

1

Introduction

Welcome to Stencyl — an exciting and fun game development tool used by many professional and amateur game developers around the world.

By the end of this book, we will know all the skills required to develop a product that exhibits all the features expected in a professionally-developed computer game.

However, before we start to create our game, we're going to learn about the Stencyl toolkit, install Stencyl, and test that everything is working as required, then experiment with some of the sample games that have been provided with the installation.

In this chapter we will learn the following:

- ◆ How Stencyl works
- ◆ How we'll learn to use Stencyl
- ◆ Why Stencyl is a great development tool
- ◆ Platforms that Stencyl runs on
- ◆ What makes Stencyl different
- ◆ Successful games created with Stencyl
- ◆ Using the free version of Stencyl
- ◆ Installing Stencyl and testing the setup

How Stencyl works

If you have purchased this book, then you may already have an idea of what Stencyl is and how it works.

However, if you are browsing online or standing in a bookstore flicking through the pages of this book, then you might want to know that Stencyl is a no-coding toolkit for creating 2D video games that will run on many different mobile and desktop devices.

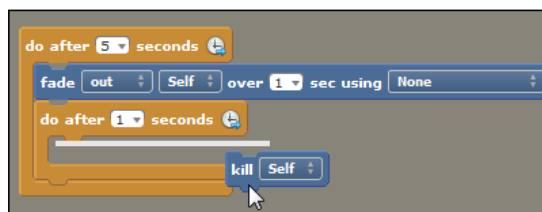
Scenes, or levels, within a game are created using Stencyl's drag-and-drop **Scene Designer**. The screenshot below shows the Scene Designer being used to modify one of the levels in the game that we'll be creating:



Downloading the color images of this book

We also provide you a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from: http://www.packtpub.com/sites/default/files/downloads/5961OT_ColoredImages.pdf

The instructions or logic for the gameplay can be created using Stencyl's **Gameplay Designer** — a clever system that utilizes building blocks which snap together to create a series of instructions that are used by the objects in our game. An example of some instructions being created in the Gameplay Designer is shown in the following screenshot:



How we'll learn to use Stencyl

All that is needed to follow the tutorials in this book are a desktop computer—Microsoft Windows, Mac OS X, or Linux will do just fine, along with the free version of Stencyl that can be downloaded from www.stencyl.com. The free version of Stencyl can be used to develop and publish Flash and HTML5 games, and it can also be used to develop and test games for desktop computers and mobile devices. If you want to publish games for platforms other than Flash and HTML5, then you'll need to pay for an annual subscription to Stencyl in order to access the additional publishing features.

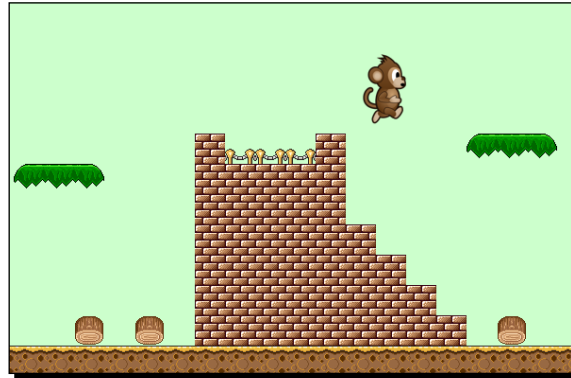
A great way to learn to use a tool such as Stencyl is to complete a practical project, and over the course of this book, we will develop a game from a blank screen right through to completion, learning all the important features of Stencyl along the way.

The game will start in a very basic form and as we progress, we will use the tools within Stencyl to add features to our game until we have a final product that demonstrates many of the characteristics that a professional game should exhibit.

Later in this chapter, we'll install Stencyl and run some test games to ensure that everything is working properly. Then, in *Chapter 2, Let's Make a Game!*, we'll jump straight in and create a game with a character that we can control as it runs and jumps across a scrolling jungle scene and interacts with an interesting environment.

We're going to experience some amazing results very quickly, so the learning process is going to be fast paced and fun!

The following screenshot was captured while developing the game in *Chapter 2, Let's Make a Game!*, so it can be seen how quickly we will be progressing through the development of our game.



Why Stencyl is a great development tool

Stencyl is a ready-to-use, complete, game development studio that can be used by anyone from beginners to game development experts to create professional-quality games; there is no requirement to purchase or install additional software. The Stencyl game development process avoids the repetitive, complicated requirements of writing hundreds of lines of computer code, and gets you started with the creation of your games without being distracted by the intricate details that traditional computer programming languages require.

Rapid prototyping and development

If you have an idea for a game, Stencyl will enable you to rapidly create a working prototype, so that you can quickly progress to creating a completed game using the great features that come built into the Stencyl toolkit, including:

- ◆ Scene Designer
- ◆ Drag-and-drop Gameplay Designer
- ◆ Resource management (sounds and graphics)
- ◆ Animation editor
- ◆ Graphics editor
- ◆ Online resource sharing

We'll be learning how to use all these features to enable us to quickly build our video game.

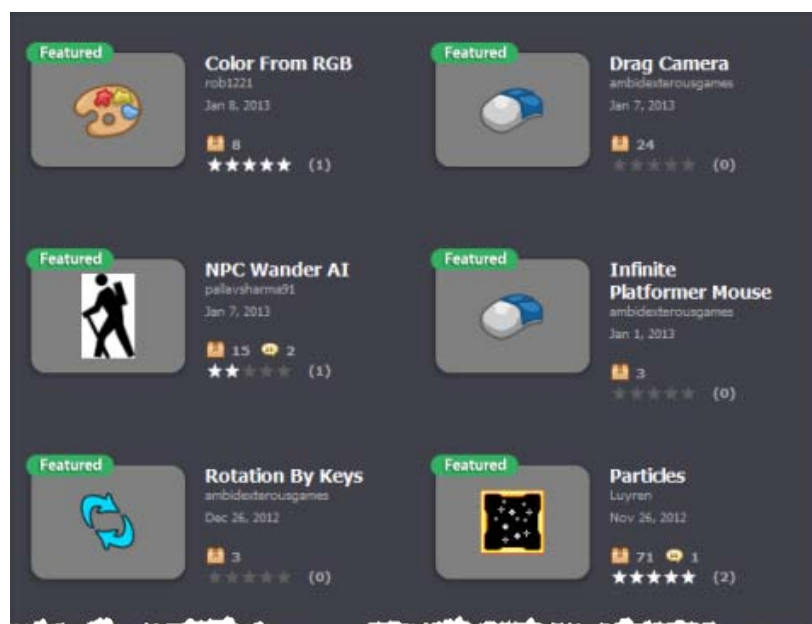
No code development

Games can be created without writing a single line of computer code, by using Stencyl's drag-and-drop Gameplay Designer. Its clever system of building sequences of commands using specially designed instruction blocks means that we can only create instructions that make sense to the computer.

If you have ever tried writing computer games using a language such as Apple's Objective-C or Adobe's ActionScript, you probably know how frustrating it can be to ensure that every bracket, period, and special symbol appears in exactly the right place. That's a problem of the past with Stencyl!

Sharing resources with other Stencyl developers

One of the many great features of Stencyl is **StencylForge** — a ready-built repository of game resources that can be used for downloading graphics, audio, gameplay rules (known as **Behaviors** in Stencyl), and utilities. You can also upload your own resources to share with others.



StencylForge is a remarkable asset to the Stencyl community, and it's one of the exciting features of Stencyl that we'll be learning to use while we develop our game.

In addition to being able to share resources using StencylForge, there is a great community of helpful Stencyl users for both beginner and experienced users at community.stencyl.com.

Platforms that Stencyl runs on

The Stencyl game development toolkit can be installed on the following desktop computer platforms:

- ◆ Microsoft Windows (XP, Windows Vista, Windows 7, and Windows 8)
- ◆ Mac OS X
- ◆ Linux

The installation process for each of the above platforms is detailed later in this chapter.

Stencyl target platforms

When we talk about target platforms, we mean the devices that our game can run on.

Games created with Stencyl can be played on most common platforms including web browsers, Microsoft Windows, Mac OS X and iOS devices (for example, iPhone and iPad), Android, and Linux. The main issues that we need to think about, when it comes to the target platform, are the screen size of the game and how our game will be controlled (that is, keyboard, mouse, or touch). Stencyl does all the hard work for you!

Currently, at the time of publication, Stencyl can target the following platforms:

- ◆ Apple mobile devices running iOS Version 4.1 and above, including:
 - iPod touch
 - iPad
 - iPhone
- ◆ Google Android devices running Android Version 2.2 and above, including:
 - Phones
 - Tablets
- ◆ HTML5 web browsers:
 - Many web browsers and devices that support the HTML5 web standard
- ◆ Microsoft Windows desktop PCs and laptops:
 - Windows XP and above
- ◆ Mac desktop operating systems:
 - Mac OS X
- ◆ Linux desktops:
 - Ubuntu distributions recommended

Targeting specific devices

If you need to target a specific device, it is important to ensure that it is supported. So, check the Stencyl website for the most up-to-date information, as details may have changed since publication.

Publishing desktop games

When creating games for desktop platforms, it's important to understand that you can only publish games for the platform on which you are running Stencyl. For example, if you are running Stencyl on a Windows PC, you can only create desktop games for other Windows PCs, and if you are running Stencyl on Mac OS X, then you can only create desktop games for other Mac OS X computers.

Publishing to iOS devices

Amazingly, if you want to publish to Apple's iOS devices using Stencyl, you don't need to own an Apple computer, which has been, until now, an expensive barrier to entry for would-be iOS game developers! The Stencyl *Mobile* and *Studio* annual subscriptions include access to the StencylBuilder service that enables Windows and Linux users to publish their Stencyl games to the Apple App Store as long as they are current members of the Apple iOS Developer Program.

Publishing to Android

Publishing to Android is a very straightforward process with Stencyl running on any supported desktop platform, and testing on an Android device does not require a developer license.

What makes Stencyl different

There are many game development tools available, but Stencyl has some stand-out features that make it very different from its competitors.

Stencyl runs on almost any desktop computer

Stencyl can be installed on nearly all popular desktop computers, and Stencyl files can be freely exported and imported between the different desktop platforms, which is very useful when working with friends or colleagues who use different platforms for game development.

Stencyl creates native code

Whichever platform we want to publish our game onto, Stencyl does all the hard work for us! It creates the game code specifically for that device, and it will not have to run the game in a wrapper that can slow our games down.

