

ELEC 1020 LAB 3:

Interactivity

with Google Web Designer (GWD)



Google Web Designer

Outcomes of this lab

1. Learn the interaction techniques in animation
2. Create an animation with interactivity and control

Attention

1. BOTH basic and creative tasks needed to be posted on Facebook as a comment on the specified post
2. Save your work to 'D:\temp'

Scoring Rubric For the Labs

max. 4 (100%) out of 4	4 (100%) Exceeds Standard	2 (50%) Meets Standard	1 (25%) Approaches Standard	0 (0%) Fails
Evaluation of your production in terms of design and technical effectiveness	On top of the standard deliverables, students showed unusual insight, complexity, originality, or creativity with their message or choice of an art form or medium to express it.	Students completed ALL required messages and reproduced the expected art form or medium.	Students partially completed the required messages and reproduced the expected art form or medium.	Students FAILED to complete any required messages and reproduced the expected art form or medium.
	Work shows rich evidence that student researched his/her OWN ideas, art form, and the techniques of his/her art form.	Work shows full evidence that students completed the required ideas, art form, and the techniques of art form expected.	Work shows partial evidence that students completed some required ideas, art form, and the techniques of art form expected.	Work shows NO evidence that students completed any required ideas, art form, and the techniques of art form expected.

Lab Contents

1. Best Practice with GWD
2. Create a **Mobile-based Project**
3. Hyperlink and Audio
4. **Events**: Interactivity and Control

Best Practice with GWD (Critical Knowledge)

