

LEVEL UP

BECOME A
GAME DEVELOPER



3

2

1



LEARN HOW TO MAKE & SELL YOUR OWN GAMES



Table of Contents

| | |
|--|-----------|
| WELCOME | 1 |
| INTRODUCTION | 2 |
| GETTING TO KNOW CONSTRUCT 3..... | 3 |
| OPENING AND USING C3 | 3 |
| SYSTEM REQUIREMENTS | 4 |
| USING AN ACCOUNT | 5 |
| C3 ON MOBILE | 5 |
| THE USER INTERFACE | 8 |
| OVERVIEW | 8 |
| START PAGE..... | 9 |
| MAIN MENU | 11 |
| MAIN TOOLBAR..... | 13 |
| VIEWS..... | 14 |
| LAYOUT VIEW | 14 |
| EVENT SHEET..... | 18 |
| BARS..... | 23 |
| PROJECT | 23 |
| PROPERTIES..... | 24 |
| LAYERS | 25 |
| ANIMATIONS EDITOR | 26 |
| SAVING/SHARING | 32 |
| MAKING YOUR FIRST GAME | 35 |
| PART 1 – GETTING STARTED | 35 |
| PART 2 – ADDING MORE OBJECTS | 37 |
| PART 3 – ADDING BEHAVIORS..... | 40 |
| PART 4 – ADDING EVENTS..... | 44 |
| PART 5 – MORE GAME LOGIC | 49 |
| PART 6 – USING INSTANCE VARIABLES | 53 |
| PART 7 – KEEPING SCORE | 56 |
| PART 8 – DISPLAYING SCORE | 58 |
| PART 9 – FINISHING UP..... | 60 |



| | |
|--------------------------------------|-----------|
| TESTING AND PUBLISHING | 61 |
| TESTING YOUR GAME | 61 |
| EXPORTING AND PUBLISHING..... | 63 |
| PUBLISHING TO ITCH.IO | 64 |
| | |
| WHAT NEXT? | 73 |
| | |
| USEFUL RESOURCES | 74 |



Welcome to Level Up

So, you've decided you want to make games using Construct 3? That's fantastic! Let me take this opportunity to thank you for choosing Construct and welcome you to this eBook that will accompany you on your first steps with the program.

The first chapters will go through how to use Construct 3, the program's interface and explain what various things do. Following on from that, we'll show you how to make your very first simple game, including publishing it!

This eBook is based on existing Construct 3 documentation, including [the Manual](#) and [Beginner's Guide](#) tutorial, both written by Scirra founder, Ashley Gullen. I've simply added some extra bits and re-packaged it into an offline resource for you!

I hope you enjoy your journey with Construct 3 and would like to thank you again for choosing Level Up.

Laura Donaghy
Editor



Introduction

Once upon a time, the world of game development was only open to those skilled in the ways of code, or graphic design. People looked on as these heroes created the early classics that we came to adore.

Fast forward to today and the landscape has changed dramatically. Development tools are now more readily available than ever and range from full blown code tools to simpler (but still powerful) block-based tools. Anyone can turn on their computer (or even their phone or tablet these days) and start working away at the game of their dreams.

The video games industry is growing at an incredible rate, and it's brilliant how anyone can become a part of it. In 2018, the video games industry in the UK alone was valued at a whopping £5.7 billion – a positive growth of 10% over 2017. And with things like eSports on the rise, this industry looks to continue that growth trend.

Here at Scirra, we're proud to be part of that growing industry, and to have been a part of it since 2011. The Construct engine started as an open source project called Construct Classic. Classic evolved into a commercial product – Construct 2. And now, after being downloaded more than 4 million times worldwide, it's evolved again into Construct 3.

But what *is* Construct 3?

Construct 3 is a browser-based game engine that allows you to create games without having to use a single line of traditional code. Your core toolkit starts with Construct's unique event system. Combine that with built-in behaviors and you can have a game up and running in minutes. It's a great place to start and can help you make some powerful games. It's cliché, but your only limit is your imagination!

So, if you want to jump into the game development world, you're in the right place. Keep this book to hand, pick up your development weapon and let's get going.



Getting to Know Construct 3

This segment will take you through Construct 3 itself, starting with how to open and use the program, and then moving on to explain what some of the core features of the interface do. After all, it's helpful to know what something does before you begin using it in earnest!

Opening and Using C3

Construct 3 runs right in your browser, so there's nothing to download or install!

To open the program, visit editor.construct.net in your browser on a supported system and Construct 3 will start. This makes it easy to switch between devices, use public computer terminals (even with strictly limited access), or painlessly deploy Construct 3 across a computer lab or office.

Despite being browser-based, Construct 3 works offline - you don't have to always have an active Internet connection, you only need to be online the first time you load Construct 3. After Construct 3 first starts, after a while, you should see a notification in the corner indicating Construct 3 is ready to work offline.

Note that if you purchase a subscription and work offline, you must start Construct 3 while connected to the Internet at least once every 7 days to re-validate your subscription. However, if you have not purchased a subscription, you can use Construct 3 offline permanently.

You can add Construct 3 to your desktop, shelf (on Chrome OS) or home screen (on Android) which creates an icon on your device to launch Construct 3 like an app.

Sometimes, **Install as app** will appear in Construct's main menu when this option is available. Simply click this menu option to install Construct 3 as an app on your device. Alternatively, if you are using Chrome, you can do this manually from Chrome's menu with the following steps:

- On Windows and MacOS choose **Install Construct 3** from the menu
- On Linux, choose **More tools** → **Create shortcut**
- On Chrome OS, choose **More tools** → **Add to shelf**
- On Android, choose **Add to Home screen**

This is a great way to reach Construct 3 more easily, and also saves space on your screen since it hides the browser address bar and tabs. It's also a much convenient way of using Construct 3 offline.

Construct 3 automatically stays up-to-date, so you don't need to worry about a thing. It will notify you when there's a new version available, in the case of beta releases it will ask if you want to update, and it'll inform you when it's updated. The Construct website also provides a list of all releases with detailed information about changes in each update, and also provides links to run



older versions in case there's a problem with an update. You can check which version of Construct 3 you're currently using by opening Construct 3's main menu and selecting About.

System Requirements

Supported Browsers

Construct 3 can run in the following browsers:

- Google Chrome 57+
 - Other browsers that use the Chrome browser engine (Chromium), such as Opera and Yandex, providing they are updated to Chromium 57+
- Firefox 55+
- Safari 11+
- Microsoft Edge 16+

Construct 3 does not support Internet Explorer. However, in Windows 10 Microsoft replaced Internet Explorer with the Edge browser, which is supported from version 16+.

Supported Operating Systems

Construct 3 can work on the following operating systems:

- Windows: Windows 7, Windows 8, Windows 8.1, Windows 10 or newer
- Mac: OS X / macOS 10.9 or newer
- Linux: 64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+
- Chrome OS: Any Chrome OS device updated to v57+
- Android: Any Android 5.0+ device with at least 1GB RAM
- iOS: Any iOS 11+ device

Note that Safari users on Mac must use Safari 11+, which is only supported on macOS El Capitan (10.11) or newer. Users on older versions of macOS/OS X can use Chrome or Firefox.

WebGL Support

Construct 3 requires the browser to support WebGL, which is a modern high-performance graphics technology for browsers. Almost all modern devices support WebGL. However, if you see a message about WebGL not being supported, try installing any available system updates, and check your graphics drivers are up-to-date.



Using an Account

When you first start Construct 3, you'll use it as a Guest. This means you are not logged to an account. Construct 3 shows your account status near the top-right corner. You can click this "badge" to show a menu with some account options. (See right.)

ACCOUNT BENEFITS

Until you purchase a subscription, Construct 3 works in a limited **Free edition** mode, as indicated by the "Free Edition" label on the account badge. The limitations of the Free edition are listed on the Construct 3 plans comparison page, where you can also purchase a subscription to remove the limits.

Guests have lower limits in the Free edition than registered users. Guests may only use up to 25 events in a project. Registering an account and logging to Construct 3 allows you to use up to 40 events, and then verifying your email address allows you to use up to 50 events.

If you purchase Construct 3, you must be logged in with the same account you purchased with to make use of the full features of Construct 3. The **Free edition** label next to your account will disappear to indicate you have an active subscription and no longer have the Free edition limits imposed.

At any time, you can click the View details option in the Account menu to open a dialog displaying more information about your account.

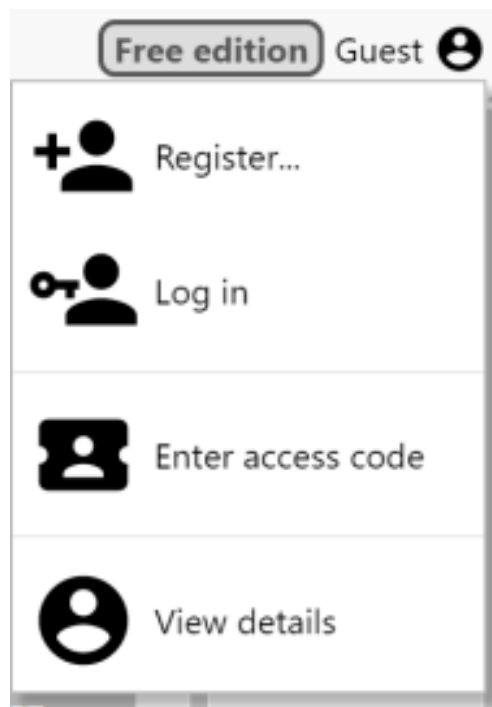
ACCESS CODES

If you have an Education subscription, you can create a time-limited access code in the subscriptions section of your construct.net profile. This allows students to use a licensed version of Construct 3 for a period of time. The access code can be used for simultaneous users, up to the number of seats your plan includes.

Once you have created an access code, share the code with your class. Students can then start Construct 3, choose the Enter access code option in the Account menu, type in the access code, and click OK. This will grant them access to the full version of Construct 3. This can also be done with a Guest account, so students do not need to register their own accounts. Once the access code expires, Construct 3 will notify the user and revert back to the Free edition.

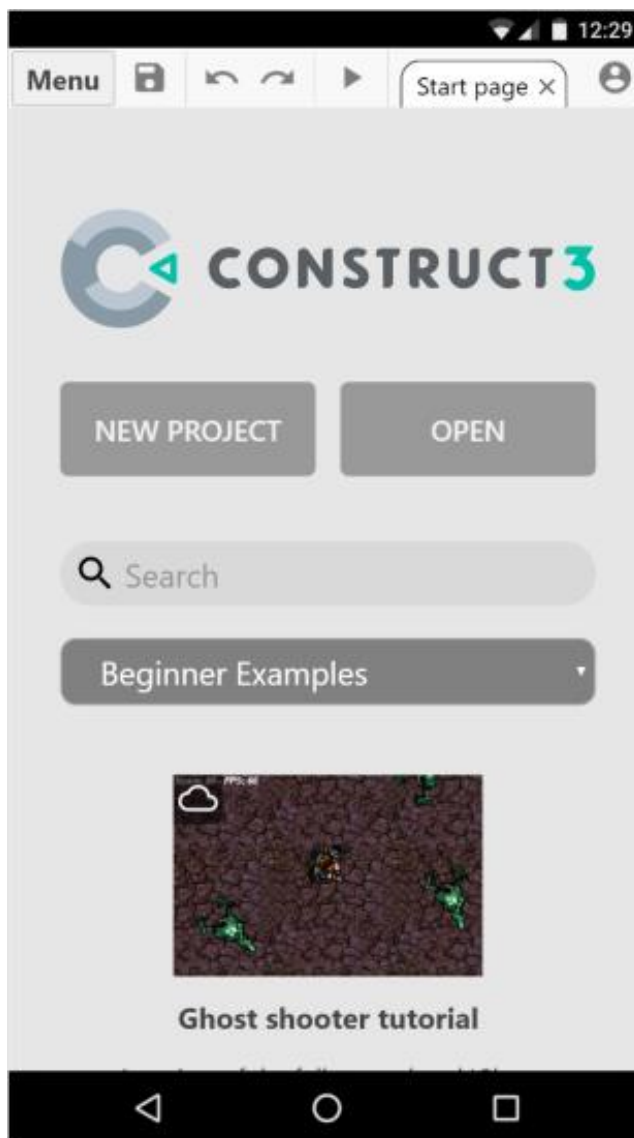
C3 on Mobile

Construct 3 works on mobile devices like phones and tablets. While Construct 3 can be used on a phone, it is much more comfortable to use a tablet device with a larger screen if you have one available.





Construct 3 adapts its appearance to better suit these devices. On mobile it will look something like this:



When using Construct 3 on a mobile device, you can use touch equivalents to mouse clicks, for example:

- Where you would click on an item, simply tap on it.
- Where you would double-click, double-tap instead.
- If you need to right-click something, or open a context menu, tap and hold on the item. After a moment a menu will appear.

ACCESSING BARS

Many mobile devices have small screens, so, to make the most out of the space available, Construct 3 hides bars by default. To access these bars, swipe in from the side and the bar will slide in. Since there are a number of bars and only two sides, you can access the other bars by



repeatedly swiping in from the side again. As you do this the previous bar will slide out and the next bar will slide in.

On the left side the sequence of bars is:

1. Properties Bar
2. Bookmarks Bar
3. Find Results bars
4. Z Order Bar

On the right side the sequence of bars is:

1. Project Bar
2. Layers Bar
3. Tilemap Bar

If you reach the end, the sequence will start again, cycling through the set of bars for that side of the screen. Bars can be closed by swiping them back the other way. The next time you swipe in from the side of the screen, you'll always get back the last bar you used that side. That helps you keep using the same bar for a while, and you can still keep swiping to switch between bars at any time.

Note that only some of Construct's bars will have their functions covered in this book. For more details about all of Construct 3's functionality, please refer to the manual.

CHANGING UI MODE

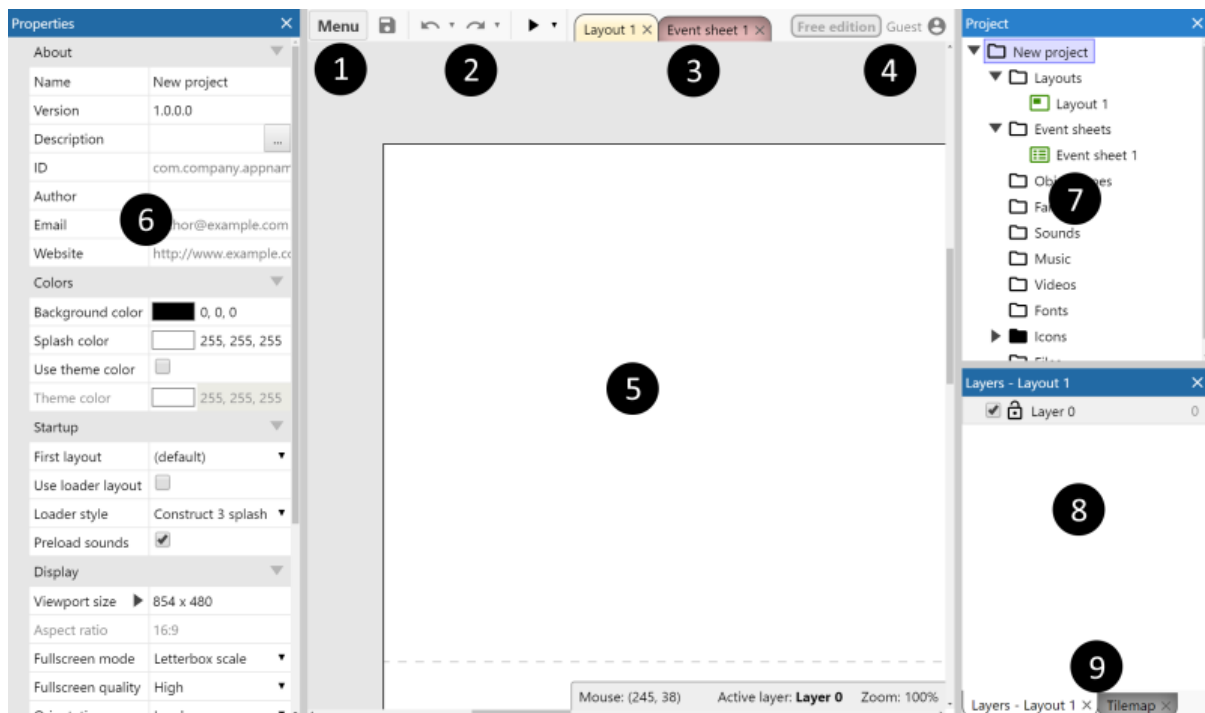
Some devices have sufficiently large enough screens to use Construct 3 in desktop mode. However, depending on the device, C3 may still use the mobile UI intended for touchscreens. You can make Construct 3 switch to desktop mode by changing the UI mode to Desktop under Construct's settings. This will always load Construct 3 using the full desktop UI. You can change this at any time, but you'll need to reload C3 for it to take effect.



The User Interface

Overview

The following image highlights the important parts of the Construct 3 user interface (UI) with numbers. Click the image to enlarge it. An overview of each part is provided below, and later in this chapter, each section will be covered in more detail. Note that initially, only the Start Page is visible. Much of the interface will not appear until you create or open a project. Also, please note the exact appearance of Construct 3 can depend on which theme you have selected.



1. Main menu button

Click this to open the main menu. This provides options for basic tasks like opening and closing projects, exporting, changing settings and so on.

2. Main toolbar

This provides shortcuts to the most commonly-used features: save, undo, redo and preview. The arrows next to some buttons provide a dropdown menu with more options.

3. View tabs

These tabs let you switch the main view between different layouts (where you place objects) and event sheets (where you define logic using the event system).

4. Account badge

This shows your account status. Click it to show the Account menu.



5. Main view

This is where the currently selected Layout View or Event Sheet View appears. The view tabs select which is visible. In this picture, it's showing an empty Layout View.

6. Properties bar

This lists all the properties for the selected item, allowing you to change settings for it.

7. Project bar

This lists everything in your project. It gives you an overview of what you've added and lets you navigate around the project as well, such as by opening layouts or event sheets to view them.

8. Layers bar

When a Layout View is open, this shows the layers on the layout.

9. Tabs

By default, the Layers bar and Tilemap bar are docked together. You can use these tabs to switch between the bars. You can drag and drop bars around to rearrange them. You can dock or tab together any combination of bars you want to customise the interface.

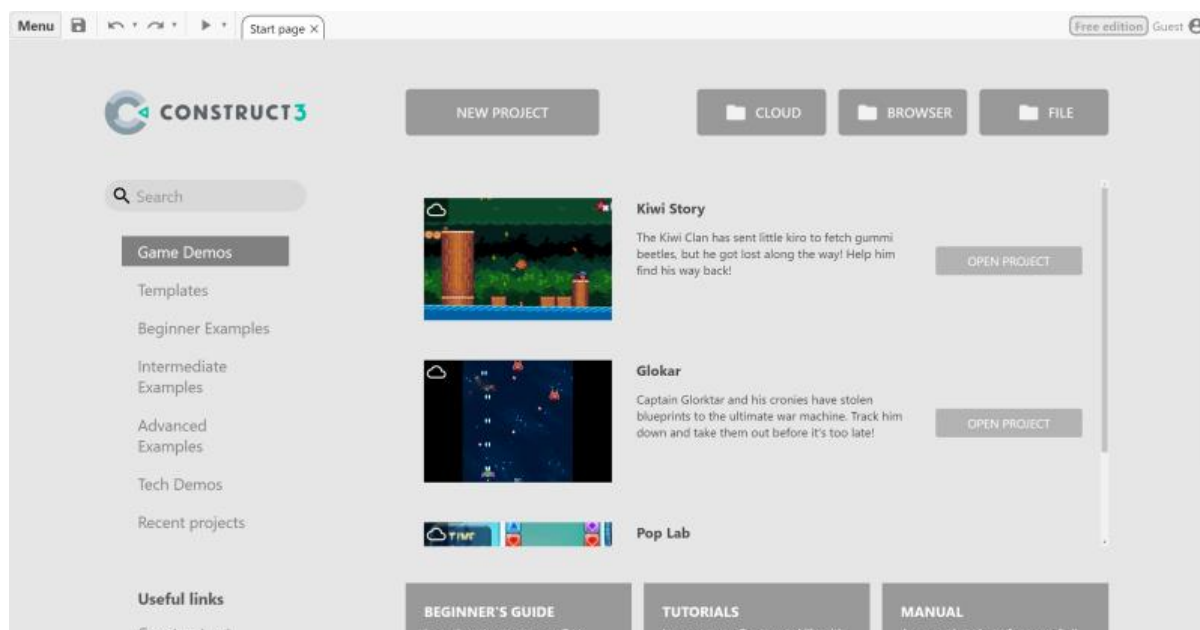
CHANGING THE THEME

You can choose a different theme to alter the appearance of Construct's interface, for example, choosing a dark theme. The current theme can be changed from Settings. Third-party themes can also be installed as addons.

