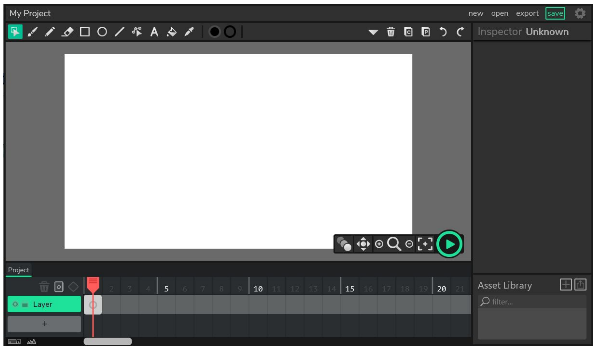
Lesson 1.2

3.7.2020

What is the Wick Editor?

DAILY OBJECTIVE

Students will be introduced to the Wick Editor, a free and open source animation and game creation environment. We'll be using the Wick Editor to create our TEALS end of semester projects.



The tool is accessible from:
www.WickEditor.com

DEFINITIONS

Here are a few terms that you might find useful today.

1. **Wick Editor:** A free, online tool for creating games, animations and interactive projects. This is the tool students will be using to create their own projects throughout the TEALS project.

2. **Point and Click Game:** An interactive game where users are tasked with navigating an environment solely by clicking with a mouse.

3. **JavaScript:** A programming language primarily designed for use on the web. This programming language is used on most modern websites to create interactions, animations, and allow the website to function. In Wick Editor, JavaScript is the primary programming language. This is not the same as Java.

4. **Canvas:** The area of the Wick Editor where creators can place drawings and images to create animations and games.

5. **Timeline:** The Timeline is the "brains" of a Wick Editor project. It contains frames that store the visuals of your animations and games.

6. **Inspector:** The area of the editor where users can view and change the properties of any selected object.

7. **Frame:** An element on the timeline that contains visual and audio information while also describing when to show or play that content.

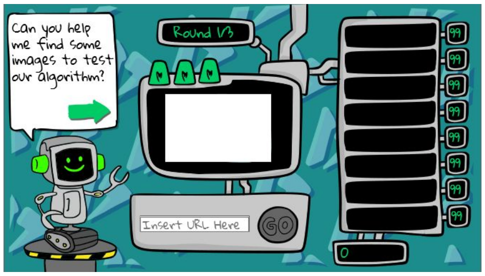
8. **Scene:** A colloquial term for a frame used to display a setting. For the purpose of this guide, scenes and frames are almost identical!

Section 1: What can we make in Wick Editor?

Let's take a look at some example projects created in Wick Editor! We'll review these as a class.

1. **"Locked in the Museum"** - A point and click adventure game, with text based puzzles that incorporates AI Text Analysis using Microsoft Azure Cognitive Services.

2. **"The Climate Lab"** - An image-finding game that challenges players to find images that an AI image Analysis program can identify!

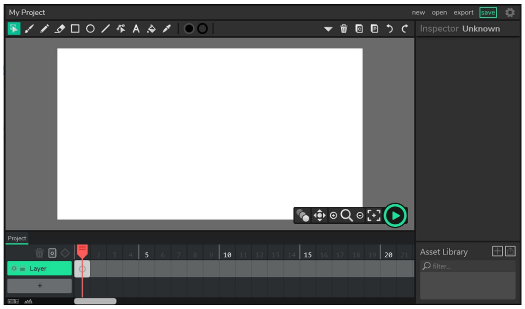


3. **"Good Dog"** - A simple role-playing game where players use their facial expressions to simulate a service dog's reaction to their owner having a medical issue.

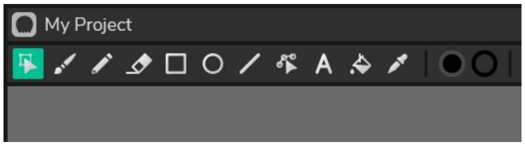
Section 2: The Drawing Tools

Every Wick Editor project starts by creating something on the Canvas.

2.1 Creating a drawing



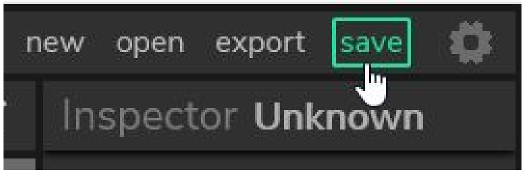
1. Open the Wick Editor by going to editor.wickeditor.com.



