Game Programming Course

Lv01_L00 – Level 01 Introduction

Introduction

Welcome to our online Game Programming Course. All of us involved in the creation of this course can remember the first time we programmed something. For most of us it was getting the computer to print out our name (or a bad word) a million times with a line or two of code! We can all remember our earliest coding projects and the excitement of having something that we created actually work on the computer.

It was even better watching friends smile or laugh and hearing them say, 'That's Awesome!', as they sat in front of our creations. Our learning continued and we figured out how to add special features, bonus rounds, and sometimes simply ridiculous stuff into our programs just for amusement. But some of these extras, especially ones that appeared simple, often required some serious learning.

After all this learning, we decided to create this online course that will try to walk you through

Game Programming Course

Lv01_L01 – Up and Running (10 minutes)

Introduction

Our first lesson will be an easy one. We will grab the free software we need for the course and download the level and lesson resources for the entire course.

Part A – Download GameMaker

The software that we use for this course is a FREE version of GameMaker Studio that is available at http://www.yoyogames.com . In the top right corner of the page you will see a link to their DOWNLOAD page.

Once on the download page you will see large links for downloading or buying GameMaker Studio. We want to DOWNLOAD the program. It is around 130 MB and currently called *GMStudio-Installer.exe*.

Once you have downloaded the software you can run the installer.
When the download completes, run the program. Select to use the free version of the program.

Part B – Download All Lesson Resources

The resources for the entire course are in a zip file on the lessons page of the website http://gameprogrammingcourse.com/lessons.html . Download the file and unzip it in a convenient location on your computer. If you don't know how to unzip the file, go to Level01_Lesson02. If you know how to unzip the file, move on to Level01_Lesson03.

That's it! Lesson One Completed!

Side Note

The free version that you just installed will limit the size of the game you can create. All the lessons in this game programming course work within these limitations. You do not need to purchase the program to complete this course.

HOWEVER, there will probably be a point in this course where you will have a game idea that requires the unlimited version of the program. The least expensive version is called GameMaker Studio Standard and it is only $49.99. In my opinion and the opinion of many students that have taken this course, the Standard Edition is well worth the money and will allow you to export games to Windows and Mac with no limits on the size of the game.

Game Programming Course

Lv01_L02 – Unzipping Zipped Files (10 minutes)

Introduction

Many of the files provided in this course are 'zipped' up so that you can download many files at once easily. If you know how to 'unzip' already, skip this lesson. To unzip files you need some free software that unzips files.

Part A – Download 7-zip

We will download and install a free program called 7-zip. It is a well known, free program that will easily unzip programs for you.

Go to
http://www.7-zip.org/
and download and install the 32 or 64 bit version of 7zip.

After the program is installed, it is easy to unzip files.
Just RIGHT-CLICK the file you want to unzip and select EXTRACT TO "filename".

Part B – Unzip allLevelResources.zip

You can now unzip the *allLevelResources.zip* folder that you downloaded in lesson Level01_02.

Notes

A zip file is a file that usually contains many files inside. When you unzip the file, you get all the files and folders inside the zip file. Zip files are a convenient way to send several files to someone.

Game Programming Course

Lv01_L03 – Watching The Videos (5 minutes)

Introduction

Videos provide most of the teaching, challenges, and solutions in this course. The videos are currently located at http://www.YouTube.com/gpchannel under the channel name *GameMaker Game Programming Course.* The lessons are titled by Level and Lesson number. The videos are HD and optimized for the large player setting or full screen viewing.

Part A – Watching the Videos in HD

Lets make sure that you are able to find the videos and optimize the viewing settings in YouTube.

Getting There

If you haven't watched a video yet, go to the GameMaker Game Programming Course channel on YouTube (http://www.YouTube.com/gpchannel) . Select a video and starting playing.

Quality Settings

The YouTube player should automatically pick the correct quality based on your connection speed, but sometimes it doesn't. In the bottom right corner of the player, click the settings button (looks like a gear cog). Make sure that 720p HD is selected.

Player Size

In the bottom right corner of the YouTube player you can select the size of the player. The videos should be watched in the large player or full screen.

Notes

The videos are recorded in 720p HD and should be very clear when your connection speed is good. If the videos are not clear, it is probably because your connection speed at the time isn't going as fast as it could. Sometimes refreshing the page fixes the problem, sometimes it doesn't. Sometimes going back to the page a few minutes later works too.

Coming Soon

All the original, uncompressed, crisp HD videos for this course will be available to you so that you can conveniently watch the videos offline on your computer, tablet, or mobile device.

Game Programming Course

Lv02_L00 – Level 02 Introduction

Introduction

You are taking this course to make a game, so lets get right to making your first game – a simple two player battle game. Players will move around the level and be able to fire balls at each other (not as violent as bullets!) and decrease the other players life until one player is victorious. We will fill it with some good beginner graphics, sounds, explosions, and a little bit of code by the end of this level.

The two player game is great for learning all about the basic game design process and to exposed you to many fundamental programming concepts that you will use for the entire course. The best part of this level is challenging a friend to a battle when you're all done.

Level 02 is a great level and it's going to get you hooked on programming!

Have fun.

Game Programming Course

Lv02_L01 – Setting Up Sprites, Objects, and the Room
(40 minutes)

Introduction
Our first step in creating our two player game will be to learn about sprites, objects, and how to create the game room with them.

Resources You Will Use
Sprites used in this lesson can be found in the *allLevelResources/Lv02* folder.