Start Page

When you first start Construct 3, it shows the Start Page which gives you a useful starting point whenever you launch Construct 3. It provides shortcuts for tasks like creating a new project, opening an existing project, a set of useful links, and a library of game demos, templates and examples that you can browse through.

The appearance of the Start Page changes depending on the size of the window or screen. It will look something like this on a desktop display. (*See overleaf.*)



The Start Page initially fills the whole window. When you create or open a project, the rest of Construct 3's interface will appear.

STARTING A NEW PROJECT

Click New project to create a new empty project. You'll be prompted for some basic details about the project to create. You don't need to enter anything though, just click Create and you'll get a new empty project with default settings.

BROWSING EXAMPLES

There are a few demo games you can check out, like Kiwi Story. Construct 3 also comes with a lot of examples showing how various features work, sorted into **Beginner examples**, **Intermediate examples** and **Advanced examples** depending on their difficulty level. You've also got the **Templates** section which has a list of projects each with a set of game mechanics already set up, like an endless runner type game for example.

You can quickly preview a demo game or example by clicking its Preview button on the Start Page. To open the project so you can see how it works, either click Open Project, or just double-click or double-tap the project. Once open you can then try it out by pressing the Preview button in the main toolbar, or by pressing F5. Once you're done, close the project by selecting **Menu** → **Project** → **Close project**. When the project closes, you'll see the Start Page again where you can choose another example project to preview or open. You can also search for examples in the **Search** field above the category list.

LINKING TO EXAMPLES

You can get a direct link to an example by right-clicking an example in the Start Page and selecting Copy direct link. This will copy a link with the example embedded, which when visited, will open Construct 3 and automatically load that project. This is a useful way to share examples.



OPENING EXISTING PROJECTS

You can open projects from three sources: Cloud (projects saved to a cloud storage service like Google Drive), Browser (projects saved to the local browser), or File (from a file on the device). You can also find previously saved projects in the Recent projects section. Note that projects saved to local files cannot be shown in this section. For more information about saving and loading projects, see the chapter on Saving and Sharing.

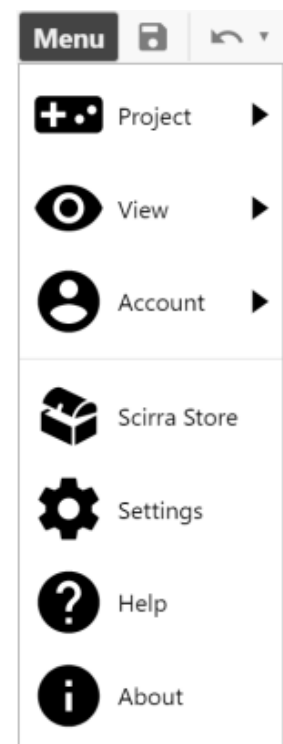
OTHER RESOURCES

There are lots of links on the Start Page to other resources to help you get started and find out more about Construct. You can find links to community resources like the forums, social media accounts for Construct where you can follow news and updates, and other learning resources like tutorials.

Main Menu

Click the Menu button next to the main toolbar to access the main menu. The exact options which appear in the menu depend on whether you have a project open, and whether or not you are logged in.

If you're using Construct 3 as a guest and you have a project open, you'll see the most options in the Main Menu and that menu's structure is as follows:



Project

Save

Saves the current project to its last saved location. If it has not been saved before, this defaults to Cloud Save.

Save As

Cloud save: save to a cloud storage service.

Save to local browser: save to the browser's storage on the device.

Download a copy: save a copy of the current project by downloading a file.

Preview

This runs a preview of the current layout.

Remote Preview

Starts a Remote Preview of the current project – this allows you to test or share projects on other devices by visiting a URL or scanning a QR code. This will be covered in more details in the Saving and Sharing chapter.

Debug

Runs a preview of the current layout with the debugger enabled.



Export

Opens the Export dialog to begin exporting the current project for publishing.

Close project

Closes the current project, prompting you to save if there are any changes.

New

Creates a new empty project – this opens the same dialog as if you created a new project from the Start Page.

Open

Opens a previously saved project from the cloud, local browser, or a local file.

View

Bars

In this submenu, you can hide and show any of the bars visible in the interface. If a bar is missing, or you accidentally close one of them, use this menu to bring it back. Anything with a tick next to it should be visible!

Start page

Ticking or unticking this option allows you to show or hide the Start Page.

Addon manager

Opens the Addon Manager to view and manage addons. You can find a lot of third-party addons to extend Construct 3's capabilities in the [Addon Exchange](#) on [construct.net](#)

Export manager

Opens the Export Manager dialog, which lists the last few exports and allows you to download them again.

Account

Register

Takes you to [construct.net](#) in order to register a new account to use with Construct.

Log in

Opens the log in dialog to allow you to use Construct with an existing account.

Enter access code

Allows you to enter an access code to allow temporary use of the full version of Construct. As previously mentioned, this does not apply for personal licenses.

View details

Opens a dialog displaying more information about the Construct account currently in use.



Scirra Store

Takes you to the Scirra Store where you can find game assets, other software and more.

Settings

Opens the Settings dialog where you can customise Construct 3's settings to work the way you prefer. You can set options in here for things like C3's theme, your UI preferences and whether or not C3 prompts you about new beta releases.

Help

Opens the Construct 3 manual.

About

Opens a dialog displaying information about the version of Construct 3 you're using, as well as credits, storage information and diagnostic details.

Main Toolbar

The main toolbar provides quick access to a few of the most commonly-used options in Construct. It appears next to the main menu button.



The main toolbar has the following buttons:

Save

Saves the current project to its last saved location. If the project has not been saved yet, this defaults to Cloud Save. You can change the default location under Construct 3's settings.

Undo and Redo

Undoes the last performed action in the editor. After pressing undo, you can then redo the action again. Click the dropdown arrow next to the button to see a list of the undo or redo actions. Selecting an item from the list will undo or redo all the actions up to the chosen item.

Preview

Runs a preview of the current layout. By default, this opens a pop-up window; you may be prompted to allow pop-ups. Click the dropdown arrow next to the button to see a list of other kinds of preview.



Views

Layout View

The Layout View is a visual designer for your objects. It allows you to set up a pre-arranged layout of objects, such as a game level, menu or title screen. But before we talk about the layout view itself, let's define what a layout is.

A layout is a pre-arranged set of objects. It can represent a game level, menu or title screen. They can be added, renamed and deleted from the Project Bar and they are edited from within the Layout View. Every layout has an associated event sheet which defines how the layout works.

Layouts contain a stack of layers and a layout must have at least one layer – try not to get **LAYOUTS** and **LAYERS** mixed up! Objects that appear on the screen do not belong directly to a layout - they belong to one of the layers in the layout.

Layouts do not have a background colour. To set a background colour, make the bottom layer opaque and set its background colour. This can be done in the Layers Bar but won't be necessary in the game we'll make later.

Layouts can also have effects applied to them, which affects all content appearing in the display.

Now you know what a layout is and what it's for, let's move on to the Layout View:



The dashed rectangle in the top left of the layout area indicates the viewport size in the layout – so this is the size of the area visible to the player at any given time. The viewport is set to this position by default, but you can change its size and orientation in the project properties.

In the bottom right-hand corner of the layout view, you'll see a small status bar with information about the current mouse position in the layout,

the current zoom level, and the current active layer. The active layer is important since it is the layer new object instances are added to. The active layer can be changed by selecting a different layer in the Layers Bar.

ADDING, MODIFYING AND DELETING OBJECTS

In Construct, objects perform most of the useful work in a project. A lot of the things you see in a Construct project are represented by objects, and there are also hidden objects for other purposes (e.g. audio playback).



When inserting a new object, typically you first choose the plugin in the dialog (e.g. *Sprite*). This then creates an object type (e.g. *TrollEnemy*). When the mouse turns to a crosshair this allows you to place the first *instance*, and you can duplicate the instance to create more of them.

Understanding the differences between them is essential to use Construct effectively, especially *object types* and *instances*.

Object types are a central part of game design in Construct. Object types define a 'class' of an object. For example, *TrollEnemy* and *OgreEnemy* could be different object types of the *Sprite* plugin. They have different animations and events can be applied separately to make them look and act differently, despite the fact they are both *Sprite* objects.

There can be multiple instances of an object type in a project. For example, you may wish for there to be four *TrollEnemy* objects awaiting the player in a game. These four instances share the same animations, images, behaviors, instance variables and events. (In the case of instance variables, each instance stores its own unique value, e.g. for health, and behaviors work independently for each instance too.)

So how do we go about adding objects into a project?

Double-click a space in the layout or right-click and select *Insert new object* to add a new object type. This will bring up the *Create New Object Type* dialog.

To create new instances of an existing object type, you can either copy and paste, click and drag the existing object while holding the *Ctrl* key, or drag and drop the object in from the *Project Bar*.

A shortcut for importing image files as *Sprite* objects is to drag and drop image files straight from a local folder into the *Layout View*. This automatically creates a new *Sprite* object type with the dragged image. This will automatically name Sprites as *Sprite1*, *Sprite2*, *Sprite3* and so on. It is recommended that you re-name your sprites to make your life easier down the line!

If multiple image files are dragged into the editor at once, the *Sprite* is assigned an animation with the dragged images as animation frames.

Dragging in animated *GIF* files this way will only use the first frame of the animation. If you want to import every frame of an animated *GIF*, use another tool to split the frames up to separate *PNG* images and import those.

Instances can be moved by dragging and dropping them with the mouse. Hold *Shift* to axis-lock the drag to diagonals. Alternatively, they can be nudged 1 pixel at a time with the arrow keys (hold *shift* to nudge 10 pixels), or coordinates can be typed in directly to the *Properties Bar*.

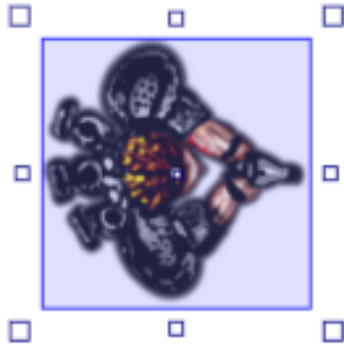
The *Delete* key or right-click *Delete* option deletes instances. Deleting all instances of an object does not remove the object type from the project. To entirely remove an object from the project it should be deleted via the *Project Bar*.

Click objects to select them. Objects cannot be selected if their layer is locked. Hold *Control* while clicking to select multiple objects or click and drag a selection rectangle to select all objects in an



area. The Properties Bar displays properties for *all* currently selected objects, so changing a property sets it for every selected object.

When a single object is selected it appears with resize handles around it.



Click and drag the resize handles to stretch the object and if you hold down the Shift key, you will proportionally resize the object. Hold control to resize relative to the object origin, which appears as a small dot on the selected object.

Rotatable objects like Sprite can be rotated by moving the mouse just outside the resize handles, away from the object. When you do this the mouse cursor will change to a rotation arrow. When you see this, click and drag to rotate the object.

Sometimes the resize handles, or rotate cursor, can get in the way of other objects. If this happens, hold Alt to temporarily hide the resize handles and disable rotation. This allows you to select another object instead of modifying the selected object.

SCROLLING AND ZOOMING

There are a few ways to scroll in the Layout View:

- You can use the vertical and horizontal scrollbars at the edges of the view.
- Scrolling the mouse wheel will scroll the view vertically. To scroll horizontally, hold the shift key while scrolling the mouse wheel.
- Hold the middle mouse button or the Spacebar and drag the mouse.

On desktop systems, middle-mouse dragging is probably the most convenient way to move around the layout, while the Spacebar option makes this method more viable for laptops with trackpads.

Zooming is useful to focus on a small area or see an overview of the entire layout. There are several ways to zoom:

- The Zoom options in the View menu when right-clicking in the Layout View
- Hold Control and scroll the mouse wheel. Hold both Control + Shift to double or halve the zoom (e.g. 100%, 200%, 400%...)
- Ctrl and + or - on the keyboard. Hold Shift to double or halve the zoom.

Press Control + 0 to return to 100% zoom.

If you're using Construct on mobile, you can scroll across the Layout View by swiping with two fingers, and you can zoom in or out by using pinching gestures.

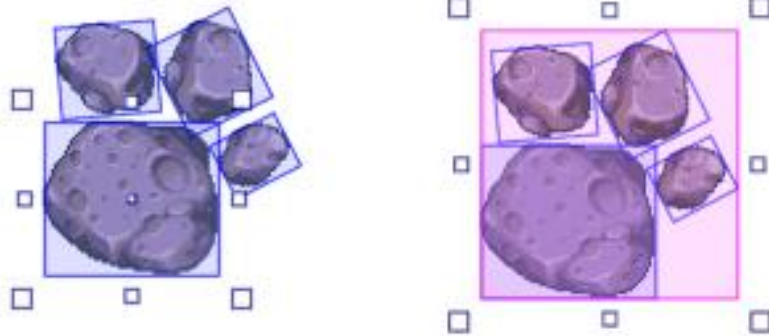
SELECTION WRAPPING

If you select two or more objects, you can wrap the selection by pressing Enter or right-clicking and selecting Wrap selection. This is a little more complicated to do on mobile, but you can hold



one object, then tap others to add them to the selection. Then simply tap and hold on one object in the selection to open the menu where you'll find the option to wrap the selection.

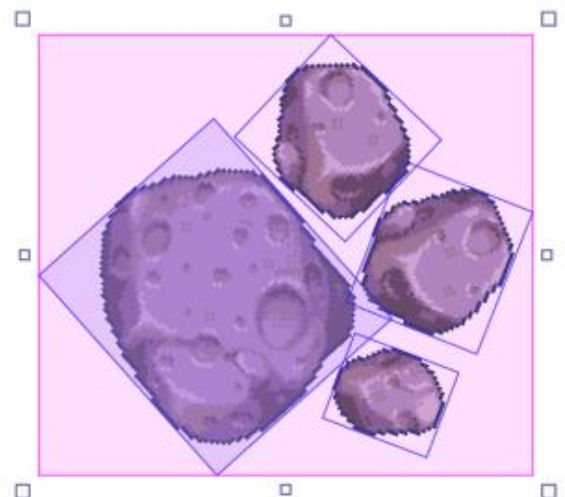
These wrapped selections will appear with a different coloured selection box, like so:



Wrapped selections can be resized and rotated as if they are one large object. For example, the selection can be enlarged and rotated, and all objects maintain their position relative to each other.

While a selection is wrapped, click any of the objects in the wrapped selection to make that object the rotation origin.

Click or tap outside of a selection to clear it.



CONTAINERS

Containers are an advanced feature that is only available in the full edition of Construct 3. They are a little beyond the scope of this eBook, however we will briefly mention them here to differentiate between types of wrapped selections.

Objects that are grouped into Containers are highlighted in yellow in the Layout View. Containers can also be set to automatically wrap their selection. If you still need to select an individual object in an automatically wrapped selection, hold Alt and click one of the objects.



MORE TO CONSIDER

As previously mentioned, each layout has an associated event sheet, which is easily accessible by either pressing Ctrl + E or right-clicking the layout and selecting Edit event sheet.

The Z Order of objects (that's the order which determines what objects sit in front of others) within a layer can be adjusted by right-clicking and selecting **Z Order → Send to top of layer** or **Z Order → Send to bottom of layer**. The full version of Construct 3 has a dedicated bar to allow advanced control over the Z Order.

Objects can be snapped to a grid for tile placement, and the collision polygons of the displayed objects can also be outlined. These features can be enabled in the layout's properties, which can be found in the Properties Bar when the layout in question is selected.

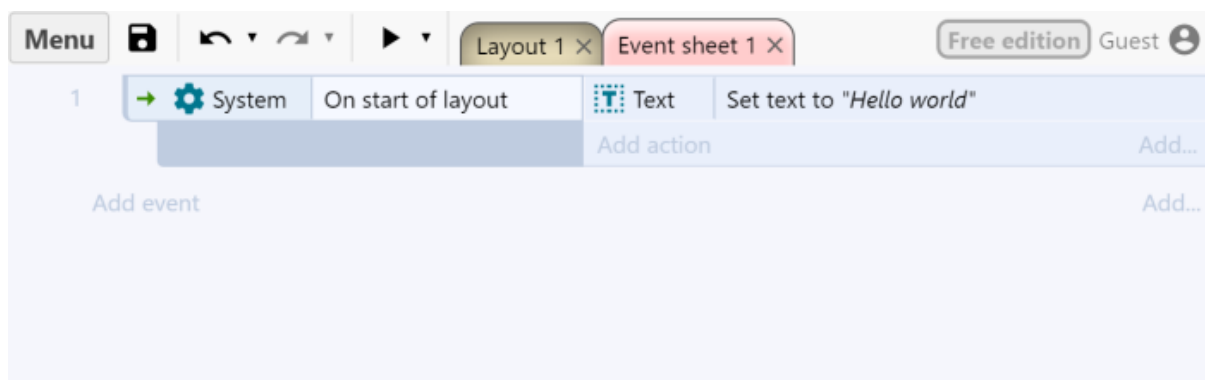
The right-click menu in the layout view also provides some alignment tools under the Align sub-menu. These allow you to quickly space objects equally or align objects along their edges. When aligning, the objects are aligned to the particular object you right-clicked.

The Animations editor can be brought up by double-clicking objects with images or animations like Tiled Background and Sprite. You can also double-click Text objects to edit their initial text in a dialog.

If you want to see any effects you've applied to objects, there's a setting under Project Properties which will display them in the layout view.

Event Sheet

The Event Sheet View is where events can be added, viewed and edited in an event sheet using the event system - Construct's alternative to traditional programming.



WHAT IS AN EVENT?

Events are designed to be easily readable and to intuitively "just work". However, they have specific, well-defined ways of working. They consist of **conditions**, which test if certain criteria are met, e.g. "Is spacebar down?". If all these conditions are met, the event's **actions** are all run, e.g. "Create a bullet object". After the actions have run, any sub-events are also run - these can then



test more conditions, then run more actions, then more sub-events, and so on. Using this system, we can build sophisticated logic for our games and apps.

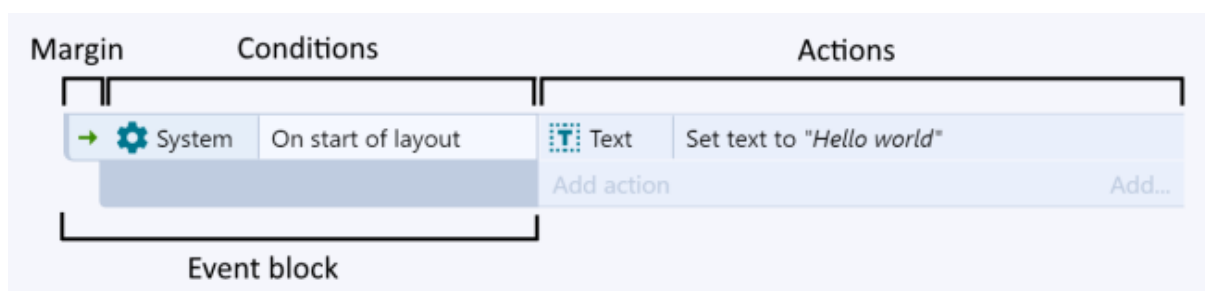
In short, an event basically runs like this:

Are all conditions met?

- Yes: run all the event's actions.
- No: go to next event (not including any sub-events).

That's a bit of an oversimplification. Construct provides a lot of event features to cover lots of different things you might need to do. However, for now, that's a good way to think about it.

The following image illustrates the key parts of an event.

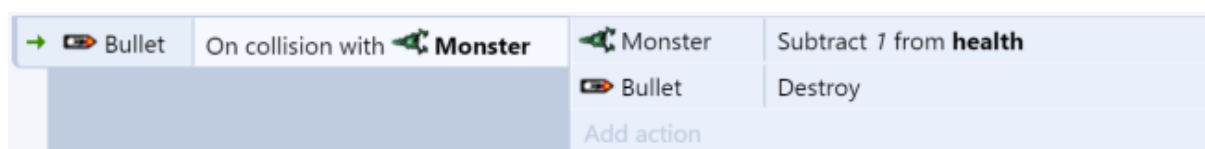


These sections are:

1. The event block, which contains the conditions. Notice the margin to the left of the condition which allows you to select the entire event.
2. The conditions, which are each listed inside the event block.
3. The actions, which are listed to the right of the event block.

Conditions and actions can be selected by clicking on them. The entire event can be selected (which also selects all its conditions and actions) by clicking the event margin, or the bottom part of the event block. The event margin can also be right-clicked to access a menu allowing things like adding conditions or sub-events.

