs is difficult on that score. For a start the Nintendo DS has considerably less computing power than a Windows machine (typically 128 times less RAM, or for me, 512 times less!).

Game Maker has an objects and events system. DS Game Maker does not: but you will see this as an advantage once you use the program for a while. To begin justification, the control that comes from DS Game Maker's way of doing things cannot be achieved in Game Maker (the system is too limiting). With room orientation you are designing games as close to programming as possible. It's a bit more difficult, since you have to track events yourself, but it really is not hard.

Because actions are added to the room, you control everything in the room from that point. You can manipulate all the sprites at once for example. The program does less work for you but at the same time you have much more control over the exact behavior of your game and you will learn much more.

The differences between Game Maker and DS Game Maker should not be considered and ideally people would consider them different products that have different ways of letting you design games. Some do not share this view and prefer the easy ride of Game Maker.

Game Maker may be easier to use initially but the program will never have the control that DS Game Maker offers even in the Free Edition. We can always educate people on how to use the software to compensate for its differences whereas with Game Maker, the product is not per se at fault, but it is limited, and there is nothing they can do.

So onwards with DS development! Now we have explored the whole issue with Game Maker, you will see that the reality of making games for the DS really is bright.



---

# A VERY SIMPLE GAME FOR YOU TO TRY

We are not going to begin with an immensely complicated game or even something like Space Invaders which is still quite a challenge for a new user. This is more of a demonstration than a game (there is no scoring). Before you begin a game, you should set out what you want to achieve (like a project proposal or idea).

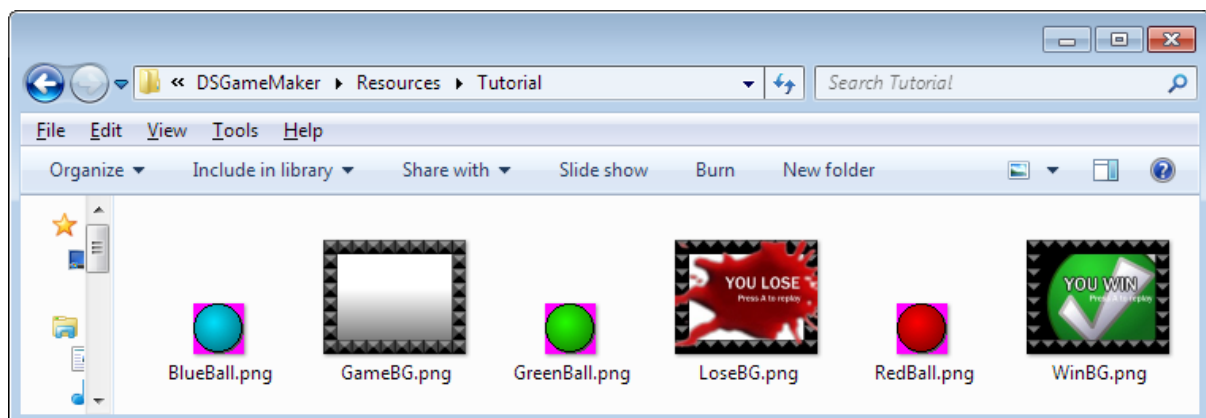
**The player must use the DS' stylus to move the green ball to the red ball, without touching the blue ball. If they complete the challenge, they are shown the 'Win' screen. If they do not complete the challenge (touch the blue ball), they are shown the 'Lose' screen.**

In order to create the game, we will need the following resources:

- 3 ball sprites (e.g. red, blue, green balls).
- Background for the game
- 'WIN' background
- 'LOSE' background

These resources are included with DS Game Maker and may be found in the folder:

**C:\DSGameMaker\Resources\Tutorial**



Now we should plan the game. If you do this you are much more likely to succeed with your project.

- Game room
  - Game background for both screens.
  - 3 ball sprites.
  - The game logic.
- Win room with logic to return to the game
  - No sprites
  - Win background on top screen
  - Logic for returning to Game room.
- Lose room with logic to return to the game
  - No sprites
  - Lose background on top screen
  - Logic for returning to Game room.

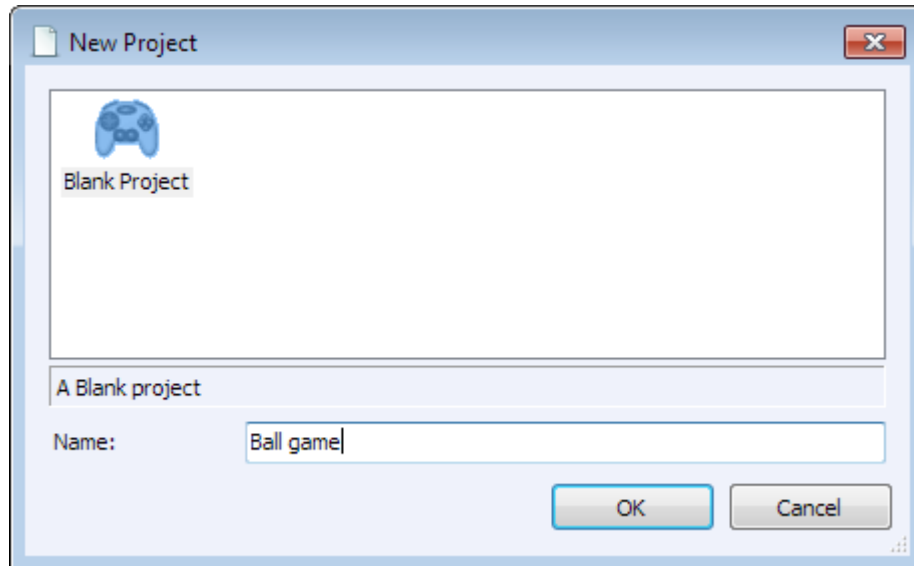
If you don't want to make the game yourself, you can load it from the Tutorial folder:

**C:\DSGameMaker\Resources\Tutorial**

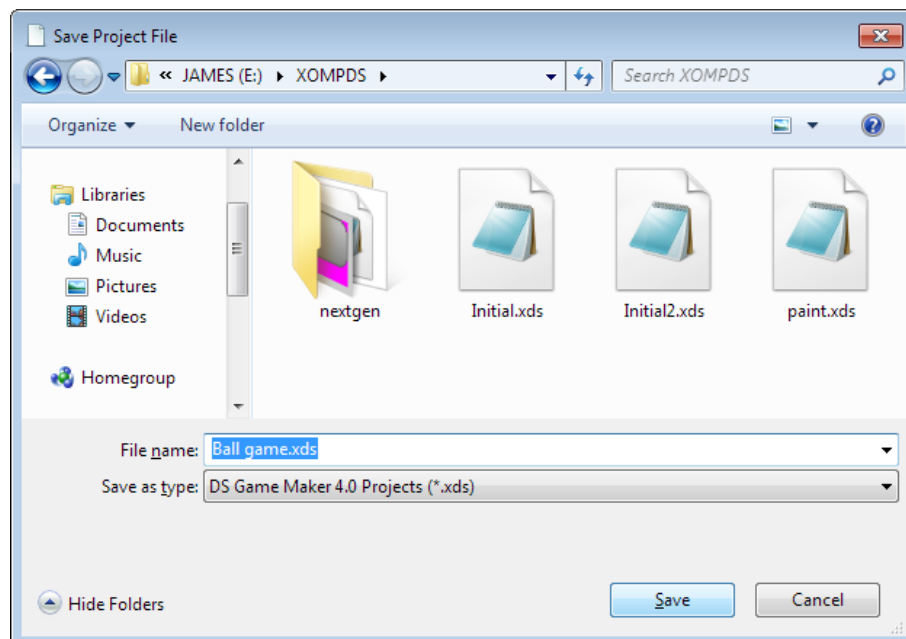
Under the name:

**Very simple ball game.xds**

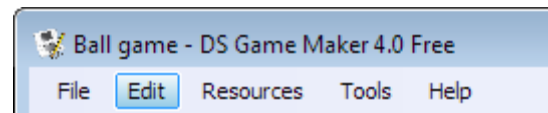
From the **File** menu select "**New Project...**".



Select the "Blank Project" template and enter a suitable name for the game into the textbox.



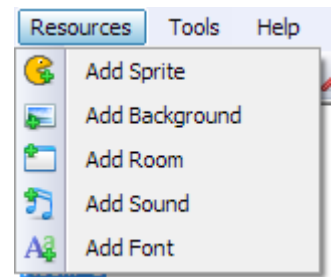
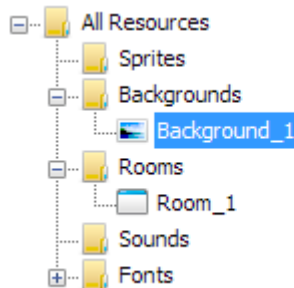
You must save the project before you can begin working on it. You can save it anywhere but keeping your projects organized is advantageous. You may wish to save it in the same folder as the tutorial graphics.



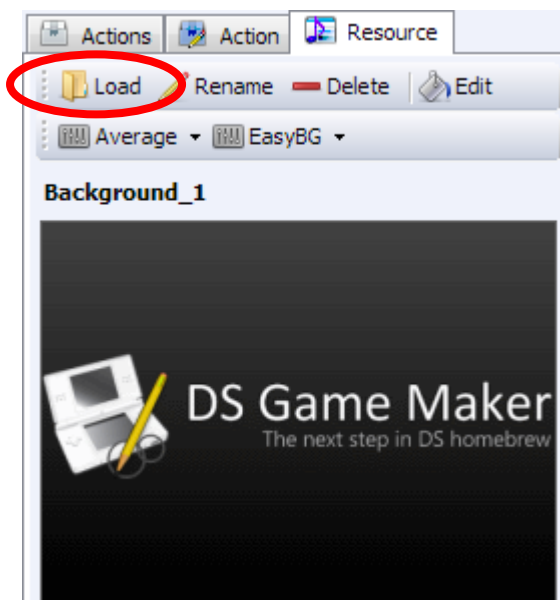
Once the above has been done correctly you will see that the title bar of the program has changed to include your game name. **Now we can begin creating the game!**

From the '**Resources**' menu in the program you can add all of the graphics that are required to make this game.

We will begin by clicking '**Add Background**'.

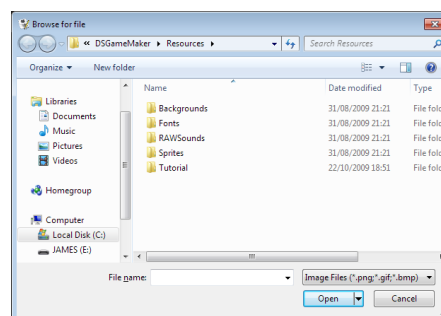


You will see now that in the resources pane at the left of the program a background has appeared. **Double click 'Background\_1'.**

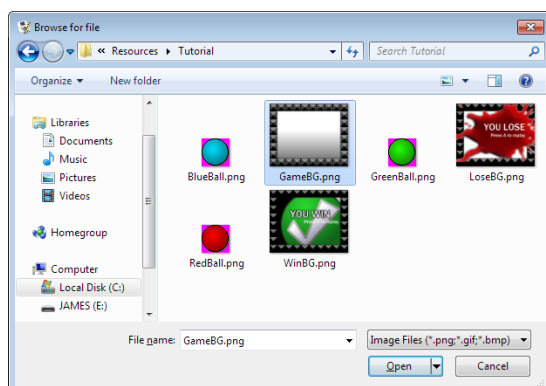


You will see that the background is now shown below the resources pane. We must now change this background so it has our own image (by default, it is the DS Game Maker logo).

To do this, click the '**Load**' button.



You are presented with a dialog to select the graphic.



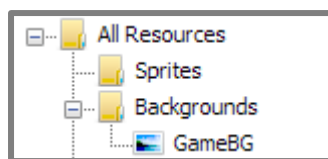
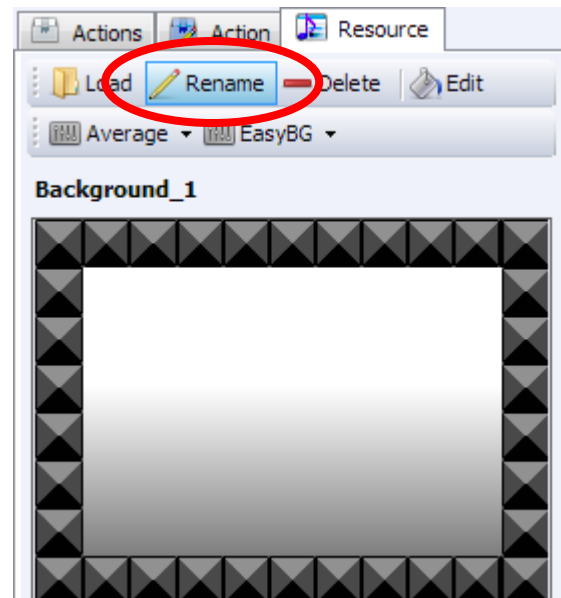
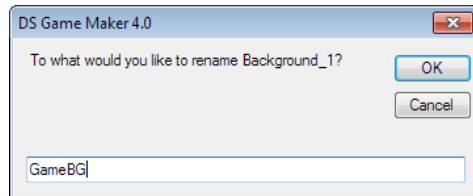
Double click the '**Tutorial**' folder.