Part A – Setting Up Sprites, Objects, and the Room

Watch the video: Lv02_L01A
Done File: Lv02_L01ADone.gmk

(The 'done file' is the result of completing a video. If you follow and imitate the programming in the above video, you will have created the done file yourself. The done file is useful for when you want to watch a video lesson but not actually code it in yourself. You can use the done file to continue onto other parts of the lesson.)

Challenge 1– Add a Second Player

Add a second player to the game – it is a two player game!
You will need to add *spr_player2, obj_player2,* and also *obj_ball2* since player2 will want to fire their own type of ball.

Challenge 1 Solution

Solution video: Lv02_L01_Ch1
Solution file: Lv02_L01_Ch1Done.gmk

Game Programming Course

Lv02_L02 – Events and Actions for Moving
(40 minutes)

Introduction

Now that we have some objects in the room, we have to get them moving around. This is where events and actions come into play. By the end of this lesson you will have objects moving and colliding.

Part A – Getting Moving with the Keyboard Event

Watch the video: Lv02_L02A

Part B – Colliding with Objects

Watch the video: Lv02_L02B
Done File: Lv02_L02BDone.gmk

Challenge 1 – Complete Second Player

Give player2 all the same behaviour that you just added to player1.

Challenge 1 Solution

Solution video: Lv02_L02_Ch1
Solution file: Lv02_L02_Ch1Done.gmk

Game Programming Course

Lv02_L03 – Creating More Objects
(30 minutes)

Introduction
You have two players moving around the room. Now it's time to fire stuff at each other! In this lesson you will learn how to create and control new objects.

Resources You Will Use
Grab and open the *CheatSheet01.html* in the Lv02 folder. It has a bunch of common programming commands that we will use in this level. Printing it out might be a good idea. Keep it close by so that you can refer to it when you are having trouble remember the exact spelling or format of a command.

Part A – Firing Balls with the Keypress Event

Watch the video: Lv02_L03A
Done File: Lv02_L03ADone.gmk

Part B – Destroying Objects

Watch the video: Lv02_L03B
Done File: Lv02_L03BDone.gmk

Challenge 1– Finish Up Player 2 and Ball Destruction

Give player2 the ability to fire their ball2.
When both types of balls hit the walls the balls get destroyed.
When ball1 hits player2, ball1 is destroyed.
When ball2 hits player1, ball2 is destroyed.

Challenge 1 Solution

Solution video: Lv02_L03_Ch1
Solution file: Lv02_L03_Ch1Done.gmk

Game Programming Course

Lv02_L04 – Jumping Objects, Image Angles, Sounds
(10 minutes)

Introduction

Sometimes it's nice to be able to pick up an object and move it to a certain location in the room. It's also nice to be able to turn sprites to face different directions. Playing sounds is pretty good too. These are the three things we'll learn how to do in this lesson.

Part A – Jumping the Players when they are hit

Watch the video: Lv02_L04A

Part B – Changing the Image Angle of sprites

Watch the video: Lv02_L04B

Part C - Sounds

Regular sounds are covered in the first part of the video and background music is covered in the last 2 minutes.

Watch the video: Lv02_L04C
Done File: Lv02_L04CDone.gmk

No Challenge this time!

Game Programming Course

Lv02_L05 – Cloning Get the Ghosts
(45 minutes)

Introduction

Now it's time for you to practice the past few lessons. You'll look at a program that we have provided. It's a 'Get the Ghost' game. Play it for a minute and then reproduce the entire game yourself!

Part A – Observe the Game

Download and play the *GetTheGhosts01.exe* for a minute or two. It's not completely a real game yet, but it will be one of the games we add on to as we learn how to program.

Watch the video: Lv02_L05A

Challenge 01 – Clone the Game

Now that you've watched the game, you're job is to clone it. Start from a brand new empty project file and reproduce the program so that it is an exact match (or almost an exact match) of what you see in my program. The sprites and sounds can be found in the Lv02 resources folder.
Remember to use the coding cheat sheet!

Challenge 1 Solution

The solution is a three part walk through of me making the game.

Solution videos: Lv02_L05_Ch1a (sprites, objects, room)
Lv02_L05_Ch1b (events and actions)
Lv02_L05_Ch1c (sounds and background music)

Solution file: Lv02_L05_Ch1Done.gmk

Congratulations! You've completed Level 02!

In the next level, we will introduce you to creating, changing, and checking your own user created game variables. Your programs will jump to an entire new level - Level 03!

Game Programming Course

Lv03_L01 – Introduction To Variables
(60 minutes)

Introduction
blah blah blah

Resources You Will Use
It's Level 03 now, so you can find all the files in the lv03 resource folder.

Part A – Creating and Changing Instance Variables

To save time, a pre-made game project has been made that will help us learn about variables. Go the resources folder and open the game project called Lv03_L01_StartHere.gmk

Watch the video: Lv03_L01A
Done File: Lv03_L01ADone.gmk

Challenge 1– Add the Following
a) Give the player a new variable called *money*. When the player collides with a coin, add one to the player's money and destroy the coin. Use a show message to confirm it is working!

b) Since the player can gain points, lets make a way for the player to lose points. Add the code so that when an arrow hits a wall (misses a ghost) the player loses 2 points. Use a show message to confirm it is working.

Challenge 1 Solution
Solution video: Lv03_L01_Ch1
Solution file: Lv03_L01_Ch1Done.gmk

Part B – Drawing Variables

Continue with the file you are working on.