Now you know what makes up an event, let's go into a bit more detail about how they work - beyond the "are conditions met? Then do this" approach. Events work by filtering specific instances that meet some conditions. The actions then run for *those instances only*. For example, consider the following event:



In this example, when a **Bullet** collides with a **Monster** the event condition is met. The specific instances of **Bullet** and **Monster** that collided in the game are "picked" by the event. Actions only run on the "picked" instances. If there are other instances of **Bullet** and **Monster** in the layout, they won't be affected by the **Subtract 1 from health** and **Destroy** actions. It would be very difficult to make good games if every bullet hurt every monster!



After an event ends, the next event begins from scratch. Its conditions will start picking from all instances again.

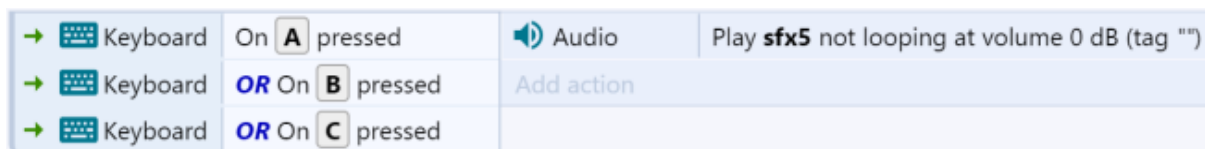
On the other hand, sub-events (which appear indented) carry on from where its parent event left off. A sub-event will further filter the instances left over by the event that came before it.

If an event has two sub-events, they both pick from the same set of instances the parent left behind - the second sub-event is not affected by the first. In other words, events at the same indentation level always pick from the same set of instances, and events at a lower indentation level are always working with the instances handed down from above.

AND/OR

As mentioned before, all conditions have to be met for an event to run. This is called a '**Logical AND**', because "condition 1 **AND** condition 2 **AND** condition 3..." must be true. However, you can change an event to run when any condition is true. This is called a '**Logical OR**', because the event will run if "condition 1 **OR** condition 2 **OR** condition 3..." are true.

Normally, event blocks work as **AND** blocks. To make an **OR** block, right-click the block and select Make **OR** block. It will then display with an **OR** between each condition, as shown below.



Be aware that because **OR** blocks run if any condition is true, it's possible the event will still run if some conditions were false and did not pick any instances. In this case, the actions will still run, but possibly with zero instances picked for any objects where no instances met the condition. If any actions are run for objects with no instances picked, nothing happens.

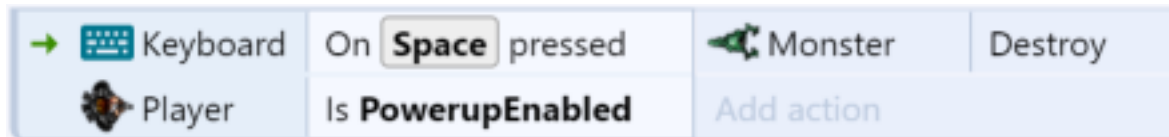
You can combine the block types by using sub-events. This allows you to build up more advanced logic, such as an **OR** block followed by an **AND** block.

EVENT ORDERING

The order of events is important. Every event is checked once per tick (about 60 times a second on most computers), and they are run from top to bottom in the event sheet. The screen is drawn once every event has been run, then the process starts again. This means if one event does something and the next event undoes it, you'll never see that anything happened.

The same applies within events: conditions are checked from top to bottom, and the actions run from top to bottom.

However, triggers are an exception. Notice that there is a green arrow to the left of **Keyboard: On Space pressed** in the following example:



This indicates the event is triggered. Rather than running once per tick, this event simply runs (or "fires") upon something actually happening. In this case, the event runs when the user hits the Spacebar key on the keyboard. It is never checked any other time.

Since triggers run upon an event happening, they aren't checked in top-to-bottom order like other events. This means the ordering of triggers relative to other events is not important (except relative to other triggers of the same type, since triggers still fire top-to-bottom).

There can only be one trigger in an event because two triggers cannot fire simultaneously. However, multiple triggers can be placed in **OR** blocks.

CREATING EVENTS

There are a number of ways to add a new event:

- Double-click a space in the event sheet, or right-click in a space to see a menu of things to add
- Click the Add event link which comes after the last event in a sheet or group or click the Add... link on the right
- Right-click an event's margin and choose an item from the Add menu

When you add a new event, the dialog that appears is for adding the first condition. To add more conditions to an event, right-click the margin or an existing condition and select **Add another condition** or use the Add... link on the right of the **Add action** link.

Actions can be added by clicking the Add action link (if it has not been hidden in the ribbon), or right-clicking the margin or an existing action and selecting Add another action.

MODIFYING EVENTS

Double-click or select and press Enter on condition or action to edit it.

Events, conditions and actions can be dragged and dropped around the event sheet. Holding Control and dragging will duplicate the dragged event, condition or action. Event items can also be cut, copied and pasted.

As with the Layout View, multiple selections can be made by holding Control and clicking different items. However, you can only have either events, conditions or actions selected at a time (e.g. you can't have both a condition and action selected at once). You can also hold Shift and click an event, condition or action to select all the items in a line between the selection and clicked item.

Press R or right-click and use the Replace object option to quickly swap objects referenced in the selection. Note that objects with references to instance variables or behaviors in the selection can only be swapped with other objects with the same instance variables and behaviors which have the same names and types.



You may find it convenient to organise events into groups, which can also be activated and deactivated as a whole. Groups and any comments you include in your event sheet can also have their text and body colours changed to help with organisation.

SCROLLING AND SCALE

There are several ways to scroll in the Event Sheet View:

- The vertical scrollbar to the right of the view
- Scrolling the mouse wheel
- Middle-clicking to pan the view
- Pressing Space, up/down arrows or page up/down

There are some options to adjust the text size in this view as well:

- Hold Control and scroll the mouse wheel
- Press Control and + or -
- Right-click and use the Event sheet → Font size menu
- Use the browser's zoom feature, but note this will scale the whole of Construct, not just the text scale in the Event Sheet View.

FINDING IN EVENTS

In the full version of Construct 3, you can search for some text in an event sheet by pressing Ctrl + F or right-clicking and selecting **Event sheet → Find...** This opens a dialog that allows you to enter text to search for, with options to look in the current sheet or the entire project, and whether to make it a case-sensitive search. (Case sensitive searches count uppercase and lowercase characters as different, e.g. "SPRITE" and "sprite" would be considered different in a case-sensitive search.) When you click **Find**, the results are displayed in the **Find Results Bar**.

However, a text search is not always appropriate for finding in events. For example, if you want to find all events referring to an object named Sprite, searching for the text *Sprite* will also return results for other names like *Sprite2*, since they also include the search term. To solve this, you can use the **Find all references** feature. This is available in many places in Construct for various kinds of things like behaviors and instance variables as well. For objects, you can right-click an object in the Project Bar and select **Find all references**. This will open the Find Results Bar with a comprehensive and precise list of all references to that object, excluding any other references that happen to include the object name. This is a great way to easily review your project with confidence the results are what you want.



Bars

Project

The Project Bar shows an overview of everything in your project. Everything is pre-arranged into folders according to object type, like Layouts, Sounds, Files etc.

If something in the project has changed, it is displayed in *italics*. When you save the project, everything reverts to normal text, indicating nothing has changed since the last save.

To view the properties of your current project, select the project item - the item at the top of the Project Bar with the name of the project. When you select it, the Properties Bar displays properties affecting the whole project. You can also right-click the project item to show a list of options and submenus.

ORGANISING PROJECTS

You can hold Control or Shift to select multiple items and drag them into a folder at the same time. However, you can only organise items into folders of the same type, e.g. you can't drag an event sheet into a layout folder.

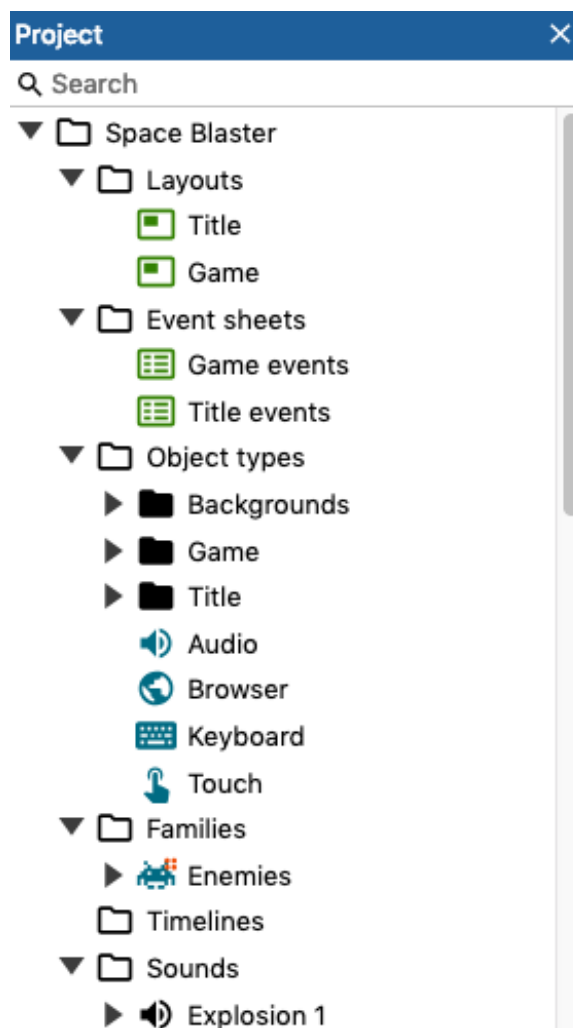
You can also further organise your project by using Construct 3's subfolder system – available in the full version. Subfolders can be added by right-clicking a folder and selecting Add subfolder. Then, you can drag and drop folders and items to organise them into the right folders.

MANAGING ITEMS IN THE PROJECT

Right-click any item in the Project Bar to show a list of options. Most items can be renamed and deleted. This is the only way to fully remove an object type from a project – if you delete something from the Layout View, it will simply remove that instance, not the whole object type.

Right-clicking a folder also has the option to add a new item to that folder, such as a new layout or event sheet. Objects are more commonly added in the Layout View, but you can still add them from the Project Bar too.

There is also a search box located at the top of the Project Bar for easy project searching.





IMPORTING FILES

You can import additional external files to the project, including web fonts. These can be categorised into Sounds, Music, Videos, Fonts, Icons or the general-purpose Files folder.

To import audio files, for example, right-click either the Sound or Music folders and select Import sounds or Import music to open the Import Audio dialog. This allows you to pick audio files from your computer and import them to the project. Construct will convert them to the necessary format to support all browsers. To play back audio in your game, make sure you add the Audio object to the project. Most other files can be imported in a similar way.

If you have the same audio file in multiple formats, it appears as a single item in the Project Bar to save space. However, it can still be expanded to show the different versions of the file. For example, if you have sfx.m4a and sfx.webm in your project, the Project Bar will show a single item named sfx, with two child items for sfx.m4a and sfx.webm.

PREVIEWING FILES

From the Project Bar, you can preview several kinds of files added to the project. Audio and video files can be played back. Web fonts can be previewed with a dialog showing some text using the font. SVG files can also be previewed. Other kinds of file can be viewed and edited using the full version's file editors.

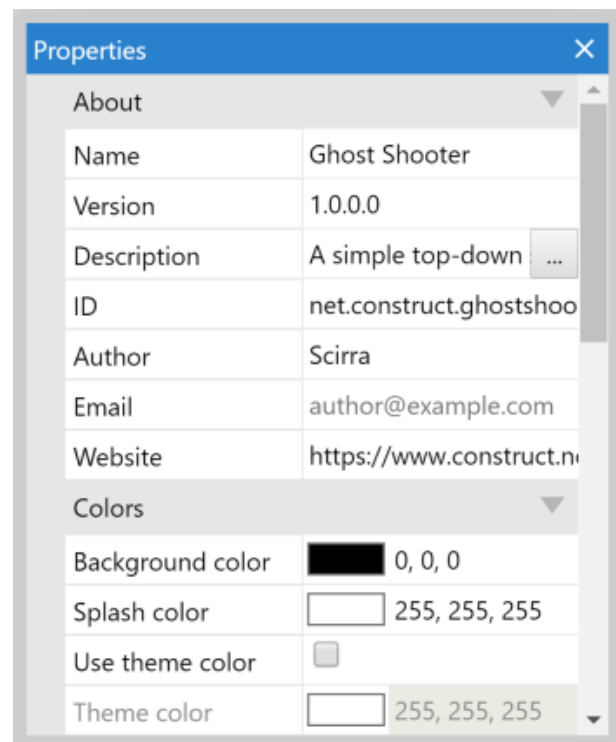
Properties

The Properties Bar is an essential part of the interface. It displays a list of all the settings you can change on whatever is selected. The picture below shows the Properties Bar displaying a project's properties.

Properties are organised into categories which can be expanded and collapsed. There are many kinds of properties, including number fields, text fields, dropdown lists and clickable links. The property name appears in the left column, and the editable value appears in the right column.

Whenever something in the project is clicked or selected, its properties display in the Properties Bar. For example, selecting objects in the Layout View or clicking items in the Project Bar shows the relevant properties in the Properties Bar.

A lot of Construct's elements have properties, including projects, layouts, layers and object instances. Many plugins, behaviors and effects have their own properties as well. There is also a





Help link displayed at the end of every property list. Click that to open the relevant manual section for those properties.

All properties also have a description which provides additional information about what the property is used for. This is displayed in a panel at the bottom of the Properties Bar – it's worth keeping an eye on this since it can contain useful hints and tips. An example is shown below.

Event sheet: Choose the event sheet to run for this layout. You can include other event sheets from this sheet.

SPECIAL FEATURES FOR NUMBER VALUE PROPERTIES

In number values, you can type calculations like **1920 / 2** and press enter to set the value to the result of the calculation (960). The syntax used is the same as expressions used in events. You can also use some basic system expressions like **sqrt(64)**.

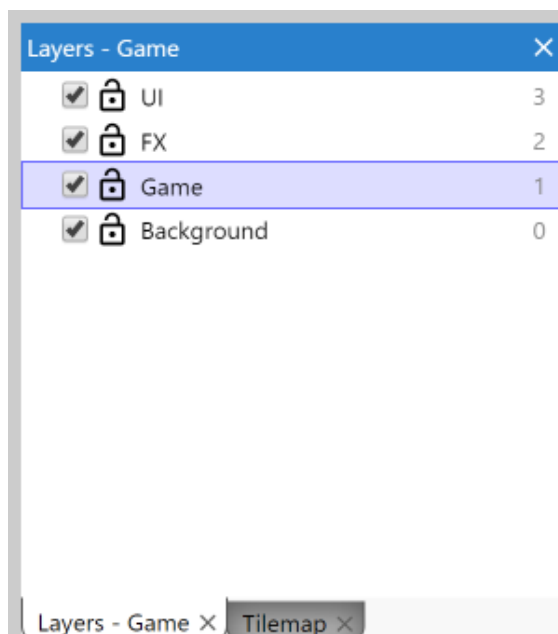
Number values can also be smoothly dragged with instant feedback in the Layout View. This is useful to try out a range of values and easily see which is best. To do this, click and drag vertically inside the number value cell. You can also hold Control or Shift while dragging to increase or decrease the rate of change. If you have trouble getting this to work, try first clicking inside the cell (which should select the text in the cell), and then click on the selected text and drag vertically.

Layers

The Layers Bar is used to add, edit and remove layers in a layout. A layer is like a sheet of glass objects are painted on to. This allows easy arrangement of which objects display in front of other objects, for example showing foreground objects in front of the background sprites. It also allows for interesting depth effects like parallax, and layers can be individually scaled and rotated as well.

Layers can be dragged and dropped in the Layers Bar to change their order. Layers at the bottom of the list are displayed at the back (e.g. background objects), and layers at the top of the list are displayed at the front (e.g. HUD objects).

Selecting a layer displays its properties in the Properties Bar and also sets it as the active layer which new objects are inserted to.





THE LAYERS LIST

Each layer in the list has the following:

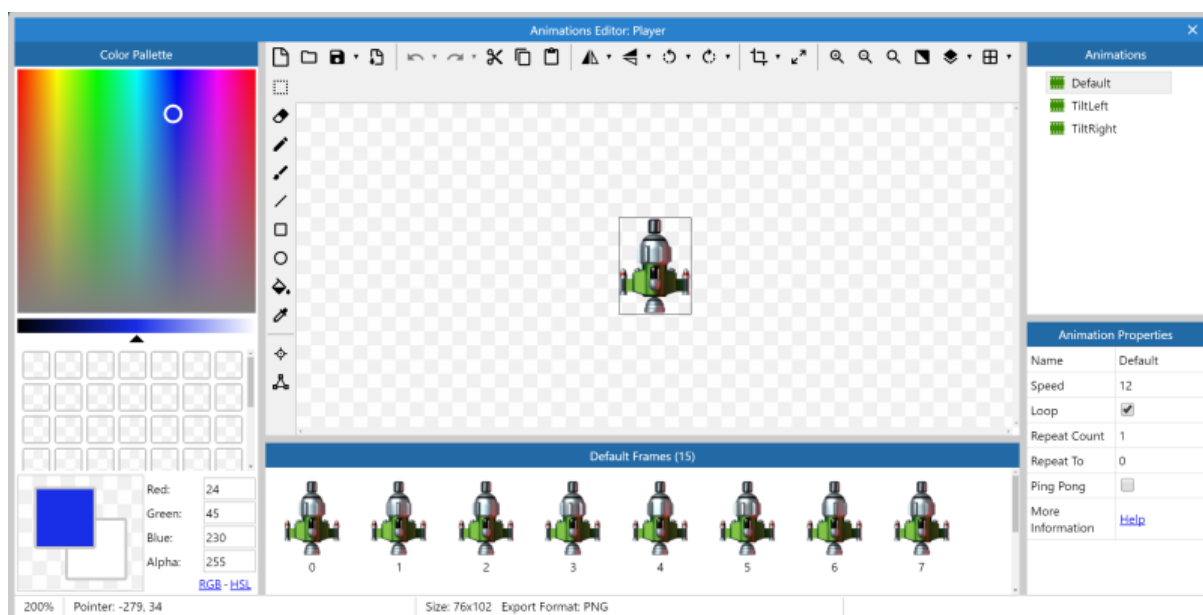
- A checkbox to toggle whether the layer is visible in the editor (this does not affect the game when previewing or exporting)
- A padlock icon which toggles the layer's locked status. If a layer is locked, objects on that layer cannot be selected in the editor. This is useful to prevent accidental selections on rarely-used layers like backgrounds.
- A number to the right which is the zero-based index of the layer (the first layer is number 0, not 1). If you need to enter a layer number in the event system, this is the corresponding number. You can also enter layer names in the event system, which is often more convenient since unlike the numbers, the names don't change if you reorder layers.

You can right-click a layer to see a menu of additional options, such as to add a new layer at the top or bottom, rename or delete layers, or shortcuts to show, hide, lock or unlock all layers (or all but the selected one).

Animations Editor

Construct 3 has a fully-featured built-in image and animations editor, used to create animations for the Sprite object. When opening for an object without animations, such as Tiled Background, the animation editing features are hidden and it acts as a normal image editor. For brevity, it is consistently referred to as the Animations Editor, even in cases where it is only editing a single image.

To open this editor, double-click an object with an image or animations in the Layout View or Project Bar.





Each pane in the Animation Editor can be resized by dragging the borders, similar to how you can with the main Construct 3 interface. This lets you customise the layout of the Animation Editor.

COLOUR PALETTE

The colour palette appears on the left and allows a colour to be picked for the drawing tools. You can choose both a primary and secondary colour with left and right-click. The pane also has a number of cells that can be used to remember a set of colours. Right-click a cell to save or use the primary or secondary colour. By default, left-clicking the cell will set the primary colour.

You can use colour codes to specify a colour too – these can in any of the following formats:

- r, g, b or r, g, b, a
- rgb(r, g, b) or rgba(r, g, b, a)
- hsl(h, s, l) or hsla(h, s, l, a)
- Hex as either #ffffff or fffffff

IMAGE TOOLS

The top toolbar in the image pane provides tools that affect the entire image, such as mirroring and flipping. If you hover over an icon on the toolbar, it will show you the name of the tool and an associated shortcut if it has one.

In order, from left to right, the tools available are:

Clear Image: this resets the image to a transparent box.

Open: imports an image from a local file. You can also choose SVG files, but these will be rastered into a bitmap at a given size.

Save: exports a copy of the current image. In the browser, this downloads the current image as a PNG file. You can use the dropdown next to the button to save the current animation, or all animations, bundled in a zip file.

Set export format: this opens the *Image Format dialog*, allowing you to choose whether the image is saved as PNG or JPEG when the project is exported. This can also be applied to the current animation or all animations. Bear in mind that Construct stores all images in the project in a lossless PNG format; images are only converted on export.

Undo and Redo: allows you to undo or redo actions – use the drop-down arrows next to each button to select multiple actions to undo/redo.

Cut, Copy, Paste: performs clipboard operations with the image.

Mirror and Flip: inverts the image on one of its axes. Mirror inverts horizontally, while flip inverts vertically. Use the dropdown next to the button to affect the entire animation.