2. Start in Wick Editor by experimenting with the drawing tools.

   Be sure to use as many as you can, including:

   a. **Brush**
      i. Create basic lines and shapes.
   b. **Cursor**
      i. Manipulate shapes on the canvas (scale, rotation, position)
   c. **Eraser**
      i. Remove parts of the drawings on canvas.
   d. **Shape Tools**
      i. Create Rectangles, Ellipses, and Lines
   e. **Text Tool**
      i. Add a title, or signature, to your project.



3. Save your work by pressing the “save” option in the upper right of the editor. You should save as often as you can, without slowing down your workflow!

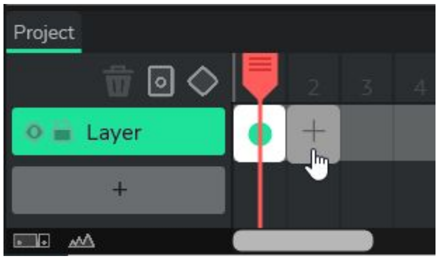
B O N U S	With any additional time, experiment with the advanced drawing tools such as the Fill Bucket, EyeDropper, and Path Cursor tools. Use them to further customize your image.

Section 3: Making an Interactive Project

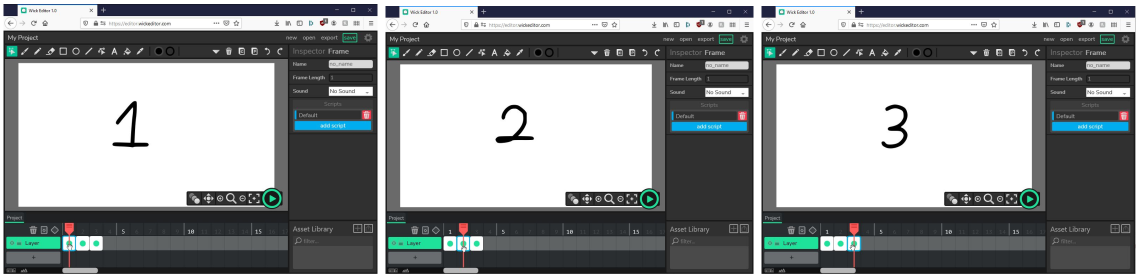
We'll move on by creating an interactive project.

3.1 Making a multi-frame animation.

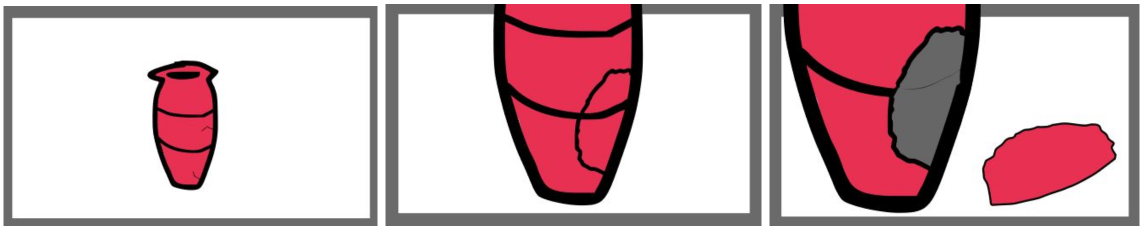
1. Start by creating 3 frames on the timeline. You can do this by pressing the "add frame" button while highlighting an empty space on the timeline.



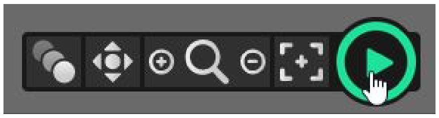
2. When creating frames, you will see new frame elements on the timeline. You can move between each frame by selecting each frame object!



3. On each frame, create an image that changes slightly from frame to frame. In our example, we've created an object (a red vase) that cracks and breaks from frame 1 to 3.



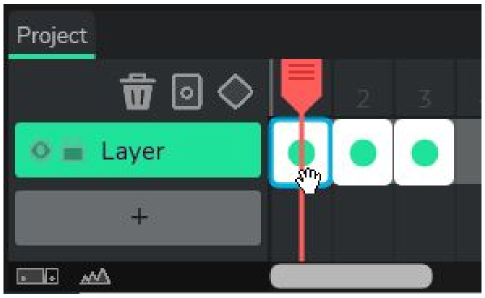
4. Once you're done, press the play button (the large green button on the bottom-right of the canvas). The three frames should play quickly, making an animation!



