



# Game Development with GameMaker

A Primer on Game Development  
and Design

—

*Second Edition*

—

Seb Cossu

Apress®

# **Game Development with GameMaker**

**A Primer on Game  
Development and Design**

**Second Edition**

**Sebastiano M. Cossu**

**Apress®**

# ***Game Development with GameMaker: A Primer on Game Development and Design, Second Edition***

Sebastiano M. Cossu  
LONDON, UK

ISBN-13 (pbk): 979-8-8688-0878-4  
<https://doi.org/10.1007/979-8-8688-0879-1>

ISBN-13 (electronic): 979-8-8688-0879-1

Copyright © 2024 by Sebastiano M. Cossu

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: James Markham  
Coordinating Editor: Kripa Joseph  
Technical review: Alynne Keith

Cover designed by eStudioCalamar

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 <https://www.apress.com/gp/services/source-code>.

If disposing of this product, please recycle the paper

*To the dreamers and hard workers.*

# Table of Contents

<b>About the Author .....</b>	<b>xv</b>
<b>Acknowledgments .....</b>	<b>xvii</b>
<b>Introduction .....</b>	<b>xix</b>
<b>Chapter 1: Overview .....</b>	<b>1</b>
The Right Tool for the Job .....	2
What Is GameMaker? .....	3
About Game Design.....	4
About Coding.....	5
How to Use This Book .....	5
Additional Content.....	9
Pricing.....	9
Installing GameMaker .....	10
Windows.....	11
Mac.....	14
Installing from Steam .....	15
<b>Chapter 2: Hello, World!.....</b>	<b>17</b>
UI Overview .....	19
Sprites.....	23
Objects .....	25
Events.....	27
Code .....	29

## TABLE OF CONTENTS

Tile Sets .....	30
Fonts .....	30
Rooms .....	31
Hello, GML! .....	32
Create Event .....	33
Left Pressed (Mouse) Event .....	38
Draw .....	39
<b>Chapter 3: Card Game (Part 1) .....</b>	<b>45</b>
The Design .....	45
A Game Design Document Primer .....	46
Memory GDD .....	47
Assets .....	50
From GDD to Development .....	52
Cards .....	52
Implementation .....	54
Deck .....	62
Array .....	63
Stack .....	65
Queue .....	66
List .....	68
Map .....	68
Priority Queue .....	69
Grids .....	70
Designing Decks .....	71
Code Loops .....	73
Making Decks .....	76
Every Day I'm Shuffling .....	79

<b>Chapter 4: Card Game (Part 2)</b> .....	<b>87</b>
Finite-State Machines (FSMs).....	87
From State Machine to Code.....	93
A Matter of Time.....	102
Play to Win!.....	109
<b>Chapter 5: Fixed Shooter</b> .....	<b>117</b>
History of the Genre.....	117
Space Gala (GDD).....	119
Story and Setting.....	120
Gameplay.....	120
Victory Conditions.....	120
Controls .....	120
Menu.....	121
Pacing.....	121
Enemies.....	121
Game Modes.....	122
Level 1 .....	122
Similar Games and Influences.....	122
Target Audience.....	122
From GDD to the Game .....	123
Assets.....	123
Making Features, Not Objects .....	130
Shooting .....	138
Designing rm_level_1 .....	149
Game States .....	150
Making HUDs.....	159
What About Victory? .....	162
Menu.....	163

TABLE OF CONTENTS

**Chapter 6: Shoot ‘Em Up!..... 173**

- Fixed vs. Scrolling Shoot ‘Em Up! ..... 174
- Space Gala v.2.0 (GDD)..... 175
  - Story and Setting..... 175
  - Gameplay..... 175
  - Victory Conditions..... 176
  - Controls ..... 177
  - Menu..... 177
  - Pacing..... 177
  - Enemies..... 178
  - Game Modes..... 179
    - Level 1 ..... 179
    - Level 2 ..... 179
  - Similar Games and Influences..... 179
  - Target Audience ..... 180
  - Assets..... 180
  - Cameras and Viewports..... 186
- Designing Color-Switching..... 197
- Inheritance ..... 201
- Color Shooting..... 206
- More Enemies ..... 209
  - Ain’t Nothing but the Blues..... 210
  - Walkers on Paths..... 212
  - Unidentified Flying... Instance!..... 216
- Super-Attack ..... 219
- How to Design a Good Shmup Level ..... 225
- Boss Fighting ..... 226
- Conclusion ..... 231