Select the '**GameBG.png**' file and click '**Open**'.

The background should have changed for you (see right). We will now give the background a better name. This helps ultimately when you have a larger game with many backgrounds and sprites.

**Click the 'Rename' button.**

In the input dialog type the name 'GameBG' and hit OK:

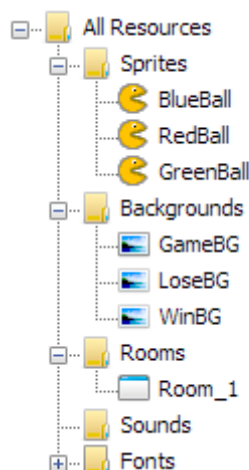


You will see that in the resources pane the background node has been updated with our newly chosen name.

- Add a new background. Load the file 'LoseBG.png' from the 'Tutorial' folder. Name the background 'LoseBG'.
- Add a new background. Load the file 'WinBG.png' from the 'Tutorial' folder. Name the background 'WinBG'.

**We will now add the sprites to the game.** Thankfully the process for this is identical to the above for adding a background. In the 'Resources' menu click the item 'Add Sprite'. A sprite named Sprite\_1 has been added to the resources pane.

*At this point you may find it useful to know that DS Game Maker, when creating a new resource, always looks for the first available numeric slot for the resource's name. If you have Sprite\_1, Sprite\_2 and Sprite\_4 in the project DS Game Maker will not jump to Sprite\_5. It will create a sprite with name Sprite\_3. With this system the program will never create 2 sprites with the same name!*



Double click the sprite in the resources pane (double click 'Sprite\_1'). Click the 'Load' button and in the dialog open the 'Tutorial' folder and select the file 'RedBall.png'. Click 'Open' to close the dialog. Now click the 'Rename' button and type 'RedBall'. Click 'OK' and perform the same checks as with backgrounds: The image for the sprite should have changed and the name in the resources pane should say 'RedBall'.

Do the same for 'BlueBall' loading the file 'BlueBall.png' and for 'GreenBall' loading the file 'GreenBall.png'.

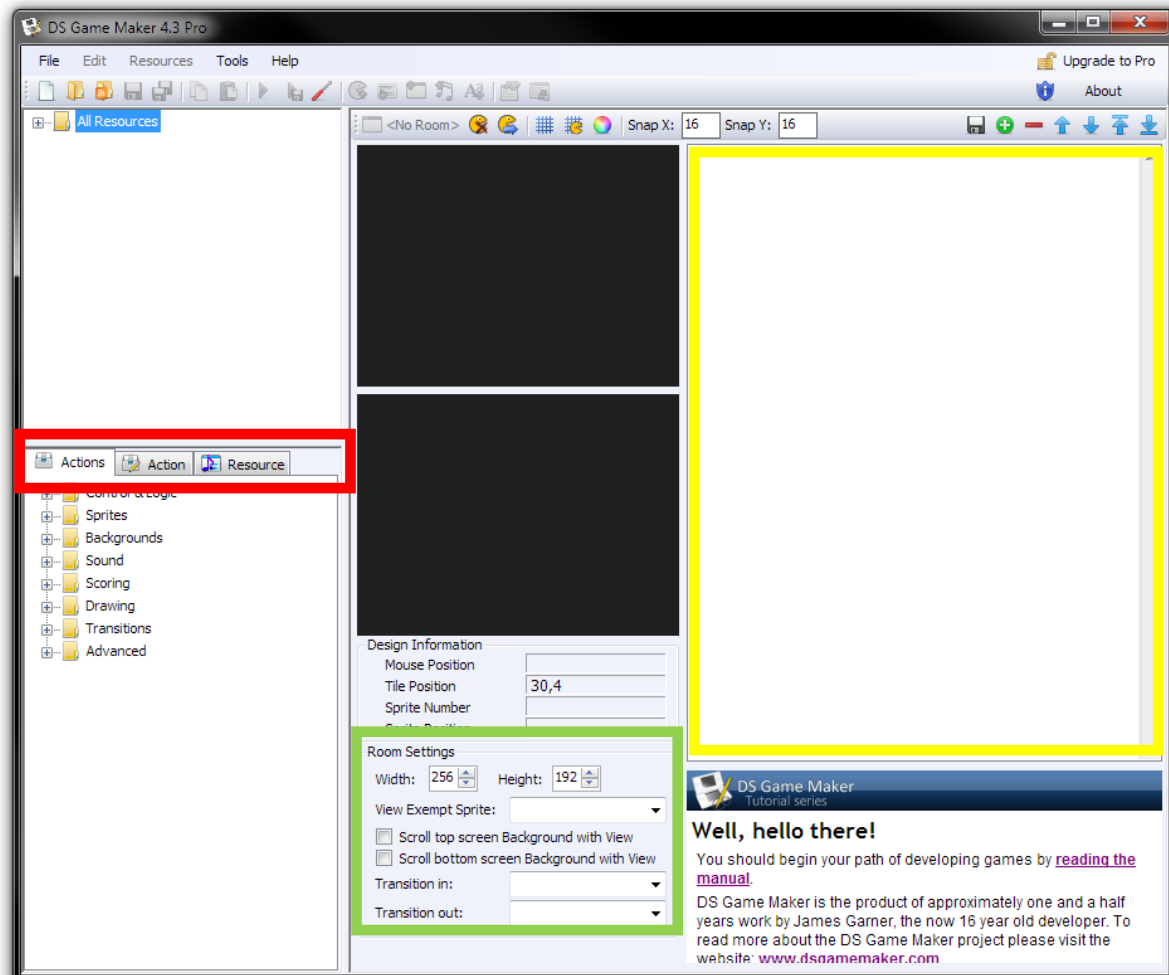
**Now you should check you have done the above correctly and have a resources pane like shown on the left.**

We will now begin to design the first room of our game. Select 'Room\_1' in the resources pane (double click it). Rename the room in the same way as a sprite or background, call it 'GameRoom'. Note that there is no space in between 'Game' and 'Room' (you cannot have spaces in resource names).