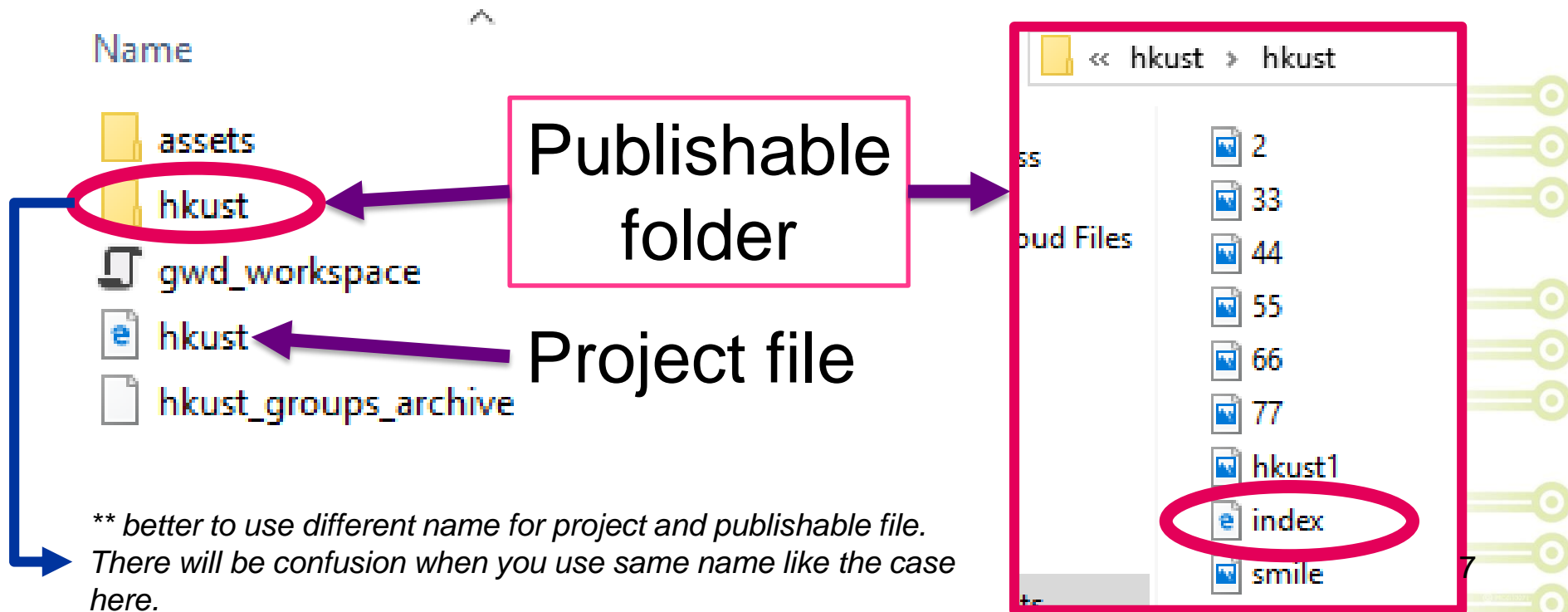
Best Practice with GWD

1. **Project Name and Publish Name** should be different.
(*Refer Slide 7*)
2. If there are nothing in preview, please make sure the elements are within the **boundaries**. (*Refer Slide 8*)
3. Check the position of the **blue seeker** before make any editing. (*Refer Slide 9*)
4. **Save** the file every few minutes.

Basic Practice 1: Project File vs Publishable File

1. When you
create your project:

2. When you
publish your project



Basic Practice 2: Workspace Boundaries

When you place your element here, it wouldn't be able to show in browser preview

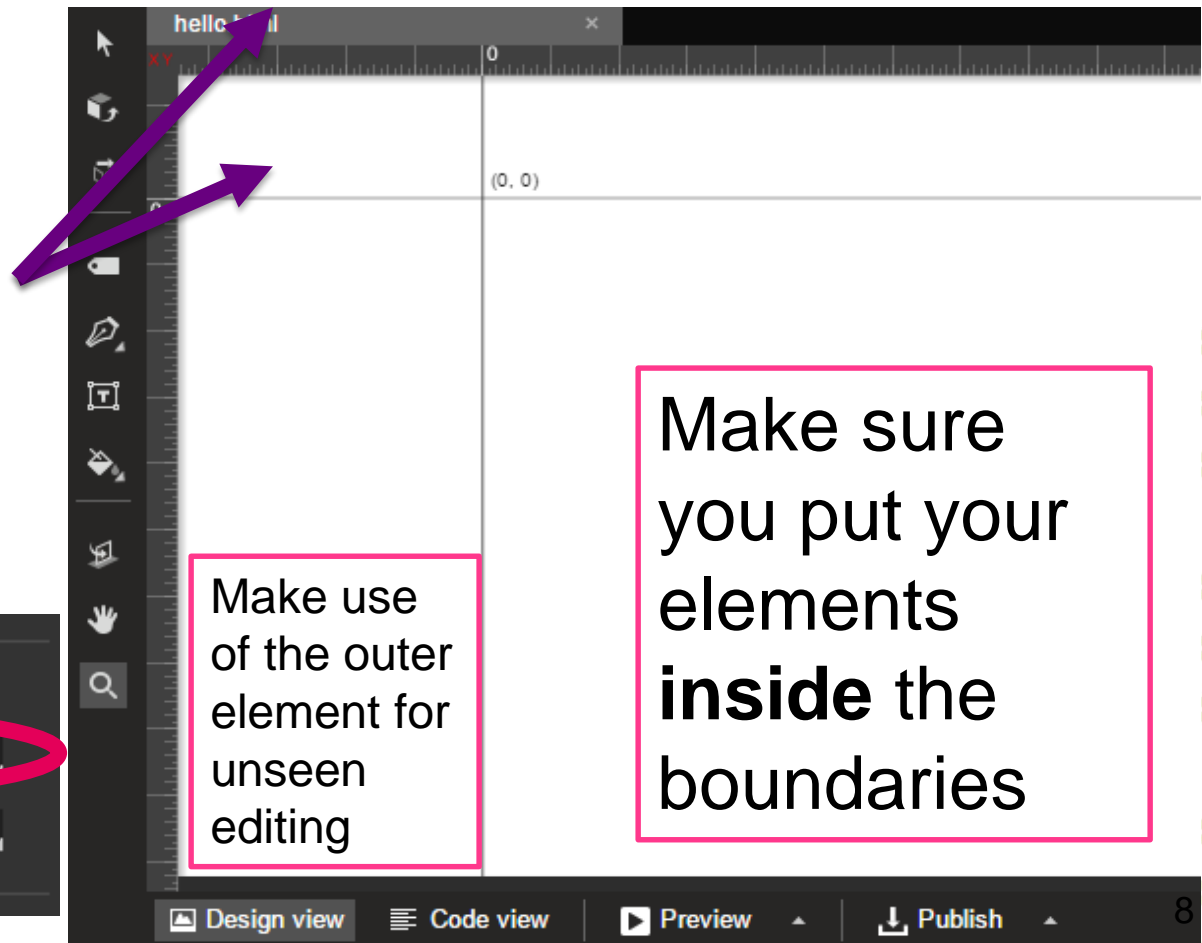
Check the properties to make sure the position of your element.