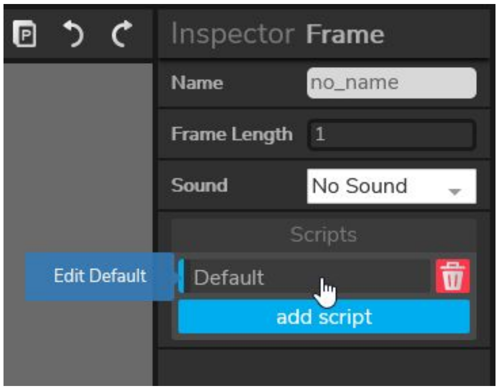
3.2 Preparing The Project For Interactions

1. To convert the project into an interactive project, we start by adding code that tells the Wick Editor to stop and wait for user input.

   Select the first frame on the timeline.



2. Next, select “default” from the Scripts menu in the Inspector on the right. The code window should now appear.

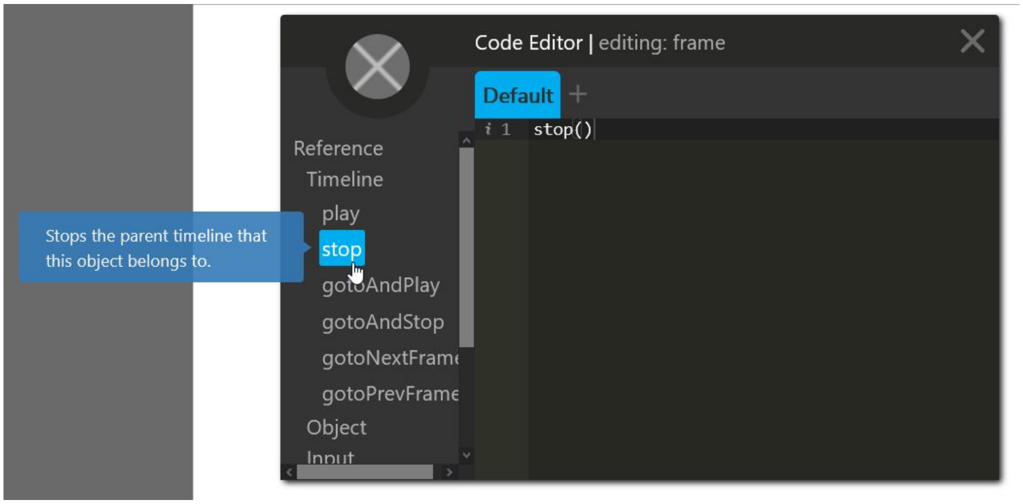


Name: no_name

Frame Length: 1

Sound: No Sound

3. We’ll add the **stop()** script to this frame. This can be done by typing the command (don’t forget the parentheses) or by selecting the **stop** command from the interactive reference on the left of the coding window.



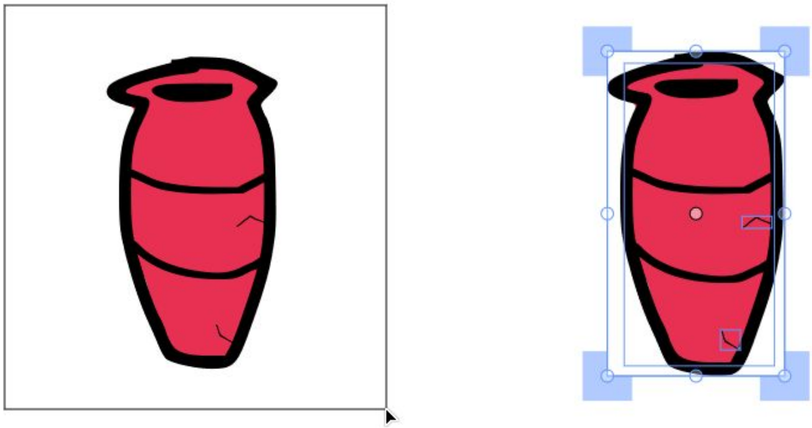
Stops the parent timeline that this object belongs to.

```
stop()
```

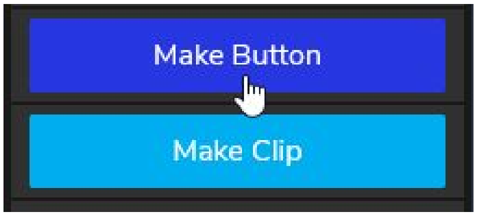
Closing the coding window and pressing the Play button now should cause the project to stop on the first frame!