Watch the video: Lv03_L01B
Done File: Lv03_L01BDone.gmk

Challenge 2– Add the Following
This challenge will have a few more tasks to code so that you can get confident with the basic use of instance variables.

a) When a player picks up gold, add 10 to their money. Remember to destroy the gold. Play sound *pickup*.

b) Give the player a variable called *life*. It should start at 100. Use the draw object to draw this variable to the screen.

When the player touches the poison object they should lose 1 life but don't destroy the poison! Watch how fast life goes down when they player touches the poison.

c) Eating apples will give the player 25 more life.

d) Touching the blue potion will bring you to 100 life instantly.

e) Code this in the WALL object. When the wall collides with the player, lower the player life by one.

Challenge 2 Solution
Solution video: Lv03_L01_Ch2
Solution file: Lv03_L01_Ch2Done.gmk

Game Programming Course

Lv03_L02 – Variables and Conditions
(75 minutes)

Introduction

Variables are great for remembering values. But the real power of variables is being able to ask questions about their values, and depending on these values, perform different blocks of code. This lesson will introduce you to the power of simple conditional statements.

Part A – Introduction to Conditions

Read the notes on variables.

We'll continue working on the variable game project from the last lesson. Open the completed Challenge 02 from last lesson, Lv03_L01_Ch2Done.gmk

Watch the video: Lv03_L02A Done File: Lv03_L02ADone.gmk
(player life, wall life)
Watch the video: Lv03_L02B Done File: Lv03_L02BDone.gmk
(money to fire)
Watch the video: Lv03_L02C Done File: Lv03_L02CDone.gmk
(limit life to 100)
Watch the video: Lv03_L02D Done File: Lv03_L02DDone.gmk
(potion increases life but has amount)

Challenge 1– Add the Following

before you test this out, change the player create event code so that the player starts with money=1000. Easier to test this way!
give ghosts life, hit 2 times to destroy.
pick up 6 coins say you got all coins
ghost keep track of how many times they have bounced off of walls, if they bounce 5 times they change direction to move left!
destroy 3 ghosts and message "Next Level" and end game

Challenge 1 Solution

Solution Video: Lv03_L02_Ch1
Solution File: Lv03_L02_Ch1Done.gmk

Game Programming Course

Lv03_L03 – Using Random Values
(60 minutes)

Introduction
blah blah blah

Part A – Setting Variables to Random Values

To save time, a pre-made game project has been made that will help us learn about random values and variables. Go the resources folder and open the game project called Lv03_L03A_StartHere.gmk

Watch the video: Lv03_L03A
Done File: Lv03_L03ADone.gmk

Challenge 1 – Random in Step Event

In the step event of the banana, pick a random number between 0 and 30. If the number is a 1, change the direction of the banana to a random value between 0 and 359. If successful, this will have the banana changing directions every second or so.

In the step event of the burger, add onto the existing code. Pick a number between 0 and 1000. If the number is 10 or less, set the x and y variables of the burger to new, random values, between 0 and 400 (the size of the room). You should see the burger 'jump' to a new position in the room every 3 seconds or so.

Challenge 1 Solution

Watch the Video: Lv03_03_Ch1
Solution File: Lv03_03_Ch1Done.gmk

Part B – Random Values, Random Actions

To save time, a pre-made game project has been made. Go the resources folder and open the game project called Lv03_L03B_StartHere.gmk
fire arrow or burger
jump to one of three locations
one of two types of collision effects on wall

Watch the video: Lv03_L03B
Done File: Lv03_L03BDone.gmk

Challenge 2 – Random Generation

Continue using the file you were just working with, Lv03_L03BDone.gmk

Three new things to add to the game,
a) When you press the “M” key, make the player randomly decide whether to start moving up, down, left, or right at a speed of 4.
b) When the player touches the chest, destroy the chest and randomly decide whether to create a coin or some gold in the same location. There should be more of a chance of a coin appearing.
c) When the ghost is hit by an arrow, there is a $1/5^{th}$ chance that the ghost is destroyed (that's a 20% chance).

Challenge 2 Solution
Watch the Video: Lv03_03_Ch2
Solution File: Lv03_03_Ch2Done.gmk

Game Programming Course

Lv04_L01 – Modifying Get The Ghosts
(60 minutes)

Introduction

Lets use what you've learned in the past few lessons and add some new features to the Get The Ghosts program that you cloned. There is no instructional video for this lesson, just a long challenge and a solution video (if you need a few hints along the way).

Challenge 1 – Add to Get The Ghosts

You can start with this file, Lv04_L01_Ch1GetTheGhostsStart.gmk
You can also run the file Lv04_L01_Ch1GetTheGhostsDone.exe

Don't skip modifications as some of them build upon each other!
These modifications are solved in the first solution video.

a) Give the player two new variables, *life* and *points*. Set the life to 100 and the points to 0. (solved at 0:00)
b) Draw these two variables somewhere in the room. Since the room has a black background, you won't see the variables being drawn out unless you switch to a different drawing color. To do this, make the first line in the Draw Event *draw_set_color(c_white)* and you will be drawing in white. Follow this line with your normal draw code. (solved at 0:47)
c) Give the ghosts a variable called *life* and set it to 2 and change the existing code so that the ghost will have to get hit twice before it is destroyed. (solved at 4:14).
d) Give the player 5 points every time they hit the ghost with their laser. (solved at 6:25)
e) Add this section of code to the event Ghost collides with Wall.