<b>Chapter 7: Designing Bosses .....</b>	<b>235</b>
Teaching and Experimenting.....	237
Motivation! .....	238
How Can We Use This?.....	240
<b>Chapter 8: Single-Screen Platformer .....</b>	<b>243</b>
Cherry Caves.....	246
Story and Setting.....	246
Gameplay.....	247
Victory Condition .....	247
Controls .....	248
Enemies.....	248
Assets .....	249
Sprites .....	249
Fonts.....	253
Sounds.....	253
How to Create a Hero .....	253
Setting the Boundaries .....	257
Everything That Goes Up Comes Down .....	262
Get a Jump on Gravity!.....	265
Climbing the Ladder.....	266
Controlling the Game Flow.....	272
HUD.....	282
How to Die.....	285
Cherry-Picking .....	288
Through Cherries, to the Star .....	289

TABLE OF CONTENTS

Level Design: The Art of Creating Worlds .....	292
Designing Caves .....	293
Level 2! .....	294
<b>Chapter 9: Scrolling Platformer .....</b>	<b>301</b>
Story and Setting .....	302
Gameplay .....	302
Victory Condition .....	303
Items .....	303
Controls .....	304
Enemies .....	305
Attack .....	305
Miscellaneous .....	306
Similar Games .....	307
Assets .....	307
spr_land .....	307
spr_skybg .....	308
spr_platform_falling .....	309
spr_platform_trampoline .....	309
spr_platform_moving .....	309
spr_octopus_green .....	310
spr_octopus_purple .....	310
spr_titlescreen .....	311
spr_coin .....	311
spr_terrain .....	312
Fonts .....	312
fnt_title .....	312
Sounds .....	312

The More You Do It ...	313
Title Screen .....	314
Tiles and Level Design .....	323
Scrolling Camera.....	333
Fixing and Re-adapting .....	337
Different Ways to Move .....	345
Gotta Squash 'Em All!.....	354
Items and Power-Ups.....	361
Coins.....	361
Cherries .....	363
Creating the First Level .....	366
<b>Chapter 10: Designing Platformers .....</b>	<b>375</b>
Controls Are Key.....	375
It's My Fault! .....	377
Keep It Simple!.....	379
Power-Ups, Items, and Gear.....	381
Interesting Collections .....	381
World Makers .....	384
Conclusion .....	388
<b>Chapter 11: Metroidvania (Part 1) .....</b>	<b>389</b>
History.....	390
Isolation (Game Design Document).....	391
Story and Setting.....	392
Gameplay.....	392
Victory Condition .....	392
Controls .....	392

## TABLE OF CONTENTS

Enemies.....	394
Attack .....	394
Skills.....	394
Maps.....	395
Inventory.....	395
Similar Games .....	396
Assets .....	396
spr_player_idle.....	396
spr_player_walk.....	397
spr_player_jump .....	397
spr_player_jump_fall/spr_player_dash .....	397
spr_player_wallslide .....	398
spr_heart.....	398
spr_warp .....	399
spr_marker.....	399
spr_upgrade .....	400
spr_cure .....	400
spr_octopus_green .....	400
spr_ground_brown.....	401
spr_checkpoint_inactive .....	401
spr_checkpoint_active .....	402
spr_bullet_heavy.....	402
spr_bullet_light .....	402
Fonts.....	403
Sounds.....	403
Creating the Platforming Base .....	404
Gamepad Support!.....	409

Gravity, No Escaping!.....	412
Making the Leap.....	414
Another Kick in the Wall.....	416
Moving Forward with a Dash.....	423
The Game Flow.....	430
Warped!.....	439
Conclusion.....	444
<b>Chapter 12: Metroidvania (Part 2).....</b>	<b>447</b>
About Maps.....	448
Map Makers, Grids, and Semaphores.....	449
Items and Inventory.....	469
Creating the Combat System.....	485
Old Enemies.....	490
Saving Maria.....	496
file_text_open_read(fname).....	498
file_text_open_write(fname).....	498
file_text_open_append(fname).....	498
file_text_write_string(file_id, my_string).....	499
file_text_close(file_id).....	499
Conclusion.....	508
<b>Chapter 13: Extra: Artificial Intelligence.....</b>	<b>511</b>
Isolation: CR – Artificial Intelligence.....	515
Change Request Document.....	515
Description of Change.....	516
Justification.....	517
Impact Analysis.....	518
Raycasting-Based Detection.....	519