Although that might sound fairly technical, it just means that our games will run at the fastest possible speeds without us having to learn a new programming language for each target platform. Stencyl takes care of all the hard work when it comes to publishing to different target devices; we can just concentrate on creating our games!

You don't need to be a coder

Probably the greatest differentiator between Stencyl and other game development tools is the way in which the gameplay instructions are built. With many game development tools, you must hand-code the instructions in a specific programming language, which can be very tedious and time-consuming. Or, with some other development tools, you are required to point and click on the instructions that you want to select in order to build rules that the objects in your game will follow, which can be very restricting when it comes to complex gameplay.

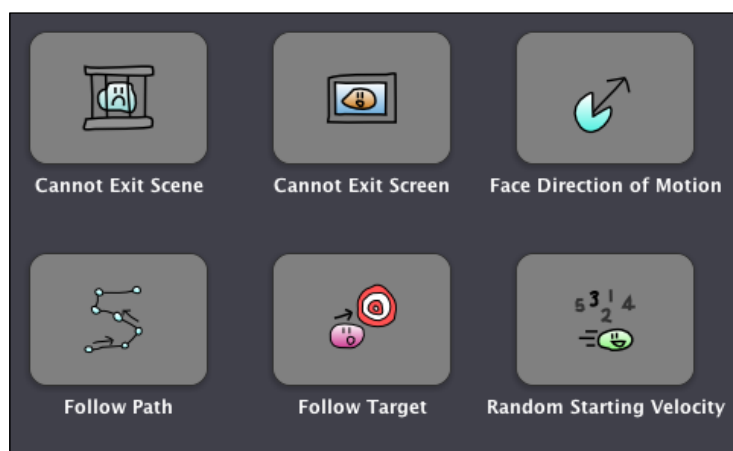
The developers of Stencyl have created a system that offers an impressive set of options when compared to other currently available game development methods. There are four different options available to game developers when using Stencyl, so we can do the following:

- ◆ Use Stencyl's built-in gameplay routines (called **behaviors** in Stencyl)
- ◆ Download existing behaviors from StencylForge
- ◆ Create our own behaviors using an intuitive drag-and-drop system
- ◆ Hand-code our own gameplay instructions using a traditional programming language called Haxe

The magic of Stencyl is that we can mix and match any combination of the above development methods!

For example, if we want to create a basic game with common gameplay features, it's very easy to use Stencyl's built-in behaviors, and it's quite possible to develop a complete game in this manner.

At the time of publication, there were more than fifty built-in behaviors, a small sample of which is shown in the following screenshot:



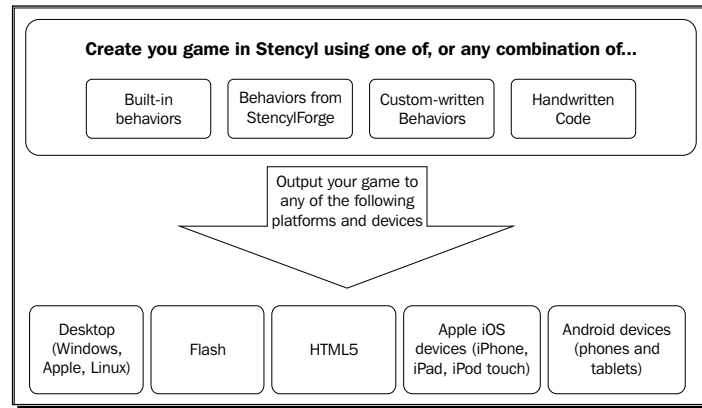
Once we understand how the prebuilt behaviors work, we can either customize existing ones or we can create our own from scratch using the drag-and-drop behavior editor.

The following screenshot shows an example of a custom-built behavior, and even if you have never seen Stencyl before, you might be able to understand some of the instructions.



Finally, more advanced developers may choose to hand-code their own special game routines using the Haxe programming language (pronounced Hex, according to the main developer). This is certainly not a requirement for creating games, and many Stencyl users have never even tried this kind of development! Most Stencyl developers use a combination of prebuilt behaviors, customized behaviors, and also behaviors that they have created themselves, and these are the methods that we are going to be using throughout this book.

When our game is finished, we just need to decide which platforms we want to target, and Stencyl will do the rest with a few clicks of the mouse!



Using the free version of Stencyl

The free version of Stencyl has very few limitations compared to the subscription versions and in practice, you can use all the important game development features and make any kind of game with the free version that you can make with the subscription versions.

The main limitations of the free version of Stencyl are:

- ◆ Only Flash and HTML5 games can be published
- ◆ A Stencyl splash screen is displayed for a few moments when a published game loads
- ◆ Custom preloaders can't be implemented (these are the loading screen and progress bar that you see when your game first loads)
- ◆ Some monetization features are limited

The great thing about the free version of Stencyl is that we can develop our games to completion, publish them for Flash or HTML5, and even sell them if we wish.

If, after publishing for Flash or HTML5, we find that a game is very popular, we can then pay for the appropriate Stencyl subscription and sell the game in the Apple App Store, or for Android devices in the Google Play store. We can learn how to use Stencyl and develop games for free to see if people like them, before paying for an annual subscription!

If you are a member of the Apple iOS Developer program, games can be created and tested for the Apple iOS devices with the free version of Stencyl, but they can't be published to the Apple App Store. In *Chapter 10, Targeting Mobile Platforms*, we'll look in more detail at the options available, and the requirements for publishing to mobile devices running iOS and Android.

Using the free version of Stencyl with this book

In order to follow the game development tutorials in this book, only the free version of Stencyl is required. Subscriptions to Stencyl are only necessary for publishing to mobile devices and desktop applications.

The following matrix shows the major differences between the free and subscription versions of Stencyl:

	Free	Indie	Mobile	Studio
Platform Support				
Flash Publishing	Yes (w/ watermark)	Yes	Yes (w/ watermark)	Yes
HTML5 publishing	Yes (w/ watermark)	Yes	Yes (w/ watermark)	Yes
Chrome Store publishing	Yes (w/ watermark)	Yes	Yes (w/ watermark)	Yes
Windows 8 Store publishing	Yes (w/ watermark)	Yes	Yes (w/ watermark)	Yes
Native desktop publishing	Yes (w/ watermark)	Yes	Yes (w/ watermark)	Yes
iOS publishing	No	No	No	Yes
Android publishing	No	No	No	Yes

Please note that the previous matrix provides only a summary of differences between the available versions of Stencyl. It is vital to check the Stencyl website (www.stencyl.com) for the latest features prior to making a purchasing decision, as the feature list may have been updated since publication.

Successful games created with Stencyl

Many successful games for Flash and mobile devices have been created with Stencyl. In some cases, they have been created by experienced developers who have switched to Stencyl to speed up the development process, and in other cases, the games have been developed by newcomers to game development, who had previously been unable to create their own games due to the complexity of traditional programming languages, or who had simply not attempted to create a game previously.

Following are three examples of successful games that were created using Stencyl:

- ◆ **Making Monkeys:** Making Monkeys is an original, fun, platform game created by Greg Sergeant of greg-anim.com. The idea of the game is that very weird-looking monkeys can use a special weapon to duplicate themselves and solve various puzzles. Making Monkeys was sponsored by Armor Games and has gained over 1 million game plays.



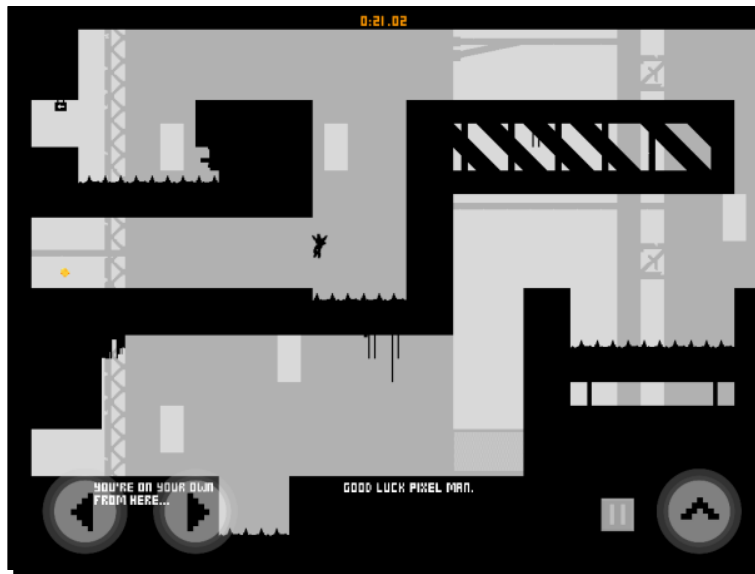
Greg has stated that it only took him a few days to create the prototype of the game and that Stencyl enabled him to try out new gameplay elements very quickly.

- ◆ **Kreayshawn Goes to Japan:** The first game that Beth Maher (www.bethmaher.com) created was called Kreayshawn: The game. The game, shown in the following screenshot, was created by her with Stencyl during a workshop project that was designed to encourage women to become involved in the indie game industry.



The development of this fan game led to Beth working with Columbia Records (part of the Sony group of companies) to produce Kreayshawn Goes to Japan, for Kreayshawn's fan website (arcade.kreayshawn.com). The game is a bright, imaginative platform game featuring Kreayshawn as the main character.

- ◆ **Impossible Pixel:** Impossible Pixel and the Fate of Destiny, created by 99 Up Games, is a challenging platform game with 93 levels, available on iPad and iPhone. This game takes advantage of the touchscreen features provided by iOS; the buttons for controlling the player can be seen in the lower-left and lower-right corners of the following screenshot:



At the time of writing, Impossible Pixel, according to the developer's figures, had been downloaded from the Apple App Store approximately 700,000 times and had reached the position of the most popular free download on the App Store in nine countries. It was also the second most popular free download in thirty other countries, including the US App Store.

Impossible Pixel includes many professional features including Game Center achievements, high score leaderboards, atmospheric retro soundtrack, and many interesting challenges throughout the game.

The Stencyl Showcase

The listed games are a very small sample of the successful games that have been published using Stencyl. The Stencyl website contains areas dedicated to showcasing games, and it's certainly worth visiting Stencyl Showcase for inspiration, and to see some examples of impressive games that have been created with Stencyl.