At this point we will also add 2 more rooms – 'LoseRoom' and 'WinRoom'. Add 2 rooms and name them appropriately. See the screenshot to the right. You can add resources both using the toolbar and menubar.



**Now we will begin adding the backgrounds and sprites into the game.**



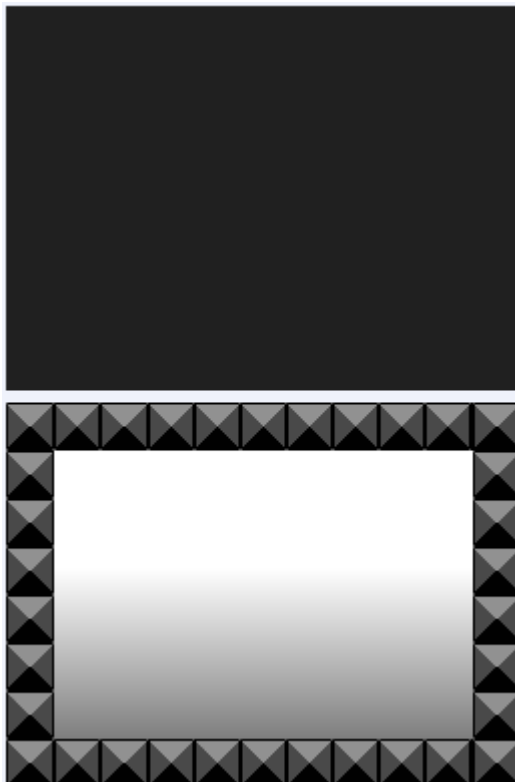
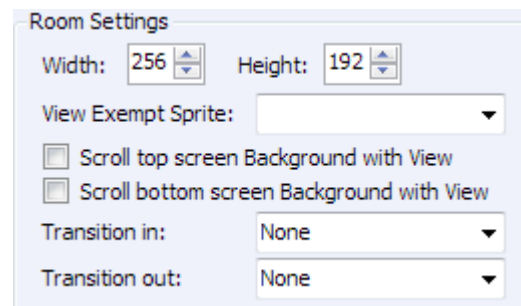
Open 'GameRoom' by double clicking it in the resources pane. You will see that this has happened as the room icon has appeared instead of a sprite or background. The interface of DS Game Maker is explained below:

1. **Room Actions List:** All of the logic (actions) that happen in the room is listed here.
2. **Room Views settings:** Adjust the room size for games where objects exceed the DS screen's visible area. We do not need this for our game.
3. **Tab switching between:**
  - a. List of actions that can be added to the room (all actions on system categorized)
  - b. Panel to edit existing action selected in room from 1, or panel to edit an action about to be added to the room.
  - c. Panel to view/edit selected resource, e.g. sprite, background, font.

Before we begin adding sprites to the game, we must set the view size to the size of the DS screen. The views system enables you to manipulate a “camera” to navigate a large room. This is useful but we do not need it for our game as we will not be adding sprites beyond the constraints of the DS screen size.

The size of the Nintendo DS screen is 256x192 pixels. This is an important fact to remember especially when designing backgrounds for your games. We will make the room the same size as the screen using the Views settings, see right.

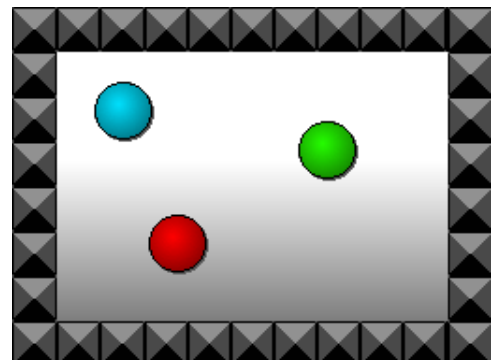
You will notice that the scrollbars on the 2 DS screens have disappeared. This is because now there is no content beyond the screen size to scroll in and out.



Double click 'GameBG' (a background). Now click anywhere in the bottom DS screen to apply the background. See left.

Now select the sprite 'RedBall' by double clicking it in the resources pane. To add the sprite to the room click anywhere on the bottom screen. Click to plot the sprite.


Do the same for the blue ball (double click BlueBall in the resources pane and click in the room to plot it). Also do this for the green ball.



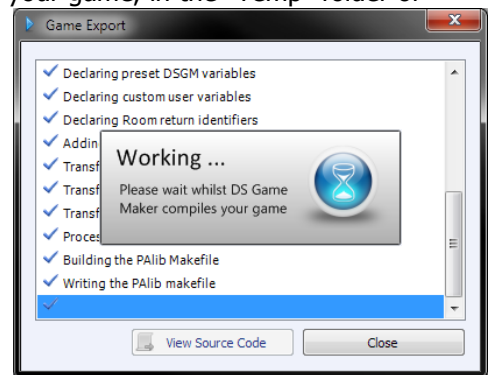
This is a good stage to test whether DS Game Maker will compile the game for us. “Compiling” is the term developers use to mean converting the game design into the finished, distributable product. It is the equivalent of creating an EXE in Game Maker. On the Nintendo DS, a similar executable system is used. The final extension of Nintendo DS games is **NDS**. This means that if you want to share your game online you will upload an NDS file of your game.



console.



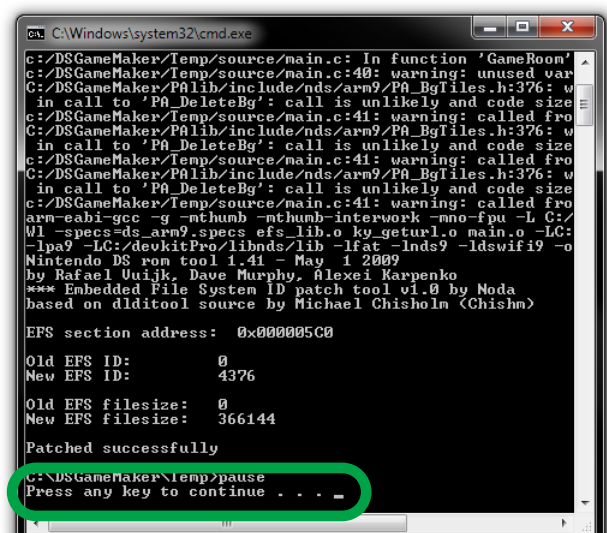
1. DS Game Maker creates the source files and C code for your game, in the "Temp" folder of the software installation directory, typically C:\DSGameMaker
2. A command line tool called 'PAGfx' runs which converts your PC format graphics into a format that works on the Nintendo DS
3. A command prompt window is launched to run the 'make' command, which compiles all of the source files and C code into the Nintendo DS executable
4. DS Game Maker launches NO\$GBA allowing you to test your game.

