ConantPhysics
Construct 2 Reference

TOPIC	PAGE					
	1	**2**	**3**	**4**	**5**	**6**
	Vocabulary	Behaviors	Objects	Events	Actions	Tips

Basic Vocab:
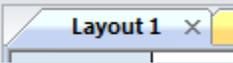
Layout	Layout 1 ✕	All of the visuals in your game
Event Sheet	1 Event sheet 1 ✕	All of the "brains" of your game
Start Page	t 1 Start page ✕	This is the introduction page. Close this once your project is started
Object	Insert New Object — Double-click a plugin to create a new object type from:	A visual based "thing" that is input into your layout
Event	Add event	A command in the event sheet that initiates an action.
Action	Add action	The resulting direction for the computer to complete when an event is triggered.
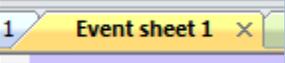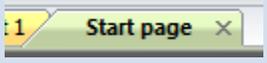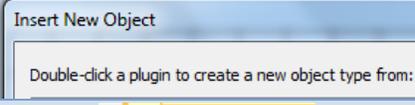
Layout Vocab

These describe options available in the left and right windows around your layout
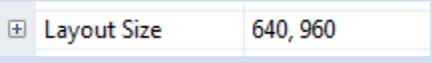
|		
Layout Size	⊞ Layout Size 640, 960	This is the size that your game can occupy (including off screen space) Edit by clicking once on the layout.
Window Size	⊞ Window Size 640, 960	This is the size that the screen will be. Your layout size can be bigger than the window size. Edit by clicking "project properties" after clicking the layout once.
Positioning objects	⊞ Position 320, 544 ⊞ Size 300, 300	The point (0,0) is the top left corner. Numbers will generally represent pixels. On an iPad 264px = 1 inch.
Layers window	Layers ⇥ ✕ ➕ 👁 🔒 ✏ 🗑 ⬆ ⬇ ☑ 🔒 Layer 0 1	Layers of artwork. If this confuses you, don't use it. It is sometimes helpful to make a background layer. (click layers in the right window)
Objects window	Objects ⇥ ✕ All 'Layout 1' objects ⊙ ⬤ Button ⃝ CircleBig	The things that you have made to interact with in your game. Tip: delete objects if there are accidental duplicates. (click objects in the right window)
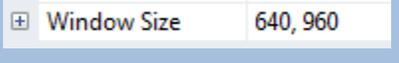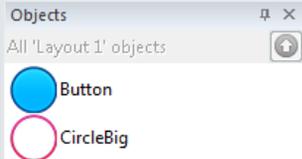
Behaviors

These are prepackaged actions for an object. These are our most commonly used behaviors
To add a behavior, click on an object once, then in the properties window at left, select "behaviors"

Drag and drop		Drag & Drop	Make an object drag and droppable. Exactly what it sounds like.
Bound to layout		Bound to layout	Do not allow an object to leave the area of play. Works well with drag and drop.
Pin		Pin	Pin an object to something else. This glues two items together to temporarily act like one.
Bullet		Bullet	This forces an object to move continuously in a straight line. Don't just imagine bullets.
Destroy outside layout		Destroy outside layout	Delete an item when it leaves the area of play. The more objects there are, the harder the computer works. Works well with bullet behavior
Physics		Physics	Forces objects to obey the rules of nature. There are a lot of options of motion for this one. Does not play nice with bullet or solid
Sine		Sine	Causes an aspect of your object to continuously change such as size, position, angle, etc. I use it to cause objects to wiggle.
Fade or Flash		Fade Flash	A visual effect that we add at the end of our simulations that give things a more finished feel. Causes an object to fade or flash
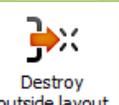
Behaviors are meant to be shortcuts to speed up your creative process. They are general actions for an object to follow, so if you are looking for a very specific application of a behavior that you cannot find, then you will likely have to cause it to happen using the event sheet.