- ◆ www.stencyl.com/stencyl/showcase/
- ◆ www.stencyl.com/stencyl/successes/

Installing Stencyl and testing the setup

We've had a look at what Stencyl can do and how we are going to learn to use it. So, let's jump right into the next vital step and install Stencyl. The following section of the book is split into four parts. The first three parts will explain how to install and run Stencyl on each of Microsoft Windows, Mac OS X, and Linux, and the fourth part will step through creating an account and quickly testing Stencyl to make sure that everything is working.

Once we have Stencyl up and running, we'll head off into *Chapter 2, Let's make a game!*, where we'll quickly progress with the development of our first game.

Installing Stencyl

Let's look at how to install Stencyl on each of the three desktop platforms that it supports: Microsoft Windows, Mac OS X, and Linux.

Microsoft Windows

Stencyl will currently install on the versions of Windows listed as follows:

- ◆ Windows XP
- ◆ Windows Vista
- ◆ Windows 7
- ◆ Windows 8

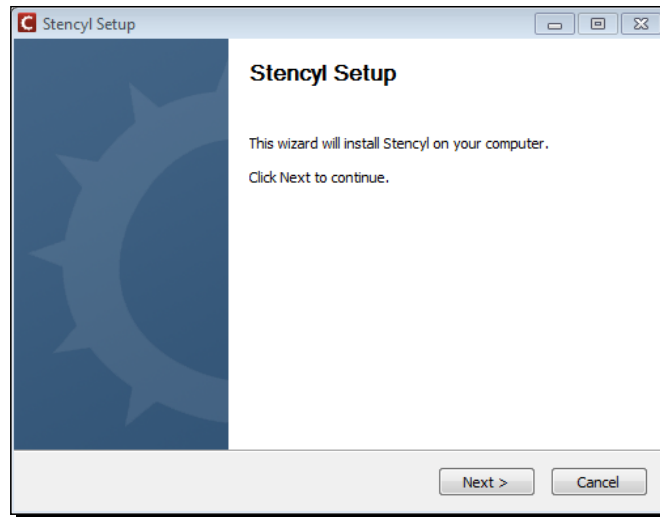
The download and installation methods are the same for each version of Windows, but the following screenshots have been captured in Windows 7.

Time for action – downloading and installing Stencyl on Windows

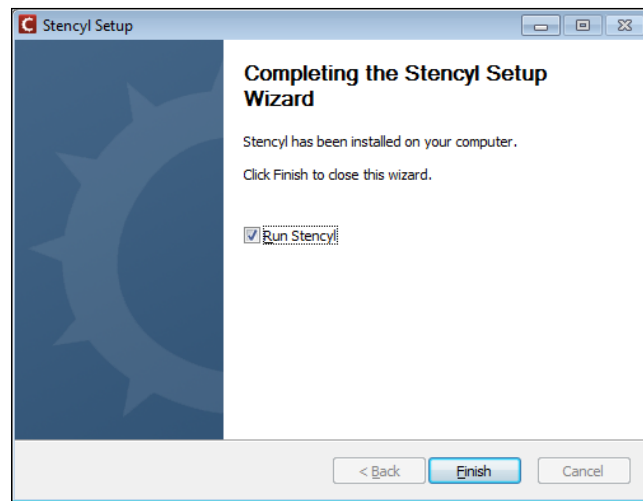
In this session, we're going to download and install Stencyl on Windows:

1. Visit www.stencyl.com.
2. Click on the **Download Now** button on the home page.
3. Depending on which web browser you are using and how you have configured it, the latest Stencyl installation file will start to download automatically, or you may have to confirm that you accept the download and specify the download location.

4. When the installation file has downloaded, find it in Windows Explorer and double-click it to start the installation process. You can see the dialog box shown in the following screenshot:



5. Click on **Next** to accept the default installation folder.
6. Click on **Install** to set the installation process running. Usually, the installation process will take less than a minute to complete, but the time may vary depending on the specification of your computer.
7. When the installation has completed, click on **Next** again to display the final dialog box shown in the following screenshot:



8. I recommend clicking on the **Run Stencyl** option to remove the checkmark so Stencyl does not run automatically.
9. Click on **Finish** to complete the installation of Stencyl.



Because of the system access that Stencyl requires, you may need to run Stencyl as the Administrator on a Windows PC when updating or the first time after purchasing a subscription. This can easily be achieved by right-clicking on the Stencyl application icon and selecting **Run as administrator**.

What just happened?

We have just downloaded the latest version of Stencyl for Windows, and the installation is now complete. If you want to delete the installation file that you downloaded, then that's fine; it's not needed any more.



When running Stencyl for the first time, we are asked to sign in. The signing in process is the same for all platforms and is detailed under the heading, *Creating a Stencyl account*, (after the Linux installation procedure in this chapter).

Mac OS X

Stencyl will currently install on all versions of Mac OS X, but it should be noted that Java 6 must be installed in order for Stencyl to work.



At the time of writing, Java 6 was a mandatory requirement for correct operation of Stencyl; Java 7 is not compatible.

The download and installation methods are the same for each version of Mac OS X, but the following screenshots have been captured in Mac OS X 10.7.4.

Time for action – downloading and installing Stencyl for Mac OS X

In this session, we're going to download and install Stencyl for Mac OS X.

1. Visit www.stencyl.com.
2. Click on the **Download Now** button on the home page.

3. Depending on which web browser you are using and how you have configured it, the latest Stencyl installation file will start to download automatically, or you may have to confirm that you accept the download and specify the download location.
4. When the installation file has downloaded, double-click on the `stencyl.dmg` file to automatically mount it and start the installation process.
5. When the application's files are ready to install, the following window will be displayed:




6. To complete the installation, click-and-hold on the **Stencyl** icon and drag it on top of the **Applications** icon. The Stencyl files will be copied into the Applications folder, and Stencyl is then ready to run!

What just happened?

We have just downloaded the latest version of Stencyl for Mac OS X, and the installation is now complete.

The `stencyl.dmg` file can now be unmounted and deleted; it's not needed anymore.

[ When running Stencyl for the first time, we are asked to sign in. The signing in process is the same for all platforms and is detailed under the heading, *Creating a Stencyl account*, (after the Linux installation procedure in this chapter).]

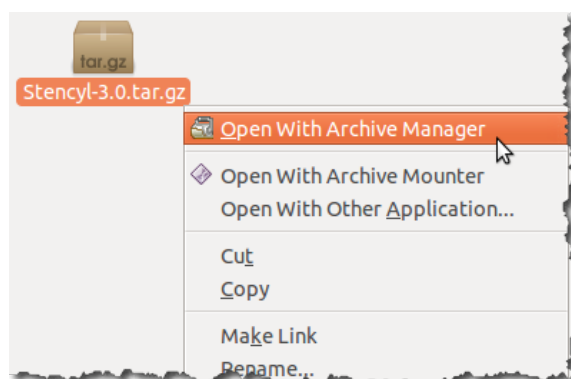
Linux

Stencyl should install on most recent versions of Linux, but Ubuntu distributions are recommended. For the following screenshots, Stencyl was installed and tested on Ubuntu Version 10.10.

Time for action – downloading and installing Stencyl for Linux

In this session, we're going to download and install Stencyl on Linux.

1. Visit `www.stencyl.com`.
2. Click on the **Download Now** button on the home page.
3. Depending on which web browser you are using and how you have configured it, the latest Stencyl installation file will start to download automatically, or you may have to confirm that you accept the download and specify the download location.
4. When the download has completed, right-click on the downloaded `tar.gz` file and select the **Open With Archive Manager** option as shown in the following screenshot:



5. Open the folder where the files were extracted.
6. Stencyl is now ready to run!

What just happened?

We have just downloaded the latest version of Stencyl for Linux, and the installation is now complete.

The Linux version of Stencyl doesn't really need installing – we just needed to extract the files to create the `Stencyl` folder so that the application files are accessible.

If you want to delete the original download file, that's fine; it isn't needed anymore.

Creating a Stencyl account

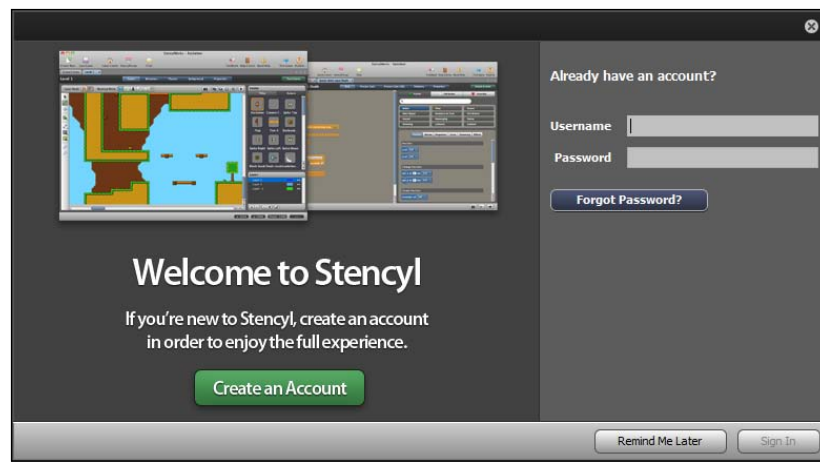
Now that we have installed Stencyl, the next step is to run a quick test to ensure that everything is working, so we can then dive into the next chapter and create our first game.

Time for action – creating an account and signing in

Although it is not a requirement to sign in to create games with Stencyl, we won't be able to publish any games, access StencylForge, or contribute to the Stencyl online forums without an account.

In order to follow the tutorials in this book, we'll need access to StencylForge, and we'll also need to be able to publish our games, so I highly recommend creating a Stencyl account and signing in.