Rotate: rotates the image in 90° either clockwise or anti-clockwise depending on which tool you use. Use the dropdown next to the button to affect the entire animation.

Crop: automatically removes transparent space around the edges of the image. This is a good idea to save memory. This action leaves a 1px transparent border to improve the



image quality at the edges. Use the dropdown next to the button to affect the entire animation.

Resize: resizes the current image. A dialog will open with options for the resize, including a checkbox to apply the resize to the whole animation.

Zoom options: the three magnifying glass type icons are your zoom controls – zoom in, out and reset zoom. These adjust the zoom level in the image editor, or you can use Control + mouse wheel.

Toggle background brightness: switch between a light and dark background for the image editor. Changing to a dark background can be useful when editing very light images.

Toggle onion skin: this feature is only available in the full version of Construct 3 and allows you to display adjacent frames translucently over the current image when editing an animation. This can help when drawing animations. The options are to display the previous frame, next and previous frames, or next and previous two frames over the current image.

Grid: toggle the display of a grid over the current image. Use the dropdown next to the button to adjust the grid settings, such as the grid size, colour and whether to snap to the grid.

DRAWING TOOLS

The side toolbar provides some tools for drawing in the image, as well as some extra Construct-specific tools for setting image points and adjusting the collision polygon. Some tools have extra settings, such as the size for the brush tool, which appear underneath the top toolbar. The following tools are available, in order from top to bottom:

Rectangle select: allows you to select rectangular sections of the image. You can then move, delete, cut, copy and paste them.

Pencil: draws with a solid square, useful at 1px size for pixel art.

Brush: draws with a soft round brush.

Line: draws straight lines. Hold shift to lock the angle to 5° increments.

Rectangle: draws a rectangle in the image, using the secondary colour as the border colour. Hold shift to draw a square.

Ellipse: draws an ellipse in the image. As with the rectangle tool, the secondary colour is used as a border. Hold shift to draw a circle.

Fill: fills a continuous area of the image with a colour. You can also choose to enable or disable flood fill to alter how the editor manages filling.

Eye dropper: selects a colour from the image. Alternatively, hold Control and click with another tool selected.



Image points: this button displays the origin and image points for your image and allows you to edit them. This switches the colour palette pane to a list of image points, allowing you to add new ones, or remove any you don't want. The origin determines the rotation point of the image and is where the X and Y coordinates of the object are aligned to. Image points can be used in the event system to refer to alternative positions. For example, you may place an image point at the end of a gun, so spawned bullets can appear at the end of the gun barrel instead of at the origin.

Collision polygon: This allows you to adjust the collision polygon – that is, the area that counts as colliding for this image. By default, Construct guesses a collision shape, but it is not always accurate, so you can click and drag the points of the collision polygon to alter its shape as you need to. Right-click on a point or in a space to display a menu of additional options for the collision polygon, such as adding and deleting points. Some objects, like Tiled Background, do not use collision polygons.

FRAMES PANE

The bottom pane displays a list of all frames in the current animation. You can add or delete frames from here or drag and drop them to alter their sequence. If you right-click a frame you have the options to duplicate or delete that frame. You can also adjust the size of the frame icons appearing in the pane by adjusting the Thumbnail size.

If you want to edit the image of a different frame, simply select it in the Frames pane. This also shows a single property in the properties pane: Duration, which is a multiplier for the amount of time to spend on the frame. For example, a frame duration of 2 will spend twice as long on that animation frame, 0.5 half as long, etc. relative to the current animation speed.

Right-click in an empty space in the pane to see additional options for managing the animation, which include:

Add frame: add a new empty frame at the end of the animation

Duplicate last: add a new frame which is a copy of the last frame in the animation

Reverse frames: invert the sequence of frames in the animation

Import frames

From files: add multiple animation frames by selecting a set of local image files to import

From strip: add multiple animation frames by selecting a local image file with multiple images placed on it (often called a *sprite strip*), and cutting it up into individual images. You must specify the number of cells horizontally and vertically, and the direction to read cells in.

ANIMATIONS AND PROPERTIES PANES

The right side of the Animations Editor shows the Animations pane, where animations can be added, edited and deleted. If you right-click in a space, you can add a new animation or add a



subfolder to organise animations. Whereas if you right-click on an animation itself, you'll see options like Preview, which shows how the animation will look in the game. If you're using the full version of Construct 3, you can also use **Find all references** for an animation to locate all its references in events.

Selecting an animation also switches which frames are showing in the frames pane and displays settings for the animation in the Properties pane. The following properties are available for animations:

Name: the name of the animation. This can also be directly edited in the Animations pane.

Speed: the rate at which to play the animation, in animation frames per second. For example, if set to 5, each frame will last 1/5th of a second. Remember that this cannot be faster than the game framerate, which is typically 60. Set the speed to 0 if you do not want the animation to play, which is useful if you want to control which frame is showing by events. You can also use negative speeds, which causes the animation to play backwards. If you do choose to do this then repeating animations should set the *Repeat to* frame at the *end* of the animation, otherwise, by default, it repeats to frame 0 (the start of the animation), causing the animation to stop after playing in reverse.

Loop: enable to infinitely repeat the animation.



Repeat count: if the animation is not looping, the number of times to repeat the animation.

Repeat to: the zero-based index of the animation frame to go back to when the animation loops or repeats.

Ping pong: play the animation alternately forwards and backwards when looping or repeating.

IMAGE POINTS PANE

When you select the Image points tool in the image editor, the left pane switches to a list of image points for the current animation frame.

| Image Points | | |
|---|--------------|--------|
| | Name | Number |
|  | Origin | 0 |
|  | Imagepoint 1 | 1 |

The Origin is a special kind of image point defining the centre of the object, or its point of rotation. It has a different icon to denote it. The term *image point* usually means "image points including the origin". Image points have a zero-based index, and the first image point (number 0) is always the origin. The origin cannot be renamed.

You can also add additional image points.

These are useful to create spawn points for other objects. Since you can create objects at image points in events, it is often useful to place an image point in places like the end of the player's gun in the image. Image points can also be given a name and referred to in events by this name.



EDITING IMAGE POINTS

Select an image point in the list and a corresponding point will appear on the image. Left click to place the point under the mouse. The arrow keys can also nudge it 1 pixel in each direction.

An image point can be quickly placed using the number pad, e.g. 1 for the bottom-left corner or 5 for centred. Alternatively, the image point can be right-clicked in the Image Points pane and an option chosen from the quick assign menu.

Right-clicking an image point in the Image Points pane also provides **Apply to whole animation** and **Apply to all animations** options. These set the image point in the same relative place in all frames in the current animation, or all frames in all animations, respectively. If an image point does not exist in all frames, this option also creates it. Holding shift while placing the image point is a shortcut for this.

DRAG-AND-DROP IMPORTING TO THE ANIMATIONS EDITOR

There are various ways you can import images by drag-and-drop into the Animations Editor window.

An image file can be dropped into the main image editing pane to replace the content of the current frame with the dropped image file. This works the same way as using the Open button on the top toolbar.

Dropping a single image file into the Frames pane will prompt you asking how the image should be treated. The image can be treated as a plain image file, in which case the image is added as a new frame in the current animation. This works the same way as the context menu option **Import → From files**. The image can also be treated as a sprite sheet, which works the same way as the context menu option **Import → From strip**.

Dropping multiple image files into the Frames pane will add a new frame for each dropped image file. This works the same way as the context menu option **Import → From files**.

You can drag-and-drop either a single image file or multiple image files into the Animations pane, and it is handled in the same way as with the Frames pane, except that the frames are added to a new animation.



Saving/Sharing

In Construct 3, there are three main places you can save your work, each of which we'll cover in more detail shortly. By default, pressing Save on a new project will save with the Cloud Save option. You can select a different option, using the Save As menu.

In Construct 3's settings, there are a few options for Saving and Backups. Here you can change Construct's default saving option if you don't want to always save to the cloud.

Save & backup

Default save location

Periodically back up active project

Backup location

Backup interval (minutes)

You can also configure C3's built-in backup system from this part of the settings menu. You can choose whether or not you want Construct to make backups for you, as well as how frequently you want them made, and where you would like them saved.

We would recommend that you back up your projects regularly to avoid losing too much work when things do go wrong. We would also recommend having multiple backups in different places to protect against things like hard drive failure or having your cloud accounts hacked.

CLOUD SAVE

You can save your work to a cloud storage service, allowing you to access your work wherever you go. Since Construct 3 runs in the browser and can be used on any device, this is a great way to ensure you can carry on from where you left off no matter which device you end up using. Many cloud storage services also provide built-in backups and file histories, helping ensure your work is safe even in the face of disaster.

Construct 3 currently supports Google Drive, Dropbox and Microsoft OneDrive. The first time you select **Menu → Project → Save as → Cloud save**, a dialog will appear asking you to choose one of the supported services. When you choose one, you'll be prompted to log to your cloud storage account, so Construct 3 has permission to save and open files from your account. Once you've entered your details they will be remembered, so you can keep using Cloud Save without having to keep entering your details.

When you press Save with a Cloud Save project, Construct 3 will save your project and upload it to your cloud storage account. The upload will continue in the background showing the upload



status in the corner of the window, allowing you to continue working on your project. Bear in mind you cannot save again until the upload completes.

Next time you use Construct 3, you can choose **Cloud open** under the Open menu to find your project again. It'll also appear in the Recent Projects section of the Start Page.

SAVE TO LOCAL BROWSER

Modern browsers allow websites to save data to your local system. This is how they remember things like your login details and settings. Construct 3 can save projects to the local browser's storage. This storage is unique to both the specific device and browser, so, for example if you save a project to browser storage on a specific laptop with Chrome, you can only find it again by using the same browser (Chrome) on the same device (that specific laptop).

By default, browsers may in some circumstances delete storage (e.g. if it uses too much space, is running out of disk space, or if the data is particularly old). To prevent your saved projects from being deleted, the first time you select **Menu → Project → Save as → Save to local browser**, Construct 3 will ask you to give permission to use persistent storage. This kind of storage is not subject to deletion by the browser, ensuring your work can be permanently saved. The browser may prompt you with a permission request to persist storage which you must accept in order to use local browser storage. You can also check the status of the persistent storage permission, as well as how much space the browser is allowed to use and how much it is using, in the About dialog.

Next time you use Construct 3, you can choose **Menu → Project → Open from local browser** to find your project again. It'll also appear in the Recent projects section of the Start Page. But remember you must use the same browser on the same device you saved with in order to access the project.

USING LOCAL FILES

For security reasons, browsers can't write files directly to your device. However, you can select **Menu → Project → Save as → Download a copy** to download your project as a local file. Construct 3 will ask if you want to change the downloaded filename; you can leave it empty to use the default. Construct 3 projects use the .c3p file extension.

Normally the file will go to your Downloads folder, but you may also be prompted to save to a different location depending on the browser – you can alter this setting in most major browsers. You can also choose a new default download location in most browsers too. Alternatively, you can usually drag-and-drop the resulting file directly out of the browser, such as from Google Chrome's downloads footer section.

Despite the name, this does not actually download a file from the Internet – all the project data is stored locally. The term *Download* refers to invoking the browser's download UI to save your project to a local file.

To open a local file, simply choose the **Menu → Project → Open local file** option. This will open a file picker to choose a file on the local system. Alternatively, you can simply drag-and-drop a



.c3p file to the Construct 3 window, and it will open. The .c3p file may also be named .zip, or be contained within a .zip file, and Construct 3 will still open it. This can be useful for some services which do not recognise or allow the .c3p file extension but allow zip files.

Bear in mind, that for the same reasons that browsers can't write files directly to disk, they also can't read files directly from your device. This means local files do not appear in the Recent Projects section of the Start Page.

SHARING PROJECTS

Sharing your project or game is a great way to get feedback on your work – it can usually help to have an outside opinion!

The easiest and quickest way to share your project, especially during development, is using Remote Preview. We'll cover this feature in more detail in the Testing Your Game chapter, but a quick explainer for Remote Preview – it allows you to preview your project on a different device by either scanning a QR Code or visiting a specific URL.

But Remote Preview isn't always the best option (it's also not available in the free edition.) Let's say you've hit a wall with your development, and you need a bit of help solving a problem. You might need to send someone your project for them to have a look at. In this case, you can either:

- Use Cloud Save, and use the storage service's sharing features
- Use the *Download a copy* option to save a local file and share that.

If you want people to play your finished project, you should export it, which produces a playable game ready for publishing on a given platform. Exporting projects will be covered in more depth in a later chapter.



Making Your First Game

Part 1 – Getting Started

Let's get started creating your first game. We'll be making the Ghost Shooter demo game, which you can access from the Construct 3 Start Page under Beginner's Templates. This is a simple top-down shooter game, designed to be used with a keyboard and mouse.

It's worth looking at this project and maybe previewing it before you dive into this tutorial so you can get a feel for the game and see what you're aiming for!

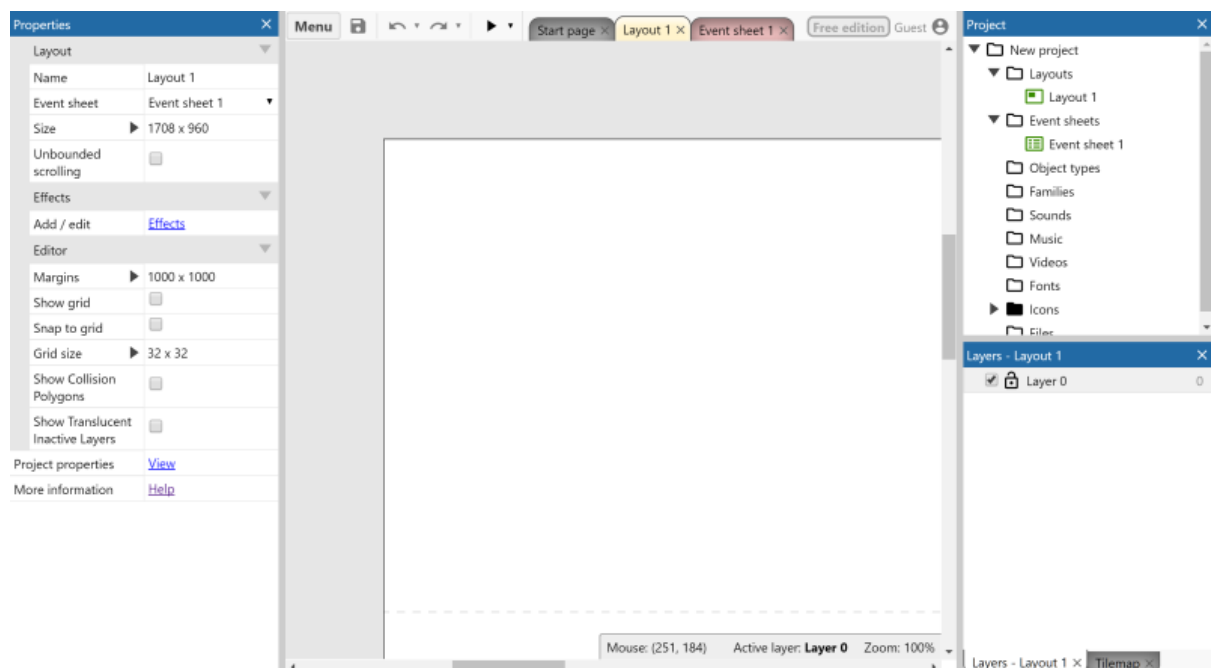
You will also need some assets for this game. You can save them by exporting them using the animations editor in the example project, by saving them from the online version of this tutorial, or you can download the asset pack from here.

Alternatively, you can make your own using the animations editor, or download something from an asset website – OpenGameArt and Kenney.nl are good starting points! You'll need:

- Something to use as a tiled background
- A player character (referred to in this as **Player**)
- An enemy character (referred to as **Monster**)
- A bullet or projectile (referred to as **Bullet**)
- An explosion (referred to as **Explosion**)

CREATE A NEW PROJECT

Click the New project button. A dialog will appear asking for some details. You don't have to change anything, but you can type in a name for your project if you like (how about *My super awesome game?*) Click Create, and you should see a new empty project, something like this.



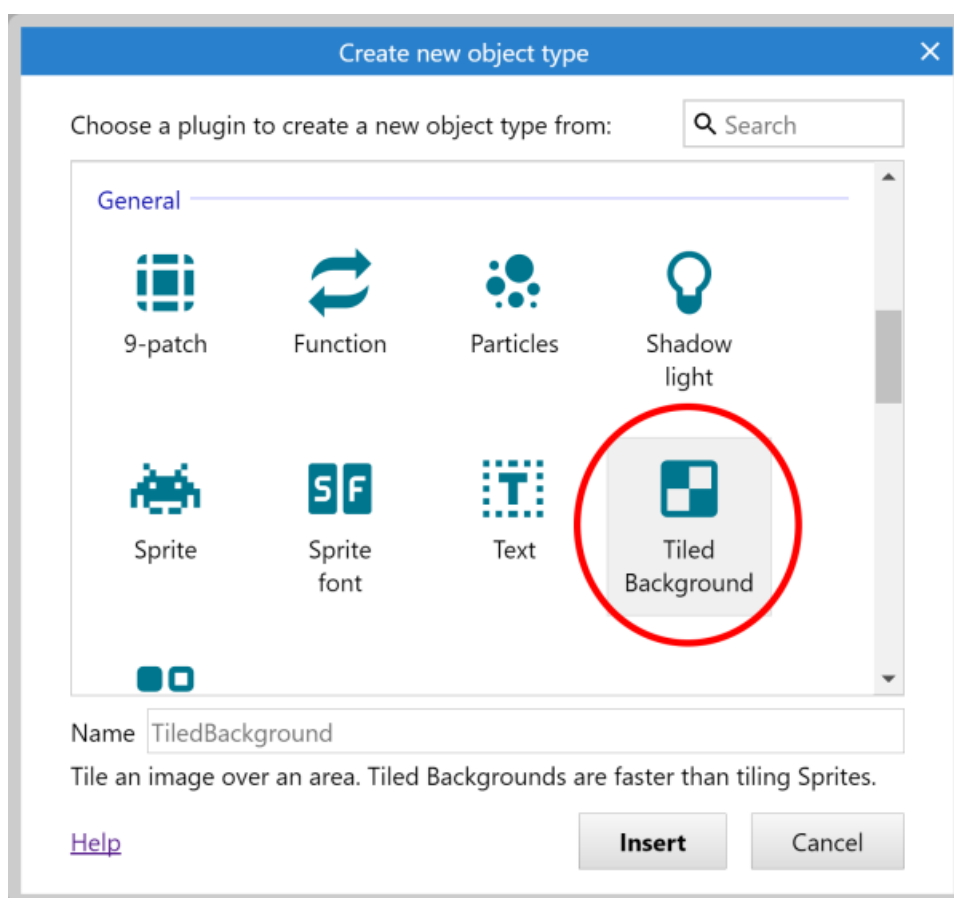


ADDING OBJECTS

Now you have a project, you'll need to start adding the objects you need to make the game. The first will be the **Tiled Background** object.

An easy way to make a background is to repeat an image over the layout, Tiled background can do this for you.

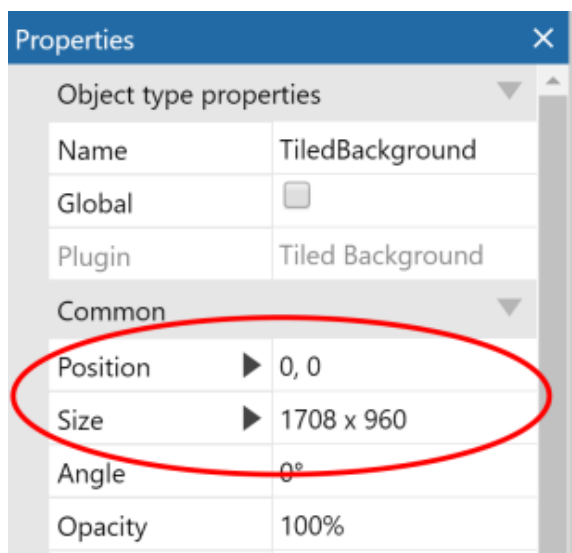
To add a new object, double-click an empty space in the layout – this will bring up the **Create new object type** dialog. (If your layout is full, you can also right-click and select *Insert new object*.) Once the Create new object dialog appears, double click the Tiled Background object. You can use the search box in the top right-hand corner to find specific objects if you don't want to scroll through the list!



