



# GameMaker Fundamentals

Learn GML Programming to  
Start Making Amazing Games

—  
Ben Tyers

Apress®

# GameMaker Fundamentals

Learn GML Programming to Start  
Making Amazing Games

Ben Tyers

Apress®

# **GameMaker Fundamentals: Learn GML Programming to Start Making Amazing Games**

Ben Tyers  
Worthing, West Sussex, UK

ISBN-13 (pbk): 978-1-4842-8712-5  
<https://doi.org/10.1007/978-1-4842-8713-2>

ISBN-13 (electronic): 978-1-4842-8713-2

Copyright © 2023 by Ben Tyers

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr  
Acquisitions Editor: Spandana Chatterjee  
Development Editor: Spandana Chatterjee  
Coordinating Editor: Mark Powers

Cover designed by eStudioCalamar

Cover image by Freepik ([www.freepik.com](http://www.freepik.com))

Distributed to the book trade worldwide by Apress Media, LLC, 1 New York Plaza, New York, NY 10004, U.S.A. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [booktranslations@springernature.com](mailto:booktranslations@springernature.com); for reprint, paperback, or audio rights, please e-mail [bookpermissions@springernature.com](mailto:bookpermissions@springernature.com).

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub (<https://github.com/Apress>). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

# Table of Contents

<b>About the Author .....</b>	<b>xv</b>
<b>About the Technical Reviewer .....</b>	<b>xvii</b>
<b>Acknowledgments .....</b>	<b>xix</b>
<b>Introduction .....</b>	<b>xxiii</b>
<b>Welcome .....</b>	<b>xxv</b>
<b>Chapter 1: Variables .....</b>	<b>1</b>
Basic Assignments.....	14
Advance Assignments.....	15
Useful Functions .....	15
Summary.....	16
<b>Chapter 2: Conditionals .....</b>	<b>17</b>
Basic Assignments.....	21
Advance Assignment.....	22
Useful Functions .....	22
Summary.....	22
<b>Chapter 3: Drawing Shapes .....</b>	<b>23</b>
Basic Projects .....	26
Advance Project.....	26
Useful Functions .....	26
Summary.....	27

## TABLE OF CONTENTS

<b>Chapter 4: More Drawing.....</b>	<b>29</b>
Basic Projects .....	41
Advance Projects .....	41
Useful Functions .....	42
Summary.....	42
<b>Chapter 5: Input and Movement .....</b>	<b>43</b>
Basic Projects .....	45
Advance Project .....	45
Useful Functions.....	45
Summary.....	46
<b>Chapter 6: Objects and Events.....</b>	<b>47</b>
Create Event.....	48
Mouse Events.....	48
Quick Summary of the Mouse Events.....	50
Destroy Event.....	51
Alarm Event.....	51
Draw Event.....	52
Step Event.....	53
Key Events .....	53
Collision Event.....	54
Draw GUI Event .....	56
Basic Projects .....	57
Advance Project .....	57
Useful Functions .....	58
Summary.....	58
<b>Chapter 7: Sprites.....</b>	<b>59</b>
Basic Projects .....	65
Advance Project .....	65
Useful Functions .....	65
Summary.....	66

## TABLE OF CONTENTS

<b>Chapter 8: Health, Lives, and Score .....</b>	<b>67</b>
Health.....	68
Lives.....	68
Score.....	69
Basic Projects .....	70
Advance Projects .....	70
Useful Functions .....	70
Summary.....	70
<b>Chapter 9: Mouse.....</b>	<b>71</b>
Cursors.....	73
Basic Projects .....	75
Advance Projects .....	76
Useful Functions .....	76
Summary.....	76
<b>Chapter 10: Alarms.....</b>	<b>77</b>
Delaying an Action .....	78
Basic Projects .....	79
Advance Projects .....	79
Useful Functions .....	80
Summary.....	80
<b>Chapter 11: Collisions.....</b>	<b>81</b>
Collisions.....	82
Collision Line.....	84
Basic Projects .....	86
Advance Projects .....	87
Useful Functions .....	87
Summary.....	87