1. Run Stencyl and the following screen will be displayed:

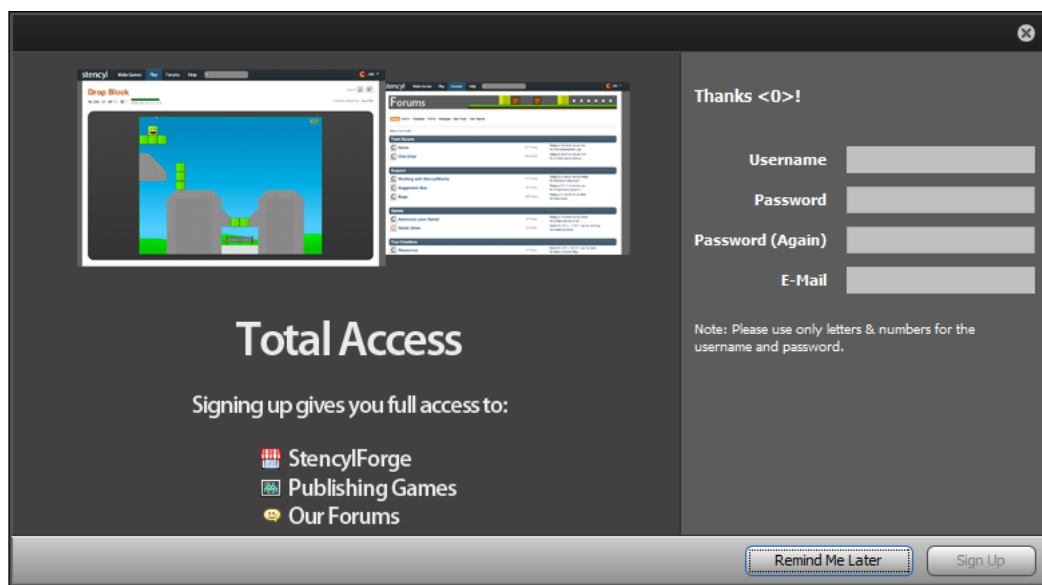


2. If you do not see the sign-in screen, ensure that you have Stencyl running, and select **File | Sign In** on the main menu at the upper-left corner of the **Stencyl** screen.



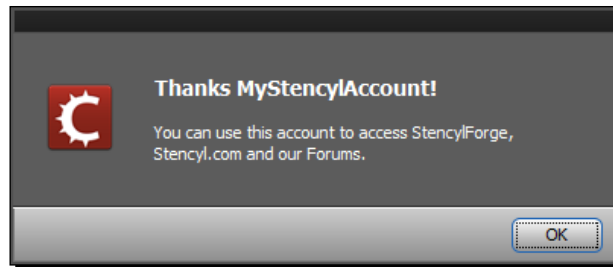
If you have already signed up for an account on the Stencyl forums, you do not need to create a separate account in order to log into Stencyl. The Stencyl application login and the Stencyl forum login use the same credentials; so, you can sign in to Stencyl without any further action!

3. Click on **Create an Account** and enter your username, password, and e-mail address in the right-hand side of the dialog box shown as follows. Note that Stencyl usernames and passwords are restricted to alphanumeric characters (letters and numbers only).



Choose your username carefully as it will be your permanent Stencyl login name and username on the Stencyl online forums!

4. When you are confident that you have entered the correct details, click on the **Sign Up** button, and after a few seconds a confirmation message will be displayed as shown in the following screenshot:



5. Click on **OK** to confirm, and we are ready to start using Stencyl!

What just happened?

We have created an account so that we can access the important features of Stencyl. The same account credentials can be used to log in to the Stencyl forums at `community.stencyl.com`.

You do not need to sign in and out of the Stencyl application each time you use it, because your login details are remembered unless you specifically choose to sign out using the **File | Sign Out** option in the Stencyl menu.

Have a go hero

Why not visit the Stencyl community forums and say hello?

Fire up your web browser and visit `community.stencyl.com`, sign in to the forums using the account details that you have just created, and have a browse.

You could go to the **Chit Chat** forum, find the topic entitled **Introduce Yourself!**, and well, introduce yourself!

Testing the Stencyl installation

Now that we have installed Stencyl, it would be a good idea to carry out a test so that we know that everything is working as it should be.

Time For action – testing Stencyl

We're going to load up Stencyl, open one of the sample games that has been provided, and run the game to make sure that Stencyl is working properly.



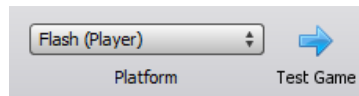
From this point onwards, all the tutorials will be the same whether Stencyl is running on Microsoft Windows, Mac OS X, or Linux, although the screenshots have mainly been taken using the Microsoft Windows version. On the rare occasion that there is a difference between the versions, it will be explained clearly in the tutorial.

1. If Stencyl is not already running on your computer, load it up and make sure that you have signed in with the account username and password that you created earlier, so you will see a screen that looks something like the following screenshot:



Don't worry if the list of games isn't identical on your screen, because the sample games packaged with Stencyl may be changed as software updates are released.

2. Double-click on the game icon for **Alien Approach** to open the game.
3. When the game has loaded, ensure that the target-platform drop-down option is set to **Flash (Player)**, and click on the blue arrow labeled **Test Game**, both of which can be found at the upper-right corner of the screen, as shown in the following screenshot:



4. Wait for Stencyl to create the game and display it in the **Adobe Flash Player** window as shown in the following screenshot (this may take up to 30 seconds, perhaps more on a very old computer).



5. Play the game using the arrow keys to move left and right, and the spacebar to fire your weapons.
6. When you have finished playing the game, close the **Adobe Flash Player** window.

What just happened?

We just loaded a game into Stencyl and compiled it for the first time! Compilation is the process of converting the Stencyl game-code into a format that will run on our target platform. The default target platform is Flash, so when the compilation process has finished, the game will be displayed in an **Adobe Flash Player** window.

It's important to be aware that the sample games are not completed projects – they are just examples that have been provided so that we can see the kind of games that can be created. We can also use the samples to understand how the games have been put together, which is a great way to learn!

Have a go hero

We've loaded and compiled one sample game and have had a play, so why not try some of the other sample games that have been provided?

Before we can open up another game, we should close the **Adobe Flash Player** window in which the previously tested game is running, and we must close the current game in Stencyl by clicking on **File | Close Game** and follow any prompts that appear.

Now, experiment with the other sample games, so that you become comfortable with the process of loading, testing, and closing games in Stencyl.

Summary

We've reached the end of this chapter!

We have examined what Stencyl is for and who will find it useful. We have also looked at why Stencyl is different from other game development tools, and we've discovered that Stencyl can run on virtually all desktop computers, and can create games for many different target platforms, including the latest Apple iOS and Android devices.

Finally, we stepped through the procedures for installing Stencyl on Microsoft Windows, Mac OS X, and Linux PCs, and then rounded off with a quick test, using a sample game to ensure that Stencyl was working properly.

We've learned a lot in this chapter, but now that our Stencyl installation is complete and has been tested, it's time to roll up our sleeves and start using Stencyl! In *Chapter 2, Let's Make a Game!* We'll very quickly build up the basics of our game, so that we will have a character that we can control as it explores the environment we are going to create for it.

2

Let's Make a Game!

In Chapter 1, Introduction, I promised that we'd have the basics of a working game in place by the end of this chapter, and I plan to keep my promise!

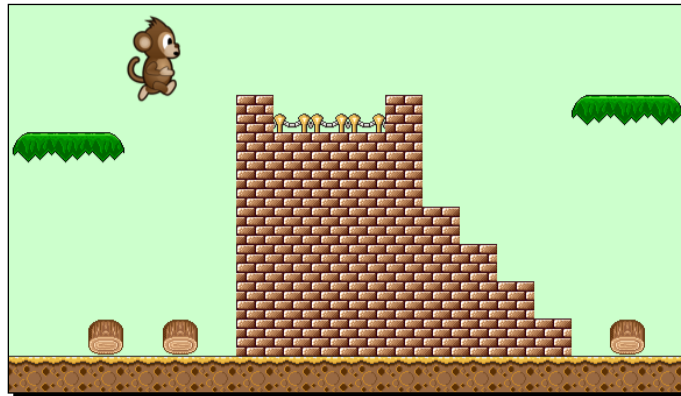
Throughout this book, we are going to create a platform game, based in the jungle, in which the player will control a cheeky monkey that will have the ability to run and jump around the scene. As our monkey character runs, the screen will scroll horizontally to display more of the jungle and, along the way, there will be various obstacles, some items to collect, and some enemies to avoid.

Although some of the preceding features will be added to our game later in the development process, when we have learned more features of Stencyl, in this chapter, we will put in place many of the basic mechanics of the game.

The skills we will be learning in this chapter are:

- ◆ Creating a new game
- ◆ Creating a blank scene
- ◆ Downloading resources from StencylForge
- ◆ Understanding the Stencyl Dashboard
- ◆ Adding an actor into a scene
- ◆ Downloading and using tiles for scenery
- ◆ Using behaviors to interact with our game
- ◆ Improving the scene mechanics
- ◆ Making the screen scroll

By the end of this chapter, our game should resemble the following screenshot:



Using the downloaded game files

When starting a practical activity in the book, we can either continue in Stencyl, using our progress from the previous tutorial or, if we are coming back to the book after a break, we can load the last version of the game that we saved at the end of the previous session.

If we want to start afresh at the beginning of a practical activity, we can import the appropriate downloaded game file into Stencyl, so we know that we are starting at the right point in the game development process!



The game files accompanying each chapter can be found in the *Support* section of this book's web page at www.packtpub.com.

Before each practical activity, there will be a note advising which Stencyl game file should be imported prior to starting the exercise, but it's not necessary to use the import files if we are following the tutorials consecutively.

It's very easy to import a Stencyl game file—just load up Stencyl and, on the main menu, go to **File | Import Game...**, then double-click the relevant file.

Let's get started!

Grab your keyboard and mouse, load up Stencyl, and let's start developing our game!

Creating a new game

We'll start by setting up Stencyl with a new, blank game to work with.

Time for action – creating a new game

We are starting with a blank game, so there's no need to import a game file!

1. Load up Stencyl on your computer.
2. Click on the **Click here to create New Game** cutout box, which appears after all the existing games on the **My Games** screen.
3. Click on **Blank Game** at the bottom of the left-hand side panel of the **Create a New Game...** dialog box.
4. Click on the **Next** button.
5. Type the name of our game `Monkey Run` into the **Name** text box under the **Basic Info** heading.
6. Click on the **Create** button.
7. Wait a few seconds while the blank game is created.
8. On the main menu, click on **File | Save**, or click the **Save Game** icon on the toolbar.

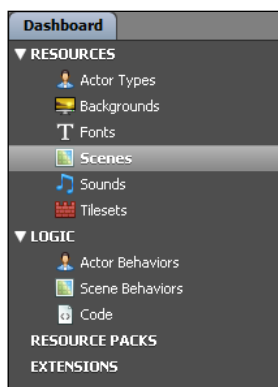
What just happened?