* {-ve} mean out of space

▼ Position and size

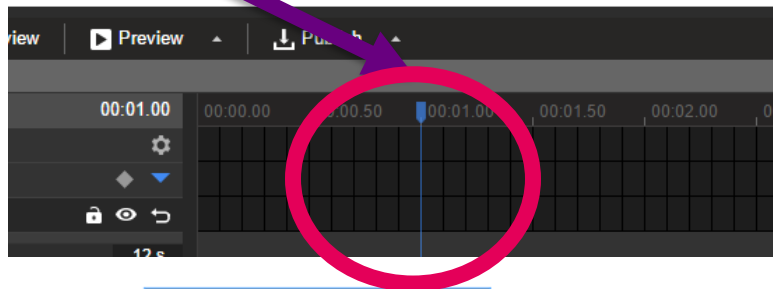
Left -316 px Top -327 px

Width 523 px Height 239 px



Basic Practice 3: Position of Blue Seeker

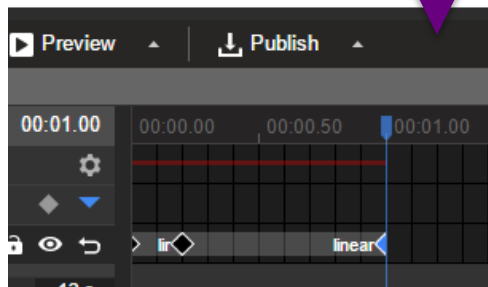
Blue
Seeker



Do not edit your element when your blue seeker is at random position



Consequence



If you move/edit your black element (e.g. move to the left), keyframe will added automatically.

Mobile-based Project

Creating a Project for mobile

1. Select Interstitial

2. Choose "Non-Google Ad"

3. Choose "Dual" orientation

4. Use custom dimension, set width to 320 and height to 480

5. Enter your fileName

The screenshot shows the AdMob mobile project creation interface. On the left, a sidebar lists ad formats: Ads, Banner, Expandable, Interstitial (highlighted in orange), HTML with Pages, CSS, Javascript, and XML. A pink box with the instruction '1. Select Interstitial' has a purple arrow pointing to the 'Interstitial' option. The main panel contains settings for the ad environment and orientation. A pink box with the instruction '2. Choose "Non-Google Ad"' has a purple arrow pointing to the 'Non-Google Ad' dropdown in the 'Environment' section. Below that, a pink box with the instruction '3. Choose "Dual" orientation' has a purple arrow pointing to the 'Dual' radio button in the 'Orientation' section. In the 'Dimensions' section, a pink box with the instruction '4. Use custom dimension, set width to 320 and height to 480' has a purple arrow pointing to the 'Custom' dropdown. The 'Width' field is set to 320 and the 'Height' field is set to 480. Below the 'Dimensions' section, a pink box with the instruction '5. Enter your fileName' has a purple arrow pointing to the 'Name' field. The 'Location' field is set to 'D:\ELEC 1020 Media Production\lab\lab 5\GWD'. The 'Animation Mode' is set to 'Advanced'. At the bottom right, there are 'OK' and 'Cancel' buttons.

Ads

Banner

Expandable

Interstitial

HTML with Pages

CSS

Javascript

XML

Environment: Non-Google Ad

Orientation: ☐ Single ☒ Dual

Dimensions: Custom Width: 320 Height: 480

Publish Size: Match ad slot size

Name:

Location: D:\ELEC 1020 Media Production\lab\lab 5\GWD

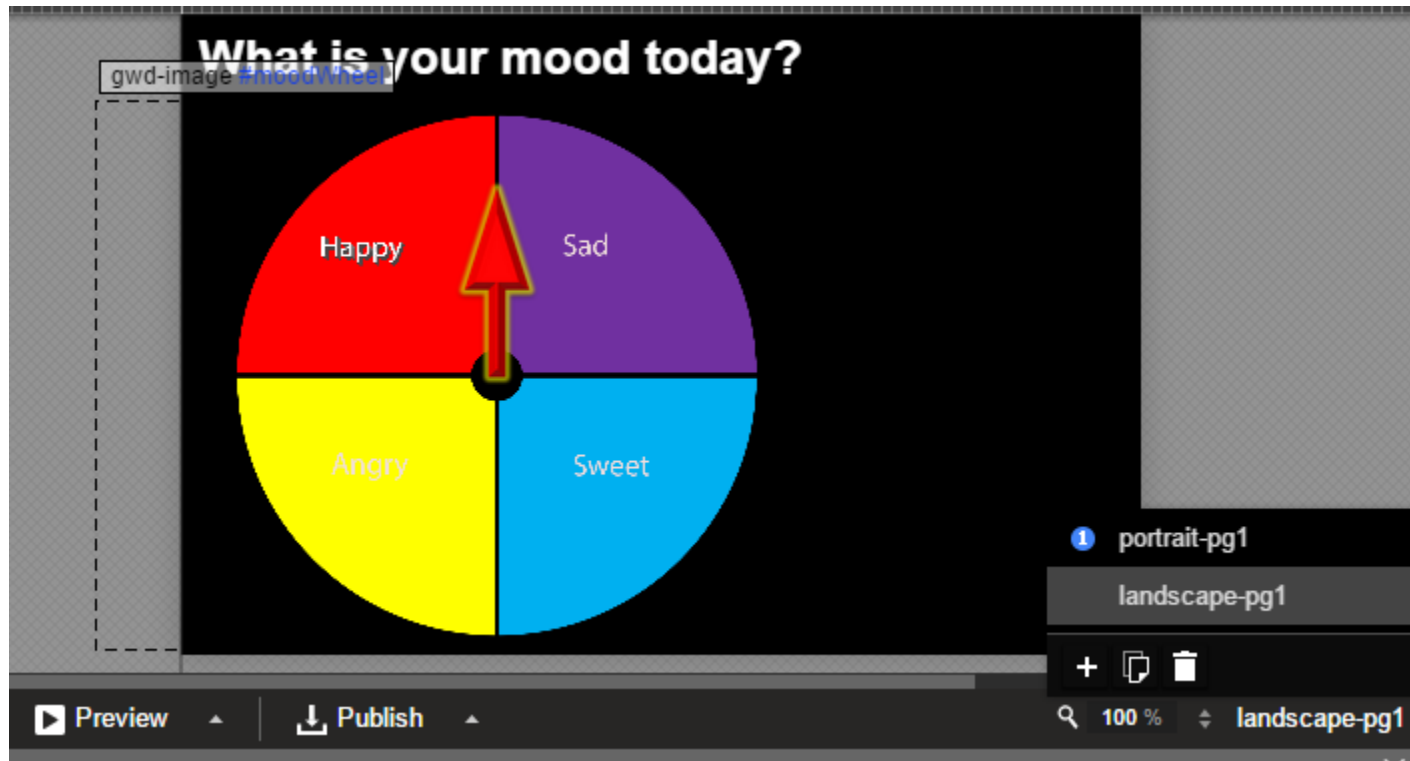
Animation Mode: Advanced

OK Cancel

Note on Mobile-based Project

1. You can specify the orientation, either
 - Portrait
 - Landscape
2. If you do your editing in the Portrait page, only the device in Portrait orientation can view your work
3. So if you would like to have your program available in both orientation, you need to do **double work.**