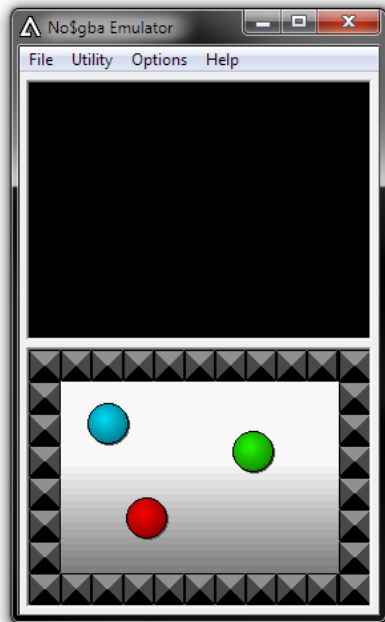


**Go ahead and click the 'Test Game' button or hit F5.**

You will see a command prompt launch, very similar to the one shown to the right here. You should ignore most of the content shown in this window. Unless you have an error you need to fix, it is of little use.

Press any key to continue, e.g. Space bar.





Congratulations if you see the Window on the left. You have successfully created your first DS game, running in NO\$GBA.

**Please see the next section 'Fixing Errors' if you were not able to do the above.**

---

## FIXING ERRORS

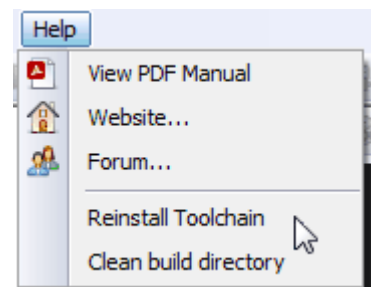
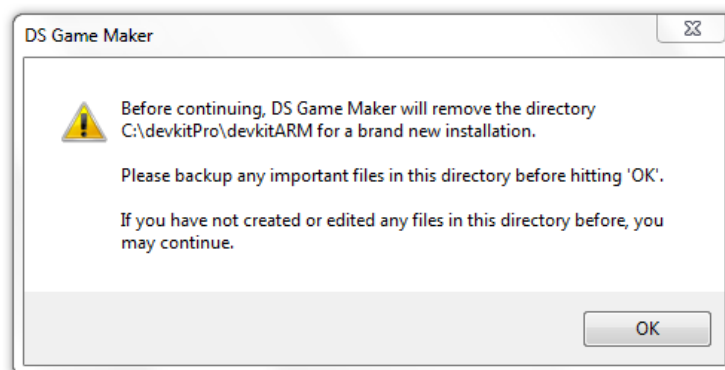
It is better to fix a broken installation here than to press on with game creation and consolidate a problem later.

**This section applies to fixing all errors in general, not errors specific to the game we are creating. The first thing you should try is compiling a completely blank game, with no graphics, actions, etc. This will identify where the problem lies.**

If the blank game compiles and an NDS file is created, you know that the error is in the game you are making.

### A BLANK GAME DOES NOT COMPILE... NOTHING COMPILES

The first thing to try is reinstalling the toolchain. Most times this corrects the error. To do this, select 'Reinstall Toolchain' from the Help menu in the program. Note that a project does not have to be loaded to do so.



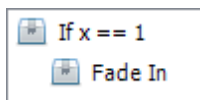
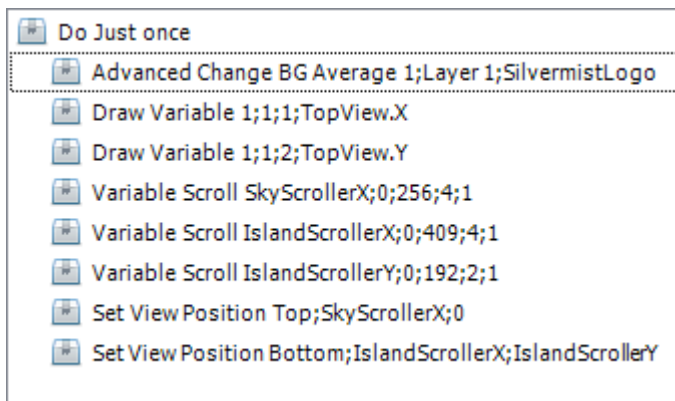


## FINDING ERRORS IN A GAME

### LOGIC ERRORS

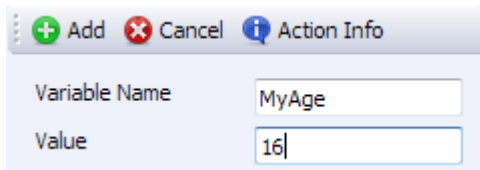
Consider the following actions list. The game will not compile because of it. The "Do Just Once" action opens a "block", hence the indentation. Every block must be closed with a corresponding action. In this instance, "End Do Just once" is the action that is needed.

You can usually tell if your game logic is correct, if the indentation at the bottom of the list is aligned with that of the top of the list.



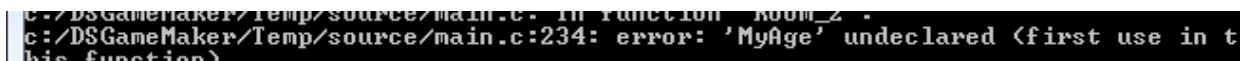
Another classic example of a logic error is with the use of the "If" action. If you use the "If" action, you must close the block with an "End If" action after "Fade In".

### REFERENCE OR DECLARATION ERRORS



Consider the following action ("Set Variable"). If you have not previously declared "MyAge" as a variable, you will get a compile error. This is because the DS will not know anything about "MyAge" so it cannot give it a value of 16.