We have created and saved a blank game, but there are no game characters, no scenes, and no gameplay instructions!

It's a basic task, but we've taken our first step into game development with Stencyl.

Creating a blank scene

Every game needs a "world" in which to play, and this is where we will start to build our game. We already know that our game is going to be based in the jungle. So, let's take the next step and create a world for our monkey to play in.



The **Dashboard** should currently be displayed on the screen and, if we look at the left-hand side panel, under the heading **RESOURCES**, we can see a list of the types of resources that can be used in our game. The fourth item in the list is **Scenes**, and it should be highlighted, which is helpful, because the next step we are going to take is to create a scene for our game.

Time for action – creating a blank scene

The game file to import and load for this session is `5961_02_01.stencyl`.

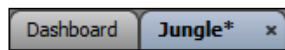
1. Click on the message, **This game contains no Scenes. Click here to create one**, that appears in the middle of the right-hand side panel.
2. Give the scene a name. I prefer keeping scene names to one word if possible, so let's call it **Jungle**.
3. In the **Background Color** section of the dialog box, click on the long, thin, white rectangle so the color-selector pops up, and change the color to **Sea Green**—hovering the mouse over a color will pop up a tooltip with the name of the color.
4. Click on the **Create** button.

What just happened?

We have just created a **scene**!

In Stencyl, a scene is most easily described as being a "level" in a game. Our game will have just one level during the initial development stages, and it's now being displayed in the Scene Editor.

If we have a look at the upper-left side of our Stencyl screen (under the toolbar), we can see that we have two tabs—one is the **Dashboard** tab, and the other is our new **Jungle** tab that represents the scene we have just created. Just in case you're wondering, an asterisk next to the name on a tab means that the work in that tab has not been saved yet—we'll save our jungle scene at the start of the next step!



Downloading resources from StencylForge

We don't have any objects in our game yet, so we can't do anything interesting with our scene. Our next step is to insert a controllable character into our game and, as mentioned at the start of this chapter, our main character is going to be a monkey—so let's find that monkey!

Time for action – downloading an actor

The game file to import and load for this session is `5961_02_02.stencyl`.

We're going to add a character into our game, right now, by downloading it from Stencyl's online resource, which is called **StencylForge**.

1. On the Stencyl menu, click on **File | Save Game** (or click the **Save Game** toolbar icon).
2. Click on the **StencylForge** icon on the Stencyl toolbar.
3. Click on **Actor Types** in the **Resources** section of the left-hand side panel.
4. In the search box at the upper-right corner of the screen, type `Stencyl Book Monkey`.
5. Press *Enter* on the keyboard and wait for the search results to appear.
6. Double-click on the **Stencyl Book Monkey** thumbnail that appears in the right-hand side panel and wait for the monkey actor's information to be displayed—this may take a few seconds.
7. Read the information under the **Description** heading.
8. Click on the green **Download** button at the upper-right corner of the screen and wait for the actor to be downloaded into our game—the download may take a few seconds to start.
9. When the actor has downloaded, the monkey animations will appear in the **Animation Editor**.
10. Save the game.



Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

What just happened?

After saving the game, we downloaded an animated monkey from StencylForge. In Stencyl, the controllable characters in our game are called **actors**, and we are now looking at the **Animations** editor which shows the available animations for the monkey.



As we can see, there are eight different animations for our monkey, which provides plenty of scope for the movement of our monkey actor within the game.

We can see that the **Waiting** animation in the left-hand side panel has a star icon attached to it, which means it is the default animation for the actor. When we later place the actor into the jungle scene, the **Waiting** animation will be displayed by default.



The actor graphics for our jungle game have all been created by the talented Vicki Wenderlich—www.vickiwenderlich.com—who has generously made the images available under the Creative Commons Attribution license. This means that anyone can use these graphics in their games—even commercial ones—as long as Vicki Wenderlich is clearly given credit for her work. We'll be having a closer look at legal matters relating to licensing and copyright in *Appendix 11, Planning, Resources, and Legal Issues*.

Have a go hero – searching StencylForge for interesting actors

In the previous *Time for Action* steps, we learned how to search StencylForge for actors. It's a good idea to become familiar with the resources that are available to us for use in our games, so have a go at browsing through the various actor resources that are available for download from StencylForge.

Open up the **StencylForge** tab, and have a good rummage around to see what interesting actors you can find! Although it is quite easy to delete unwanted actors from a game, if you want to download some more actors, to have a look at their animations, I would recommend creating a blank game to experiment with.

Understanding the Stencyl Dashboard

The Dashboard in Stencyl is the control center—it's where we will find all the resources in our game, whether they are scenes, actors, or any of the other resources that we'll be working with throughout the game development process.

It's a good idea to become familiar with Dashboard, so click on the **Dashboard** tab and note that the **Actor Types** heading and the **Scenes** heading both have a digit **1** next to them. This is telling us that there is currently one actor and one scene in our game, so it's a useful way to know, at a glance, what resources we have in a game that we are developing.

Adding an actor into a scene

We have a jungle scene, and we have downloaded our monkey actor, so let's do the right thing and put the monkey in the jungle using Stencyl's **Scene Designer**!

Time for action – adding an actor to the jungle scene

The game file to import and load for this session is `5961_02_03.stencyl`.

We need to ensure that the Animation Editor is displaying the monkey animation so, click on the **Actor Types** heading in the **Dashboard**, then double-click the **Stencyl Book Monkey** thumbnail in the main panel.

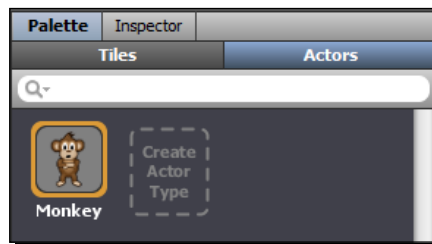
1. Click on the green **Add to Scene** button at the upper-right corner of the screen.
2. Click on the green **Jungle** scene that is shown in the **Choose a Scene** dialog box.
3. Click on the **OK** button to display the scene.
4. Move the mouse cursor (which shows the monkey) into the center of the green jungle scene.
5. Click on the mouse once to place the monkey into the scene.
6. Press *Escape* on the keyboard.
7. Click **File | Save Game** to save the game (or click the **Save Game** icon on the toolbar).

What just happened?

We have placed the monkey actor into the jungle scene, which was as easy as selecting the scene and then clicking on the mouse where we wanted the actor to appear. We only have one scene in our game, so it wasn't a difficult choice to make!

We are currently looking at the **Scene Designer**, which shows the design of the scene, and, to the right of the scene view, we can see the **Actors** in the **Palette** tab.

Currently, we can only see one actor in the palette—the monkey—because this is the only actor in our game. As we add more actors, they will be added to the **Actors** palette, and then we can click on any actor and add it to the scene as required.



If we wanted to place several copies of the actor into the scene, then we could have done so by moving the mouse and clicking on it each time we wanted to add a copy of the actor. It's not strictly necessary, but pressing *Escape* cleared the monkey actor from the mouse cursor so we didn't accidentally place more than one copy of the actor into the scene.

If we accidentally add more than one monkey to the jungle scene, it is easy to remove the unwanted actors—ensuring that the *Escape* key has been pressed, click on the unwanted actor and press the *Delete* key to remove it.

Testing the game

So far, we have a scene with a single actor placed on it, so now would be a good time to test the game to make sure that we know what we have achieved in the development process!

Time for action – testing the game

The game file to import and load for this session is `5961_02_04.stencyl`.

Fortunately, Stencyl makes it very easy—and quick—to test our games:

1. Click the **Test Game** icon on the right-hand side of the toolbar.
2. Watch the progress messages that appear while the game is prepared.

3. The Flash Player window opens so that we can see our game.
4. Press the arrow keys on the keyboard. Don't worry—nothing should happen!
5. Close the Flash Player window.

What just happened?

We tested our game, but it wasn't very exciting to see, was it?

However, when we clicked on the **Test Game** icon, Stencyl saved the game file, put together all the information that we have provided and compiled it into a Flash game that was displayed in the Flash Player window.

When we pressed the arrow keys on the keyboard, absolutely nothing happened because we haven't provided Stencyl with any information about how the monkey actor will be controlled by the player!

The other problem we have is that we have no scenery in our scene. If we don't make the scene interesting, it's not going to be a very interesting game, so let's make that our next task.

Downloading and using tiles for scenery

Currently, our monkey is floating in midair, which doesn't present the player of our game with a very interesting scene! We need to build some scenery, made from tiles, so that our monkey can run and jump on something.

Once again, we're going to visit StencylForge and search for some resources that we can use in our game, so let's get on and do that:

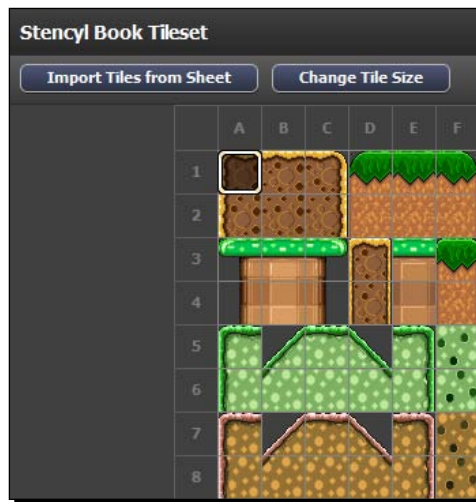
Time for action – downloading tiles from StencylForge

The game file to import and load for this session is `5961_02_04.stencyl`.

1. If the **StencylForge** tab is still available at the top of the screen, then click on it. If the StencylForge tab is not visible, click on the **StencylForge** icon on the toolbar to open it up again.
2. In the left-hand side panel, under the **MEDIA** heading, click on **Tilesets**.
3. In the search box, found at the upper-right corner of the screen, type `Stencyl Book Tileset`, and press *Enter*.
4. Double-click on the thumbnail image for the **Stencyl Book Tileset** and read the information under the **Description** heading.
5. Click on the **Download** button at the upper-right of the screen.
6. Wait a moment for the tileset to download.

What just happened?

In the same way that we downloaded the monkey actor from StencylForge, we have just downloaded a **tileset**—a collection of predesigned tiles that will help us build our jungle scene.



The tileset that we are using in our game is based on a design by a Stencyl user called Ceric. The tileset was generously released by Ceric as a **public domain** resource, which means that we can use the graphics for absolutely any purpose we desire, without any licensing requirements or copyright-related issues.