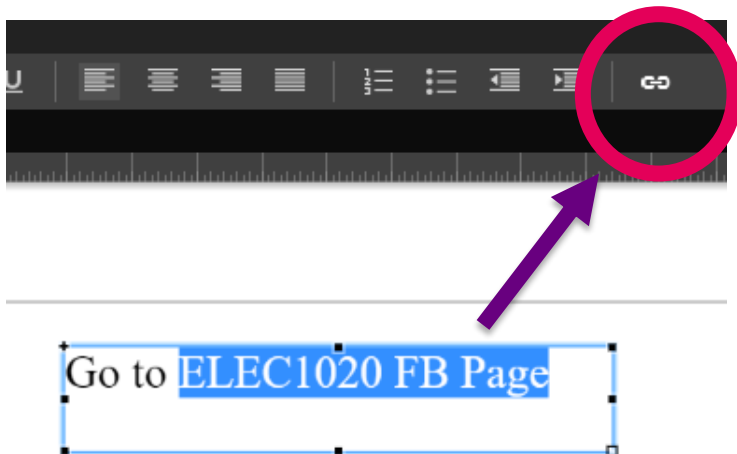
Switch between Portrait and Landscape



Hyperlink and Audio

Insert Hyperlink: Active Control

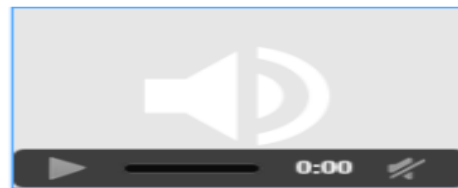
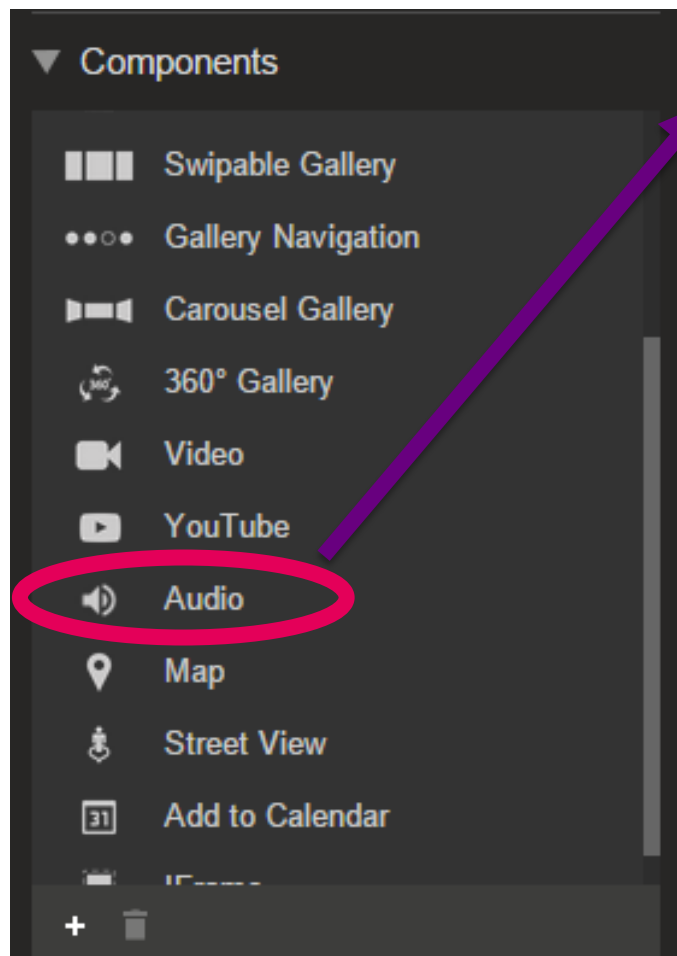
Create hyperlink

A screenshot of the 'Edit Link' dialog box. The 'Text to display' field contains 'ELEC1020 FB Page'. The 'Link to' section has a 'Web address' field containing the URL 'https://www.facebook.com/2016springelec1020/'. This URL is circled in red. Below the 'Web address' field is an 'Email address' field and a checkbox labeled 'Open this link in a new window'. At the bottom right are 'OK' and 'Cancel' buttons.

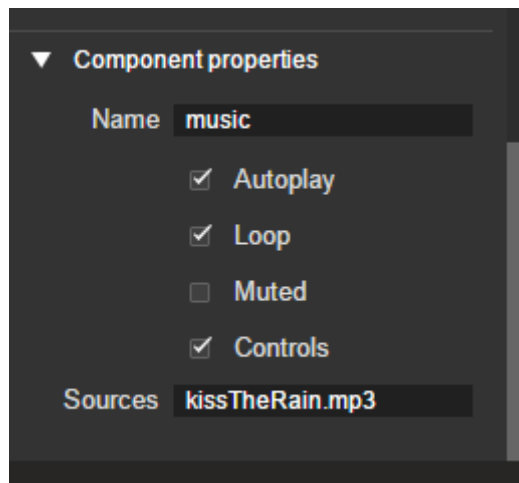
Code generated automatically in the code view

```
<body>
  <div class="gwd-div-17hg gwd-gen-tx0tgwdanimation"></div>
  <p class="gwd-p-16h6">Go to <a href="https://www.facebook.com/2016springelec1020/">ELEC1020 FB Page</a>
</p>
</body>
```

