

Introduction

Welcome to Flowlab! Create games right from your browser and export apps for iOS, Android & Windows.

This Guide will help you get started, and by the end of it, you will not only understand Flowlab but also create games with nothing holding you back.

Objects

In Flowlab, every game contains one or more Levels, which within themselves have game Objects.

You can use game Objects to build game levels, players, enemies, and everything else that makes up your game world.

Objects are defined by three main components that determine how it looks and behaves:
A Sprite, its Physics, and some optional Behavior logic.

Creating an Object

Create a new object by clicking anywhere on the grid, and select [Create](#) from the wheel menu.

It will open the object panel, displaying the available properties for the newly created Object.



Object Properties

The first property to set when creating a new object is its Type.

[The Type](#) is a label that describes what sort of Object this is.

All the objects of the same Type will share the same image sprite, physics, and behavior.



Tip: Carefully name your types to keep the game organized.

All objects also get a name property when added to a level.

Many copies of the same Object can share the same Type, but every Object name is unique in the game world.



Object Type:
Slime
Object Name:
Bob



Object Type:
Slime
Object Name:
Jill



Object Type:
Slime
Object Name:
Phil

Physics Properties of an Object

The physics properties determine how Objects move and react to forces or collisions when the game runs.

- **Movable:** Whether this Object can be moved around during the game or pulled by gravity. When unselected, the Object is static and won't move by physical forces, such as another object bumping into it.



- **Affected by Gravity:** Whether the world gravity forces apply to this Object. Uncheck this box to prevent the Object from falling in the direction of the gravity force.



- **Is Solid:** Whether this Object passes through others. If checked, it will automatically "enable collisions" on this Object and display the option to "enable rotation".



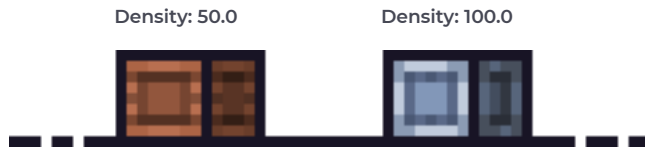
- **Enable Collisions:** Whether this Object has collisions. Behaviors such as collision, proximity, and raycast can use these events, so this Object can be detected by - and detect when colliding with others.



- **Enable Rotation:** Whether this Object can rotate, roll, and fall over from physical force, instead of always being locked to the x/y axis. This option is only available if the Object is solid.



- **Density:** Increase or decrease the mass of the Object. This option is only available if the Object is movable.



- **Bounce:** Setting a higher bounce means that this Object will bounce off when colliding with other Objects.



- **Friction:** Setting a high friction value will make other Objects slow down when in contact with it.



Think of friction like this:

- High friction, objects "walk on a hard surface", losing momentum.
- Low friction, objects "walk on ice", keeping momentum.

- **Forward Direction:** By default, the forward direction is to the right; you can change this when you want behaviors to rotate or add forces in a specific direction. The forward property can be adjusted to either up, down, or left instead, if desired.

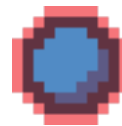


Collision Shape:

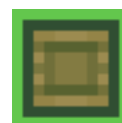
Only available if the Object has enabled collisions or is solid.

There are four options to choose from when selecting a collision shape:

Circle: For movable objects, the default is a circle, which means that this Object will be treated as a circle to detect collisions.



Rectangle: Non-moveable Objects will default to a rectangle, which will be the smallest rectangle containing the Object's entire sprite.



Capsule: Similar to the rectangle shape but with clipped corners. Specialized for game characters, it has default friction on the bottom and top part but low friction on the sides.



Polygon: This shape is an automatically generated polygon with the shape of your sprite.



Additional Properties

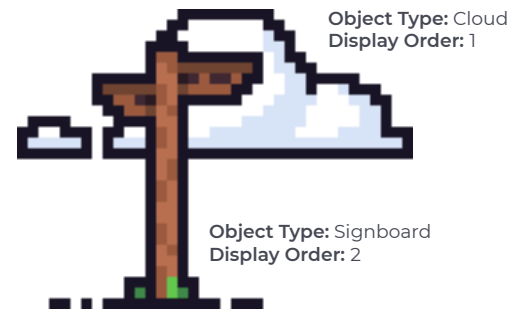
Behavior Reset:

- By default, all of an object's logic resets when the level loads (Reset on level start).
- By selecting keep on level restart, its behaviors won't reset and will be preserved instead.
- By selecting keep between levels, this Object will be retained in the game world even when switching between levels.

Display Order:

This determines where to sort the Object's sprite when rendering the game.

Suppose this Object will overlap with other objects and pass in front of them, then assign a higher value. Otherwise, give a lower value.