TABLE OF CONTENTS

Octopus, Behave! .....523

Patrol.....525

Chase .....527

Conclusion .....530

**Chapter 14: Designing Fun Games.....533**

Document Your Design!.....533

Respect Your Game .....534

Keep Your Players Immersed .....535

    Autonomy .....536

    Competence .....537

    Relatedness.....538

Having Fun Means Learning .....539

Conclusion .....541

**Chapter 15: What’s Next? .....543**

Itch.io .....543

GOG.....547

Humble Store .....549

Steam.....550

End Game.....552

**Index.....553**

# About the Author



**Sebastiano M. Cossu** is a software engineer and game developer, currently employed in Electronic Arts. Video games have always been his greatest passion, and he began studying game development at an early age. Sebastiano started working with GameMaker in 2002 and has worked with every version of the software since then.

He is the author of *Game Development with GameMaker Studio 2* and *Beginning Game AI with Unity*, both published by Apress.

# Acknowledgments

I would like to express my deepest gratitude to my family, whose unwavering support has fueled my lifelong passion for video games. Your encouragement and belief in me have been the cornerstone of my journey.

To my closest friends, thank you for sharing this passion with me and for your continuous support. Our shared love for video games has been a constant source of inspiration and motivation.

To everyone who purchased the first edition of this book, thank you. I hope it was helpful in achieving your goals, and your feedback has been invaluable in shaping this second edition.

Lastly, to all the dreamers aspiring to be the game developers of tomorrow, this book is for you. May your journey be as rewarding and fulfilling as mine has been. Keep dreaming, keep creating, and never stop playing.

# Introduction

When I was a kid, the game industry was rapidly growing, becoming bigger and more advanced. There was little room for homemade games or one-person companies, like those that had initially expanded the industry. Creating games was turning into an elite activity, with most gamers drawn to big AAA productions, complex worlds, cutting-edge graphics, and intricate mechanics.

This was when I discovered GameMaker, a very accessible game engine that allowed anyone to create simple 2D games without a programming or software engineering background, thanks to its visual programming and easy-to-learn scripting language. GameMaker was surrounded by a prolific and inclusive community, where amateurs and passionate gamers met in virtual chat rooms (mostly IRC) to exchange knowledge, provide feedback, and help each other create their own video games. It was an environment of passion and creativity, where everyone could turn their dream of making their own video games into reality. Many kids who grew up in this environment have become today's professional game developers in both the indie and AAA industries.

In recent years, game engines for beginners and amateurs have become much more powerful, and the gaming industry has shifted its focus from huge AAA productions to indie and smaller products. GameMaker itself has grown and evolved, playing a significant role in today's indie games landscape. Consider titles like *Undertale*, *Hotline Miami*, *Spelunky*, *Risk of Rain*, *Hyper Light Drifter*, *Gods Will Be Watching*, and *Katana Zero*, all created with GameMaker by small teams or even solo developers.

## INTRODUCTION

This book will guide you through learning the basics of GameMaker and designing and developing your first video games. We will embark on a journey to understand the fundamentals of game development, analyzing successful games to learn valuable lessons for creating fun and engaging experiences. We will explore various game genres, designing and developing them while understanding and internalizing the principles of game design and development, using the power and accessibility of GameMaker. Buckle up and prepare to embark on an exciting journey into game development, where you'll dive deep into the essential skills and creative processes needed to craft your own games.

# CHAPTER 1

## Overview

“How can I make video games?” This is a question I asked many times to a lot of people (and mostly to Google) when I was a kid. The desire to create games is something that nearly every gamer happens to have at a certain point. It’s something that is common between all media consumers, from books to movies to video games: we try to create the things that make us feel good. We dedicate a lot of time to video games, and they give us strong emotions and wonderful stories in return. Sometimes they help us in hard times – like if a piece of software can understand us better than a person – and sometimes they just entertain us when we are bored or when we just need some fun. We give them time, and they give us emotions and wellness in return.