direction = direction + 180

It will make the ghost appear to bounce off of walls when the collide with walls once we get them moving. (Solved at 7:48)
f) Make the boss randomly produce a ghost every few seconds. The ghost that is made should be moving to the right. (Solved at 8:47)

These modifications are solved in the second solution video.

g) Make a new object called *laserred* and get the ghosts to fire, every second or two, red lasers at an angle of 260-280 degrees every second or two. Remember to set the image_angle of the laser so it looks correct in the game. (Solved at 0:00)
h) Since there are no walls located along the bottom of the screen, red lasers just keep on going and going. We need to destroy them when they leave the room. There is an event called Other->Outside Room that is called when an object leaves the room. Add an instance destroy method inside this event for the red laser. (Solved at 3:30)
i) When the player gets hit by the red laser, destroy the laser and reduce the player's life by 10. (Solved at 4:54)

j) Just after you take 10 off the player's life, check to see if the player's life is below or equal to 0. If it is, show the message, "Game Over! Click to Restart!" and then use the line *game_restart()* to have the game restart itself. (Solved at 6:20)
k) Create a new object called *healthpack.* When the player collects a health pack it should be destroyed and the player should get +25 onto their life and make sure that the player's life doesn't go over 100. Add a health pack or two neaer the player to test it out. (Solved at 8:05)
l) Last modification! When a ghost is destroyed, there should be a 1 in 10 chance that it drops a health pack downward. If a health pack goes outside the room, it should destroy itself. (Solved at 11:15)

Challenge 1 Solution
The solution is split into 2 videos

Watch the Video: Lv04_01_Ch1
Watch the Video: Lv04_01_Ch1b
Solution File: Lv04_01_Ch1_GetTheGhostsDone.gmk

Complete Game Code
Lv04_L01_Code_GetTheGhosts

Game Programming Course

Lv04_L02 – Modifying Two Player Game
(60 minutes)

Introduction

Lets get some more practice using the concepts taught in the previous few lesson and modify our Two Player Game project from Level 2. There is no instructional video for this lesson, just a long challenge and a solution video (if you need a few hints along the way).

Challenge 1 – Add to the Two Player Game

You can start with this file, Lv04_L02_Ch1TwoPlayerGameStart.gmk
You can also run the file Lv04_L02_Ch1TwoPlayerGameDone.exe

Instead of reading all the modifications below, run the .exe file above and observe the game. What modifications have been made? Try to program as many of the observed modifications as possible without reading the list below. If you think you have all of them done, check the list below and see how you did!

These modifications are solved in the first solution video.

a) Give each players a variable called *hp* and set it to 100. Draw these variables to the screen. (Solved at 0:00)
b) When player 1 gets hit by player 2's ball, decrease their life by 25. Repeat for the player 2 object. (Solved at 3:20)
c) Right now the players are jumped back to their starting position when hit once. Change this behaviour so that the players don't jump back unless the hit points variable reaches 0 or below. Also set the player's hit points back to 100 after they jump back. (Solved at 4:25)

These modifications are solved in the second solution video.

d) Give each player a new variable called *ammo* and set it to 10. (Solved at 0:00)
e) When the player presses the key to fire, check to make sure that they have some ammo (more than zero!). If they do, let them fire and decrease the ammo by one. Draw both players ammo values to the screen. (Solved at 0:35)
f) Make a new object called *ammobox* . When a player collects an ammo box, their ammo will increase by 10. Place a few in the room and test. (Solved at 4:00)
g) Go back to where you jump the players. Make the ammo variable reset to 10 when players are jumped. (Solved at 6:20)
h) Make a new object called *producer*. A producer doesn't need a sprite, it will sit in the room invisible. A producer will randomly fire out ammo boxes, in any direction, every couple of seconds. Place a single producer in the middle of the room. (Solved at 7:30)
i) Make sure that ammo boxes will stop when they hit walls. (Solved at 9:53)

Challenge 1 Solution
The solution is split into 2 videos

Watch the Video: Lv04_02_Ch1
Watch the Video: Lv04_02_Ch1b
Watch the Video: Lv04_02_Ch1c
Solution File: Lv04_01_Ch1_TwoPlayerGameDone.gmk

Complete Game Code
Lv04_L02_Code_TwoPlayerGame

Game Programming Course

Lv05_L01 – Counters and Alarms for Timing
(60 minutes)

Introduction

In the previous lesson/s you used random numbers in the step event to make objects perform tasks at random times. But sometimes you want an object to perform a task at specific time intervals. This lesson will cover two popular methods of making objects 'time' their actions.

Part A – Using Counters

Watch the video: Lv05_L01A
Donefile: Lv05_L01ADone

Challenge 1– Redo the Video without watching

Grab the pre-made starting file Lv05_L01_StartHere.gmk.
Re-add the these three tasks that were showing in the lesson video.

a) paddle will fire a ball every one second
b) player is limited to firing every 15 steps
c) bomb will explode 2 seconds after it is created

Challenge 1 Solution

Solution: Watch the video again! This is very commonly used, useful code to know. Practice until you can do it without using the video for help.