Insert Audio: Synchronicity



The component properties located at the bottom of the properties panel



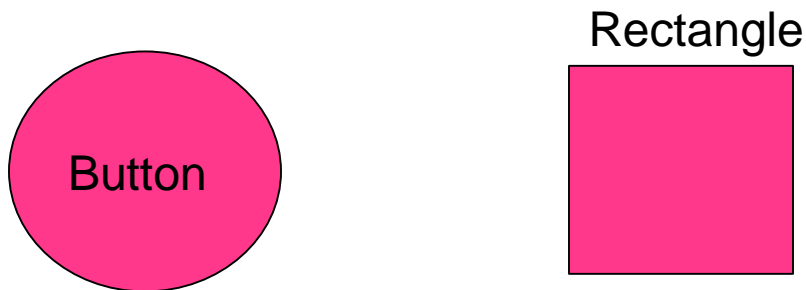
1. Drag it to the workspace

2. Set the sources of the audio

Event: Interactivity and Control

What is an interaction

- You trigger something, and it responds



- You have to define:
 1. Which object to be trigger? Button or Rectangle?
 2. Triggered by what? Clicking? Hover? Time?
 3. Triggering what? Button or Rectangle?
 4. After actions? Move? Play audio?

What is an interaction

- With one time line, how to have interaction?
 - We need label!

Label: start

Label: game start

Label: game end



- Triggered: by time
- Triggering: timeline back to start
- After actions: play animation

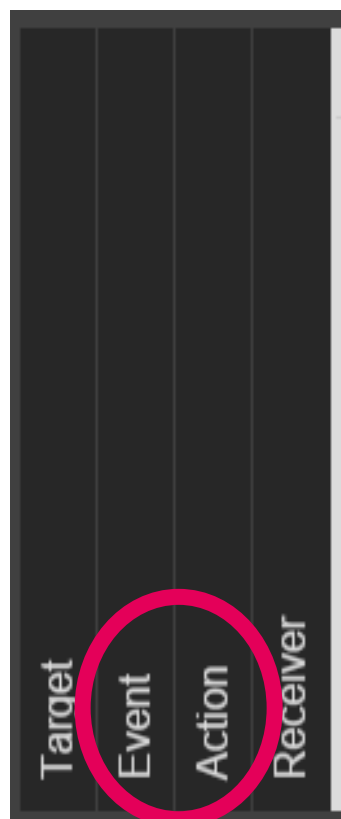
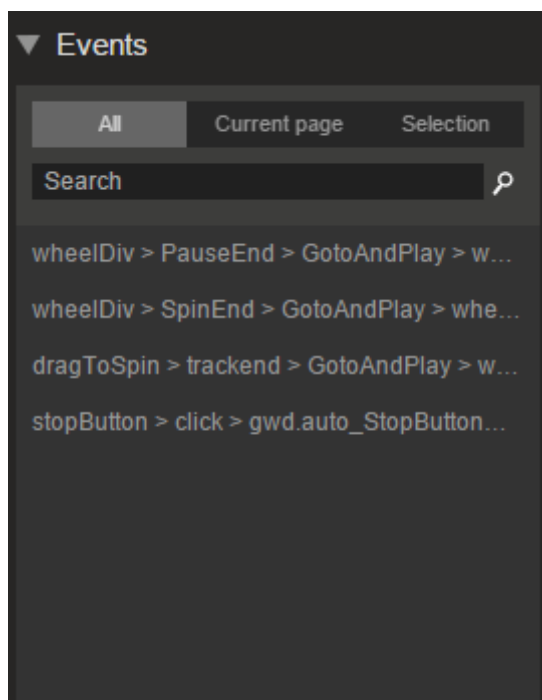
What is Event?

Define the interactions of GWD

- How an event is triggered:
 - timeline (when it reaches certain time)
 - gesture (**external input** from user)
 - button Control (**external input** from user)
- Example 1:
 - Triggered: when user click on a link/button
 - Triggering: an object
 - After actions: rotating
- Example 2:
 - Triggered: when an audio end
 - Triggering: itself (the audio)
 - After actions: replay

Event and Action

- Event Panel



Event:

Defines the next state of the program when the **Action** is the trigger.