We're currently looking at the Tileset Editor, which enables us to modify the tileset's graphics along with various other aspects of tile management, but we don't need to make any changes right now—for our purposes, the tiles are just fine as they are.

Now that we have downloaded the tiles, they are available for us to use in our game.

Have a go hero – searching StencylForge for tilesets

Search StencylForge and see which other tilesets are available—but don't download any of them into our game! If you would like to experiment with downloading some tilesets, it would be a good idea to create a blank game to practice with.

Adding tiles into the scene

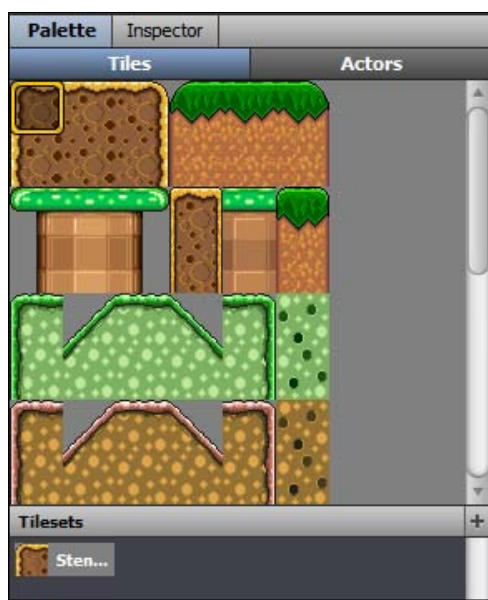
Adding tiles into the scene is usually a fun process—it's when we really start to see how the game will look—and we can experiment to our heart's desire until we get the layout just right.

Time for action – adding tiles into the scene

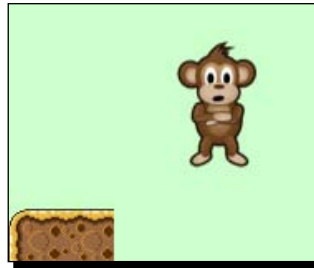
The game file to import and load for this session is `5961_02_05.stencyl`.

We're currently looking at the Tilesset Editor, but we're not going to be using it in this chapter, so we can close the tab and ensure that we are viewing the Scene Designer.

1. Close the **Stencyl Book Tileset** tab by clicking on the small **x** icon in the tab.
2. Close the **StencylForge** tab.
3. Click on the **Jungle** tab to display the green jungle scene. If the **Jungle** tab isn't visible, click on the **Dashboard** tab, click on **Scenes** in the left panel, and then double-click on the **Jungle** scene thumbnail image in the right-hand side panel.
4. In the **Palette** panel, on the right of the screen, click on the **Tiles** palette button, to display the tileset as follows:



5. The tile in the upper-left corner of the tileset is already selected, so move the mouse cursor to the lower-left corner of the green jungle scene and click the mouse once to place the tile.
6. In the **Tiles** palette, click on the second tile present in the top row of the tileset to select it.
7. Place the newly-selected tile by clicking on the green jungle scene, immediately to the right of the first tile, so the lower-left corner of the jungle scene looks like the following screenshot:



What just happened?

We've started to design the layout of the jungle scene, but, so far, we have only placed two tiles.

Designing a scene using tiles is a straightforward process—we just click in the **Tilesets** palette to select a tile, and then we click on the scene to place the tile.

Our next step is to complete the basic layout for the jungle scene but, before we do that, we're going to learn to make use of the design tools that are available to us.

Working with tiles

If we had to click once on the scene *every* time we wanted to place one tile, the fun of designing a scene would soon turn into frustration! Let's have a look at some tips that will help us to speed up our placement of tiles in the Scene Designer.

Deleting tiles from a scene

To delete a tile from a scene, we first need to ensure that we do not have a tile attached to the mouse cursor, and this is done by pressing the *Escape* key on the keyboard. We can then click on any tile and press *Delete* on the keyboard to remove it.

Replacing existing tiles in a scene

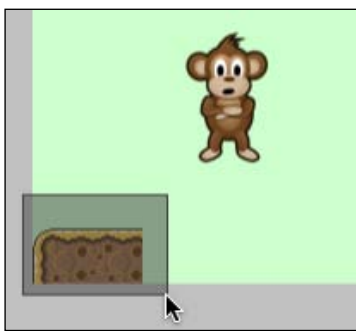
If we want to replace an existing tile, we don't need to delete it first—we can simply place a new tile over the top of it.

Moving tiles in a scene

We can move a single tile by clicking on the tile in the scene and then dragging it to a new location or, once we have clicked on a tile, we can use the arrow keys to position it. Again, we need to press *Escape* first, to ensure that we do not have a tile selected in the tile palette.

Selecting multiple tiles

We can select multiple tiles by clicking-and-dragging a selection box around the tiles that we want to select. The trick is to ensure that we start the click-and-drag on an unused part of the scene—or start the click-and-drag outside the edge of the scene—as shown in the following screenshot:



When we draw the selection box and let go of the mouse button, the selected tiles will be highlighted as a group, so they can be dragged or deleted together, which is much quicker than moving or deleting one tile at a time!



Placing multiple copies of tiles

If we really want to speed things up, we can place multiple copies of the same tile on the scene, by "painting" the tiles into place, rather than clicking once to position each individual tile. To paint a tile onto a scene, we select the required tile from the **Tiles** palette and then click-and-drag the mouse to cover the areas where we want the tile to appear.

We can also select multiple tiles from the **Tiles** palette by clicking-and-dragging on the tiles that we want from the palette. We can then single-click to add that block into the scene or, we can "paint" the block onto the scene by clicking and dragging, just as we can do with a single block.

Have a go hero – experimenting with tiles in the scene

The game file to import and load for this session is `5961_02_06.stencyl`.

Using the skills that we have discussed earlier, we can now go ahead and experiment with placing tiles in the scene. Go wild—after all, we're in a jungle, but don't spend too much time making the perfect scene—this is just an experiment, and our next task will be to delete our experimental tiles!

Try experimenting with the following techniques in our jungle scene:

- ◆ Placing a tile over the top of an existing tile
- ◆ Adding a single tile
- ◆ Painting a single tile onto the scene
- ◆ Deleting single and multiple tiles
- ◆ Moving single and multiple tiles
- ◆ Painting blocks of multiple tiles into the scene

The more we experiment with the tools, the more familiar we will become with them, and the easier it will become to design our scenes in the future!

It really doesn't matter what our scene looks like after experimenting with the tile-placement techniques—we're going to tidy it up shortly!

Finalizing the initial design

We've had fun experimenting with tiles in the Scene Designer, but we need a scene layout that we can work with for the rest of the tutorial, so the next task is to clear up the scene.

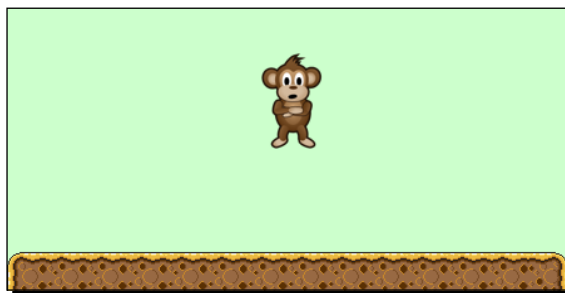
Have a go hero – tidying up the scene

The game file to import and load for this session is `5961_02_06.stencyl`.

We have now discovered all the tools that we need to place tiles into a scene, so let's delete our experimental tiles and set up our scene for the rest of the tutorial.

We need to do the following:

- ◆ Delete the experimental tiles that we have just placed in the scene.
- ◆ Ensure that the jungle scene looks like the following screenshot.
- ◆ Save the game!



We're going to keep the layout very basic to start with, as it will allow us to build up our game without becoming distracted by too many tiles—later on, we'll be adding to the scene to make it more challenging for the player.

Reviewing our progress

We've already made a lot of progress and have discovered many of Stencyl's game-design features, so let's review what we have achieved so far—we have:

- ◆ Created a blank game
- ◆ Created a game scene
- ◆ Downloaded an actor from StencylForge
- ◆ Added an actor to a scene
- ◆ Downloaded a tileset from StencylForge
- ◆ Added tiles to a scene
- ◆ Modified the layout of tiles within a scene

We now have a basic scene layout for our game, but there is currently a vital element of the game that we still need to add.

Using behaviors to interact with our game

We can't really call our current creation a *game*, because there isn't any interaction! Players need to be able to react to what is happening in our game, or it won't be much fun so, in this part of the book, we are going to implement the following gameplay elements:

- ◆ Controlling the monkey with the keyboard
- ◆ Scrolling the scene as the monkey runs

Working with behaviors

Behaviors are the instructions, or rules, for our game. Currently, we have a scene and an actor, but we haven't implemented any rules relating to how our game will work.

Think of behaviors as the answers to questions that are asked when certain events occur in our game, such as the ones listed here:

- ◆ What happens when the player presses keys on the keyboard?
- ◆ What should happen when my actor reaches the edge of the screen?
- ◆ What happens when my player actor bumps into an enemy actor?

The behaviors in our game can listen out for these mentioned **events**, and can then apply the rules that we have specified. Understanding events is a very important aspect of game development—hundreds of events will occur within our game every second, and it is up to us, as developers, to decide how we will respond to those events.

The good news is that Stencyl, as we might now expect, will manage all the hard work for us. All we need to know is that events will occur, and that our behaviors will manage how our game reacts to those events.

Adding behaviors

We are now going to start adding some behaviors into our game so that it will start coming to life! First of all, let's get that monkey moving.

Time for action – attaching a behavior to an actor

The game file to import and load for this session is `5961_02_06.stencyl`.



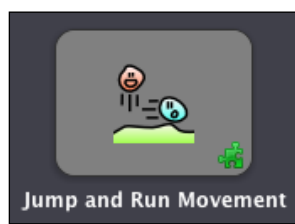
We will be working with the monkey actor, so ensure that it is currently on display in the **Animation Editor**.

Stencyl comes packaged with many useful behaviors—and we first need to tell Stencyl to attach the required behaviors to the relevant objects within our game. In this case, we need to attach a behavior to the monkey actor, so that we can control it with the keyboard.

1. Click on the **Stencyl Book Monkey** tab to display the monkey in the Animation Editor.
2. Click on the **Behaviors** button in the row of buttons at the upper-center of the screen.