3.3 Adding Interactive Elements

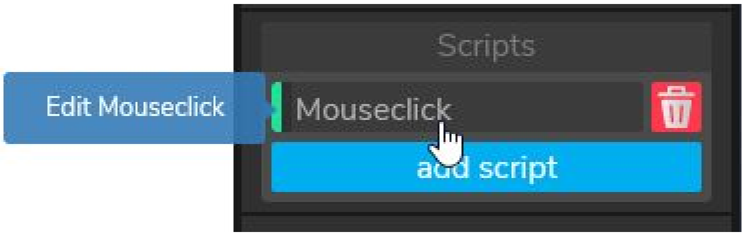
1. Next, we'll add an interactive button to the first frame. Select your object using the Cursor tool. Create a selection box by clicking and dragging from an empty area of the Canvas.



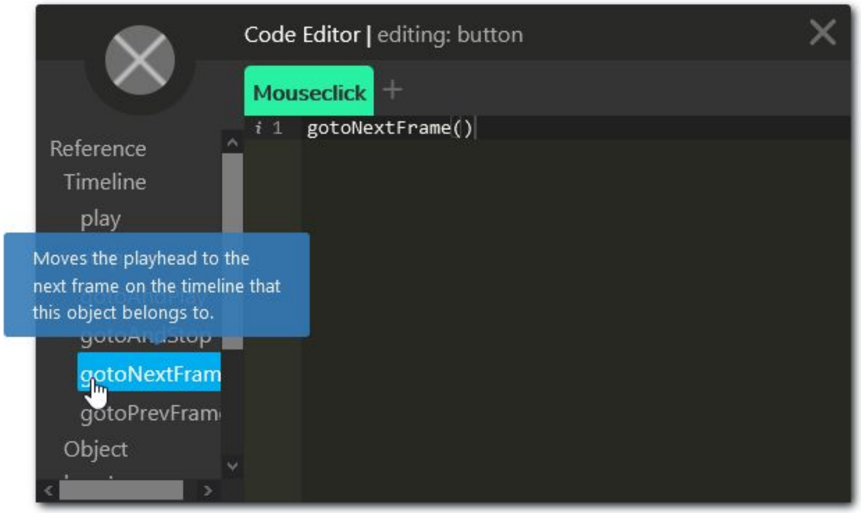
2. With the object selected, convert this object to a button using the **"Make Button"** option in the inspector.



3. With the object converted to a button, and that object selected, there should be a new option for a **mouseclick** script within the inspector on the right of the screen.



4. Open the **mouseclick** script, and insert the command **gotoNextFrame()** by typing the command, or selecting the command from the interactive reference.



```
gotoNextFrame()
```

Moves the playhead to the next frame on the timeline that this object belongs to.

5. Closing the code editor and playing the project will now allow you to click the interactive object, and it should bring you to the second frame! With this command, we now have the basis for an interactive storybook.

3.4 Challenges

You now have the basics of interactive projects under your belt! We'll move on to creating some more complex projects.

C 1	Never get stuck on a frame. Add a button to each frame that allows you to return to the previous frame using the gotoPrevFrame command. Remember, a button can be ANY drawing, or image, you make!
C 2	More than one ending. Add new options that allow a user to move through your project in a non-linear way. i.e. Frame 2 can move to either frame 3 OR frame 4, depending on which button is pressed. Use the gotoAndStop(İ) command to jump to any frame number!

3.5 Code Review

Here’s a list of all the commands you’ve used today in your challenges.

Command	Description
stop()	Stops the playhead of the animation until some input causes the project to play again or move to another frame.
gotoNextFrame()	Moves the playhead ahead to the next frame.
gotoPrevFrame()	Moves the playhead backward one frame to the previous frame.
gotoAndStop(i)	Moves the playhead to frame i and stops the timeline on that frame.