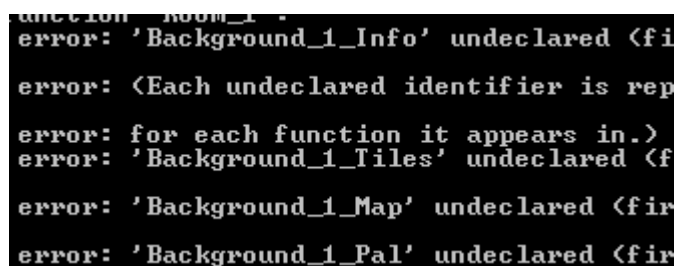
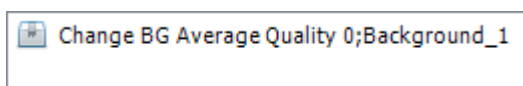
It is easy to tell when you have forgotten to declare a variable:



In the compile command prompt window, you will see something like the above.

**If you rename a resource, e.g. "Sprite\_1" -> "PlayerSprite", DS Game Maker will not update your actions for you.**

Consider the following example. In this action, we have told the DS to load "Background\_1" on the bottom screen. However, "Background\_1" either does not exist, has been deleted or has been renamed.



As you can see from the error on the right, "Background\_1" is referenced in many places along with the buzzword "undeclared". Go ahead and fix this by removing the action, renaming a background to "Background\_1" or selecting another background in the action, e.g. "Background\_2" (which must exist).

## WARNINGS

If you see a warning like the below, do not panic. Warnings do not stop your game from compiling. Their purpose is to inform you about inefficiencies in your game – areas of your logic that could be improved. There is a clear difference between warnings and errors. If you get an error, no NDS will be created – compilation will have failed. Warnings do not stop game creation, but heed the warning, as it may be a clue as to why your game may not function correctly when played.

```
warning: unused variable 'DSGMSouldFade'
```

The above is a fine example of a warning that does no harm at all. It is simply informing you, as the game creator, of a possible "waste" of resources. The compiler sees the variable "DSGMSouldFade" being declared but does not see it being used. The variable name indicates that it is part of the DS Game Maker engine. You are safe to ignore this.

## NOT ENOUGH VRAM

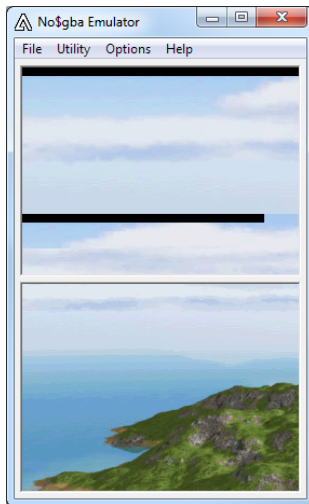


It was made clear above that the DS does not have a lot of temporary memory (RAM). If you see an error like the one on the left, you are loading too many sprites and backgrounds at once.

You can fix this by removing sprites on the screen you are not using, combining backgrounds you have on different layers into 1 image, using MP3 files instead of RAW files and **not using the "Create Sprite" action.**

In this example the VRAM error is triggered by attempting to show the on-screen keyboard, which is in itself a background.

## GRAPHIC ERRORS



As can be seen on the top screen, the background has a glitch. It looks fine inside DS Game Maker (see right).

The problem is that the background is not correctly sized.

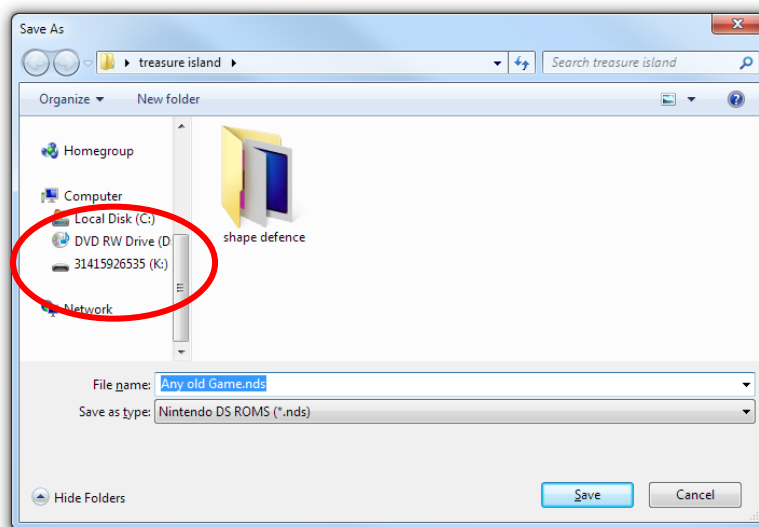
For a background that is designed to fill the screen, it should have the dimensions of 256x192 pixels. If you are scrolling the background vertically it should be 256 pixels high. The width in pixels should be a multiple of 256 pixels, for example, 256, 512 or 768 pixels wide.



## TESTING THE GAME ON THE NINTENDO DS

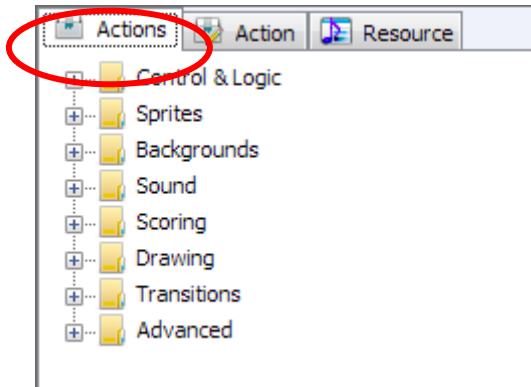
Before moving on to create the game, you should test you can run the game on your DS. To do this we will use the second compile button, "Create .nds File and Save", shortcut F6.

**Now connect your flashcart memory card to your computer, either by using an inbuilt memory card slot or with a MicroSD to USB adapter.**



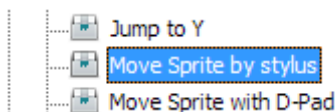
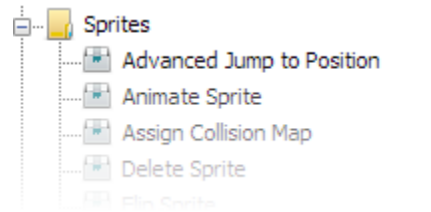
We are presented with a standard "Save As" dialog. On the left, as circled, select your memory card. For me, the drive is "K:/". Hit the "Save" button.