Part B – Using Alarms

Watch the video: Lv05_L01B
Donefile: Lv05_L01BDone

Challenge 2 – Add these tasks

Continue adding to the file from the end of the video lesson Lv05_L01BDone.gmk
Add the following using an alarm.

a) use an alarm to make a ball add 30 degrees to its direction every 15 steps. This should make it move in a circle-ish (not a real word) shape.
Use the line *direction = direction + 30* in your code.
b) After the old man presses the "D" key, he will self destruct in 3 seconds.
c) Make it so that the left and right arrow keys make the man move at a speed of 1 to the left and right. Now limit the use of the keys so that they only work every 3 seconds. For example: if they press the right arrow key, they will move right and then the left or right arrow key will have no effect until 3 seconds has past. Test out the executable file if you're not sure you understand!

Challenge 2 Solution
Solution Video: Lv05_L01_Ch2
Solution File: Lv05_L01_Ch2Done.gmk

Challenge 3 – Repeat the last Challenge with counters!
Remember those last three tasks you did with alarm? Repeat them with counters instead. Task two (the destruction of the man) might be trickier than you think, be creative! So open up Lv05_L01BDone.gmk again and see if you can repeat these tasks with counters.

Challenge 3 Solution
Solution Video: Lv05_L01_Ch3
Solution File: Lv05_L01_Ch3Done.gmk

Game Programming Course

Lv05_L02 – Global Variables vs Instance Variables
(30 minutes)

Introduction

We've been making good use of variables in the past few lessons. The variables we have been using are called *instance* variables because each instance (object you create in the game) keeps the variable inside of itself. There are other types of variables in programming. One type is a *global* variable. Global variables are not owned by any individual object. They are owned by the entire game and any object is free to make them, change them, or check them. This lesson will introduce you to global variables and when to use them.

Part A – Global Variables, Why we need them.

Watch the video: Lv05_L02A

Part B – Using Global Variables to replace common values

Watch the video: Lv05_L02B
Donefile: Lv05_L02BDone

Challenge 1– Add some more Global variables

Continue with the file from the videos Lv05_L02BDone.gmk
Add the these tasks.

a) make a global variable to keep track of what level the player is currently on. The first room is level 1. When the player touches the door and advances to the next room, increase the level variable by one. Also draw this variable to the screen.
b) make a global variable to keep track of the number of times the player has hit a ghost with a laser. Increase this variable when ghosts get hit by lasers. Draw to screen.
c) make a global variable that will keep track of how fast the player laser will move when created. When the player has earned 5 ghost hits, the speed of the laser will increase to 12.

Challenge 1 Solution

Solution Video: Lv05_L02_Ch1
Solution File: Lv05_L02_Ch1Done.gmk

Game Programming Course

Lv05_L03 – More About Conditional Statements
(60 minutes)

Introduction

In the past several lessons we have used simple *if* statements to check the values of variables to help us decide whether or not to execute some code. While these simple conditions work well, sometimes you need slightly more complicated conditions in your code. In this lesson we will take a look at some of the extra options you have when asking *if* ...

Part A – Reading and Questions Time!

This is one of the rare topics in this course where reading some notes and answering some concepts questions will help you quite a bit. The notes,concept questions, and solutions are bundled into one file for you.

Read the Notes: Lv05_L03_Reading01.pdf
Contains
1) Basic Operators
2) Using AND and OR
3) Using if, else if, else

Watch the video: Lv05_L03_A
(reviews the reading notes)

Try the Concept Questions Lv05_L03_ConceptQuestions01.pdf
Check your answers are at the end of the question file!

Part B – Coding Challenges

These five coding challenges come with videos explaining what to do. Watch the video and then use the pre-made starting game project to complete the challenge. Good luck!

Challenge 01 – Fire Toggle and Ammo

There is a pre-made start file: Lv05_L03_StartHereA.gmk
There is a demo video of this challenge: Lv05_L03_Ch01Ex

When the player presses the T key, this should toggle the value of the variable selected from 1 to 2 to 1 to 2 and so on. Once this is working, change the draw method so that it shows ROCK when selected is equal to 1 and ARROW when selected is equal to 2. Once this is working, the player should fire the appropriate item, a rock or an arrow. Once this is working, make sure that the player can only fire the selected item if they have enough of those items (the variables rock and arrows are already made).

Solution Video: Lv05_L03_Ch1
Done-File: Lv05_L03_Ch1Done.gmk

Challenge 02 – Keys and Locks
There is a pre-made start file: Lv05_L03_StartHereB.gmk
There is a demo video of this challenge: Lv05_L03_Ch02Ex

The player can open the chest if they have collected either of the keys. To open the chest, use the code with obj_chest { sprite_index = spr_chestOpen } . This will change the sprite of the chest to an open chest graphic.

The player can destroy the red lock only if they have collected the gold and silver key. Having just one key won't don't it!

The player can destroy the door if the player has either of the keys and has collected all 5 coins that are in the room.

Solution Video: Lv05_L03_Ch2
Done-File: Lv05_L03_Ch2Done.gmk

Challenge 03 – Target Points
There is a pre-made start file: Lv05_L03_StartHereC.gmk
There is a demo video of this challenge: Lv05_L03_Ch03Ex

The player has a points variable. When the player has between 0-9 points, show "Noob" on the screen. When they have between 10-19 points show "Not Bad". When between 20-29 show "Good", and 30 and higher show "Awesome!".

Once this is working, make it so that a pear, cherry, or strawberry is produced from the target when the target is hit from an arrow. Set the odds to 10% chance of pear being produced, 30% chance of cherry being produced, and 60% chance of strawberry being produced. When you test your program, make sure that these approximate odds are working!