## TABLE OF CONTENTS

<b>Chapter 12: Rooms .....</b>	<b>89</b>
Room Editor .....	89
Basic Projects .....	93
Advance Project .....	93
Useful Functions .....	93
Summary.....	94
<b>Chapter 13: Backgrounds .....</b>	<b>95</b>
Setting Up a Background .....	95
Parallax Effect.....	97
Basic Projects .....	101
Advanced Project .....	101
Useful Functions .....	101
Summary.....	102
<b>Chapter 14: Sounds .....</b>	<b>103</b>
Sound Effects.....	103
Music .....	105
Pausing and Resuming Audio .....	106
Basic Projects .....	107
Advanced Projects .....	107
Useful Functions .....	108
Summary.....	108
<b>Chapter 15: Splash Screens and Menu.....</b>	<b>109</b>
Splash Screen Example .....	109
Basic Projects .....	110
Advanced Project .....	110
Useful Functions .....	111
Summary.....	111

## TABLE OF CONTENTS

<b>Chapter 16: Random .....</b>	<b>113</b>
Random Numbers .....	114
Other Random Uses .....	115
Basic Projects .....	116
Advanced Project .....	116
Useful Functions .....	117
Summary.....	117
<b>Chapter 17: AI .....</b>	<b>119</b>
Basic Enemy AI .....	121
Enemy Shooting .....	122
Basic Projects .....	124
Advanced Projects .....	125
Useful Functions .....	125
Summary.....	125
<b>Chapter 18: INI Files .....</b>	<b>127</b>
INI File Contents.....	127
Loading Data.....	128
Saving Data.....	128
Basic Projects .....	129
Advanced Project .....	129
Useful Functions .....	129
Summary.....	130
<b>Chapter 19: Effects .....</b>	<b>131</b>
Effects .....	131
Effect Layer.....	134
Basic Projects .....	137
Advance Projects .....	138
Useful Functions .....	138
Summary.....	139

## TABLE OF CONTENTS

<b>Chapter 20: Loops.....</b>	<b>141</b>
Repeat Loop .....	142
While Loop .....	142
For Loop .....	142
Do Loop.....	143
Basic Projects .....	144
Advance Projects .....	144
Useful Functions .....	144
Summary.....	145
<b>Chapter 21: Arrays.....</b>	<b>147</b>
Array Example.....	147
Drawing Array Contents .....	149
Array-Based Shop.....	151
Basic Projects .....	155
Advanced Projects .....	156
Useful Functions .....	156
Summary.....	156
<b>Chapter 22: DS Lists .....</b>	<b>157</b>
Creating a DS List .....	157
Sorting an Array .....	158
Removing a Value.....	159
Adding a Value .....	159
Basic Projects .....	161
Advanced Project .....	161
Useful Functions .....	161
Summary.....	162

## TABLE OF CONTENTS

<b>Chapter 23: Paths .....</b>	<b>163</b>
Path Example .....	163
Creating a Path in Code .....	164
Drawing a Path.....	166
Basic Projects .....	169
Advanced Projects .....	169
Useful Functions .....	169
Summary.....	170
<b>Chapter 24: Functions .....</b>	<b>171</b>
Creating a Function.....	172
Calling a Function .....	173
Basic Projects .....	174
Advanced Projects .....	175
Useful Functions .....	175
Summary.....	177
<b>Chapter 25: Tilesets.....</b>	<b>179</b>
Setting a Tileset .....	179
Placing Tiles.....	181
Basic Project.....	182
Advanced Projects .....	183
Useful Functions .....	183
Summary.....	183
<b>Chapter 26: Timelines.....</b>	<b>185</b>
Creating a Timeline .....	185
Make a Timeline Play .....	189
Basic Projects .....	191
Advanced Projects .....	191
Useful Functions .....	191
Summary.....	191

## TABLE OF CONTENTS

<b>Chapter 27: Views.....</b>	<b>193</b>
Setting a View .....	194
Advance Projects .....	198
Useful Functions .....	198
Summary.....	198
<b>Chapter 28: MP Grids.....</b>	<b>199</b>
Setting a Grid .....	199
Adding Objects to a Grid .....	199
Basic Project.....	201
Advance Project .....	202
Useful Functions .....	203
Summary.....	204
<b>Chapter 29: Sequences.....</b>	<b>205</b>
Creating a Sequence.....	206
Setting a Scale Curve.....	212
Basic Project.....	216
Advance Project .....	216
Useful Functions .....	217
Summary.....	217
<b>Appendix A: Game – Plane Game.....</b>	<b>219</b>
Plan Out Main Game Features with a Pen and Paper .....	221
Source-Suitable Graphics and Audio .....	222
Set Up Splash Screen Room .....	223
Set Up Menu Room .....	227
Set Up Game Information Room.....	233
Set Up Gameover Room .....	235
Set Up First Level.....	238
Create HUD.....	245
Create Main Player.....	249