## CONTINUING THE GAME



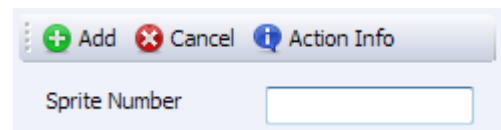
We will continue by adding an action to move the green ball along with the stylus. The action for this is "Move Sprite by stylus". To add an action we need to go to the right panel with the list of all the actions we can add.

The action we want is categorized under 'Sprites'. Click the plus to expand this category:



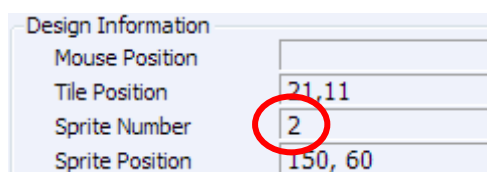
Find the action in the list and **double click it to add it to our room.**

You will see that the actions list has been replaced with a form as shown right. This action has arguments; this means we need to give it information for it to work, like parameters. In this case, the action needs to know which sprite should be moved by the stylus. We know this is the **green ball**.

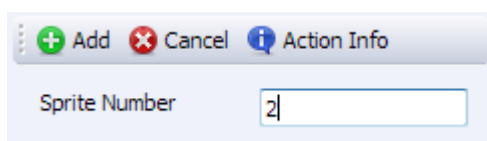


When we plotted our sprites the green ball was the 3<sup>rd</sup> one to be placed. So it has an index of 2:

Plotted order	Sprite number (index)
1 <sup>st</sup>	0
2 <sup>nd</sup>	1
3 <sup>rd</sup>	2

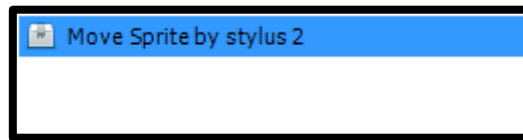


If you have lots of sprites this is confusing. You may use the table of data in DS Game Maker to tell you the sprite number by hovering over the sprite with your mouse.

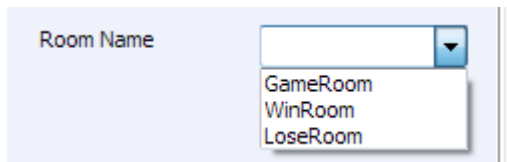


So go ahead and enter 2 in the box for Sprite Number and hit 'Add'.

Hurrah! Now you should see the action in the room actions list:



Now we will add some very simple logic for checking collisions. Firstly we will add 3 actions to check if the player has won. We will begin by adding the action 'If Collision', under also the 'Sprites' category. For screen of course we select 'Bottom' (where both our sprites are). Imagine 'Sprite 1' to be the player and 'Sprite 2' the enemy, and us detecting a collision between them. For Sprite 1 Number type 2 (our green ball). For Sprite 2 Number type 0 – our red ball.



Next add the action 'Switch Room', under 'Control and Logic'. You will see you can select a room to which to switch if the collision happens. As it is the aim of the game to touch the red ball, select 'WinRoom'.

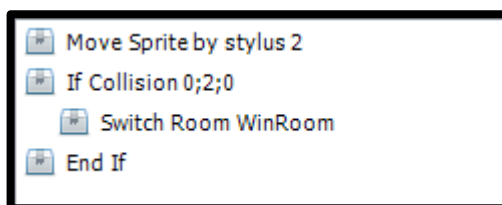
Finally, we add the 'End if' action, to finish our block. **This action appears right away in the list.**

There is now just one thing to do before we test our work, we must apply our 'Win' backgrounds to 'WinRoom'. Open up 'WinRoom' by double clicking it in the resources pane on the left. Double click 'WinBG' under 'Backgrounds' and click once in both screens to apply it.

**Now let us press F5 to test the game!**

If all is well, there is no compile error and you may move the green ball with the stylus. When the green ball contacts the red ball the win screen is shown.

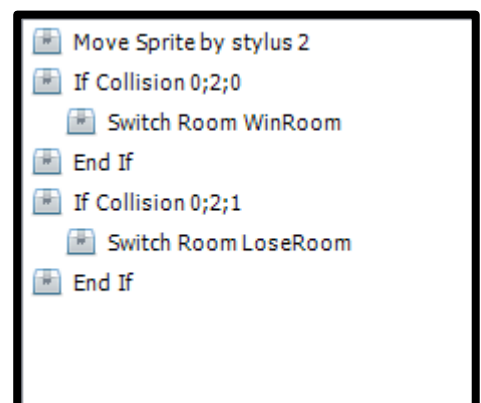
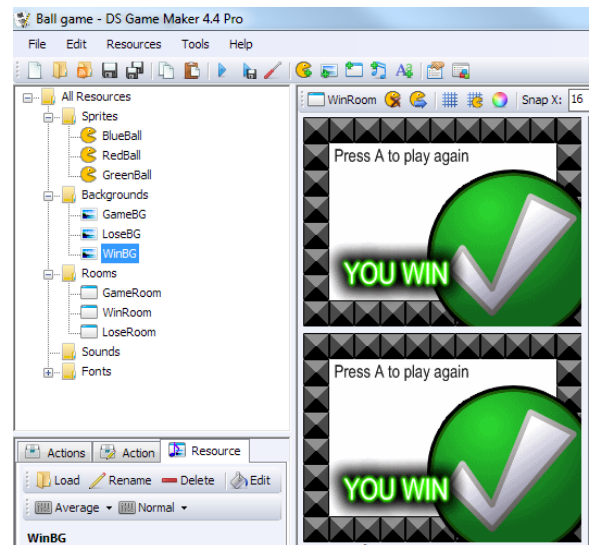
The final actions list for 'GameRoom' is as follows:



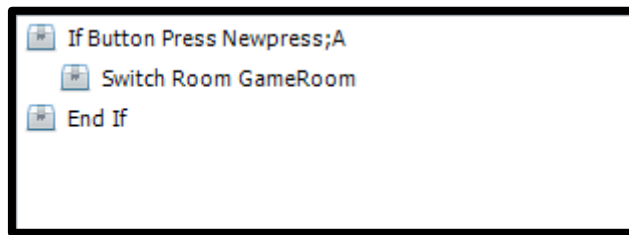
Now we need to do the same for the blue ball and get the room to switch to 'LoseRoom'.. the actions you can probably figure out on your own. The list should look now like the list shown right.

We need again to apply the 'LoseBG' background to the screens in 'LoseRoom'.