Solution Video: Lv05_L03_Ch3
Done-File: Lv05_L03_Ch3Done.gmk

Challenge 04 – Color Squares
There is a pre-made start file: Lv05_L03_StartHereD.gmk
There is a demo video of this challenge: Lv05_L03_Ch4Ex

When you detect that the player is in different areas of the screen, write the following test on the screen using the draw object provided. Draw each line at a slightly different height so that when two are showing up they don't overlap.

In the blue areas on the left: "Blue Side"
In the large top right square: "Top Right"
In the gray area: "Gray Part"
In the tiny center square: "Center!"
In any of the four large corners: "Corner!"

Solution Video: Lv05_L03_Ch4
Done-File: Lv05_L03_Ch4Done.gmk

Challenge 05 – Cycling
There is a pre-made start file: Lv05_L03_StartHereE.gmk
There is a demo video of this challenge: Lv05_L03_Ch05Ex

The player has a variable called *item*. Pretend the player has 5 different items to cycle through. When they press the right arrow, add 1 to item. But if the value of item goes past 5 you should put it back to 1. When the press the left arrow key cycle down by subtracting 1 from item. If the item variable goes below 1 then bring it to 5. As you continuously press an arrow key, you should be cycling through the item values.

When this is working, add some code inside the draw event so that you show the actual names of the items.

Solution Video: Lv05_L03_Ch5
Done-File: Lv05_L03_Ch5Done.gmk

Game Programming Course

Lv06_L01 – Sprites and Images
(60 minutes)

Introduction

Graphics are important part of any video game. In this lesson we're going to learn how to do some basic sprite tasks that will give you more control over how a sprite behaves on screen. You will definitely use what you learn today in your own games.

Part A – Common Sprite and Image Options

Watch the video: Lv06_L01A
Watch the video: Lv06_L01B
Donefile: Lv06_L01ABDone.gmk

Challenge 1 – Using Sprite Index

Add the up and down direction to the player with the appropriate animated sprites.

Add a chest object into the room. When the player touches the chest object, it should change to an open chest sprite. If the player touches the chest again, it should change to a closed chest again. Now here's the problem: if the player stands on top of the chest, the chest will rapidly open and close and you won't even be able to see the changing sprites take place. Solution: Use an alarm on the chest so that it can only be opened/closed every 3 seconds. This is a nice challenge that reviews alarms! Check the solution file or video if this one stumps you.

Challenge 1 Solution

Watch the Video: Lv06_L01_Ch1
Donefile: Lv06_L01_Ch1Done.gmk

Challenge 2 – Growing Health

Modify the health object so that it grows bigger quickly until it is twice its normal size. Then it returns to it's original size and the pattern repeats forever.

Now try to modify this even more! Make it so that this effect doesn't start happening until the user clicks the health pack. So before the user clicks the health pack, it just sits there with one size. Then after clicked it starts to do size change effect on the sprite over and over. Hint: an extra variable and and an if statement will help.

Challenge 2 Solution

Solution Video: Lv05_L01_Ch2
Solution File: Lv06_L01_Ch2Done.gmk

Game Programming Course

Lv06_L02 – Sprite Editor and Sprite Sheets
(45 minutes)

Introduction

So you think you can draw? We'll open up the sprite editor built into GameMaker and make our own animated sprites. But if you are like me, you would rather just download some free graphics from the Internet. The sprite editor will also let you create animated sprites from sprite sheets.

Part A – Very Basics of the Sprite Editor

Watch the video: Lv06_L02A
Donefile: Lv06_L02ADone.gmk

Challenge 1 – Draw Something!

Take a few minutes and draw your own sprite. You don't have to make it a masterpiece. Maybe a ghost from Pacman, a space invader, or a really good stick figure trying to run. Test it out in your game. No video solution to this one.

Part B – Using Sprite Sheets

Watch the video: Lv06_L02B
Donefile: Lv06_L02BDone.gmk

Challenge 2 – Street Fighter

Start with the pre-made game project: Lv06_L02_Ch2StartHere.gmk
In the Level 06 Resource folder there is a nice sprite sheet for Ken, one of the main characters for the famous game Street Fighter. Use this sprite sheet to create the sprites for the following:

spr_fire, spr_standing, spr_punch, spr_kick, spr_crouch

Each animated sprite uses one row of the sprite sheet. The individual sprite squares are all 80 pixels X 80 pixels (easy sheet to work with).

After you have the 5 animated sprites, use the standing sprite and the kicking sprite and your own code to imitate the behaviour shown in the video below.

Watch the video: Lv06_L02_Ch2Ex

Challenge 2 Solution

Solution Video: Lv06_L02_Ch2
Solution File: Lv06_L02_Ch2Done.gmk

Game Programming Course

Lv06_L03 – Advanced Sprite Control
(60 minutes)

Introduction

Now that you know how to switch sprites around you have to learn how to plan and control the various sprite animations that you might assign to one object. In this lesson we will develop Ken, our Street Fighter so his behaviour is more like the real Ken in the arcade. It will be an excellent review of conditions and variables mixed with sprite control.

Part A – image_index

When animated sprites are playing, a variable called *image_index* is keeping track of which frame the animation is on. If the current image index is 4, and you switch to another sprite, you will switch sprites and still be on frame 4. This can have weird effects. Sometimes you will want to set the image index back to 0 when you switch sprites that have some sort of animation link to each other.

Watch the video: Lv06_L03A

Part B – Controlling Actions / Sprites

We are going to build on our street fighter so that we can control the way he fires, punches, kicks, crouches, and stands. It will require a combination of variables, well thought out conditions, and sprite control.