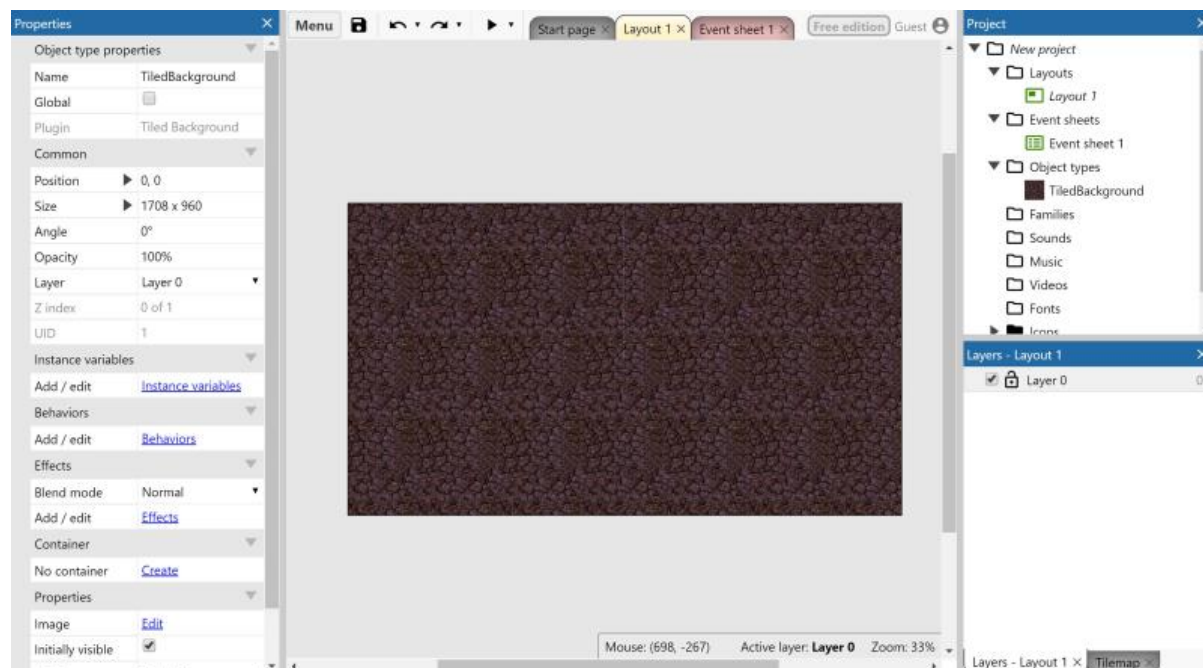
The mouse will turn to a crosshair for you to indicate where to place the object. Click somewhere near the middle of the layout. The image editor will then open for you to draw or import the image to tile. If you're going to use an external image, then click the folder icon to load an image from your computer, find your chosen file and select it.



Close the image editor by clicking the X in the top-right. Now you should see your tiled background object in the layout. Let's resize it to cover the entire layout. Make sure it's selected, then the *Properties Bar* on the left should show all the settings for the object, including its size and position. Set its position to 0, 0 (the top left of the layout), and its size to 1708 x 960 (the size of the layout - this is the default size, based on twice the size of the viewport).



Your tiled background should cover the entire layout now, and if you're using the assets from the online version of this tutorial, your layout should look something like this:



If you click the Preview button in the main toolbar - the preview window should pop up showing your tiled layout. This is a quick way to see how your game will look as you're going along!

Part 2 – Adding More Objects

Before you continue, the tiled background should be locked. As you create and move around objects on top of it, it's easy to accidentally select or modify the background. Since you don't need to change the background any more, locking it makes it unselectable, so it won't get in the way. To lock it, right-click on the tiled background and select Lock → Lock selection. (If you do want to change it later on, simply right-click and select Lock → Unlock all.)



ADD THE INPUT OBJECTS

Double-click in a space (this can be anywhere since the tiled background is locked) to add another new object. This time select the **Mouse** object, since you'll need mouse input. Do the same again for the **Keyboard** object.

Note that these objects don't need placing in a layout. They are hidden, and automatically work across the entire project. Now all layouts in our project can accept mouse and keyboard input.

THE GAME OBJECTS

It's time to add our game objects! You'll need your player character, enemy character, bullet and explosion images for this. For each of these images, we will add a Sprite object.

Sprites simply display an image, which you can move about, rotate, resize and optionally animate. Games are generally composed mostly out of sprite objects.

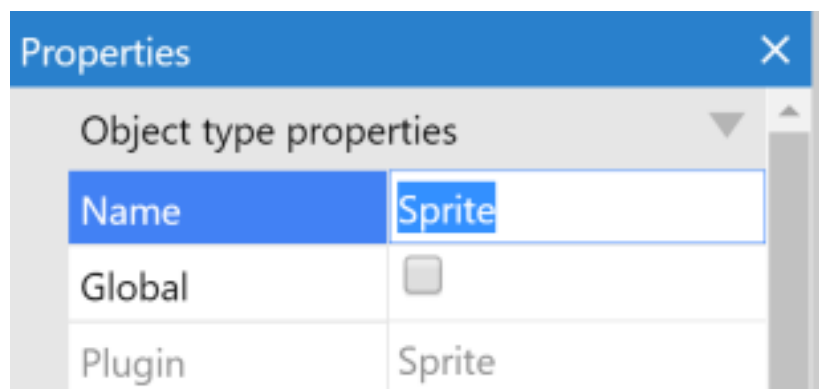
Let's add each of the above four objects as sprite objects. The process is similar to inserting the Tiled Background:

1. Double click to insert a new object
2. Double click the Sprite object.
3. When the mouse turns to a crosshair, click somewhere in the layout to place it
4. The image editor pops up. Click the *Load image* button and load one of the four images.
5. Close the image editor. You should now see the object in the layout!

As mentioned in the chapter covering the Layout View, a quick way to create sprite objects is to drag and drop the image file to the layout view, then Construct will create a Sprite with that image for you.

These objects will be called **Sprite**, **Sprite2**, **Sprite3** and **Sprite4**. That's not very useful - things will quickly get confusing like this. Rename them to something like **Player**, **Monster**, **Bullet** and **Explosion** as appropriate.

You can do it by selecting the object, then changing the Name property in the properties bar:





Be sure to drag each image in one at a time though - if you drag all four files in at once, Construct will make a single sprite with four animation frames.

You can place the **Player** and **Monster** sprites anywhere on the layout. But you'll need to move the **Bullet** and **Explosion** sprites to somewhere off the edge of the layout - we don't want to see them when the game starts.



Part 3 – Adding Behaviors

Behaviors are quick ways to make an object act a certain way. For example, you can add a *Platform* behavior to an object, and the *Solid* behavior to the floor, and you instantly can jump around like a platformer game. You can usually do the same in events, but behaviors are much quicker! Construct has a wide range of behaviors, but here are a few that we'll use in this tutorial.

8 Direction movement: this lets you move an object around with the arrow keys. It will do nicely for the player's movement.

Bullet: this simply moves an object forwards at its current angle. It'll work great for the player's bullets. It'll also work nicely to move the monsters around - since all the movement does is move objects forwards at some speed.

Scroll to: this makes the screen follow an object as it moves around (also known as scrolling). This will be useful to keep the view centred on the player.

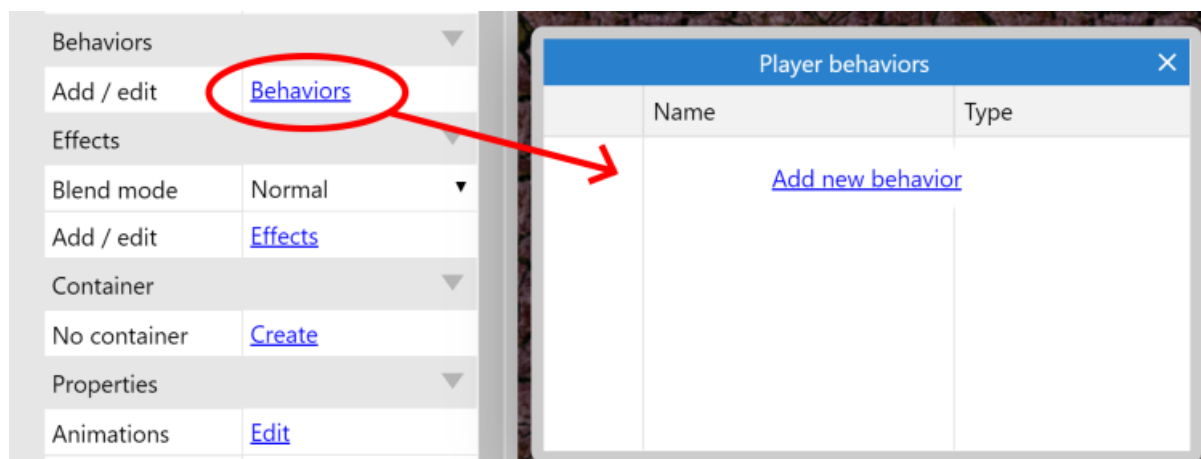
Bound to layout: this will stop an object leaving the layout area. This will also be useful on the player, so they can't wander off outside the game area!

Destroy outside layout: instead of stopping an object leaving the layout area, this destroys it. It's useful for our bullets. Without it, bullets would fly off the screen forever, always taking a little bit of memory and processing power. Instead, we should destroy the bullets once they've left the layout.

Fade: this makes an object fade out, which we will use on the explosions.

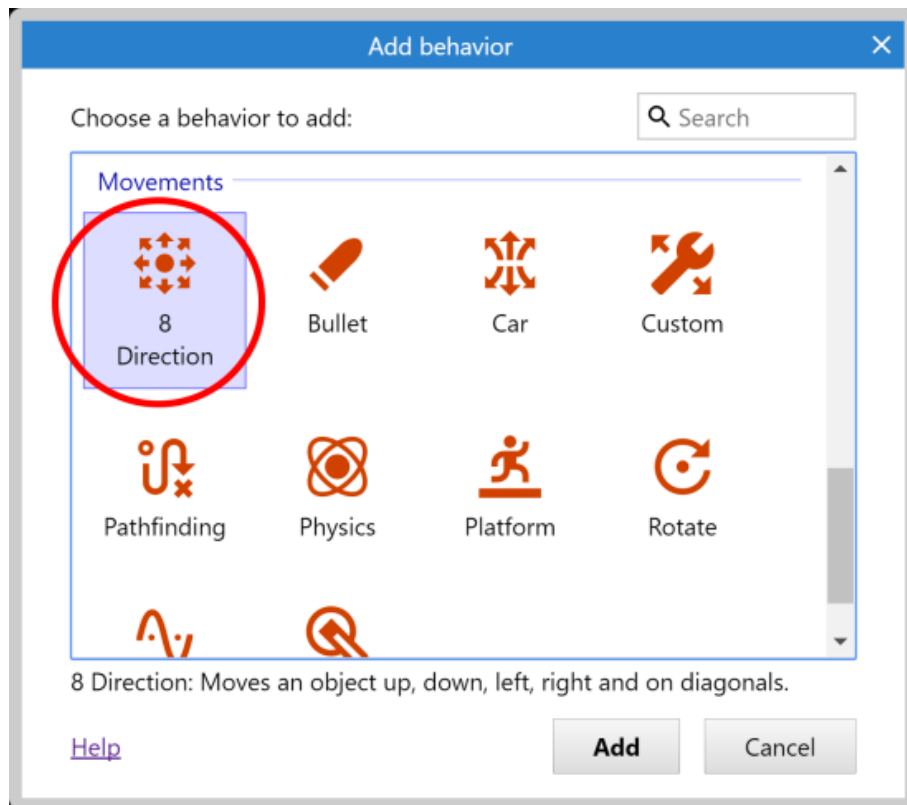
HOW TO ADD A BEHAVIOR

Let's add the 8 Direction movement behavior to the player. Click the **Player** object to select it. In the Properties Bar, you should see the Behaviors category. Click the Behaviors link there, and the Behaviors dialog for the **Player** will open.

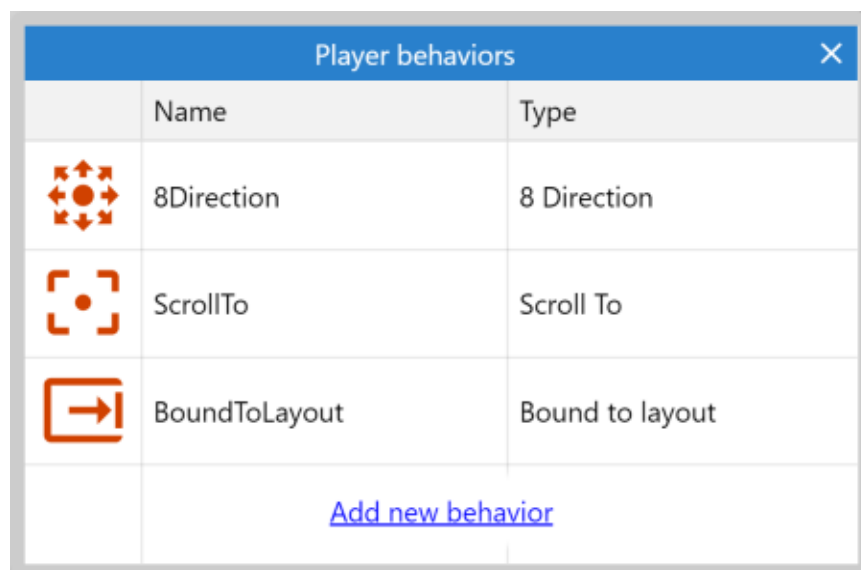




Click Add new behavior in the behaviors dialog, then double-click the **8 Direction** icon to add it.



Do the same again and this time add the **Scroll To** behavior, to make the screen follow the player. Then also add the **Bound to Layout** behavior, to keep them inside the layout. The behaviors dialog should now look like this:



Close the behaviors dialog and try pressing Preview to run the game so far! Once the preview starts, you'll find that you can already move around with the arrow keys, and the screen follows



the player. You also can't walk outside the layout area, thanks to the Bound to Layout behavior. This is what behaviors are good for - quickly adding common features.

ADDING THE OTHER BEHAVIORS

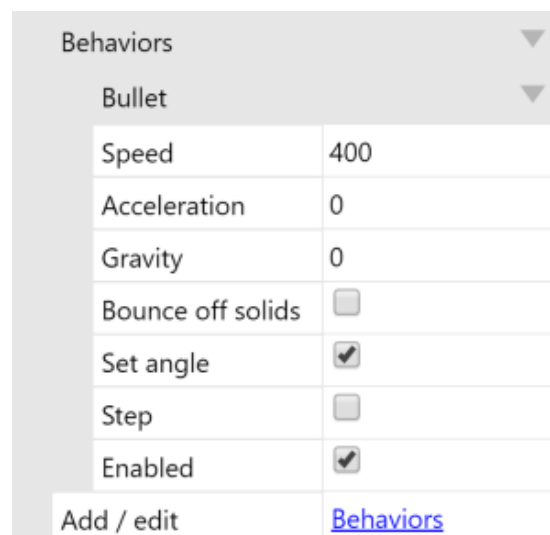
We can add behaviors to the other objects by the same method - select it, click the *Behaviors* link to open the behaviors dialog and add some behaviors. Let's add these other behaviors:

1. Add **Bullet** and **Destroy outside Layout** to the **Bullet** object (no surprises there)
2. Add **Bullet** to the **Monster** object (because it just moves forwards as well, only at a slower speed)
3. Add the **Fade** behavior to the **Explosion** object (so it gradually disappears after appearing). By default, the Fade behavior also destroys the object after it has faded out, which also saves us having to worry about invisible explosion objects clogging up the game.

If you run the game, you might notice the only thing different is any monsters you can see suddenly shoot off rather quickly. Let's slow them down to a leisurely pace. Select the **Monster** object. Notice how since we added a behavior, some additional properties have appeared in the properties bar.

This allows us to tweak how behaviors work. Change the speed from 400 to 80 (this is in pixels travelled per second).

Similarly, change the **Bullet** object's speed to 600, and the **Explosion** object's **Fade** behavior's Fade out time to 0.5 (that's half a second).



CREATE SOME MORE MONSTERS

Holding Control, click and drag the **Monster** object. You'll notice it creates another *instance*. This is simply another object of the **Monster** object type.

Object types are essentially 'classes' of objects. In the event system, you mainly deal with object types. For example, you might make an event that says "**Bullet** collides with **Monster**". This actually means "*Any instance of the **Bullet** object type collides with any instance of the **Monster** object type*". As opposed to having to make a separate event for each and every monster. We'll cover more on object types vs instances later. For now, a good way to think about it is: *different types of enemy are different object types, then the actual enemies themselves (which there might be several of) are instances of those object types.*

Using Control + drag, create 7 or 8 new monsters. Don't place any too close to the player, or they might die straight away! Remember you can zoom out with Control + mouse wheel down if it helps and spread them over the whole layout. You should end up with something a bit like this.

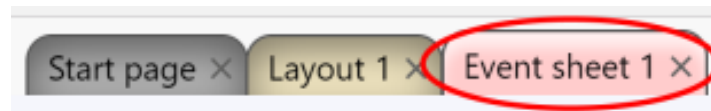


Now it's time to add our custom logic with Construct's visual method of programming - events!



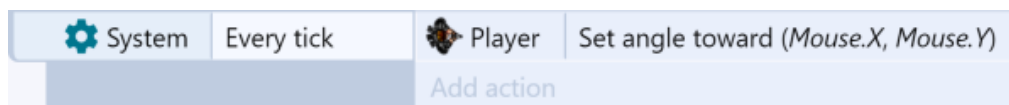
Part 4 – Adding Events

First, click the Event sheet 1 tab at the top to switch to the *Event Sheet View*. A list of events is called an *Event sheet*, and you can have different event sheets for different parts of your game, or to help with organisation. Event sheets can also "include" other event sheets, allowing you to reuse events on multiple levels, for example, but we won't need that right now.



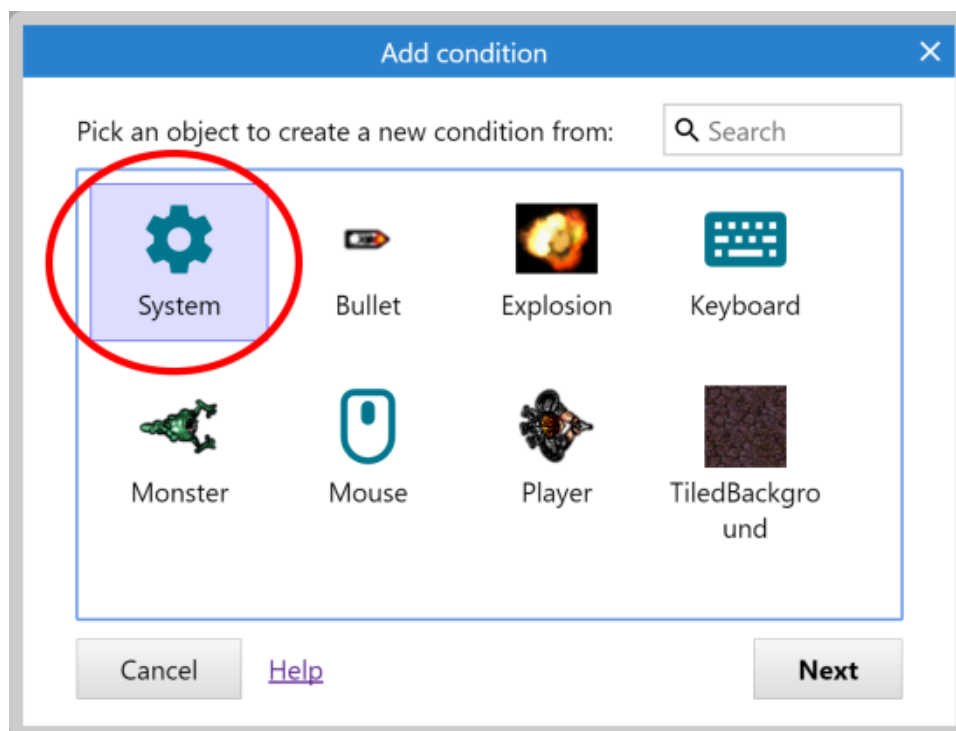
YOUR FIRST EVENT

We want to make the player always look at the mouse cursor. It will look like this when we're done:



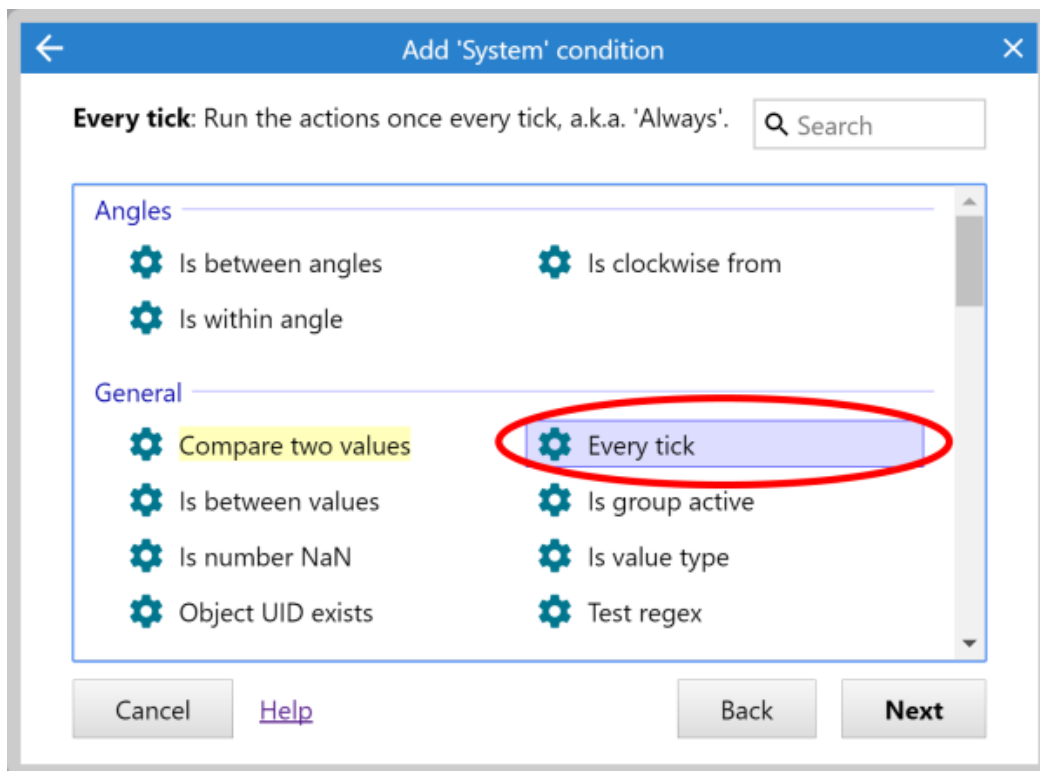
Remember a tick runs every time the screen is drawn, so if we make the player point at the mouse cursor position every tick, they'll always appear to be facing the mouse cursor.