Driven by their fascination with the power of video games, many gamers become game developers. These developers are often among the most passionate you’ll find. Their mission goes beyond just creating software—they aim to share emotions and build entire worlds.

In this chapter, you will be introduced to the tools and topics that we are going to cover in this book. You will also learn how to install GameMaker on your PC or Mac, the first necessary step to start making great games!

# The Right Tool for the Job

Video games are a very special kind of medium. They can be just fun pastimes or very intense experiences. They can teach us concepts, they can train us on specific activities, they can stimulate our creativity and problem-solving, and entertain us with great stories. These special software use graphics, music, gameplay, and technology to do all this and much more!

There was a time when the requirement to make even very simple text-based applications was to learn some very complex CPU-specific programming language called Assembly. Some of the simplest old games you can think of, like *Rogue*, *Pitfall*, *Super Mario Bros. (SMB)*, or *Wolfenstein 3D*, are made fully or partly in Assembly.

Fortunately, we live in a time in which game development is way more accessible, to the point in which sometimes coding is not even strictly required. Today, to make games, we have at our disposal a number of software specialized in making games, called game engines.

Game engines offer a number of features that simplify the process of making games, like the possibility to show an image, play a sound, get keyboard input, and so on.

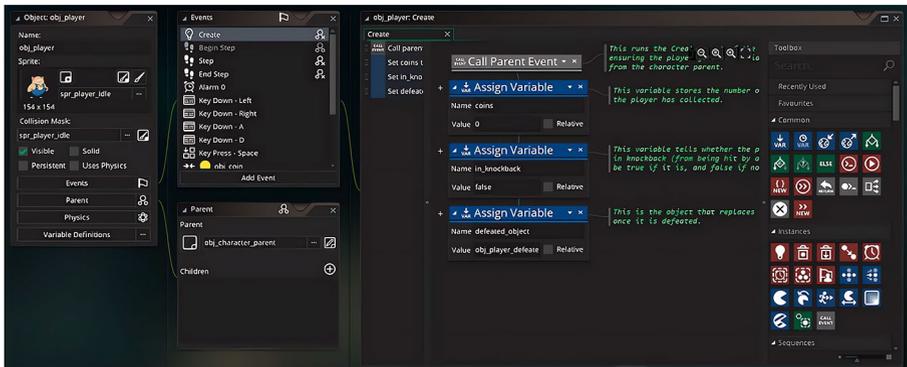
In this book, we will explore *GameMaker*, a professional yet easy-to-use game engine which offers two ways of making games: a proprietary scripting language and a no-coding drag-and-drop system. We will opt for the proprietary scripting language GML (*GameMaker Language*) to reach the full potential of the engine and to have maximum flexibility. But don't worry, this is not a programming manual, but a game development manual. Our focus is not only on how to code our games, but also on game design good practices and how to deliver fun gaming experiences. You will learn how games are made using real-world video games as study cases.

# What Is GameMaker?

**GameMaker** (*GM*) is a game engine that's perfect for both beginners and professionals. It supports 2D and 3D game development and allows you to create games with either a visual programming approach called *Drag and Drop* (DnD) or with a proprietary scripting language called GameMaker Language (*GML*). Let's take a closer look at those options.

**DnD** is a system that allows you to create game logic by combining blocks representing objects and functionalities. The specific order of those blocks defines an algorithm. DnD is the right choice for everyone that wants to explore game development and game design, but has no interest in learning how to code.

In Figure 1-1, you can see an example of DnD programming in GameMaker. From left to right, we have the object's properties, the list of the object's events, possible parent or children classes (more on that later), the DnD code blocks related to the selected event, possible comments related to the code blocks, and finally the list of possible object code blocks that we can drag and drop into the event.



**Figure 1-1.** An example of DnD programming with GameMaker

**GML** (*GameMaker Language*) is GameMaker's own scripting language. It's a highly specialized scripting language, which means that it's very easy to use, but it's also quite powerful. It's the best choice if you want maximum control on the features of your game and their implementation.

Many indie developers choose GameMaker for its ease of use and completeness of features. Some examples of great indie games made with GM are *Undertale*, *Hyper Light Drifter*, *Spelunky*, *Hotline Miami*, *Katana Zero*, *Forager*, *Nuclear Throne*, *Gunpoint*, *Nidhogg 2*, *Risk of Rain*, and so on.