## TABLE OF CONTENTS

Create Player Weapons .....	251
Create Bonuses .....	258
Create Enemies .....	261
Create Enemy Spawning System .....	273
Collisions, Effects, and Audio .....	275
<b>Appendix B: Programming Challenges .....</b>	<b>293</b>
CHALLENGE 1 – Slowly Increase Score.....	294
CHALLENGE 2 – Mouse over Menu Buttons .....	295
CHALLENGE 3 – Planes on Menu .....	296
CHALLENGE 4 – Weapon Trail Effects.....	297
CHALLENGE 5 – Random Game Play .....	298
CHALLENGE 6 – Bonus Bird .....	299
CHALLENGE 7 – Voice on New Highscore .....	300
CHALLENGE 8 – Weather Effects.....	301
CHALLENGE 9 – Player Auto Move .....	302
CHALLENGE 10 – Music Control.....	303
CHALLENGE 11 – End of Level Boss.....	304
CHALLENGE 12 – Bonus Round.....	305
CHALLENGE 13 – Draw Score As Text .....	306
CHALLENGE 14 – New Enemy .....	307
CHALLENGE 15 – Sliding Menu Buttons.....	308
CHALLENGE 16 – Enemy Crash in Water.....	309
CHALLENGE 17 – Shield Bonus .....	310
CHALLENGE 18 – Room Fade In/Out Effect.....	311
CHALLENGE 19 – Laser Weapon .....	312
CHALLENGE 20 – Radar Showing Enemies .....	313
CHALLENGE 21 – Floating Score Text.....	314
CHALLENGE 22 – Knockback When Shooting .....	315
CHALLENGE 23 – Player Flash When Hit .....	316
CHALLENGE 24 – Read Score As Voice .....	317

## TABLE OF CONTENTS

CHALLENGE 25 – Mouse Cursor Change.....	318
CHALLENGE 26 – Shooting Cool Down.....	319
CHALLENGE 27 – Basic Weapon Upgrade .....	320
CHALLENGE 28 – Mega Weapon .....	321
CHALLENGE 29 – Hoop to Fly Through.....	322
CHALLENGE 30 – Coins Bonuses .....	323
CHALLENGE 31 – Logo Change Color.....	324
CHALLENGE 32 – Friendly Hot Air Balloon.....	325
CHALLENGE 33 – Water-Based Enemy.....	326
CHALLENGE 34 – Dangerous Blimp .....	327
CHALLENGE 35 – Difficulty Settings.....	328
CHALLENGE 36 – Fast Move Bonus.....	329
CHALLENGE 37 – Exploding Mega Bomb .....	330
CHALLENGE 38 – Combo Kill Effect.....	331
CHALLENGE 39 – Slowly Change Menu Button Size .....	332
CHALLENGE 40 – Double Damage.....	333
CHALLENGE 41 – Wobbly Logo.....	334
CHALLENGE 42 – Plane Player Crash.....	335
CHALLENGE 43 – Better Collision Detection .....	336
CHALLENGE 44 – Text with Border .....	337
CHALLENGE 45 – How to Play Information.....	338
CHALLENGE 46 – Shop System.....	339
CHALLENGE 47 – On-Screen Keyboard.....	340
CHALLENGE 48 – Name with Highscore.....	341
CHALLENGE 49 – Shoot Weapon to Target .....	342
CHALLENGE 50 – Create Lots of Effects.....	343
CHALLENGE 51 – Increasing Difficulty .....	344
CHALLENGE 52 – Quit Game Confirmation.....	345
CHALLENGE 53 – Fuel Control System.....	346
CHALLENGE 54 – Heat Seeking Weapon .....	347

## TABLE OF CONTENTS

CHALLENGE 55 – Character Selection.....	348
CHALLENGE 56 – Player Achievements.....	349
CHALLENGE 57 – Engine Sounds .....	350
CHALLENGE 58 – Hit Enemy Damage .....	351
CHALLENGE 59 – Mega Super Weapon .....	352
CHALLENGE 60 – Mini Helper Plane.....	353
CHALLENGE 61 – Moving Text.....	354
CHALLENGE 62 – Enemy Incoming Alert .....	355
CHALLENGE 63 – Menu Plane Effect.....	356
CHALLENGE 64 – Heat Distortion Effect.....	357
CHALLENGE 65 – Screen Shake.....	358
<b>Appendix C: Purchasing.....</b>	<b>359</b>
Logging in or Creating a New Account.....	359
Subscription Options.....	360
Free .....	360
Creator.....	361
Indie.....	361
Enterprise .....	361
Summary .....	361
Installing .....	363
Downloading.....	363
Starting a New Project.....	363
The IDE.....	364
Menus.....	364
Assets.....	367
Sprites.....	368
Importing a Single Image .....	368
Importing a Sprite Sheet .....	370
Importing a Sprite Strip .....	373
Resizing Sprites.....	374
Sprite Origin .....	376