Watch the video: Lv06_L03_B
Done-File: Lv06_L03_BDone.gmk

Challenge 01 – Punch and Crouch

Start with the pre-made game project: Lv06_L03_Ch1StartHere.gmk
So far you have Ken standing, kicking, and firing. Time for you to add the punch and crouch. Notice that crouching is a little different then the other actions. The player doesn't press a key once to crouch, they hold down a key to crouch (so use keyboard event, not key press).

Solution

Watch the video: Lv06_L03_Ch1
Done-File: Lv06_L03_Ch1Done.gmk
Code-File:

Challenge 02 – All on Your Own

It might have been easy to add in punching and crouching when the kicking and firing are already in the file. Delete all your code and try to get all 5 moves working from a fresh file without following the videos!

Challenge 03 – Getting Hit
Open the pre-made file Lv06_L03_Ch3StartHere.gmk.
Watch the demo Video: Lv06_L03_Ch3Ex

Now that we have Ken making his moves properly, lets wrap up this project file with Ken getting hit by fire objects that are fired at him.
Notice the object called *globs* that creates a global hit point variable for Ken.
The globs object is also drawing the hit points to the screen.
There is also a tower object that is firing fire at Ken.

You add the following,
If Ken is crouched, he won't take damage. If Ken is firing he will take double damage. Any other position is normal damage (5 points).
Add an explosion object when Ken gets hit. The explosion sprite sheet is in the Level 06 resource folder. The explosion object should destroy itself when it's animation is over.

Game Programming Course

Lv06_L04 – Drawing Sprites in the Draw Event
(20 minutes)

Introduction

So far we have drawn the value of variables to the screen using some sort of draw object that we add to the room. We have also changed an object's sprite using the *sprite_index* variable. In this lesson we will explore another way of drawing variables and sprites out.

Part A – More About the Draw Event

We have left the draw event blank for most of our objects. When you leave the draw event blank the program will draw the object's sprite for you automatically. Sometimes, however, it is good to use an objects draw event to draw variables and sprites out. Check out these examples.

Watch the video: Lv06_L04_A
Watch the video: Lv06_L04_B
Done-File: Lv06_L04_ADone.gmk
Done-File: Lv06_L04_BDone.gmk

Challenge 01 – Draw Potion Life

Try changing the way that the health potion is drawn out.
First, draw the amount of the potion near the potion. Use draw event of potion.
Second, make the potion look bigger when it has an amount of 100 and get smaller as the amount of potion decreases. Try using the image_xscale and image_yscale variables to do this (ex. image_xscale = amount/100)

Part B – Draw Sprite Extended

Sometimes you need your sprite to be drawn using some extra properties like image_angle, alpha transparency, and scaling. For this you have to use the draw_sprite_ext method. Check out the video

Watch the video: Lv06_L04_C
Done_File: Lv06_L04_CDone.gmk

Challege 02 – Fading Lasers

Start with the pre-made game project: Lv06_L04_Ch2StartHere.gmk
You'll notice that the boss and the laser fire at angles and everything works alright. Here's what we want to add: Give the lasers a variable called *strength* and set it to 1.0 when they are created. In the step event, make the strength variable decrease by 0.02 (yes, just a little). If the strength variable gets to zero or below, destroy the laser. Test this out.
Now add the draw event to the laser object. Inside of it, use the *draw_sprite_ext* method to draw the sprite out at the proper angle, scaling, etc. BUT when it comes time to fill in the *alpha* parameter, use the *strength* variable.

Since the strength variable is constantly decreasing, the drawn sprite's alpha will decrease (fade out).

Solution Video: Lv06_L04_Ch2
Solution File: Lv06_L04_Ch2Done.gmk

Game Programming Course

Lv07_L01 – Using Scripts/Methods
(30 minutes)

Introduction

This lesson is about using *scripts/methods*. Methods are named chunks of code that might perform a task, send you back some values, or both. You can use methods that have been written by other programmers and you can use methods that you have written. Methods can save you coding time, make your code easier to read, and help you organize your code. GameMaker has many methods that have already been written for you to use so lets take a look at some examples.

Part A – Introduction to Methods

Watch the video: 07-01-A

Challenge 1– Find, Use, and Test

Use the project 07-01-X1Start.

Add extra key press events into the player object and try out using some of the following methods. Read about the methods in the help file to find out more about them.

a) when the player presses 'R' use the *random* method to generate a number and show it with a message.
b) when the player presses 'M' use the motion_add method and see what happens.
c) when the player presses 'E' use the *position_empty* method and check if x=112, y=208 is empty or not (it has a hamburger so it shouldn't be!) . If it is empty, show a message saying empty.
d) when the player presses the 'D' key, use the *point_direction* method to find out the direction toward the upper left corner (x=0, y=0). Once you know this direction, fire an arrow toward this direction.
e) when the player presses the 'C' key, use the *point_in_circle* method to see if the player is within a 100 pixel radius circle that is positioned in the center of the room. The room is 800X600 so the center of the room is at (400,300). If the player is at the center of the room, make their points go up by 1000.

Challenge 1 Solution

Solution Video: 07-01-X1
Solution File: 07-01-X1Done.gmk

Game Programming Course

Lv07_L02 – ID's as Return Values
(30 minutes)

Introduction

In the last lesson you learned that some methods will return values back to you. Sometimes those values can be the ID's of other game objects. When you know the ID of a game object you have the power to control it or interact with it. Confusing? Just watch the video!