GameMaker offers an *IDE* (Integrated Development Environment) containing all the tools you need to develop your game. The IDE includes a file browser to manage resources, a text editor to write GML code, a visual programming interface to create functionalities with DnD, a 2D graphic editor to create and edit images, an animation editor to make flipbook and skeletal animations, a compiler (YoYo Compiler, aka YYC) to export your games to many platforms in machine code, an interpreter to run and debug your games using the GameMaker virtual machine, and everything else you may need to create your game.

## About Game Design

Gaming is something that I always did with an inquisitive mind. I always played games (and I still do it now) asking myself questions like “*How is this made?*”, “*How does it work?*”, “*Is this fun? Why?*”, “*Why is it not fun?*”, and most importantly “*How can this be more fun?*” Only much later I realized that what I was doing all my life was (to some degree)

### **Game Design.**

**Game Design** is the process of imagining games, planning and defining all their main features. Game designers can focus on a single aspect of a video game, or they can direct or supervise the whole game by making sure that there is coherence and consistency between the

various components of the game and that the overall experience is at least honoring the user's expectations.

To be a game designer, most of the time it means to be able to ask the right questions and come up with the right answers. It's a constant learning process made of testing, experimenting, fine-tuning, and even questioning ideas, mechanics, and every component of a game.

## About Coding

Coding could be described as the act of writing a list of commands in a specific computer language, to instruct a machine on how to respond to some specific input provided by the user. This list of commands (or instructions) is translated by a software called *compiler* into *machine code*, which is the "native" language of a computer. The result of this translation is a binary file (a file made of ones and zeroes), which can be executed. The binary file is the software itself – the game, in our case. On Windows systems, binary (executable) files are called *EXE* files.

The projects in this book are created with GML. You can find a detailed reference manual about GML and everything concerning GameMaker in GameMaker's online documentation at [manual.gamemaker.io](http://manual.gamemaker.io).

## How to Use This Book

This book is designed to guide beginners into the world of game development one project at a time, exploring development techniques and GameMaker's features little by little. So if you're a beginner, the advice is to read the book cover to cover.

However, if you already have game development foundations or you're not new to GameMaker, feel free to use this book as a collection of game projects that can be explored in any order you prefer.

Here is an overview on the chapters and the topics they will cover:

1. **Overview:** The chapter that you're reading! An introduction to game development and to the topics covered in the book.
2. **Hello, World!:** In this chapter, you will create your first simple project with GameMaker learning some basics about the software.
3. **Card Game (Part 1):** In this chapter, we will design and develop the first version of a memory card game – a game about pairing matching cards. Card games are a good starting point to create some solid game development foundations.
4. **Card Game (Part 2):** This chapter will conclude the development of Memory, the card game about pairing matching cards started in Chapter 3. At the end of this chapter, you will have completed your first game!
5. **Fixed Shooter:** This chapter will be dedicated to the creation of a fixed shooter game inspired by classics like Space Invaders and Galaga. Enemies and bullet system are only two of the many topics that this project will cover.
6. **Shoot Em Up!:** In this chapter, we will extend the work done in Chapter 5 introducing new features inspired by classics like Ikaruga, R-Type, and Tyrian. This is also the chapter in which we will develop our first boss fight.

7. **Designing Bosses:** This chapter covers some interesting in-depth analysis of boss fights design taking as examples real-world video games that created memorable boss fights.
8. **Single-Screen Platformer:** In this chapter, we will explore the design and implementation of platformer games by creating a single-screen platformer game. You will learn how to create a platforming system from scratch and to design levels and enemies.
9. **Scrolling Platformer:** In this chapter, we will build on the ideas introduced in Chapter 8 and develop a scrolling platformer, one of the most famous and long-lasting game genres. Topics covered include gameplay features like power-ups, different kinds of enemies, a simple combat system, and different types of platforms that will allow us to get creative designing levels.
10. **Designing Platformers:** In this chapter, we analyze the history of platformers and how they evolved in the years. Considerations about how to make a platformer fun and challenging are the main topic of the chapter. There is an in-depth analysis of masterpieces of the genre like Super Mario games that will help us understand the golden rules for a good platformer.
11. **Metroidvania (Part 1):** In this chapter, we cover the design and implementation of the first part of a *metroidvania* game system. Main features of this genre are exploration, platforming, and combat.