You can ignore this setting if the game's objects are not allowed to overlap or if their display order is not relevant.

Parents:

Adding a parent to an object will automatically add all of the behaviors from the Parent.

This option is useful for grouping, organizing, and to help cut down on the number and complexity of behaviors needed in your games.

An object can only have one parent, but a parent can have any number of children.

When an object has a parent:

1. The parent's behaviors run first, then the child's.
2. Collision (or Proximity) behaviors that trigger on a parent will also trigger on their children.
3. Children Objects will also receive messages sent to the parent.

How this can help simplify your games:

1. Have fewer duplicate behaviors by adding the behaviors to a parent type, and all the children will inherit that logic.
2. Instead of sending a message to many different object types, set them a common parent, and target the parent type.
3. Have fewer collision/proximity triggers: to handle collisions against many objects types the same way, give them a common parent, and trigger against the parent.

Object Library:

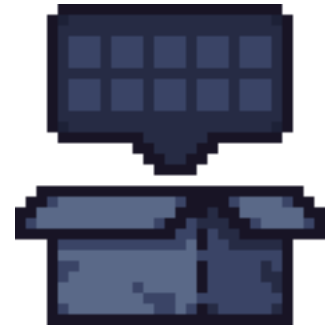
Upon creating an Object, it's added to the game's Object Library.

Clicking the Library button in the bottom toolbar opens up the library panel. From here, all of the game objects can be accessed and edited.

Clicking on an object in the library will open up the properties panel for this Object, with two additional buttons: add to level and delete all.

By selecting "add to level", you will be able to clone and place the Object into the level by clicking and dragging.

Selecting "delete all" will delete the object type from the library and all instances from the game entirely.



Tip: While in edit mode, you can mouse over+backspace to quickly delete various objects from the Game World. You can use the library to re-add these objects.



Logic and Behaviors

Bring your Objects to life with Behaviors - they are what makes them move, jump, shrink, or even change color.

The catch? You don't need programming experience to get started.

Behaviors are simple bits of logic added to your game objects. Together, they create unique and sophisticated game mechanics.

To add behaviors to an object, click it and select "Edit". From the object panel, click on "Behaviors", and the behavior editor for that Object will open.

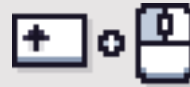


How Behaviors Work

The behavior editor is where all of your game's logic gets built.

On the left side of the editor is a menu with all behaviors, grouped by category. To add new behaviors to your Object, click one of the behavior buttons, making it appear in the editor window.

Tip: You can arrange behaviors by clicking, holding, and dragging them. You can also use shift+click to move a behavior individually.



Clicking on any behavior will open its properties panel, where you can edit its associated properties. The properties panel bottom also has a “Delete” button to remove this behavior and any of its links.

Tip: In each behavior’s properties panel, there is a “What’s This” button to tell you more information on that behavior.



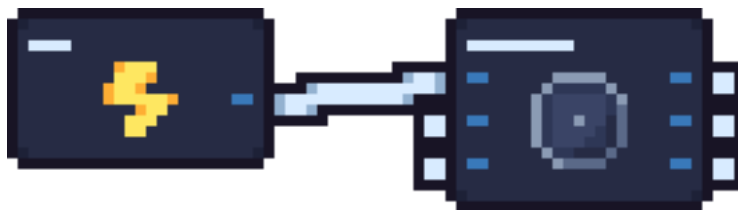
Linking Behaviors

Behaviors can be linked together to build advanced game logic and interactions.

Each behavior has a set of Ports on its left and right sides. Ports on the left side are input ports, and ports on the right side are output ports.

To add a link between two behaviors, click on an output port, and drag a link to another behavior’s input port. The link will turn green when it connects properly and can be released. When there is a link you no longer want, click on it to delete.

Whenever a behavior is triggered, it will send a message on its output links, triggering any connected behaviors.



There is a “Once” trigger behavior in the picture above - it activates once the game starts.

The “Once” behavior’s output connects to a “Sound Effect” behavior’s input: Meaning that this sound plays once when starting the game.

The “Sound Effect” behavior output doesn’t link to any other behaviors. Still, if it were, it would trigger after the sound effect starts playing.

Behavior Categories

- **Triggers:** Initiated by events occurring as the game plays, such as key presses or object collisions. These trigger behaviors will, in turn, trigger other behaviors that make up your game logic.



- **Logic & Math:** Perform arithmetic or control the flow of your logic.



- **Components:** Perform a specialized task, such as playing a sound or adding a force to the Object.



- **Properties:** Allow access to update or extract the properties of a game object directly.



- **GUI:** Add User Interface widgets to the game world, such as bars, windows, and labels.



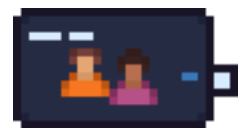
- **Game Flow:** Affect the game's flow by restarting, pausing the game, or changing levels.