An example of this would be the drag and drop behavior. Sometimes there will be areas that you do not want an object dragged over (maybe the instructions). Just using the behavior will not be enough to fully accomplish your goal. You would have to create an event in the event sheet that triggers when your conditions are met. (dropped object overlaps>move object)

Don't be afraid to use a behavior in an unusual way! In one of my simulations, I had data points on a graph acting like bullets. This caused a graph that appeared to be progressing forward in real time.
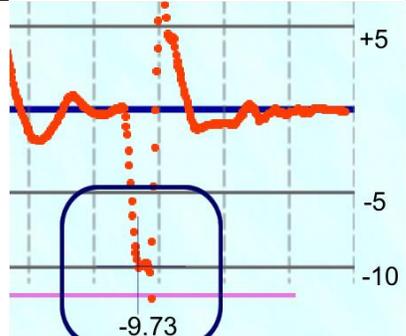
Objects

Below is our list of most commonly used objects in Construct 2.
To add an object, double click a blank space on your layout.

Sprite	Sprite	This is any artwork that does something. The most common object. When in doubt, use a sprite.
Text	Text	Text that is displayed. If you want words, this is it.
Text box	Text box	Not to be confused with text, this requests the user to input information This is an input box
Tiled background	Tiled Background	Used for large pictures that repeat. This will perform slightly better than a sprite in terms of computer run speed.
Keboard or mouse	Keyboard Mouse	Informs the computer that there will be a mouse or keyboard input to listen to.
Touch	Touch	This allows touch devices to work, and it also allows you to use acceleration data on the device. A mouse qualifies as touch.
Particles	Particles	A visual effect to simulate fireworks, explosions, water, etc. It will continually spray small sprites out in a pattern of your choosing.
Slider Bar	Slider bar	This create a slide bar that the user can adjust the value of.

There are lots of objects that do very unique things. Before you try any of the objects not on this list, look them up on the scirra.com website. They will do very unique things that you may or may not be able to control correctly.

One example would be the "button" object. At first appearance, it seems that it is the natural choice for creating a button. However, in all of the apps that we have created, we have never used the "button" object. It is much harder to control than a simple sprite, so we never use it. All of our buttons are sprites.

When in doubt, make it a sprite.

Sprite

"Events" in the event sheet

These are the cause for things to happen. They must come first in the event sheet.
To add an event, click "add event" in your event sheet.
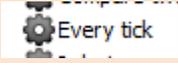
System> every tick	Every tick	Always complete your action.	Example: Enemy always faces the player
System> Compare variable	Compare variable	Compares a number to an existing variable. Activate when true.	Example: Player win when score is 10
System> Every x seconds	Every X seconds	Initiate an action every set amount of seconds.	Example: Create a new enemy every 5 sec
System> Repeat	Repeat	Repeat an action several times in a row, very quickly	Example: Create 20 enemies at the start
System> On start of layout	On start of layout	Activate as soon as the game starts.	Example: Start by randomly placing coins.
Sprite> On collision	On collision with another object	Activate when a sprite hits something	Example: Win game by hitting finish line
Sprite> Compare X or Y	Compare X Compare Y	Activate when the position of a sprite is a particular value.	Example: Unlock when a key is put in the door
Slider Bar> On changed	On changed	Activate when the slider is moved	Example: Adjust game values with a slider
Text Box> On text changed	On text changed	Activate when the user inputs a value.	Example: Calculate input when text is entered
Touch> On touched object	On touched object	Activate when clicked	Example: All buttons would be done this way
Keyboard> On key pressed	On key pressed	Activate when a particular key is pressed	Example: Reset when "r" is hit
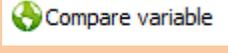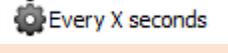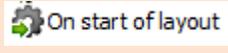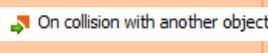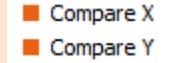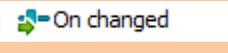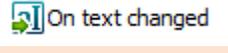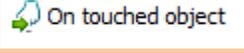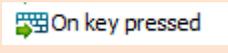
→ 👆 Touch	On touched start	▬ Se
⚙ System	Live = 0	▬ Se
		⚙ Se
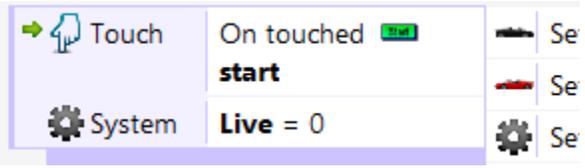
You may link two different events together to be more specific with your simulation. To do this, drag a new event on top of an existing one. The end result would look like the picture at left.
Both conditions must be true in order to activate.