3. Click on the message **Click here to choose a Behavior to attach to this Actor Type** that appears in the main panel.
4. In the left-hand side panel of the **Choose a Behavior** dialog box, click on the **Controls** option that appears under the **FROM YOUR LIBRARY** heading.
5. Scroll down the list of behaviors that appears, until you can see the thumbnail image for the **Jump and Run Movement**, then double-click on it.



What just happened?

We needed to find a behavior that would allow us to control the monkey with the keyboard, so we first selected the tab for the monkey actor, and then we chose the **Jump and Run Movement** behavior, so it is now attached to the monkey actor.

The instructions in the newly-attached behavior will now apply to the monkey actor but, before we test our game, we need to provide the behavior with some information, so that it knows exactly what to do.

Configuring behaviors with Attributes

Some behaviors must be provided with information so that they can do their job but, at the moment, the **Jump and Run Movement** behavior doesn't have all the information that it needs.

The configuration information that we supply to a behavior are called **attributes**, and an attribute could be almost any type of data such as a number, text, a type of actor, or even a list of items.

As we can see, if we scroll down the configuration information for the **Jump and Run Movement** behavior that we have onscreen, there are quite a lot of attributes needed to configure this behavior!



Fortunately, the **Jump and Run Movement** behavior only needs us to provide the following attributes to get started:

- ◆ The keys that will control the actor.
- ◆ The animations that will be used when the actor moves.

Our next task is to specify these attributes so that we can move the monkey with the keyboard and see the correct animations as the monkey moves around the scene.

Time for action – configuring the behavior

The game file to import and load for this session is `5961_02_07.stencyl`.

Ensure that we can see the attributes for the *Run and Jump Movement* behavior by opening the monkey actor and clicking on the **Behaviors** button at the upper-center of the screen.

Let's get the keyboard and animation information into the behavior:

1. Click on the drop-down arrow next to the heading **Left Control** and select the item labeled **left**.
2. Click on the drop-down arrow next to the heading **Right Control** and select the item labeled **right**.
3. Click on the drop-down arrow next to the heading **Jump Control** and select the item labeled **up**.
4. Scroll down the configuration screen so that the section with the red warning triangle icons can be seen.
5. Click on the **Choose Animation...** button next to the first icon, which is labeled **Idle Left Animation**.
6. In the **Choose Animation** dialog box that appears, click on the thumbnail for the monkey's **Idle Left** animation.
7. Click on **OK** to confirm your choice.
8. Click on the **Choose Animation...** button next to the second icon, which is labeled **Idle Right Animation**.
9. In the **Choose Animation...** dialog box that appears, click on the monkey thumbnail image labeled **Idle Right**.
10. Click on **OK** to confirm your choice.

What just happened?

We have selected the keys that we are going to use to control the monkey—arrow left, arrow right, and arrow up—and we have specified which animations are going to be displayed when the monkey actor is idle facing left or idle facing right.

However, we haven't quite finished configuring the *Jump and Run Movement* behavior yet—we need to specify the last few animations, and that's our next task.

Have a go hero – configuring the remaining animations

We have already configured the first two animations, so go ahead and configure the final four:

- ◆ Run left
- ◆ Run right
- ◆ Jump left
- ◆ Jump right

Just click on the **Choose Animation...** buttons and select the appropriate animation from the dialog box.

If we make a mistake when choosing the animations, it doesn't matter! We can simply click on the **Choose Animation...** button again, and select the appropriate animation.

Save the game!

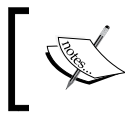
Always save the game after changes have been made!

Testing the game

Now that we have attached a behavior to the monkey, and we've configured the behavior to respond to the required key presses and display the appropriate animations, it's time to test our progress so far.

When developing a game, it is advisable to test often—perhaps even after every significant change that is made to the game. This allows us to see when things have gone wrong, and to be able to easily step backwards through any changes and fix anything that we might have done wrong. Of course, testing also enables us to see what has gone *right*, so we can enjoy the progress that we are making in the development process!

It doesn't cost anything to test a Flash game in Stencyl, and it's a very quick process, which usually takes a few seconds, so why not test as often as we can?



This is a *deliberate mistake* alert! It can be very frustrating to follow the steps in a tutorial, only to find that we do not experience the results that we are expecting.

Be aware that when we test the game this time, it's not going to work quite how we expect it to. However, don't worry—it will only take a few moments to fix the problem, and it's a good opportunity to learn that sometimes things go wrong when developing a game, so we can have the experience of putting them right!

Time for action – testing the game to find a problem!

The game file to import and load for this session is `5961_02_08.stencyl`.

Let's test the game so we can see what's going to go wrong!

1. Save the game.
2. Click the **Test Game** icon on the toolbar.

3. Wait for the game to compile and display in the Flash Player Window.
4. Press the left and right arrow keys to move the monkey and press the up arrow key to try to make the monkey jump.

Clearly things aren't working as we would expect!

What just happened?

We've experienced our first **bug**—however, it's not a fault with Stencyl—we just haven't set the game up quite right!

Although the left and right arrow keys are making the monkey move in the correct direction, the monkey is floating in midair, the animations for running left and right aren't working, and the jump button isn't doing its job.

It's almost impossible to develop a useful computer application—including a game, without introducing a mistake, or bug, during the development process, so when this happens, we need to find the problem and fix it!

Improving the scene mechanics

To fix our bug, we need to make a single change to the game. Currently, there is no gravity to make the monkey fall to the ground, so we need to make an adjustment to a setting in the Jungle scene.

Time for action – adding gravity to the Jungle scene

The game file to import and load for this session is `5961_02_08.stencyl`.

Gravity in a game enables actors to fall or float, depending on the configuration that we specify, and we need the gravity in our game to allow the monkey actor to fall—just as it would in a real jungle—so let's make that change to the Jungle scene:

If the **Jungle** tab isn't visible at the top of the screen, we'll need to open up the Jungle scene first.

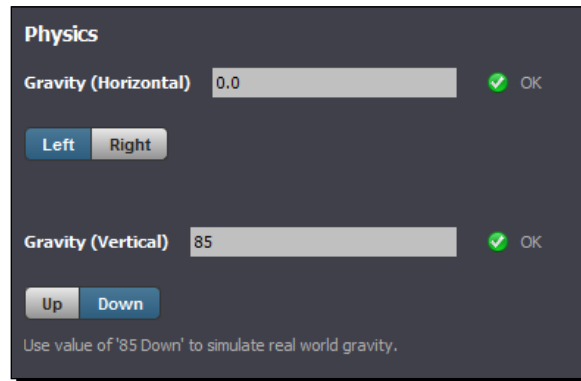
1. Click on the **Dashboard** tab.
2. Click on **Scenes** under the **RESOURCES** heading in the left-hand side panel.
3. Double-click on the thumbnail image for the **Jungle** scene.

Now that we can see the **Jungle** scene, we can change the required setting:

1. Click on the **Physics** button in the row of buttons in the upper-center of the screen, as shown in the following screenshot:



2. Change the contents of the text box next to the label **Gravity (Vertical)** so that it contains the number 85 as shown in the following screenshot:



3. Click on the **Test Game** icon on the toolbar to save and test the game.
4. Wait for the game to compile and display in the Flash Player window.
5. Use the left, right, and up arrow keys to control the monkey.
6. Try to move the monkey off the left or right edge of the scene.

What just happened?

We added gravity to our game, so now the monkey actor's behavior can do its job properly!

The Jump and Run Movement behavior needs the scene to have a gravity setting before it will work properly, because the running and jumping actions rely on the monkey being on the "ground"—which is represented by the tiles in our game.

If we have a closer look at the **Physics** settings on the screen, we can see that gravity can be set vertically and horizontally, which can provide some very interesting gameplay in the appropriate places!

We set the vertical gravity value to 85, because this number most closely represents the effects of real gravity on our actor, but do remember that we are only using a game physics engine—not a scientific simulation of real life!

Keeping an actor in a scene

While testing the game, it may have become apparent that there is another problem—when the monkey actor completely leaves the scene, by running off the sides, it disappears and never returns!

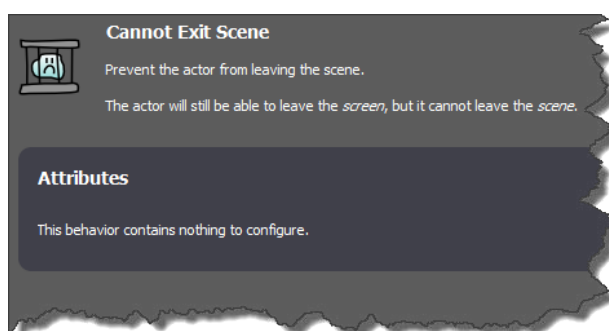
If that problem didn't occur during the last test, then test the game again right now, and make the monkey run off the side of the scene so that it disappears.

Time for action – attaching another behavior to the actor

The game file to import and load for this session is `5961_02_09.stencyl`.

We can easily prevent actors from leaving a scene by attaching another behavior, so our next task is to do just that!

1. Ensure that the **Stencyl Book Monkey** tab is active.
2. If the **Stencyl Book Monkey** tab isn't currently open, go to the **Dashboard**, click on **Actor Types** under the **RESOURCES** heading, and double-click on the thumbnail image for the monkey.
3. Click on the **Behaviors** button at the upper-center of the screen.
4. Click on the **+ Add Behavior** button at the very lower-left corner of the screen.
5. In the left-hand side panel of the **Choose a Behavior** dialog box, click on the **Motion** option that appears under the **FROM YOUR LIBRARY** heading.
6. Find the **Cannot Exit Scene** behavior in the main panel, and double-click it. Be careful not to select the similarly named **Cannot Exit Screen** behavior—it doesn't do exactly what we need!
7. Read the information on the behavior screen that is now being displayed:



8. Click the **Test Game** icon to see if the monkey can still leave at the edges of the scene, then close the Flash Player window.

What just happened?

We've attached a second behavior to the monkey actor, and this additional behavior is designed to prevent the actor from leaving the scene. Hopefully, we could see that the behavior did its job when we tested the game!

Unlike the **Jump and Run Movement** behavior, which had some information that needed to be configured, the **Cannot Exit Scene** behavior does not have any attributes that can be changed, so we were able to jump straight into testing the game, as soon as the behavior had been attached to the actor.