## TABLE OF CONTENTS

Fonts .....	377
Creating a Font .....	377
Objects .....	378
Assigning a Sprite .....	379
Events.....	381
GML Code .....	384
Sounds .....	385
Importing .....	385
Sound Settings .....	386
Rooms .....	387
Adding Instance.....	387
Background .....	389
Layers.....	391
Rooms Order.....	392
Paths .....	394
Functions .....	394
Sequences .....	395
Shaders.....	395
Tilesets.....	395
Timelines.....	395
Target.....	395
Naming Conventions .....	396
<b>Index.....</b>	<b>399</b>

# About the Author

**Ben Tyers** is an expert GameMaker user, developer, coder, and trainer, with over 10 years of experience in GameMaker. He has authored several books on GameMaker for game application developers. He runs a popular gaming website, which features accessible games that can be played by gamers with physical disabilities, with around one new game added every month. He has learnt GML coding to a reasonable level and has picked up many skills, tips and tricks, and methodology for making games in GameMaker.

# About the Technical Reviewer



**Mark Alexander** has been a technical writer for the last 12 years, starting his career working for YoYo Games in Scotland, where he was in charge of the GameMaker documentation for almost ten years. After that, he moved on to work at Asobo Studio in France, where he is doing the technical documentation for the Microsoft Flight Simulator SDK.

In his spare time – apart from reading sci-fi, collecting Lego, and playing video games – he creates and publishes small indie games and also creates assets for people to use in GameMaker. Any time he has left after that, he dedicates to the administration of the GameMaker Community Forum.

# Acknowledgments

Various Audio and Graphics Used in Game and Challenges: Permission Given to Use by Yoyogames.com. Please Do Not Redistribute These Assets.

Graphics:

Aeroplane: Pzuj/Opengameart.org - CC0

Animated Boy Sprites: Bevouliin.com/Opengameart.org - CC0

Arrow: oglSDL/Opengameart.org - CC0

Asteroids: Jasper/Opengameart.org - CC0

Bag: Flowly/Opengameart.org - CC0

Ball: Gothicfan95/Freepik.com

Barrel: Manenwolf/Opengameart.org - CC0

Beach Scene: Tgfcoder/Opengameart.org - CC0

Bee: Carriejanesart/Opengameart.org - CC0

Birds: Bevouliin.com/Opengameart.org - CC0 && CC-BY 4.0

Blue Character: Kenny/Opengameart.org - CC0

Boats: Kutejnikov/Opengameart.org - CC0

Bomb Sprite: Znevs/Opengameart.org - CC0

Boss Missile: Bevouliin.com/Opengameart.org - CC0

Boss Plane: Bevouliin.com/Opengameart.org - CC0

Bullets Set: Bevouliin.com/Opengameart.org - CC-BY 3.0

Buttons: Game Developer Studio/Opengameart.org - CC-BY 3.0

Crosshair: hc/Opengameart.org - CC0

Caveman: Hapiel/Opengameart.org - CC0

Cloud: Bananaowl/Opengameart.org - CC-BY 3.0

Clouds: Bevouliin.com/Opengameart.org - CC0

Crate: Rfc1394/Opengameart.org - CC0

Crocodile Character: Bevouliin.com/Opengameart.org - CC0

Dinosaur: Pzuj/Opengameart.org - CC0

Donuts: Game Developer Studio/Opengameart.org - CC-BY 3.0

Eggs: Ogrefewart/Opengameart.org - CC0

Elements: Game Developer Studio/Opengameart.org - CC-BY 3.0

## ACKNOWLEDGMENTS

Explosion: Stumpystrut/Opengameart.org - CC0  
Extra Character Sprites: Bevouliin.com/Opengameart.org - CC0  
Firenall: Cethiel/Opengameart.org - CC0  
Firework: Jellyfish/Opengameart.org - CC0  
Fuel Collectable: Freestockimages/Opengameart.org - CC0  
Fuel Gauge: JM.Atencia/Opengameart.org - CC-BY 3.0  
Fish: Sylly/Opengameart.org - CC0  
Forest: Ansimuz/Opengameart.org - CC0  
Gems: Wenrexa/Opengameart.org - CC0  
Grass Sky Backgrounds Images: Segel/Opengameart.org - CC0  
Green Horn Zombie Sprites: Bevouliin.com/Opengameart.org - CC0  
Green Zombie: Segel/Opengameart.org - CC0  
Guns & Bullets Images: Kay Lousberg/Opengameart.org - CC0  
Heart/Lives Sprite: Cdgramos/Opengameart.org - CC0  
Missile Enemy: Bevouliin.com/Opengameart.org - CC0  
Laser: Rawdanitsu/Opengameart.org - CC0  
Missile: JM.Atencia/Opengameart.org - CC-BY 3.0  
Mountain Background Images: Ansimuz/Opengameart.org - CC0  
Mountain Images: Ansimuz/Opengameart.org - CC0  
Mouse Icons: Qubodup/Opengameart.org - CC0  
Padlocks: Rawpixel.com/Freepik.com  
Plane Game Background: Bevouliin.com/Opengameart.org - CC0 && OGA-BY-3.0  
Rocks: Bevouliin.com/Opengameart.org - CC0  
Rotating Coin Sprites: Puddin/Opengameart.org - CC0  
Rpg Character: Curt/Opengameart.org - CC-BY 3.0  
Sea Background: Donte/Opengameart.org - CC0  
Shinny Coins: Bevouliin.com/Opengameart.org - CC0 && CC-BY 4.0  
Space Background: Satur9/Opengameart.org - CC0  
Space Parallax Images: Ansimuz/Opengameart.org - CC0  
Spaceship: Bevouliin.com/Opengameart.org - CC-BY 3.0 & CC0  
Spiky Land Monster: Bevouliin.com/Opengameart.org - CC0  
Star: Jellyfish/Opengameart.org - CC0  
Tileset: Buch/Opengameart.org - CC0  
Top Down Zombie Images: Riley Gombart/Opengameart.org - CC0  
Underwater: Game Developer Studio/Opengameart.org - CC-BY 3.0

## ACKNOWLEDGMENTS

Walking Green Monster Sprites: Bevouliin.com/Opengameart.org - CC0  
Wall: Beaterator/Opengameart.org - CC0

### Audio:

Beep: Thisusernameis/Freesound.org - CC0  
Bird Sound: JackWilDesign/Freesound.org - CC0  
Bell Sound: Se2001/Freesound.org - CC0  
Coins (long) dj997/Freesound.org - CC-BY 3.0  
Dance Music: Davejf/Freesound.org - CC0  
Explosion Sound: Derplayer/Freesound.org - CC0  
Firework: Rudmer\_rotteveel/Freesound.org - CC0  
Gun Sound: Mrrap4food/Freesound.org - CC0  
Heavy Beet Music: Jpmusic82/Freesound.org - CC0  
Machine Gun Sound: Pgi/Freesound.org - CC0  
Melody Music: Drfx/Freesound.org - CC0  
Menu Sound: Apenguin73/Freesound.org - CC0  
Nuke Sound: Tristanluigi/Freesound.org - CC0  
Ouch: Legnalegna55/Freesound.org - CC0  
Ping Sound: Wilhellboy/Freesound.org - CC0  
Plane Game Level Music: Rezoner/Opengameart.org - CC-BY 3.0  
Plane Game Menu Music: Cynicmusic/Opengameart.org - CC0  
Plane Game Sound Effects: Subspaceaudio/Opengameart.org - CC0  
Random Sound 1: Mosrod/Freesound.org - CC0  
Random Sound 2: Littlerainyseasons/Freesound.org - CC0  
Random Sound 3: Elfstonepress/Freesound.org - CC0  
Random Sound 4: Alanmcki/Freesound.org - CC0  
Random Sound 5: Jeremysykes/Freesound.org - CC0  
Rocket Sound: Derplayer/Freesound.org - CC0

# Introduction

A note from the author:

*Congratulations!*

*You are about to learn the basics of GameMaker and potentially start a career in game making.*

*This book is an introduction to the game-making process, an introduction to GameMaker, and other considerations when making your first game.*

*GameMaker is a powerful piece of software for making games. This book only covers the basics but is a great place to start. This book uses the updated 2022 edition.*

*Best of luck with your game-making endeavors.*

*Ben*

# Welcome

The purpose of this book is to provide you with some of the knowledge that I have acquired. I make no claim that I'm the best coder or designer, but I do have a proficient understanding that I would like to instill in other budding game makers.