Let's start making this event. Double-click a space in the event sheet. This will prompt us to add a condition for the new event.



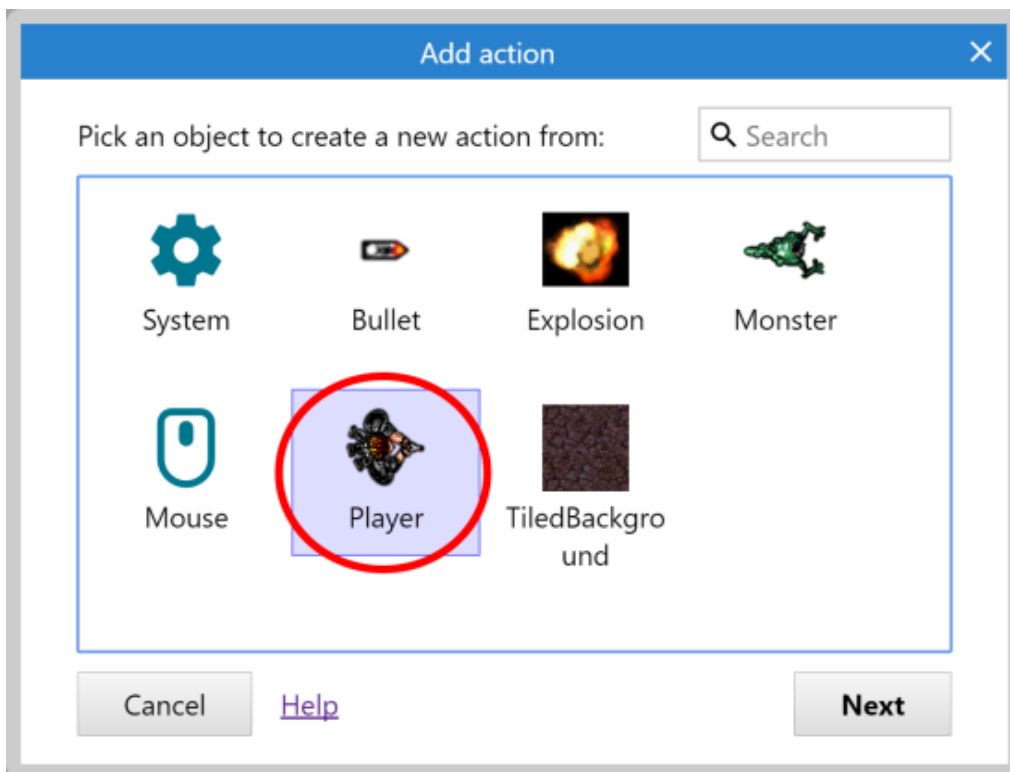


Different objects have different conditions and actions, depending on what they can do. There's also the System object, which represents Construct's built-in features. Double-click the System object as shown. The dialog will then list all of the System object's conditions:

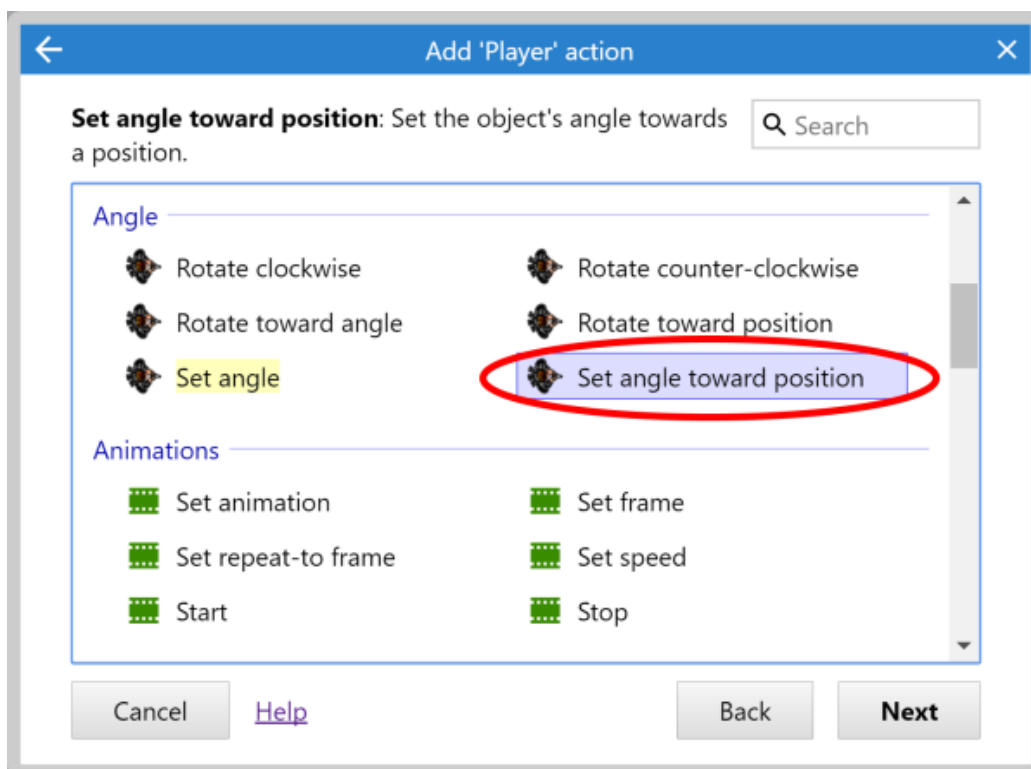


Double-click the *Every tick* condition to create an event with it. The dialog will close, and the event is created. However, you'll notice that it has no actions.

Now we want to add an action to make the player look at the mouse cursor. Click the Add action link to the right of the event. (Make sure you get the *Add action* link, not the *Add event* link underneath it which will add a whole different event again.) The Add Action dialog will appear:



As with adding an event, we have our same list of objects to choose from, but this time for adding an action. Try not to get confused between adding conditions and adding actions! As shown, double-click the **Player** object, for it is the player we want to look at the mouse cursor. The list of actions available in the **Player** object appears:





Rather than set the player's angle to a fixed number of degrees, it's convenient to use the Set angle towards position action. This will automatically calculate the angle from the player to the given X and Y co-ordinate, then set the object's angle to that. Double-click the *Set angle towards position* action in the list.

Construct now needs to know the X and Y coordinate to point the player at:

Parameters for Player: Set angle toward position

X: X position to set angle toward.

X 0

Y 0

Cancel Help Find Expressions Back Done

The X and Y fields are called the parameters of the action. Conditions can have parameters too, but *Every tick* doesn't need any.

We want to set the angle towards the mouse position. The Mouse object can provide this. Enter **Mouse.X** for X, and **Mouse.Y** for Y. These are called *expressions*. They're like sums that are calculated. For example, you could also enter **Mouse.X + 100** or **sin(Mouse.Y)** (although those particular examples might not be very useful!)

This way, you can use any data from any object, or any calculation, to work out parameters in actions and conditions. It's very powerful, and a sort of hidden source of much of Construct's flexibility.

If you get an error that says: "Mouse is not an object name", make sure you added the Mouse object! Go back to Part 2 and check under *Add the input objects*.

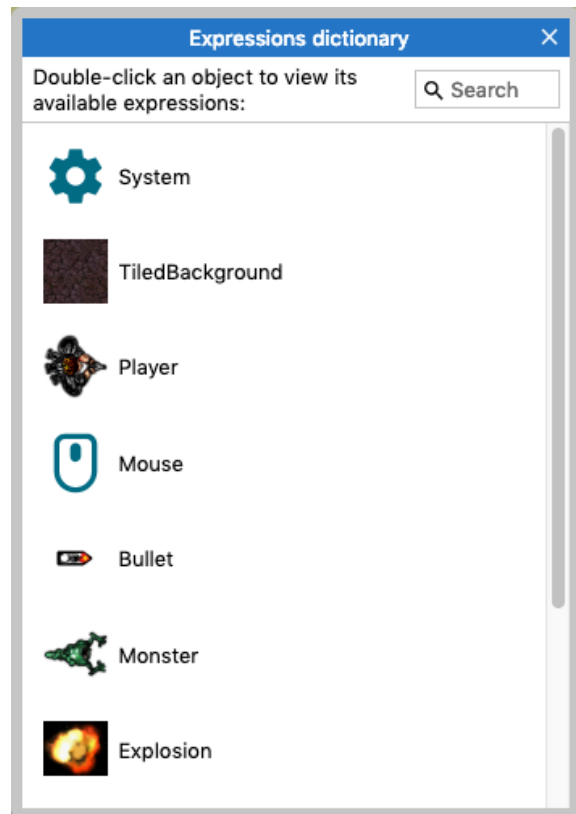


You might be wondering how you'd remember all the possible expressions you could enter.

Firstly, you might notice Construct shows some lists as you type. This is called *autocomplete* and helps show what you can type at each point in an expression.

Secondly, there's also the *Expressions Dictionary* which lists them all. If there's room on your screen, it will appear faded out, so it doesn't distract you until you need it.

If it's not there, you can click *Find expressions* to open it. You can double-click an object in the Expressions Dictionary to see a list of all its expressions. If you double-click an expression, it will also insert it for you, saving you from having to type it out.



Anyway, click Done on the parameters dialog. The action is added! As you saw before, it should look like this:



There's your first event! Try running the game, and the player should now be able to move around as before, but always facing the mouse cursor. This is our first bit of custom logic.



Part 5 – More Game Logic

If each event is described in as much detail as before, it's going to be quite a long tutorial. Let's make the description a little briefer for the next events. Remember, the steps to add a condition or action are:

1. Double-click to insert a new event or click an *Add action* link to add an action.
2. Double-click the object the condition/action is in.
3. Double-click the condition/action you want.
4. Enter parameters, if any are needed.

From now on, events will be described as the object, followed by the condition/action, followed by any parameters. For example, the event we have just inserted could be written:

Add condition System → Every tick

Add action Player → Set angle towards position, and for X: **Mouse.X**, Y: **Mouse.Y**

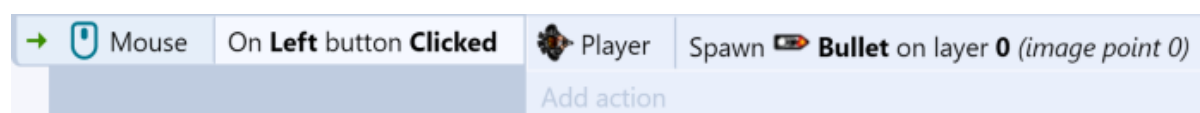
GET THE PLAYER TO SHOOT

When the player clicks, they should shoot a bullet. This can be done with the *Spawn an object* action in **Player**, which creates a new instance of an object at the same position and angle. The Bullet movement we added earlier will then make it fly out forwards. Make the following event:

Condition: Mouse → On click → Left clicked (the default)

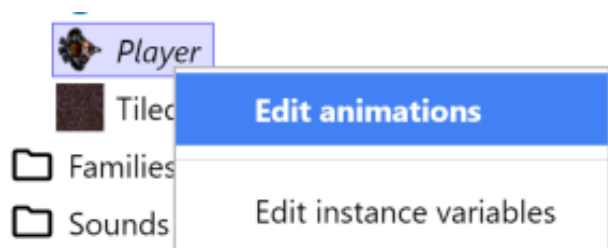
Action: Player → Spawn another object, then for *Object*, choose the **Bullet** object. Leave the other parameters as they are.

Your event should now look like this:



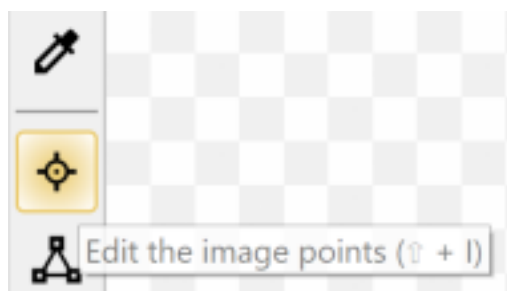
If you run the game, you can shoot bullets! However, you may notice the bullets shoot from the middle of the player, rather than from the end of the gun. Let's fix that by placing an image point at the end of the gun. An image point is just a position on an image that you can spawn objects from, and we can reference it in the *Spawn another object* action.

Right-click the **Player** in the Project Bar and select Edit animations.





The image editor for the player reappears. Click the origin and image points tool:

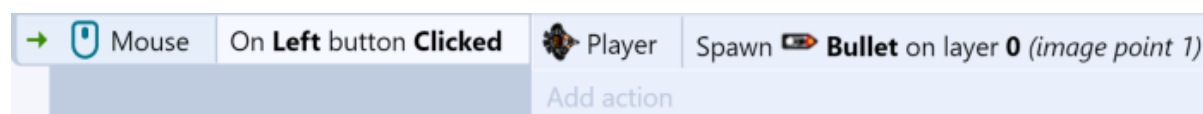


The side pane will then turn into a list of image points – you can read more about the image points pane in the chapter covering the Animations Editor.

Notice the object origin appears in the list. That's the "hotspot" or "pivot point" of the object. If you rotate the object, it spins around the origin. That's also what's used when you spawn an object at image point 0, as our action did. We want to add another image point to represent the gun, so right-click in the list and select Add a new image point. A new item appears in the list, and an icon appears over the image to indicate where this image point is. Left-click at the end of the player's gun to place the image point there:



Close the image editor. Double-click the *Spawn an object* action we added earlier and change the *Image point* to **1**. The event should now look like below - note it says *Image point 1* now:



Preview the game again. The bullets now shoot from the end of your gun! The bullets don't do anything yet, though. Hopefully, however, you'll start to realise that once you get to grips with the event system, you can put together your game's logic very quickly.

Let's make the bullets actually kill monsters. To do this, you'll need to add the following event:

Condition: Bullet → On collision with another object → Monster.

Action: Monster → Destroy

Action: Bullet → Spawn another object → Explosion



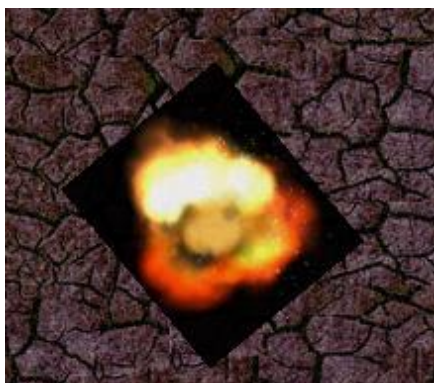
Action: Bullet → Destroy

Here's what the finished event looks like.



THE EXPLOSION EFFECT

Preview the game and try shooting a monster. If you're using the same art assets as this tutorial, then you'll notice that the explosion has that big black border!



You might have predicted it'd look like that from the start and wondered if our game was really going to end up like that! Don't worry, it won't. Click the **Explosion** object in the Project Bar and its properties will appear in the Properties Bar on the left. In the *Effects* section, set its Blend mode to Additive. Now try the game again.



Why does this work? Without going too much to the nuts and bolts, ordinary images are pasted on top of the screen. With the additive blend mode, each pixel is instead added (as in, summed)



with the background pixel behind it. Black is a zero pixel value, so nothing gets added - you don't see the black background. Brighter colours add more, so appear more strongly. It's excellent for explosions and lighting effects.

MAKING MONSTERS A LITTLE SMARTER

Right now, the monsters just wander off the layout to the right. But let's face it, that wouldn't make much of a game! So, let's make them a bit more interesting. First of all, let's start them at a random angle.

Condition: System → On start of Layout

Action: Monster → Set angle to **random(360)**

They will still wander off forever when they leave the layout, never to be seen again. Let's keep them inside. What we'll do is point them back at the player when they leave the layout. This does two things: they always stay within the layout, and if the player stands still, monsters end up coming right for them!

Condition: Monster → Is outside layout

Action: Monster → Set angle toward position X: **Player.X** Y: **Player.Y**

Here are what the two finished events look like.

| | | | |
|------------|--------------------|------------|--|
| → ⚙ System | On start of layout | 👁 Monster | Set angle to <i>random(360)</i> degrees |
| | | Add action | |
| 👁 Monster | Is outside layout | 👁 Monster | Set angle toward (<i>Player.X</i> , <i>Player.Y</i>) |
| | | Add action | |

Run the game. If you hang around for a while, you'll see the monsters stay around the layout too. And they're all going in random directions. It's hardly AI, but it'll do!

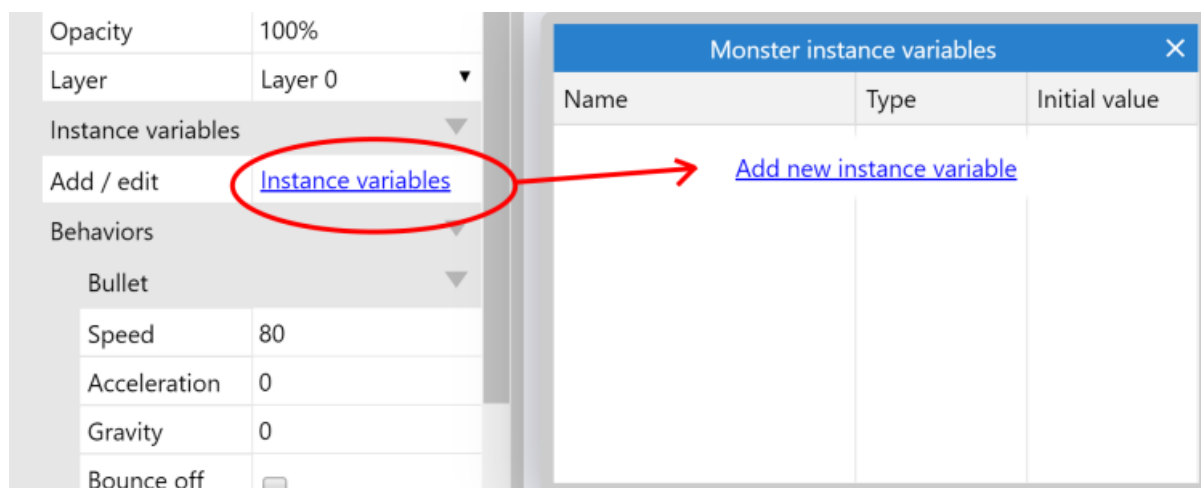
Now, suppose we want to have to shoot a monster five times before it dies, rather than instant death like it is at the moment. How do we do that? If we only store one "Health" counter, then once we've hit a monster five times, all the monsters will die. Instead, we need *each* monster to remember its *own* health. We can do that with instance variables.



Part 6 – Using Instance Variables

Instance variables allow each monster to store its own health value. A variable is simply a value that can change (or *vary*), and they are stored separately for each instance, hence the name *instance variable*.

Let's add a *health* instance variable to our monster. This works somewhat similarly to adding a behavior. Click **Monster** in the Project Bar - you can also switch back to the layout using the tabs at the top and select a **Monster** object. This will show the **Monster's** properties in the Properties Bar. Click Instance variables to open the Instance Variables dialog.



You can add as many instance variables to an object as you like, but we only need one for the monster. Click Add new instance variable. The following dialog appears for adding an instance variable.