We will start creating this project by using the concepts studied in the previous chapters and introducing new concepts like exploration skills (dash and wall jump).

12. **Metroidvania (Part 2):** This chapter concludes the metroidvania project. You will learn how to implement all the defining features of a metroidvania including on-screen maps, a checkpoint system, a shooting system, and an inventory and equipment system.
13. **Game AI:** This chapter introduces the concept of AI in video games and teaches one very effective (and still very used) game AI technique: FSM-based AI.
14. **Designing Good Games:** This chapter concludes our game design exploration introducing some principles and techniques that can help you on your journey to make great games. We will analyze famous and successful games to understand what they did good, why they are considered masterpieces, and how we can use this knowledge to design good games.
15. **What's Next?:** Our journey ends with a little guide on how to go forward in your game development career. We will explore the available options to sell or distribute your game as an indie developer on the most popular digital games stores.

## Additional Content

This book is heavily based on the use of GameMaker and revolves around the projects proposed in every chapter. So, if you're having some problem following the instructions or you just want to see the working project before you start, you can take a look at the source code on GitHub (via the book's product page located at <https://github.com/Apress/Game-Development-with-GameMaker-2nd-Edition> for more detailed information, please visit <https://www.apress.com/gp/services/source-code>).

## Pricing

GameMaker comes in different flavors (see Table 1-1) depending on your needs and if you are a professional or amateur developer. Below, you can find a useful table that can help you out making your decision based on your own needs.

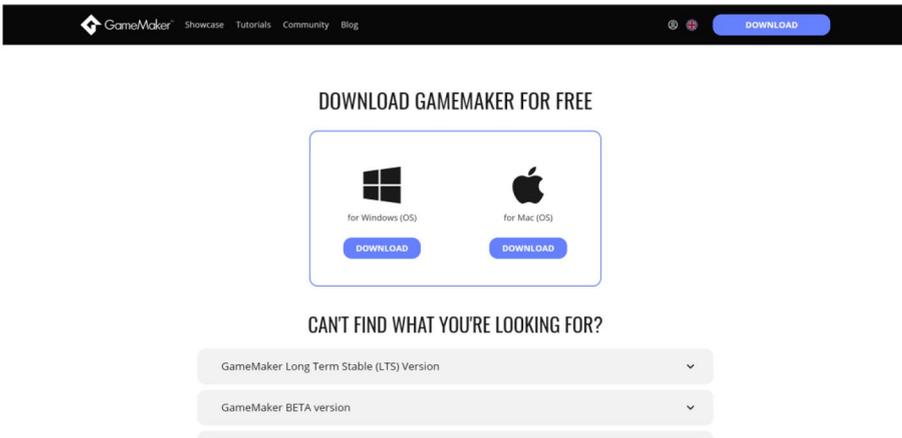
My suggestion is to go with the free version while you are learning or developing your game and only switch to the professional or enterprise version if you are ready to sell your game.

**Table 1-1.** GameMaker comes in different flavors. This table shows a list of all the possible licenses that GameMaker offers

License	Price	GX.games Export	Desktop Export	Web Export	Mobile Export	Console Export
Free (Non-commercial)	X	✓	✓	✓	✓	X
Professional	\$99.99	✓	✓	✓	✓	X
Enterprise	\$79.99 monthly - \$799.99 yearly	✓	✓	✓	✓	✓

# Installing GameMaker

Installing GameMaker is as easy as going to the official website (<https://gamemaker.io/en/download>) and pressing the download button (Figure 1-2).



**Figure 1-2.** *GameMaker: Studio 2 download page*

To be able to use GameMaker, you will have to create an Opera account in which you will have to log in when running GameMaker.