The book is broken down into 29 main chapters and three appendixes: one that pulls together the knowledge you gained from the book to create a game, one offering programming challenges, and one covering how to get set up with GameMaker and its IDE.

The main chapters each introduce a separate programming concept and explain what it is used for and how it is applied, along with some examples. Each of these chapters then finishes with some assignments.

The game chapter details how to make a simple shooting game.

You are then presented with 65 programming challenges, each of which adds a new feature to the game you have just made.

## Source Code

All source code in this book can be downloaded from [github.com/apress/gamemaker-fundamentals](https://github.com/apress/gamemaker-fundamentals).

## CHAPTER 1

# Variables

When using GameMaker, there are two main types of variables that you will use often. They are strings and numbers (reals and integers). You can do many different things with variables. They can be used for performing calculations and drawing information on the screen. You will use variables for such things as

- Keeping track of score, health, and lives
- Drawing the values of score, health and lives
- Processing data and performing actions based on their values
- Using values in calculations
- Making an object move
- Displaying values, as text or graphically
- Saving the player's progress
- Setting the difficulty of a game or level
- Sending values for online highscores
- Setting initial positions of objects and how they move
- Checking for a weapon's availability or power
- Drawing text to explain what a player must do
- And many more

---

**Note** There are several variable types such as instance, local, and global. You will focus mainly on instance and global variables.

---

Some variables automatically have global scope, such as health, lives, and score.

---

**Note** It should be noted that these preceding three variables may be deprecated in the future, so you'll need to create your own variables with global scope if you wish them to be accessible game wide, for example, `global.my_score`.

---

Global variables have scope across the whole game and can be accessed, read, and changed from anywhere else within your game. For example:

```
global.target=100;
```

Generally you would set your global values at the very start of the game, for example, some code that is run on your game splash screen:

```
/// @description Set up global values
lives=6;
score=0;
health=100;
global.target=100;
global.enemies_to_kill=20;
room_goto(room_menu);
```

---

**Note** Failure to declare any global value prior to using it will result in an error and crash your game. As such, these are generally set at the start of the game, usually in splash screen room; this is explained later in this book.

---

If you want to make another variable global in scope, you can put `global` in front of the variable, for example, `global.level`.

Instance variables, for example, `x` and `y`, and `hp`, are generally accessible for the object that set them. (Actually, you can access and change instance variables from other objects, but we won't go into that just now.)

Local variables have the scope of the code block that is using them, for example:

```
var size=5;
```

A basic example usage would be the following:

```
for (var loop = 0; loop< 20; loop++)
{
    draw_text(20+(loop*25),30,loop);
}
```

which would draw the numbers 0 through 19.

An advanced example may look something like this:

```
/// @description Shooting bullet control
if can_shoot_bullet
{
    var bullet=instance_create_layer( x,y,"bullets",obj_player_bullet);
    //spawn the bullet
    bullet.speed=4;//set the speed of bullet
    bullet.direction=image_angle;//match direction to that of the plane
    bullet.image_angle=image_angle;//make the bullet point to direction of
                                    movement
    can_shoot_bullet=false;
    alarm[0]=game_get_speed(gamespeed_fps);
    global.shots_fired++; //update shot count
    audio_play_sound(snd_player_bullet,1,false);
}
```

This would mean that the game would not remember the bullet variable once the code block is completed.

Local variables will mainly be used for data and structures that are no longer required once some data has been processed, and for variables that are outside the scope of the code block. These types of variables help prevent memory leaks.

A memory leak is, for example, when your program keeps creating new items (i.e., variables, or instances) without properly removing them, gradually eating up available memory space – which will crash your game and most likely crash the device that is running it.

## CHAPTER 1 VARIABLES

The basic code for drawing text is

```
draw_text(x_position,y_position,text);
```