The 'Add instance variable' dialog box contains the following fields and buttons:

- Name: Health
- Type: Number
- Initial value: 5
- Description (optional):
- Buttons: Help, OK, Cancel

Type *Health* for the name, leave *Type* as Number, and for *Initial value* enter 5 (as shown). This starts every monster on 5 health. When they get hit, we'll subtract 1 from the health, and then when health is zero, we'll destroy the object.

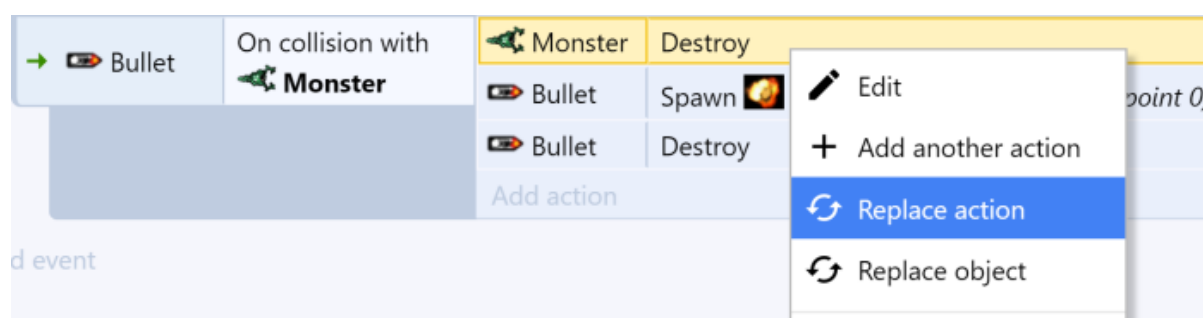


Once you're done click OK. Your variable now appears in the instance variables dialog and in the properties for the monster as well. You can quickly change initial values in the properties bar, but to add or remove variables you'll need to open the instance variables dialog. Also, note every object in the layout can have unique instance variable values set as well, so you could, for example, start every monster with a different amount of health.

CHANGING THE EVENTS

Switch back to the event sheet. Right now, we're destroying monsters as soon as the bullet hits them. Let's change that to subtract 1 from its health.

Find the event that reads **Bullet: On collision with Monster**. Notice we've got a "destroy **Monster**" action. Let's replace that with "subtract 1 from health". Right-click the "destroy **Monster**" action and click Replace action.



The same dialog appears as if we were inserting a new action, but this time it'll replace the action we clicked instead. Choose **Monster** → Subtract from (in the *Instance variables* category), choose the instance variable "health", and enter 1 for the value. Click Done. The event should now look like this:

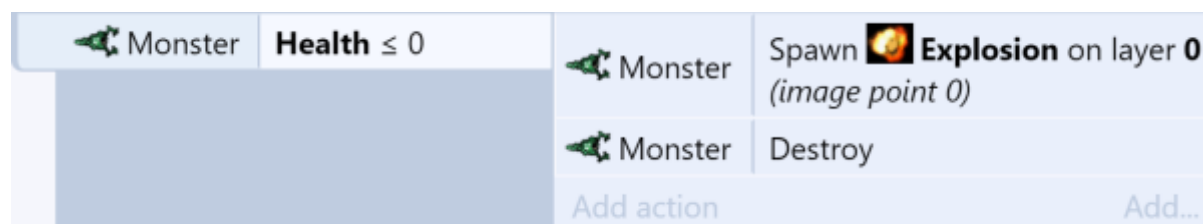


Now when we shoot a monster, they lose 1 health, and the bullet explodes, but we haven't made an event to kill monsters when their health reaches zero. Add another event:

Condition: Monster → Compare instance variable → Health, Less or equal, 0

Action: Monster → Spawn another object → Explosion

Action: Monster → Destroy





Why "less or equal 0" rather than "equals 0"? Suppose we added another more powerful weapon which subtracted 2 from health. As you shot a monster, its health would go 5, 3, 1, -1, -3... notice at no point was its health directly *equal to zero*, so it'd never die! Therefore, it's good practice to use "less or equal" to test if something's health has run out.

Run the game. You now have to hit monsters five times to kill them!



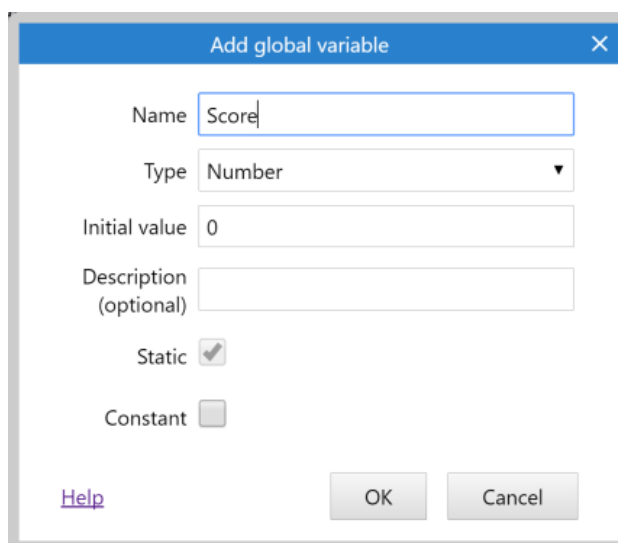
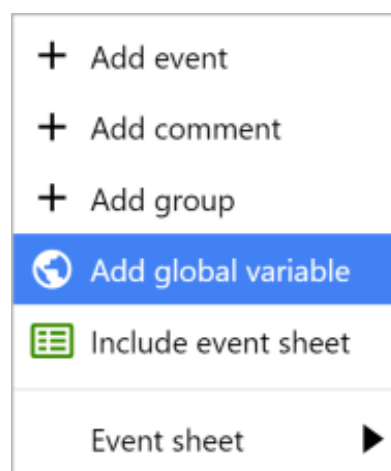
Part 7 – Keeping Score

Let's have a score, so the player knows how well they've done. We'll need another variable for this. You might think "let's put the score as one of the player's instance variables!". That's not a bad first idea, but remember the value is stored "in" the object. If there are no instances, there are no variables either! So, if we destroy the player, we can no longer tell what their score was because it was destroyed with the player.

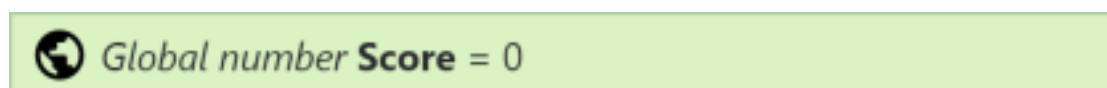
Instead, we can use a global variable. Like an instance variable, a global variable (or just "global") can store text or a number. Each variable can store a single number or a single piece of text. Global variables are also available to the entire game across all layouts - convenient if we were to add other levels.

Right-click the space at the bottom of the event sheet and select Add global variable.

Enter Score as the name. The other field defaults are OK, it'll make it a number starting at 0.




Now the global variable appears as a line in the event sheet. It's in this event sheet, but it can be accessed from any event sheet in any layout.



There are also *local variables* which can only be accessed by a smaller "scope" of events, but we don't need to worry about that right now.



Let's give the player a point for killing a monster. In our "**Monster**: health less or equal 0" event (when a monster dies), click Add action, and select System → Add to (under *Global & local variables*), and choose Score with a value of **1**. Now the event should look like this:

| Monster | Health ≤ 0 | | |
|---------|------------|--|---|
| | | Monster | Spawn  Explosion on layer 0 (image point 0) |
| | | Monster | Destroy |
| | | System | Add 1 to Score |
| | | Add action Add... | |

Now the player has a score, which increases by 1 for every monster they kill - but they can't see their score! Let's show it to them with a text object.

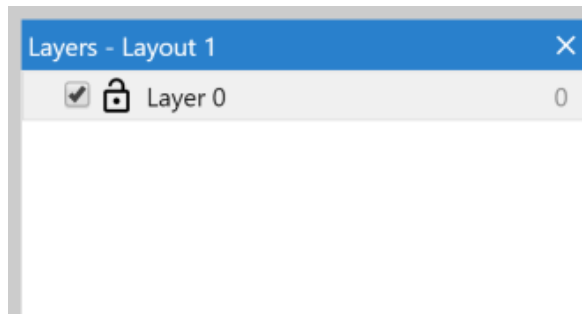


Part 8 – Displaying Score

To display the player's score, we'll use a Text object. However, we want to show it on a fixed position on-screen. The view follows the player, and we don't want the score to disappear as the player walks away! To fix this, we'll need to add a new layer.

ADDING A LAYER

To add a layer, you'll need the Layers bar, which by default is in the lower right-hand corner of the editor.



You should see Layer 0 in the list.

Construct counts starting from zero, since it works better like that in programming.

Right-click in the Layers Bar and select Add layer at top. (Be sure to add it at the top, not the bottom, because we want the score to display on top of everything else!) When you

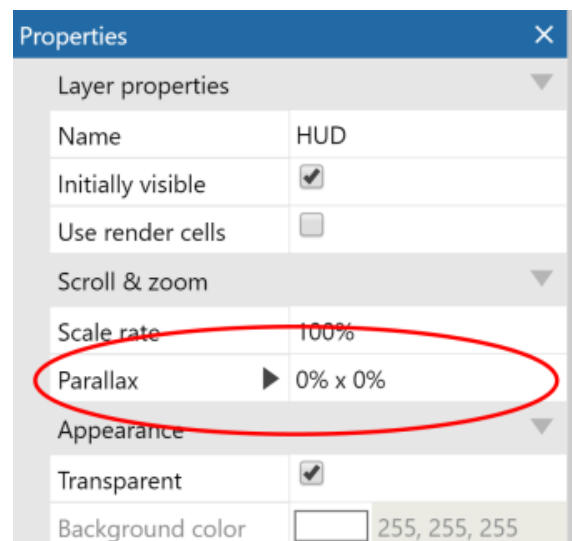
add it, you can immediately type in a name. Enter HUD, which stands for Heads-Up Display - which is a term used for all the on-screen information.

Now, make sure the 'HUD' layer is selected in the Layers Bar. This is important - the selected layer is the *active layer*. All newly added objects are added to the active layer, so if it's not selected, we'll later accidentally add our Text object to the wrong layer. The active layer is shown in the status bar in the bottom-right corner of the Layout View - it's worth keeping an eye on.

PARALLAX

The HUD should always stay in the same place on the screen. By default, layers scroll as the view moves. To keep them on the screen, we can use the layer Parallax setting. Parallax allows different layers to scroll at different rates for a sort of semi-3D effect. If we set the parallax to zero, though, the layer won't scroll at all - ideal for a HUD.

Since you have the HUD layer selected, its properties should be showing in the Properties Bar. Set the Parallax property to 0 x 0 (that's zero on both the X and Y axes).



Now we have a layer where we can place objects that appear in a fixed place on-screen! However, we don't have any objects on it yet.



ADDING THE TEXT OBJECT

Switch back to the Layout View using the tabs at the top. Make sure the HUD layer is selected in the Layers Bar, to ensure the Text object is added to the right layer. Double-click a space to add another object. This time pick the Text object.

Try not to confuse the Text object, which displays some text, with the Text input object, which is a box the user can type some text to (e.g. for a form).

Place the Text object in the top left corner of the layout. It's going to be hard to see if it's black, so in the Properties Bar, make it bold, italic, yellow, and choose a slightly larger font size. Resize it wide enough to fit a reasonable amount of text. It should look something like this:



Text objects won't display any text in the game if they are too small to fit the first word inside their box. If you don't see any text appear, try resizing the Text object larger.

Switch back to the event sheet. Let's keep the text updated with the player's score. In the Every tick event we added earlier, add the action Text → Set text.

Using the & operator, we can convert a number to text and join it to another text string. So for the text, enter:

```
"Score: " & Score
```

The first part ("**Score:** ") means the text will always begin with the phrase *Score:*. Any pre-defined text has to appear in double-quotes in expressions. The second part (**Score**) is the actual value of the Score global variable. The & joins them together to one piece of text.

Run the game and shoot some monsters. Your score is displayed, and it stays at the same place on the screen!



Part 9 – Finishing Up

We're nearly done. Let's add some final touches.

Firstly, let's have some monsters regularly spawning. Otherwise, once you've shot all the monsters, there's nothing left to do. We'll create a new monster every 3 seconds. Add a new event:

Condition: System → Every X seconds → 3

Action: System → Create object, Monster, layer 0 (that's the main layer), X: **LayoutWidth + 100**, Y: **random(LayoutHeight)**

This will make the position a random place along the right edge of the layout which is where monsters will come from. We also create them 100 pixels further to the right, so the player can't see them suddenly appear at the edge, they will instead come in from the outside.

Finally, let's have ghosts kill the player.

Condition: Monster → On collision with another object → Player

Action: Player → Destroy

SOME IDEAS TO TRY

Want to go a bit further? Here are some ideas for some extra things to add:

- Make the player get points for hitting monsters as well as killing them. You can adjust the score gained for each case.
- Make the monsters slowly speed up over time, so they're harder to hit and avoid.
- Add another kind of enemy!
- Add an alternative kind of weapon, which uses a different mouse button or keyboard control.
- Add the Audio object, import some sound files, and add sound effects or music.
- Add a title screen. Use the System object *Go to layout* action to switch between them.
- Introduce some scenery or obstacles in the level design.
- Add a "Game over" screen or make something else happen when the player dies.

You can always look through the examples on the C3 Start Page for some inspiration!



Testing and Publishing

Testing Your Game

To test your game during development, you can preview it by clicking the "play" icon in the main toolbar, by selecting Menu → Project → Preview, by right-clicking a layout in the Project Bar and selecting **Preview**, or by pressing F5. This will start your game from the current layout. We recommend testing your projects frequently, as well as having regular backups, so if you accidentally break your project, it's easy to go back without having to remember everything you did!

By default, starting a preview opens a pop-up window. You may see a message that the pop-up was blocked. Clicking **Try again** normally works, but to permanently prevent the message appearing you may need to change your browser's settings. Usually, an icon or message will appear somewhere in the browser UI indicating a pop-up was blocked; clicking this should provide a way to always allow pop-ups for the current website.

In Menu → Settings, you can choose different preview modes. The three options are:

Pop-up window: as described above, opens a pop-up window to run the project in.

Browser tab: opens a new browser tab to run the project in.

Dialog: opens a dialog inside the Construct 3 UI to run the project in. This does not use a new browser window so is not subject to pop-up blockers and does not include other browser UI features like the address bar. However, it cannot appear larger than, or outside of, the Construct 3 window.

If you select **Preview** again with a preview already running, the existing preview window or dialog will restart and begin previewing the latest version of your project.

Let's say you want to test multiplayer and you need a couple of preview windows open. Your best options are either to have the project open in multiple browsers – this also allows you to see if there are any differences when running your game in Firefox vs Chrome, for example.

Or, if you're using the full version, you can use Remote Preview which is covered in more detail later in this chapter.

OTHER PREVIEW TYPES

In the main toolbar, there is a dropdown arrow next to the Preview button that shows a menu with more preview options. These can also be found in the Menu → Project submenu, or by right-clicking the project name in the Project Bar.

DEBUG LAYOUT



This runs the current layout in a special debug mode. The debugger is a special development tool which helps you inspect the state of the game (such as the value of expressions and variables). It also provides diagnostic tools such as advancing the game frame-by-frame, changing values, destroying objects, setting breakpoints in events, and more. This can bring invaluable insight to how your game is working, particularly if you run to a problem.

PREVIEW PROJECT

This simply starts a preview from the first layout in the project. This is either the first layout that appears in the Project Bar, or whichever layout is set in the *First layout* project property.

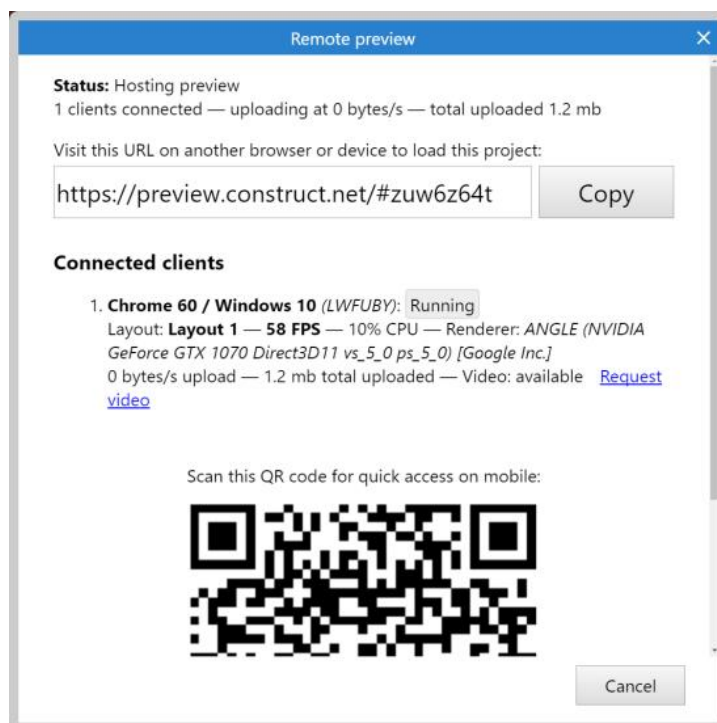
REMOTE PREVIEW

This allows you to preview your project on a different device. It is also useful for testing different browsers on the same device. Starting a Remote Preview does not actually directly run your game. It will open a dialog that provides a URL you can use to load the game, or a QR code to scan. All you need to do is open the URL on another browser or device, or share the URL with someone else, or scan the QR code, and the game will start to load and run in the browser. The game is loaded directly from your device using a peer-to-peer connection; it is not uploaded anywhere else but is still accessible from anywhere on the Internet. The game is no longer available from the provided URL as soon as you close the Remote Preview dialog. You can open the Remote Preview dialog to its own window to help keep it out of the way, by right-clicking on its caption and selecting Open to pop-up window.

Once the game starts running, they appear in the Remote Preview dialog as a connected client. You can have multiple copies of the game running simultaneously.

You can view some basic system details and real-time performance information for connected clients, including their browser & OS, which layout they are on, the framerate and approximate CPU usage (and approximate GPU usage if available), and their graphics hardware. You can also click Request video to see a video stream of their game.

Like with a standard preview, you can update a remote preview by selecting the **Remote preview** option again. This updates the version of the project available at the same URL. Any existing clients will have to reload their browser to see the new version of the project.





Clients who are viewing your project via Remote Preview will see notifications in the following situations:

- When the host updates the project, clients will see a notification indicating an update is available. They must reload their browser to load the new version.
- When the host closes the Remote Preview dialog, the remote preview ends. Clients will see a notification that the host disconnected. Clients can continue to run the project (they are not cut off), but if they reload the project will no longer be available.
- When the host starts or stops video, the client will be notified.

Remote Preview allows you to instantly share your project to anyone in the world with an Internet connection. This is particularly useful if you have remote testers, playtesters or reviewers.

On the other hand, you can use the remote preview URL on the same device for cross-browser testing, such as using Remote Preview to test your project in Firefox while Construct 3 runs in Chrome. In this case, data is not sent over the Internet and is only transferred across the local system.

Similarly, if you Remote Preview to a device on the same local area network (LAN), such as a mobile device, most browsers will try to establish a local connection for data transfer ensuring the project can load at a much faster LAN speed rather than transferring via the Internet.

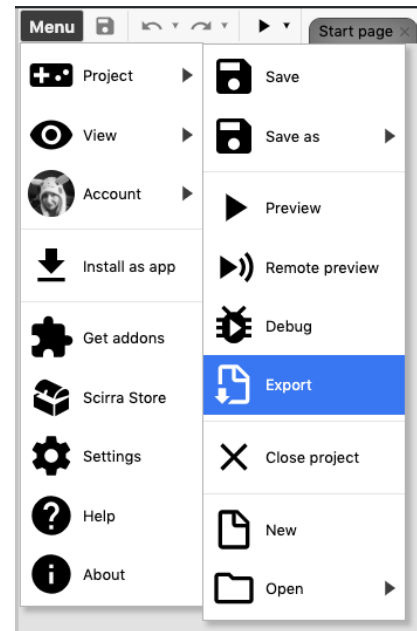
Exporting and Publishing

Construct 3 allows you to export to a range of platforms, such as the web (HTML5), Android and iOS (via Cordova), desktop apps (via NW.js) and Windows UWP (Universal Windows Platform) apps. Each platform has an accompanying tutorial to guide you through how the exporter works and covering any exporter-specific settings – you can find all of these in the tutorials section on construct.net. Bear in mind that in some cases fees may be involved, such as for developer subscriptions to allow uploading to app stores, or for hosting web content if you don't already have a server.



To publish a project, use the Menu → Project → Export option. Every kind of export has three common settings:

- **Deduplicate images** will search the entire project for identical images and remove the duplicates. This helps save memory and reduce the download size by removing redundant images.
- **Recompress images** will recompress all the PNG images in the project with enhanced compression. This can take a long time, but often significantly reduces the download size of the exported project. This step is lossless, so it is guaranteed to preserve the quality of all your artwork.
- **Minify script** will obfuscate and compress the main JavaScript file for your exported project. This also helps reduce the download size, improves load time, and makes it significantly more difficult to reverse-engineer the project.