The sheer volume of options for your simulation makes things very intimidating. The only way for you to break through this is to create a safe space for yourself. In this case, find a few events, and try to use them a lot. You will find that you eventually outgrow those events, and will adopt a few new ones. This natural growth will continue until you are "fluent" in Construct 2 Events.

"Actions" in the event sheet

These are the follow up instructions for an event. These describe what should happen as a result.
To add an action, click "add action" after an existing event.

General Object Actions

Object> rotate	■ Rotate clockwise ■ Rotate counter-clockwise ■ Rotate toward angle ■ Rotate toward position	Various methods of spinning your object	Example: Turn an object to face player
Object> Set visible	■ Set visible	Make an object visible or invisible	Example: Hide a button after its pressed
Object> Destroy	■ Destroy	Delete an object	Example: Destroy an enemy
Object> Spawn another	■ Spawn another object	Give birth to a new object	Example: Create a bullet out of a gun
Object> Set position	■ Set position ■ Set position to another object	Move to a new position	Example: Move a sprite when its clicked
Object> Set size	■ Set scale ■ Set size ■ Set width	Change the size of an object	Example: Shrink a clock when time runs

System Actions

System> Create object	Create object	Create a new object in a designated spot	Example: Create a object on a button press
System> Restart Layout	Restart layout	Reset your game to the start	Example: Restart button
System> Add to	Add to	Add to an existing global variable	Example: Add to the score
System> Wait	Wait	Pause for a set time before executing the next action	Example: Wait, then display a message

Behavior Specific Actions

|			
PIN BEHAVIOR >pin to object	Pin to object	Glue one object to another	Example: Stick two picture together
PHYSICS BEHAVIOR >set gravity	Set world gravity	Change gravity, or make it zero.	Example: Top down view should have zero gravity shown
PHYSICS BEHAVIOR >set velocity	Set velocity	Set an object's speed	Example: Move an object
SINE BEHAVIOR >set active	Set active	Start or stop the sine behavior	Example: Shake a button when its pressed

Text Actions

Text >set text	Set text	Command the text to display something	Example: Display the score
Text >append text	Append text	Add something to the end of existing text	Example: Show the order items were selected

Just like with events, pick out a few actions that you rely most on. Explore when you are ready.

General Tips
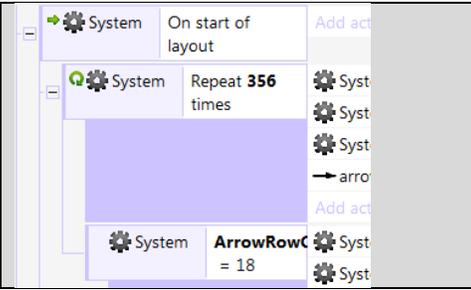
Try different combinations of events, actions and behaviors. Exploring is a great way to learn.

Be as specific as you can.
    Imagine the computer is Forrest Gump: Dumb, but very good at following instructions.
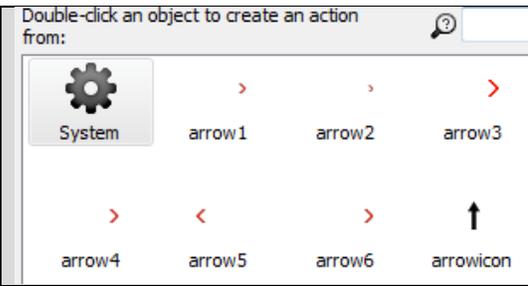
Be specific with your file names. Just calling something "button" will get very confusing when you have several buttons.

The computer keeps track of details for every object. If you are typing an object name, add a "." to the end of the name and the computer will show what information it knows. (see right)
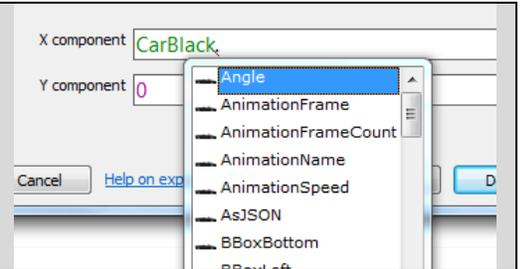
Make Art Last.

Some projects get shown to students partially finished. Ugly projects that function can still teach.

Scirra.com has a great forum with lots of helpful people. Someone has run into the same problem as you before, and they probably posted about it on the forum.