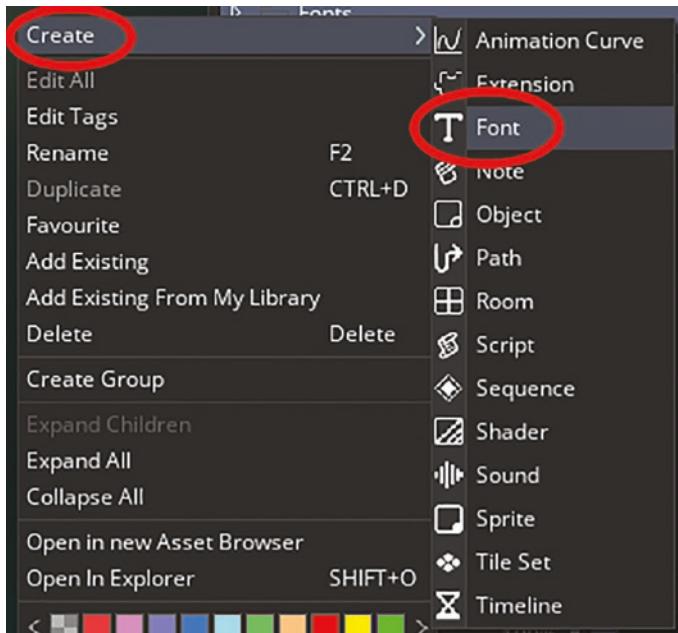
It's good practice to remember to set the font and formatting prior to drawing any text.

---

**Note** Text formatting (font, colors, alignment) will be continued for any future drawing of text until you tell the program otherwise, so get into the habit of setting it before drawing anything.

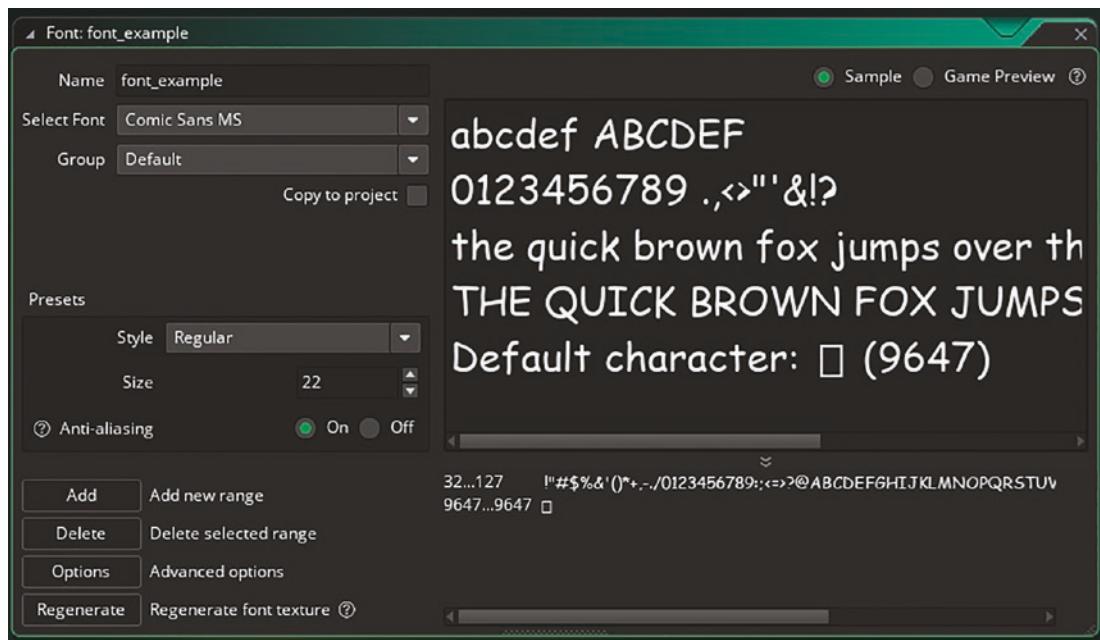
---

You can create a font, as shown in Figure 1-1.



**Figure 1-1.** Creating a font asset

First, let's set up a font to use for this chapter, as shown in Figure 1-2.