Other than the Opera account, here are some system requirements that you need to meet to be able to run GameMaker:

**Minimum Spec:**

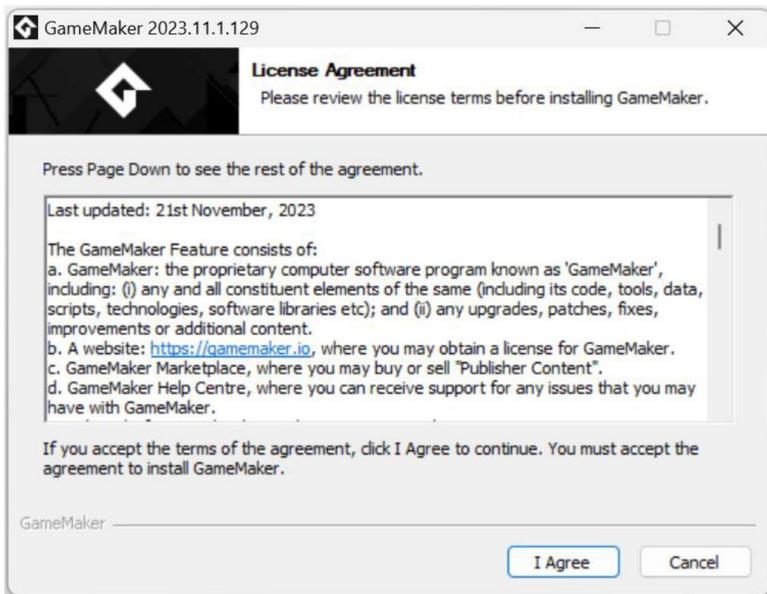
- Windows 7 with SP1 or Mac OS Big Sur
- Dual Core CPU
- 2GB RAM
- OpenGL 4-compliant onboard graphics
- HDD (at least 3GB free disk space)

Recommended Spec:

- Windows 10/11 or macOS Monterey
- Quad Core CPU
- 8GB RAM
- Dedicated graphics card
- SSD (at least 3GB free disk space)

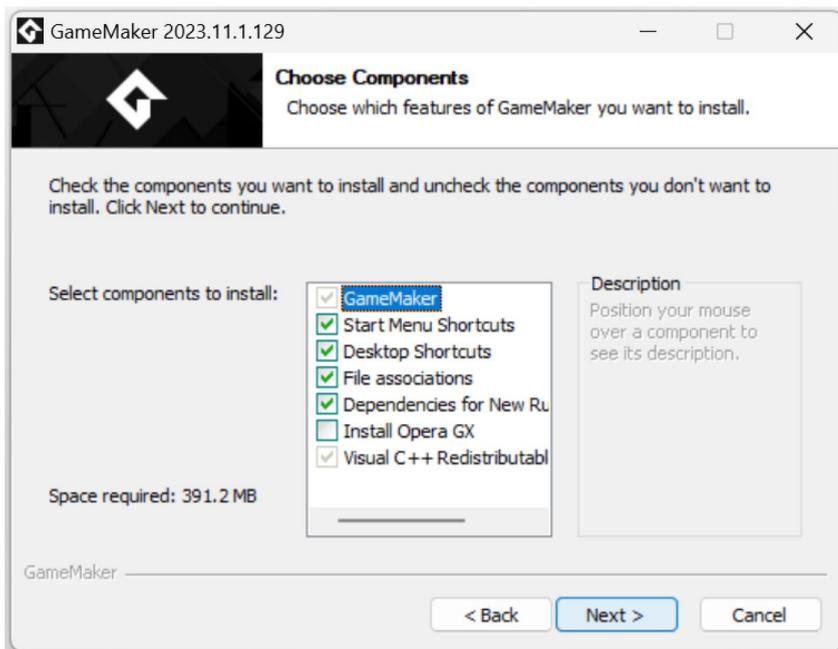
## Windows

When you execute the installer client, you will be prompted to a license agreement (Figure 1-3) that you need to accept to use the software. Note that if you're updating to a new version, you will be asked if you want to delete the previous version before continuing the installation process.



**Figure 1-3.** Windows installer's License screen

After that, you will be taken to the Choose Components screen (Figure 1-4) where you can check the additional components to install and some other options like creating Start Menu shortcuts, Desktop shortcuts, or choose to associate YYP and GML files with GameMaker. If you're in doubt, the default choice is the safe one. Don't overthink this step, as you will be able to install the missing package or set any of those options even after the software is installed.



**Figure 1-4.** Windows installer's Choose Components screen

Clicking Next, you will be asked to choose the destination of the installation (Figure 1-5). If you don't know what to do, just stick with the default choice. It will install GameMaker in your main disk.