Increasing the width of the gameplay area

Currently, our monkey only has a very small area in which to play, as he is restricted to moving within the confines of our game window which is quite small—640 pixels wide and 480 pixels high. This is a common screen size for Flash games, which is why it is the default size for Stencyl games.

It might be possible to create an interesting jungle scene in such a small space but, we can make the game more interesting for our players, if we make the gameplay area much wider, so let's change the settings for the Jungle scene to make it three times as wide.

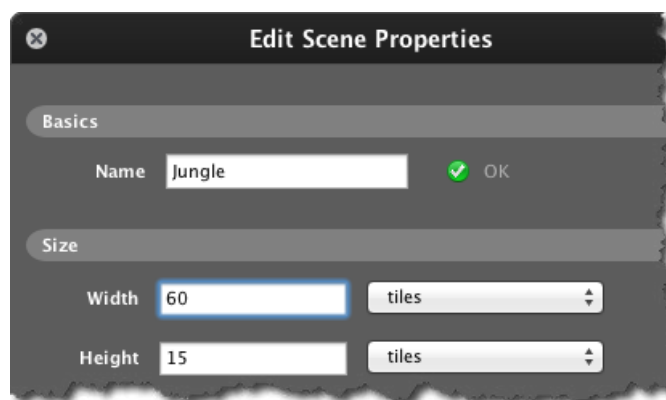
Time for action – increasing the width of the scene

The game file to import and load for this session is `5961_02_10.stencyl`.

We're going to change the width of the scene, so we need to display the jungle scene's properties:

- 1.** On the **Dashboard**, click on the **Scenes** heading.
- 2.** Double-click the thumbnail image for the **Jungle** scene.
- 3.** Ensure that the **Scene** button is selected in the row of buttons at the upper-center of the screen.
- 4.** Click on the **Properties** button in the same row of buttons.

5. In the **Edit Scene Properties** dialog box, under the **Size** heading, change the **Width** from 20 to 60.



6. Click on **OK**.
7. Test the game by making the monkey run past the rightmost edge of the scene, and then close the Flash Player window—expect a problem to occur when the monkey leaves the edge of the screen!

What just happened?

We've increased the width of the scene from its original 20 tiles to 60 tiles—three times its original width. We can see that, as soon as we confirmed the changes to the width of the scene, it immediately increased in size—it's now so wide that we need to use the horizontal scroll bar at the bottom of the Scene Designer in order to see the whole scene!

However, we have a problem because, when the monkey ran off the edge of the screen, he disappeared, never to be seen again.

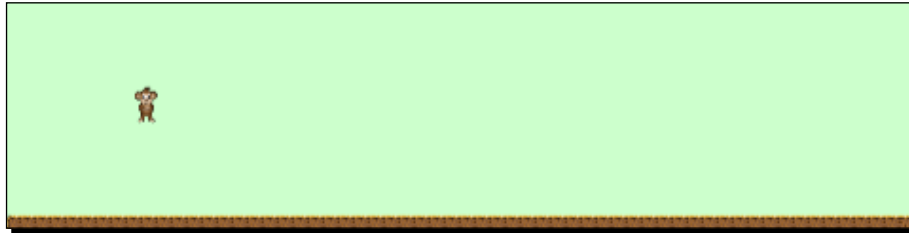
Another problem that may not immediately be apparent, is that there aren't enough tiles for the monkey to run along—when he ran past the edge of the screen, the poor monkey fell off a cliff!

Our next task is to fix both of these problems, starting with the tiles:

Have a go hero – adding more tiles to the scene

The game file to import and load for this session is `5961_02_11.stencyl`.

We already know how to add tiles, and we're already looking at the Scene Designer, so go ahead and add enough tiles to reach the far, right-hand side edge of the scene, as shown in the following zoomed-out screenshot:



When designing scenes that are too wide to fit on our screen, we can either scroll along the scene using the horizontal scroll bar, or we can zoom in and out of the scene using the zoom utility buttons in the toolbar at the left-hand side of the Scene Designer. Clicking on the zoom tools and then clicking on the scene, enables us to zoom in and out as required.

When we have added the extra tiles, we can test the game again and, this time, instead of running off a cliff when we hold the right arrow key down, the monkey will run off the side of the scene and disappear but, if we hold the left arrow key long enough, he will run back into the scene so we can see him again.

That just leaves us to solve the problem of the disappearing monkey—we'd like to see him, wherever he is, on our wide scene!

Making the screen scroll

Rather than losing our monkey off the side of the screen, we're going to ensure that the screen scrolls automatically as he runs.

Once again, behaviors come to our rescue—in just a few clicks, we will have a scrolling scene for our monkey to enjoy.

Scrolling through a scene in Stencyl is easily managed by using the **Camera Follow** behavior, and we attach the behavior to the actor that we want to follow—in our game, that's the monkey.

Time for action – attaching the Camera Follow behavior

The game file to import and load for this session is `5961_02_11.stencyl`.

1. Go to the **Dashboard** tab and click on the **Actor Types** heading.
2. Double click on the **Stencyl Book Monkey**.
3. Click on the **Behaviors** button in the row of buttons at the upper-center of the screen.
4. Click on the **+ Add Behavior** button at the lower-left corner of the screen.
5. In the **Choose a Behavior** dialog box, click on the **Game** option in the left-hand side panel, under the heading **FROM YOUR LIBRARY**.
6. Locate the **Camera Follow** behavior and double-click on it.
7. Do *not* change the scroll speed attribute in the behavior configuration screen!
8. Test the game and try to make the monkey run off the right edge of the screen—run around and see how far the monkey can go!

What just happened?

Attaching the **Camera Follow** behavior to an actor forces the scene to scroll whenever the actor tries to pass the center of the current screen view. Once our actor reaches either the leftmost or rightmost edge of the scene, it will stop scrolling, which allows our actor to move to the edge of the scene.

It would be a good exercise to test our game again, just to be sure that we are familiar with the way in which the Camera Follow behavior allows the actor to move around the wide jungle scene.

Adding some interesting scenery

Although we now have the basic game mechanics in place, we don't yet have a very interesting design for our platform game—our monkey can only run to left and right, but there are no obstacles!

Our next task is to add some detail to the scene, starting with additional tiles to make the scene more fun for our players:

Have a go hero – adding some interesting tiles to the scene

The game file to import and load for this session is `5961_02_12.stencyl`.

We already know how to add tiles to a scene but, before starting, we just need to make sure that:

- ◆ We are viewing the **Jungle** scene in the Scene Designer
- ◆ We have clicked on the **Tiles** button in the palette, in the panel on the right-hand side of the screen, so we can see the tileset

Go ahead and add some tiles from the tileset—it really doesn't matter too much, what the scene looks like, as long as the monkey can run around and jump without too many obstructions. There's an example of the left-hand side part of the scene in the following screenshot:



Don't forget to scroll along scene in the Scene Designer, and add platforms and obstructions along the whole length of the scene. Remember that, to view the whole width of the scene, we can use the zoom tools on the left-hand side of the Scene Designer.



Do include the green pillars at the leftmost and rightmost edges of the scene, as shown in the following screenshot, which shows the whole scene, as these pillars are used in a later session.



Fine-tuning the level design

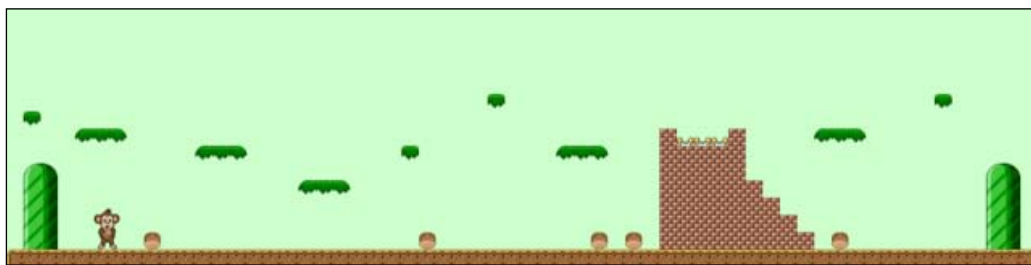
When creating a platform game we'll often find that it is too hard, too easy, or just too frustrating!

The skill of the game designer is to ensure that the players of our game will find it fun—there's no point making it ridiculously difficult, because the player will give up and move on to another game very quickly. Equally, we need to make sure that the game is not too easy to complete, otherwise the game will be finished very quickly.

While designing the layout of the game, it is important to test our progress very frequently. Make sure that it is possible for the monkey to navigate the whole length of the scene, and try to introduce some interesting challenges along the way.

The following screenshot shows an example of a level design which is not too difficult, but which has some interesting challenges. For example, if the player decides to make the monkey take the easiest route, and run along the floor of the jungle, then it will soon hit the walls of an old ruined temple, and have to run back to jump up the platforms to get over the ruins!

If this was a time-based platform game, with a countdown and bonus points for finishing the level quickly, then this is the type of challenge that can encourage a player to play the same level over and over until they can complete it in the shortest time, and gain maximum bonus points!



Finding game testers

The life of a game developer can be an isolated one! It's very easy to become lost in the development of a game without taking time out to look at it from someone else's point of view.

For this reason, it's a good idea to find someone else to test our game designs for us—it could be a friend, colleague, or a family member, but it is important to watch and listen to others while they test the game and, most importantly, make notes about what they say, so that we can improve the design of the game and make it more fun!

Summary

Wow! We've learned a lot in this chapter.

We created a blank game and added a jungle scene. We then downloaded a monkey actor from StencylForge and added a behavior to control it with the keyboard.

The next step was to use StencylForge again, to find an appropriate tileset to create some interesting content for our Jungle scene.

To ensure that the actor couldn't fall off the edge of the scene, we attached the **Cannot Exit Scene** behavior, and then we attached the **Camera Follow** behavior to the monkey, so it could run and jump along the whole length of the scene, while the scenery automatically scrolled to keep up with it.

In the final part of the chapter we made the platform game design more interesting and considered some challenges that might make our game more fun to play.

We've made quite amazing progress in just one chapter—we have all the basic mechanics in place for our platform game. However, there are two vital elements missing—we need some enemies to avoid, and we also need a challenge for our monkey to complete, so let's head off into *Chapter 3, Detecting Collisions*, and really make our game come to life!

The game file named `5961_02_13.stencyl` represents how our game should look at this point in the book.