Each option helps optimise the exported project but can make the export take longer. It is recommended to enable all three when exporting the final finished project for publishing. However, if you are simply doing a trial export, you may wish to disable them to speed up the process.

Most exporters will prompt you for additional settings specific to that exporter. For example, the Android exporter has a setting for the minimum supported Android version.

When the export finishes, you will be provided with a link to download a zip file containing the final exported project. You can also view some statistics about the export, such as how much download size was saved by image recompression (if it was enabled). The Export Manager can also be used to find the last few exported projects and download them again.

There are plenty of tutorials on [construct.net](https://www.construct.net) covering various export routes, but we're going to cover three here – each of which you can use to sell or monetise your games.

Publishing to Itch.io

“itch.io is an open marketplace for independent digital creators with a focus on independent video games. It’s a platform that enables anyone to sell the content they’ve created. As a seller you’re in charge of how it’s done: you set the price, you run sales, and you design your pages. It’s never necessary to get votes, likes, or follows to get your content approved, and you can make changes to how you distribute your work as frequently as you like.”

itch.io is also a collection of some of the most unique, interesting, and independent creations you’ll find on the web. We’re not your typical digital storefront, with a wide range of both paid and free content, we encourage you to look around and see what you find.”

That blurb is from itch.io’s About Us page. The platform is a wonderfully eclectic mix of games, assets, prototypes and more and is a fantastic place to post your games.



Getting Started

The great thing with itch.io is that it's pretty simple to get a game up there. You will also need to put together an associated page, but itch's tools are fairly easy to navigate.

You can use two of Construct's export options for publishing on itch – HTML5, which will allow users to play your game directly in the browser, or NW.js which will create a desktop app which people can then download.

Uploading games to itch is free, and you'll need an account with itch to upload a game, but it's straightforward to sign up. If you want to monetise your game, then you'll need to configure payment options for your account too. We'd encourage you to familiarise yourself with itch and its settings before uploading your first game!

Your Page

Each game you upload to itch has its own corresponding page – an extension of the URL you're assigned when you create an account. As with any game storefront these days, itch has thousands of projects, and your page is one way to make yours stand out.

There are page and content guidelines available on itch.io's website, and while there is a summary of the key points for pages below, we'd recommend having a read through the whole thing. There's lots of useful info available to itch users that covers pretty much everything the service has to offer.

When designing the page for your game, you should consider the following:

Don't publish before you're ready

When you press the button to publish your page, everyone can see it – nobody is going to be very impressed if your page is full of missing images or details when they come to view your game. You can enable your page in draft mode, which is very useful for seeing what it would look like but save that public setting for when you're definitely ready to go! It should also be noted that as soon as you set your page to public, itch will include your page in the 'Recently added' list – this only happens once, you page cannot be put back at the top of the Recently added list, even if you unpublish and republish your page.

Use accurate metadata

The metadata you attach to your game allows it to be categorised by itch, so it appears on the relevant browse pages on the site. If you're deemed to be purposefully adding irrelevant metadata, you could be flagged as spam, and nobody will be allowed to visit your page!

Tag responsibly and label correctly – especially adult content

Tagging helps itch categorise your project, so it may be tempting to throw as many tags in as possible. Again, this is frowned upon and may result in your page being delisted. Keep your tags relevant, and if there is a suggested tag that fits, use it! Suggested tags



are designed by itch to make things easier to find. You should also avoid including tags which are synonyms, already used as classifications in your game's metadata or are the same as the name of your project. That's what the title is for!

Also, if your project has adult content, this must be labelled – itch has the option to allow users to block adult content, so those users won't be happy if something incorrectly labelled suddenly appears on their feed!

Add a cover image and screenshots

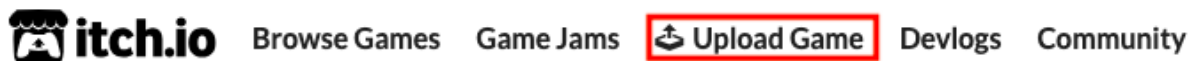
Including a good cover image for your game is a must – it's likely to be the first thing someone sees in itch's browse pages! If you want, you could even upload a GIF as your cover image! Screenshots are optional, but it's always worth including a couple somewhere on the page to give users an idea of what the game looks like.

Don't be obnoxious

This might sound rather blunt and can be quite subjective, but it nevertheless is an important point. Itch prides itself on the fact it's an ad-free site and that the user experience is generally very good – so they have requested that if you do need to show adverts, please don't make them obtrusive or misleading.

And while it's one thing to have a game that's designed to shock people, it's a very different thing to plaster that shocking imagery all over the place. Try and keep anything uber-offensive or intrusive off of your public page.

So, now we've covered some do's and don'ts, let's look at the actual process for creating a project on itch.io.



On itch's homepage, you'll see a toolbar at the top of the page – here you should see a link to Upload Game. (If you can't see this, it may be that your account is not set up as a developer account.)

Clicking Upload Game will take you to the Create New Project page where you can start filling in all the details for your game's page, as well as uploading the game files and any screenshots you might have. The Create New Project page is quite long, but we'll briefly walk through everything it involves!



Make sure everyone can find your page

Review our [quality guidelines](#) before posting your project

Title

Project URL

Short description or tagline

Shown when we link to your project. Avoid duplicating your project's title

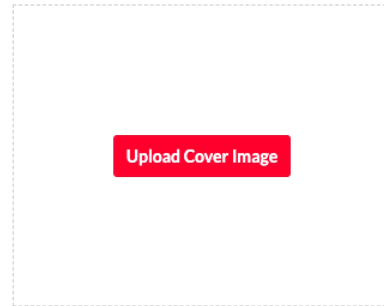
Classification

What are you uploading?

Kind of project

TIP You can add additional downloadable files for any of the types above

Release status



The cover image is used whenever itch.io wants to link to your project from another part of the site. Required (Minimum: 315x250, Recommended: 630x500)

Gameplay video or trailer

Provide a link to YouTube or Vimeo.

Screenshots

Screenshots will appear on your game's page. Optional but highly recommended. Upload 3 to 5 for best results.



To start with, you'll be asked for some fairly basic information, your game's title, a tagline perhaps and what you're classifying it as. You can also set what kind of project it is, so is it an HTML5 game you play in your browser, or one that you can download? This category will have an effect on how you monetise your game, but more on that shortly. On the right-hand side of the page, you'll see the options to add your cover image, a link to a trailer and any screenshots you have. Both the trailer and the screenshots are optional, but they're nice to have!

Pricing

 \$0 or donate Paid No payments

Someone downloading your project will be asked for a donation before getting access. They can skip to download for free.

Suggested donation — Default donation amount

Uploads

TIP Use [butler](#) to upload game files: it only uploads what's changed, generates patches for the [itch.io app](#), and you can automate it. [Get started!](#)

 or

File size limit: 1 GB. [Contact us](#) if you need more space

Your next two options are very important – pricing and uploading your game!



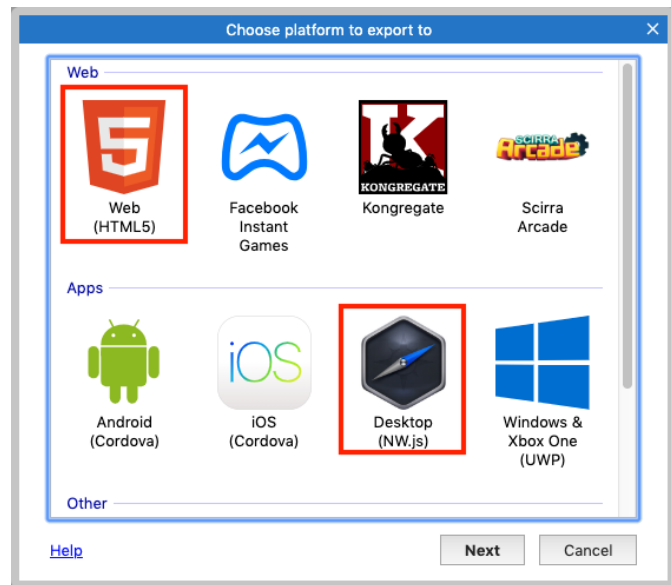
If you've set up your billing controls in your account, then you'll be able to sell your game! As you can see, itch offers three options for pricing – Free with an optional donation, a set price, or no payments. Bear in mind though that currently, if your game is an HTML5 game, you can only set it to be free or free with optional donations. If you want to charge a fixed price, it needs to be a downloadable game.

And now for the most crucial bit – uploading your game files!

Your Game

As previously mentioned, you have two options in the Construct 3 exports that are compatible with itch.io – HTML5 and NW.js. Once you're happy with your game, and you've decided how you're going to make it available on itch, it's time to export it!

Under the Project menu, you'll find the option to Export your project. Click this, and the platform selection dialog will appear. This window will show you all of the available export options within Construct 3. You only need to be concerned with two of these options. If



you're planning on having your game playable in the browser, then you'll want to select the orange shield icon in the top left corner. That's your standard, web (HTML5) export. If you want your game to be downloadable and run as its own desktop program, then you'll want the black hexagon icon – this is for NW.js desktop apps.

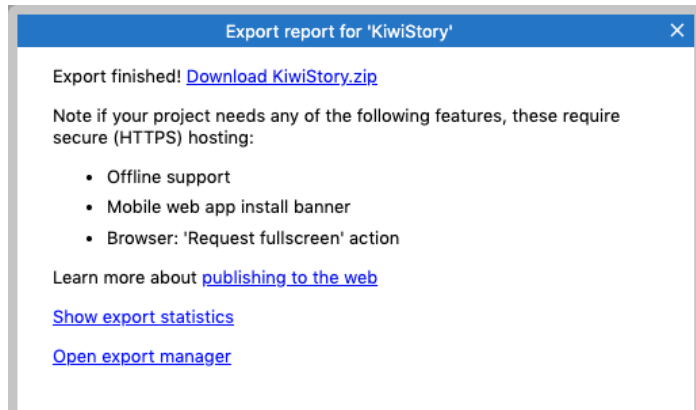
Both export options will produce a ZIP file which you'll then need to upload to itch.

HTML5 Games

Let's say you've decided to go for the play-in-browser approach. HTML5 is probably the easier of the two options as far as getting your game running is concerned. You'll need to decide whether or not you're going to use the three common export options mentioned above – as this should be your final (at least for a while) export, we'd recommend that you enable all of them, and let the export take a little longer.



Once your export is complete, a dialog will appear to inform you thusly and give you a link with which to download your ZIP file. If you lose the file, you can also access it from the export



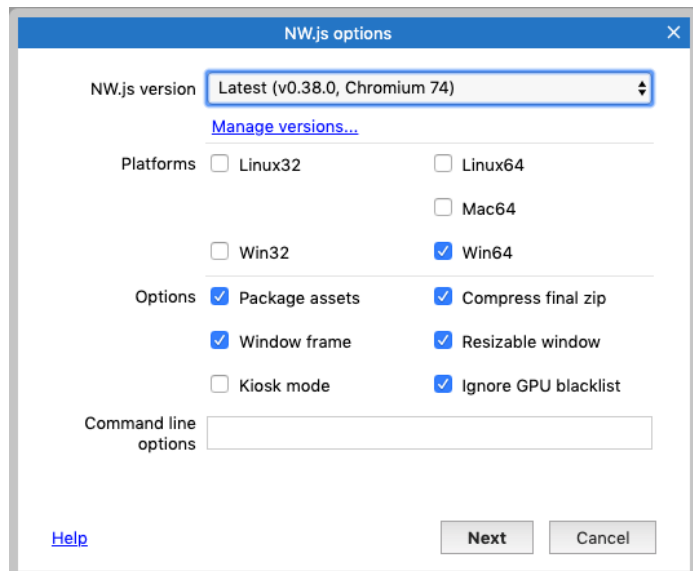
manager.

You'll notice that this dialog mentions features that need secure hosting. You won't need to worry about these as it should all be handled by itch.io. Itch, like most major websites, is already secured – denoted by the https at the start of the URL.

If you want some statistics on your export, like how much RAM you've saved by deduplicating images or how long the export actually took to complete, then click 'Show export statistics'. You'll get a little window with some useful nuggets of data.

NW.js

When it comes to exporting for NW.js, there are a few more options for you to play around with. After selecting NW.js from the platform selection dialog, you'll be presented with the NW.js options dialog. Here you can change thing around to customise your export.



When exporting for itch.io, it can be very tempting to export to all the available platforms in one go.

However, while this may save time, it can be risky. If you unzip the file on an operating system that doesn't match its destination, you run the risk of having broken file permissions.

Windows generally recognises any file with the .exe extension as something that can be run. This differs to both macOS and Linux, which require a special 'executable' file permission to be set. So, Windows is usually unaware of these special permissions, and if you transfer files across from Windows to macOS or Linux, then chances are those files won't have those special permissions and won't run. This is just one example, file permissions across operating systems can be a tricky business.

Let's say, for example, you're working on a Windows machine, but you want to create a build for MacOS. The good news is, Construct sets all the correct permissions for the files inside the ZIP. This means it's safe to transfer the zipped file between systems as you please. However, as soon



as you extract the files, they should no longer be moved between operating systems as you may lose the permissions.

Going back to the example macOS project exported on a Windows machine – when you get your ZIP file out of Construct, you should move it to a macOS machine before unzipping it. That way, your files should remain untouched, and your game should work.

If you do decide to export multiple NW.js platforms at once, let's say you simultaneously export Linux, macOS and Windows builds in one ZIP, make copies of your ZIP and send a copy to each operating system before extracting. This means you only need to do one export, but by transferring the zipped files, you retain any file permissions.

The files in the exported folders will differ depending on which platform you're exporting for. MacOS and Linux both export single files – just the executable you need to run your game. Windows exports are somewhat more complex. You'll have a folder with a whole bunch of files which may look confusing to you but imagine what a potential user will think opening that folder for the first time!

So, it might be worth 'beautifying' your files before uploading them to itch. This is by no means an exhaustive list, but some recommendations include:

- If your game has just been exported with the filename NW, then you should probably change that to the actual name of your game.
- You may find that your executable file still has the standard NW.js icon. You can use a tool like Resource Hacker to change the icon of the executable to your own custom icon.
- Create a README file. This could be included as a separate download on your itch page as well as included in the ZIP file. The README should explain the basics of your app, perhaps include controls, maybe credits – treat it like a mini manual. You should probably also include instructions for how to run your game and basic troubleshooting – particularly for the Windows builds which require the most up to date version of DirectX to run.
- If you're feeling particularly adventurous, you could tackle building an installer for your game. Though this is a highly advanced option, will require a good coding knowledge and knowledge of Windows Installer would probably help too. Not for the faint-hearted!

Whichever export method you've chosen, your ZIP now needs to be uploaded to itch. Simply click the red 'Upload files' button, navigate to where you've saved the ZIP, select it and upload. It's as simple as that!

Next Steps

Now you've uploaded your files, you can finish filling in the rest of the details on the new project page.

The first big section is the Description – this is where you can go wild, telling everyone about your game! What is it about, what are the controls, what inspired you to make it? There's so much you could put into this section, but the ultimate goal is to describe your game and make people want to play it. You can also embed videos and images into this section too if you wish.



If you're looking for inspiration, check out the featured games list on itch, or try searching for something similar to what you want to publish!

After the description, you can choose a genre for your game. This is a drop-down menu with pre-existing options, so pick the one that best suits. You can always add additional genres in the tags section.

Details

Description – This will make up the content of your game page.

<> ¶ B / ⌵ ☰ ☷ ↻ ≡ 📺 🖼️

Genre
Select the category that best describes your game. You can pick additional genres with tags below

No genre ▼

Tags – [Tips for choosing tags](#)
Any other keywords someone might search to find your game. Max of 10.
Avoid using the genre or platforms provided above.

Click to view options, type to filter or enter custom tag

App store links
If your project is available on any other stores we'll link to it.

+ Steam + Apple App Store + Google Play + Amazon App Store
+ Windows Store

Talking of tags, as mentioned in the opening of this section – tag wisely. You're allowed a maximum of ten tags, and if you're deemed by itch to be tagging inappropriately, then your page may suffer as a result. Again, perhaps look at other games on the site for inspiration and think about what makes your game unique before tagging it.

Next, you'll have the option of adding any links to your game on other storefronts, like Steam for example. Or if you've got a mobile version of your game on Android or iOS, this is your opportunity to link to those versions.

And you're almost done with your project's page! Just three more sections to sort out. The first of which you can leave blank if you wish. Itch allows you to set a 'custom noun' for your project. The default is 'Game', but you could set it to something like 'Experience' or 'Visual novel' if game doesn't quite fit. This is something you can always trial – type in a word, see what it looks like in draft and change it if you don't like it.



Your next option covers what kind of community you want to have around your page. You've got three choices – disabled, comments or discussion board. They have a brief explainer on the 'Create a Project' page, and they all have their own uses

Custom noun
You can change how itch.io refers to your project by providing a custom noun.
Leave blank to default to: 'game'.

Community
Build a community for your project by letting people post to your page.

Disabled

Comments – Add a nested comment thread to the bottom of the project page

Discussion board – Add a dedicated community page with categories, threads, replies & more

Visibility & access
Use Draft to review your page before making it public. [Learn more about access modes](#)

Draft – Only those who can edit the project can view the page

Restricted – Only owners & authorized people can view the page

Public – Anyone can view the page, you can enable this after you've saved

Save & view page

We wouldn't recommend going with disabled – it's good to have somewhere for your users to leave feedback without having to navigate to a separate page. Comments will probably suffice for most projects. This simply allows people to write comments, which will then appear at the bottom of your page, where you can reply to them.

If you're planning on creating a community for your game, then you could use the

discussion board option. This option allows you to build a small forum for your game essentially. You or your users can post threads, have discussions and really get together over your game. This may take more work in terms of moderation, but if you're community-minded for your project, this is a good starting point.

Your final option on this page is your project's visibility. As we've previously mentioned, don't set this to public until you're sure your page is ready. Until then, use Draft to view all your changes, then perhaps use Restricted access to allow certain people to view the page and give you feedback. You'll notice in the above screenshot that Public is greyed out – this will remain so until you've saved your page at least once.

If you feel like you've filled in all the required information and made your page the best it can be, then switch your visibility to public and unleash your creation upon the world!



What Next?

So, there you have it – you’ve created and perhaps even published your first game! But what now? Well, a good first step would be to work through a few more tutorials, so you can get to know more of Construct 3’s capabilities.

Construct is a hugely powerful tool, and we’ve only just begun to scratch the surface with this book. A good next step is a tutorial called *How to Learn Construct 3? Next Steps for Beginners* which walks you through some of the example projects available in Construct 3, explaining some of their core components. There’s a link in the resources section.

One of the best ways to learn to make games, is to make games. Join up for gamejams or just set yourself little challenges – a popular one is to make a game a week! Or perhaps you’ve seen a mechanic in a game that you really like. Why not try and recreate it in Construct?

And if you find you need help, your fellow Constructors are a good bunch. Sign up to [our forums](#) and you can tap into their knowledge whenever you need a bit of help!

All the time you’re using the software, you’ll be picking up new ways to use it and be constantly expanding your gamedev knowledge. Who knows? Perhaps your creation will be the next big hit. Best of luck!



Useful Resources

Our website, construct.net, has plenty of documentation to accompany on your game development journey with Construct 3. [The manual](#) is definitely worth a bookmark, as it is the full documentation for C3, written by us. It covers all of C3's documentation in quite good detail. It's updated frequently, particularly as new features hit the stable builds.

If you want something that explains more about *how* to do something in Construct, then take a look at our [Tutorials Section](#). This area of our site is full of resources written by us and the Construct Community. There are thousands of tutorials covering Construct 3 and its predecessor Construct 2 – often the two are interchangeable, though it should be noted that there are several key features that Construct 3 has over Construct 2.

As an extension of the Tutorials Section, there are also [Tutorial Courses](#) available on our website. These are 'playlists', if you like, a group of tutorials that logically follow on from one another to teach a topic.

This book goes through what we like to call our [Beginner's Guide](#), and there is a resource that we think carries on from this. [How to learn Construct 3? Next Steps for Beginners](#) looks at several of the example and template projects available in Construct 3 and breaks them down for you.

Of course, some people prefer videos to written texts, so don't be afraid to explore YouTube. There are tons of Construct videos covering all manner of topics on YouTube. There are even some embedded within tutorials on our own site! And you can always subscribe to our [official YouTube channel](#).

As mentioned in the publishing section of this book, itch.io has a large bank of FAQs to help creators publish their projects on the platform. We'd thoroughly recommend taking the time to look through them if you intend to use itch. [Creator FAQ – itch.io](#).