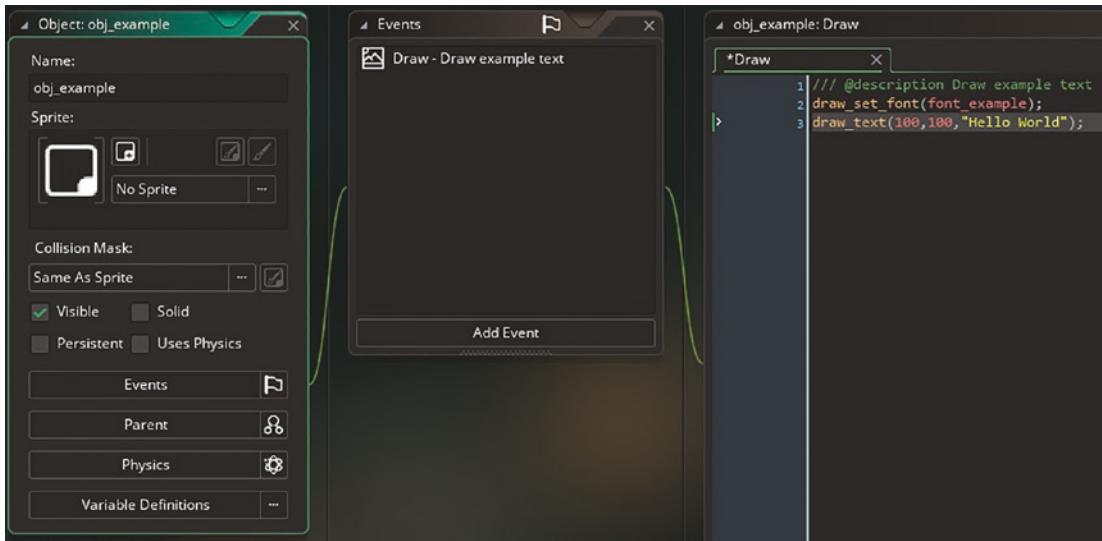


**Figure 1-2.** Setting a font to be used for this chapter

A real working example would be: To draw text “Hello World” at position 100x100. Create an object, **obj\_example**, with the following code placed in the **Draw Event**, as shown in Figure 1-3.

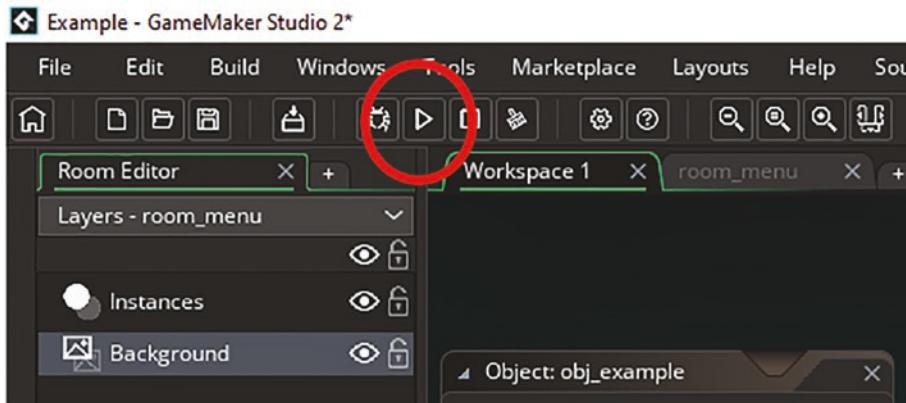
```
/// @description Draw example text
draw_set_font(font_example);
draw_text(100,100,"Hello World");
```

## CHAPTER 1 VARIABLES



**Figure 1-3.** Draw event code

Drag an instance of this object into the room onto the **Instances** layer and click the run button, as shown in Figure 1-4.



**Figure 1-4.** Showing where to click

You'll then see the text drawn onscreen, like that in Figure 1-5.



**Figure 1-5.** Showing some example text drawn

To draw a variable with a number (an integer), for example, add a **Create Event** and put the following code:

```
/// @description Set up a value
value=20;
```

And in the **Draw Event**, put the following code:

```
/// @description Draw Value
draw_set_font(font_example);
draw_text(200,200,value);
```

Test the game and check the expected value is drawn.

Next, we'll combine some variables, so change the **Create Event** to

```
/// @description Set up values
text="My Name Is Ben";
age=36;
```

Set **Draw Event** to combine these values, noting that you change the value of age to a string so they can be combined without error:

```
/// @description Draw example text
draw_set_font(font_example);
draw_text(100,100,text+" My age is "+string(age));
```

## CHAPTER 1 VARIABLES

If you are drawing just text or a numerical variable as a separate statement, then the preceding is not required, so, for example, the following would work fine:

```
line="Hello World";
draw_text(30,30,line);
value=4500;
draw_text(30,60,value);
```

When run, you'll see the following on your screen, as shown in Figure 1-6.



**Figure 1-6.** Drawing text and variables combined

Numbers can be an integer, such as 5, or a real which includes decimals, such as 11.8. Double-click on **obj\_example** in the resource tree. Change the **Create Event** code to

```
my_age=43;
```

You can of course add strings together (called concatenation), for example:

```
first_name="Ben";
last_name="Tyers";
my_name=first_name+" "+last_name;
```

You can do mathematical operations on numbers and then draw the result, for example, the following where if you were to draw the value of **total\_cost** on the screen it would show as 45:

```
cakes=9;
cost=5;
total_cost=cakes*cost;
```