Part A – ID's as Return Values

Watch the video: 07-02-A

Challenge 1 – Stop Far Ghost

Use: 07-02-X1Start

a) when the player presses F, find the *ghost* object that is furthest from the player using the instance_furthest method. Make this ghost stop moving.

Challenge 1 Solution

Solution Video: 07-02-X1
Solution File: 07-02-X1Done.gmk

Part B – More About Using ID Values

Watch the video: 07-02-B

Challenge 2 – Run Away Walls

Use: 07-02-X2Start

a) the Step event occurs 30 times a second by default. Add this code into the Step Event of the player. Find the id of the nearest wall. Then find out how far away this wall is using the point_distance method. If the wall is within 100 pixels of the player, make the wall move away from the player at a speed of 2 (hint: away from the player is the direction from the player to the wall object!).

Challenge 2 Solution

Solution Video: 07-02-X2
Solution File: 07-02-X2Done.gmk

Game Programming Course

Lv08_L01 – Writing Methods/Scripts
(30 minutes)

Introduction

In the last level you practised using methods/scripts that are already made for you. In this level you will learn how to write your own scripts. You will also start to understand why writing your own scripts can improve the readability, re-useability, and efficiency of your code.

Part A – Creating Simple Scripts

Watch the video: 08-01-A
DoneFile: Lv08_01_A.gmk

Challenge 1 – Make Some Scripts

Use: 08-01-A.gmk to start

Lets add three scripts to the current game project. These scripts basically take existing code and place it in a script.

a) The ghost object currently has code to fire two different objects, a laser and a rock. Create two scripts called *fireLaser* and *fireRock* that can be used in the program to simplify the existing ghost firing code in the step event.

b) The player currently gets points when one of their arrows hits a ghost. Create a script called *givePlayerPoints* that will replace the current point code.

Challenge 1 Solution

Solution Video: 08-01-X1
Solution File: 08-01-X1Done.gmk

Game Programming Course

Lv08_L02 – Scripts with Arguments
(30 minutes)

Introduction

In this lesson you will build upon your basic script writing by creating scripts that require *arguments (or sometimes called parameters).* Arguments and parameters are pieces of information that you give a script in order for it to complete its task. Designing a script that uses arguments usually gives the script more flexibility in the way that it can perform it's task. Adding arguments to your scripts can greatly reduce the amount of code you have to type with larger projects. Check out the videos to see how.

Part A – Scripts and Arguments

Watch the video: 08-02-A
DoneFile: Lv08_02_A.gmk

Challenge 1 – Make Some Scripts

Use: 08-012-A.gmk to start
Take the existing project and modify it to include these three scripts.

a) There are several objects that can give the player hp; apples, hamburger, and potions. Take a look at the existing code when a player collides with these three objects. Replace the increasing hp code with a script called *givePlayerHealth* that requires one argument: the amount of hp to give the player. The script should give the player the specified amount of hp and also check to make sure that the player doesn't go over 100 hp. Then replace the code in the apple, hamburger, and potion with your script.

b) The player currently gets 1 point for hitting a ghost with an arrow. Write a script that allows you to give the player any amount of points. The script will require one parameter which represents the amount of points to give the player. The script will check to see if the player has reached 10 points, and if they have go to the game over room. Once you have this script written, use it to give the player 1 point when they hit a ghost and 5 points when they destroy a ghost. Test it out!

c) Write a script called *dropRandomItem* that requires no parameters, and creates either an apple, burger, or potion when a ghost is destroyed (after being hit by a player arrow). Add this script in the appropriate location in the program.

Challenge 1 Solution

Solution Video: 08-02-X1
Solution File: 08-02-X1Done.gmk

Game Programming Course

Lv08_L03 – Scripts that Return Values
(30 minutes)

Introduction

In Level 07 we took a quick look at some GameMaker methods/scripts that return values back to you after you use them. In this lesson you will learn how to make your methods return values back to you. It will be an important concept to understand because in future lessons you will write scripts that do this when you are challenged with more complicated tasks in your games.

Part A – Returning Values

Watch the video: 08-03-A
DoneFile: Lv08_03_A.gmk

Challenge 1 – hpStatus and getBarColor

Watch the video: 08-03-X1ex that previews the challenge.
Use: 08-03-A.gmk to start

Here are two tasks to practice a bit of value returning.

a) Give the player a new variable called *healthStatus*. *healthStatus* is either "healthy" (50-100 hp), "warning" (25-49 hp), or "danger!" (below 25 hp). Use the draw object to draw this variable above the health bar.

Now write a script called *hpStatus* that returns the appropriate word based on the player's hp. This script requires no arguments.

Each time you set, give, or take away hp from the player (in the givePlayerPoints and damagePlayer scripts) you should call the script so that it determines the appropriate value for *hpStatus.*

Hint: you'll use a line like healthStatus=hpStatus() somewhere in there!

b) The health bar is currently always drawn in white. Write a script called *getBarColor* that requires one argument; the player's hp. It will use the argument and return a color so you can draw the health bar in an appropriate color (green if 50 or above, orange if 25-49, and red if below 25).

To return a color in your script you can use a line like this: return(c_red)

Once the script is made, can you figure out how to use it in the draw object to easily set the color of the health bar rectangle when it is drawn? If not just peek at the solution video. This is a nice example of how scripts can be used.

Challenge 1 Solution

Solution Video: 08-03-X1
Solution File: 08-03-X1Done.gmk

Level 09