- **Mobile device:** Used to complement the experience on exported apps.



- **Multiplayer:** Used to complement the experience on multiplayer Games.



- **Behavior Bundles:** Your library of saved bundles, used to add pre-made logic to your game.



Sprite Editor

When creating a new object type, a simple block sprite is generated for it automatically.



Every object type has an image associated with it, called a Sprite.

You can edit this sprite using the built-in sprite editor. To open it, click on the object edit panel's image, saying "click to edit".

Creating, Uploading, and Browsing Sprites

From the Sprite Editor, you can edit Sprites using the editing tools or select a new sprite.



To upload a sprite from your computer, click the Upload button and select the local image you wish to upload.

If you wish to use a built-in sprite instead, click Browse and choose one of the Asset packs or the displayed list of the previously added sprites.

Your new sprite image automatically saves when you click OK.

Animations

From here, you can also create new frame-by-frame animations. You can play these animations by selecting them using the "Animation" behavior.

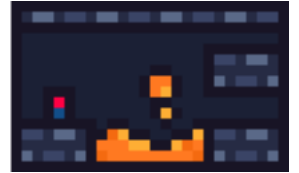


Find more information on the "Animation" behavior here:

https://flowlab.io/behavior_handbook/#properties-animation 

Game Levels

Games usually contain one or more sections or areas called “levels”, where the player will progress through them as they advance.



Levels can also be screens for menus, cut scenes, or end credits scenes.

Game Levels Panel

Inside the game levels panel, you can add new levels, delete, switch between them, or even change its name, background color, and display order.

The “New Level” button will add a new level to the game, and the Load button will switch to that level.

To edit the background color of a level, click on the color swatch next to each level name.

Switching Levels with Behaviors

While the game is running, you can switch levels using the “Next Level” behavior.



Find more information on the “Next Level” behavior here:

[https://flowlab.io/behavior_handbook/#game-flow-next-level.](https://flowlab.io/behavior_handbook/#game-flow-next-level) 

Game Layers

Access the layers panel by clicking the Layers button in the bottom toolbar. Each Game Layer has its object library that contains only the objects created in that layer.

There are three “layers” in the game world where objects can live:

Game World Layer:

The game world layer is the foreground of the game, where the action and game mechanics occur.



Background Layer:

The Background layer is behind the Game World layer. Background objects have no physics such as gravity or collisions and do not contain behaviors.



When editing the background layer, other layers will become translucent to provide a better view.

User Interface Layer:

The User Interface layer is in front of the Game World layer. Objects in the user interface layer also have no physics, but they can have behaviors.



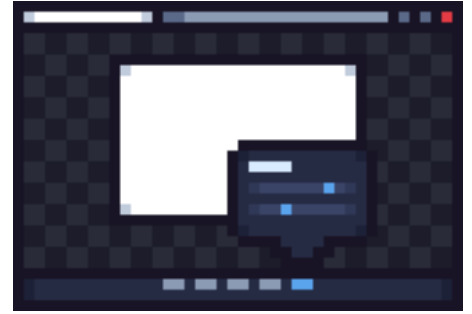
The User Interface Layer is the best place to put buttons, menus, or health bars - since they will render on top of the game and will not scroll.



Game Settings

Clicking on the Settings button in the bottom toolbar will open up the game settings panel.

From here, you can choose a name for the game, change the screen width and height, make it an online multiplayer game, or even allow your team to access and edit your game.



You are free to change all of these settings at any time.

Running the game

Flowlab has two modes: edit mode and play mode. Edit mode is where you make changes to the game, and play mode runs the game for testing or playing.

Edit Mode



Play Mode



To switch to play mode and test the game at any time, use the triangle Play button on the bottom toolbar.

Tip: The escape key it's a useful shortcut to toggle between the play and edit mode quickly.

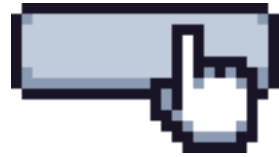


Sharing and Exporting

The simplest way to share a Flowlab game is to share its link.

Click on the “Copy Share URL” button next to each game on the [My Games](#) page to copy the game’s play link.

To embed a game on an external website, click on the “Copy Embed Code”, then paste the code into any web page to embed the game directly.



Exporting

You can also export your games as desktop or mobile apps if you have an Indie, Education, or Studio account.

You are free to distribute all your games through any store such as Steam, Apple, Google Play, and Amazon Appstore.

To export as a desktop or mobile app, go to the My Games page and click the Export button.

Once the app-building completes, a download button will appear next to the exported game.



The End

And that’s about everything you need to know to get started!
Ready to create a game and bring your ideas to life?