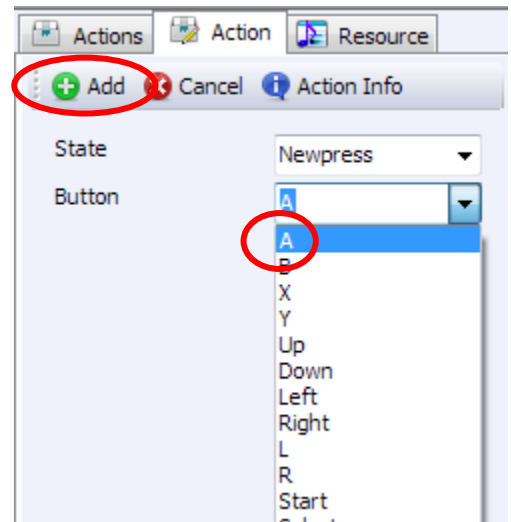
**Test the game again to make sure you can now win and lose.**



Finally, the player needs a way to get back to the game after winning or losing. The instruction on the background says 'Press A to play again'. We will use the action "If Button Press", shown right. Add the Switch Room and End If actions:



Add the same 3 actions to 'LoseRoom'.



**If everything works, congratulations – you just made your first DS game. Although this game is very simple, we have actually explored many elements of creating games with DS Game Maker, including adding resources, adding actions, switching rooms and compiling our completed game. Now this is hopefully clear, we may explore how to do much more with DS Game Maker.**

---

## EDITING EXISTING ACTIONS

To demonstrate editing actions that have been already been added, we will change the controls in the game to use B instead of A, to return to the game room. Open the room 'GameRoom' and find the action 'If Button Press Newpress;A'. To edit the arguments of this action – its properties, double click it. You are presented with a form much like the form seen when adding an action. Here you may change the button to be used, from A to B. Click 'Save' when your changes are complete. The action will automatically update in the list, and we are done.

## CONTINUATION OF THE MANUAL

Now that we have created a simple game, this manual will not instruct you any longer how to add actions or edit actions, or add resources such as sprites and backgrounds. Adding other resources like sounds and fonts is exactly the same as the process for the resources we have added thus far... use the Resources menu and select 'Add Font', 'Add Sound' etc. or use the individual buttons on the toolbar.

---

# SOUND

In creating a video game, sound is another way to make the game seem more realistic and engaging. If anything sound can set the mood of a game. Used well and you can excite the player and create emotion. Of course, sound can be used in much simpler ways, sometimes to alert the player of a certain action, for example loss of health, or the gain of a new pickup. Playing a sound when clicking a button with the Stylus is a nice feature, which informs the user their action was noticed.

The ability to use sound is one of the main benefits of using the Pro Edition of DS Game Maker. You can make games with innovative game play, but to perfect a game, sound is often the last thing to add, and completes a game, in more ways than one.

## 2 TYPES OF SOUND

On the Nintendo DS, there are basically 2 types of sound, and they each have distinct uses.

### MP3 SOUNDS

MP3 type sound is primarily for music. MP3 music can loop and it is of high quality. It can take about a second to buffer (the time between calling the music and it playing), so it is no good for sound effects.

### RAW SOUNDS

RAW is the DS' equivalent of WAV sounds. This type of sound plays instantly and the filesize is very efficient when using short, 1-3 second sound effects. Anything longer though, and the memory used in the game escalates.

To summarize:

- Music, dialog: MP3
- Short effects (beeps, etc): RAW

## ADDING AND PLAYING A SOUND

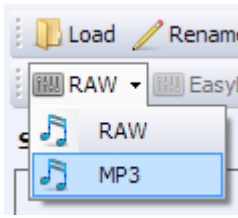
**Go ahead and add a sound to a new, blank project.** There is no need to rename the sound, but as previously mentioned, it is advised later on, to keep the project organized, to make it easy to correct bugs.

We will begin by playing a simple RAW sound when the player presses the 'A' button. As the default type of a sound is RAW, click 'Load' to select a sound file. You are presented with an open file dialog, as for loading sprites, backgrounds, etc. Open the folder 'RAWSounds' – you are presented with a long list of RAW sounds already included with DS Game Maker. You can select any sound, for the purpose of this tutorial we will be using 'explosion.raw'.

To play a RAW sound effect, we use the action 'Play RAW Sound'. This is very simple to use. To stop the sound from playing over and over again, we will use a control block, like so:

```
If Button Press Newpress;A
    Play RAW Sound Sound_1
End If
```

Test the game. Press A to hear the explosion sound play. In a nutshell, that is how to play sound effects.



To play an MP3 file, we use the action 'Play MP3 Sound'. You must change the type to 'MP3' before you are able to load an MP3 file (see left). Now click 'Load' and open the folder 'MP3Sounds' where you will see a range of royalty-free MP3 files for you to use in your game. They are all loopable which means when the music ends it does not sound odd when the beginning repeats again. For the purpose of this example, we will use the file 'Spies.mp3'.

It would be more accurate to name the action we must use 'Stream MP3 Sound'. This is because the action must 'always be in use' for the file to play (it cannot be used just once in a control block). Consider the following:

```
If Button Press Newpress;B
    Play MP3 Sound Sound_2
End If
```

Unfortunately you will find this does not work, but the following *does*:

```
Play MP3 Sound Sound_2
```

To correct this we must use a *variable*. These are covered later in this manual, but for the purpose of this example here is a very basic explanation. A variable holds a value that can be changed at any time. So it is like having a name for a certain number or piece of text, so that you can use the number etc. multiple times in your game (and only have to change 1 declaration), and also so you can change something that happens. Some examples are the X and Y positions of every sprite and the X and Y positions of the *View* (again, covered later). You can 'GET' and 'SET' these values.

This is how to play an MP3 Sound when a button is pressed:

```
Declare Room Global DoPlay;Boolean (true/false);false
If Button Press Newpress;B
    Set Variable DoPlay;true //Tell DS every frame to play the sound
End If
If DoPlay //Should we play the sound?
    Play MP3 Sound Sound_2
End If
```

Note there are the actions 'Stop MP3' and 'Make MP3s loop' that you can use as well. You can use both RAW and MP3 sounds together at the same time, like so:

```
Play MP3 Sound MyBackgroundMusic
If Button Press Newpress;L
    Play RAW Sound GunShoot
End If
```

## USING YOUR OWN SOUNDS

**Thanks for reading what I've written so far. The next revision of the manual, with this section, will be available soon.**