Example

Event: spin the wheel

Action: Gesture dragging

Learning Outcome for today's basic task

Add & configure the following events:

1. Timeline event
 - Important note: “Add label” & “Add Event” (*Refer Slide 24-28*)
2. Gesture interaction
 - e.g.,. Mouse over (hover), Drag (Track End) (*Refer Slide 30-33*)
3. Button control
 - Create a button and insert a tap area. (*Refer Slide 35-40*)
4. CSS style
 - Set the element's style (Eg. Opactiy, width) according to the event (*Refer Slide 42-44*)

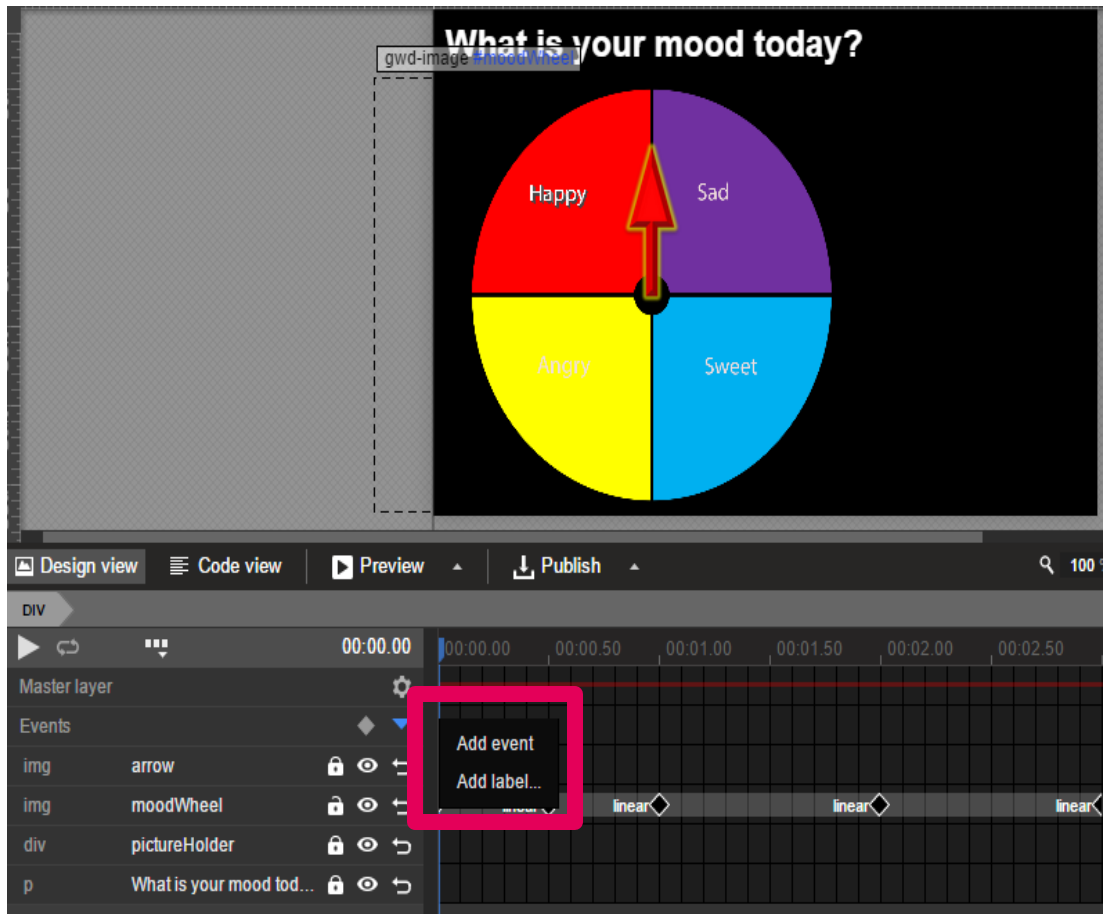
Demo Template 1

Please download the demo template at the URL below:

- <http://ow.ly/Zwu92>
- Unzip it to D:\temp\lab5

Timeline Event

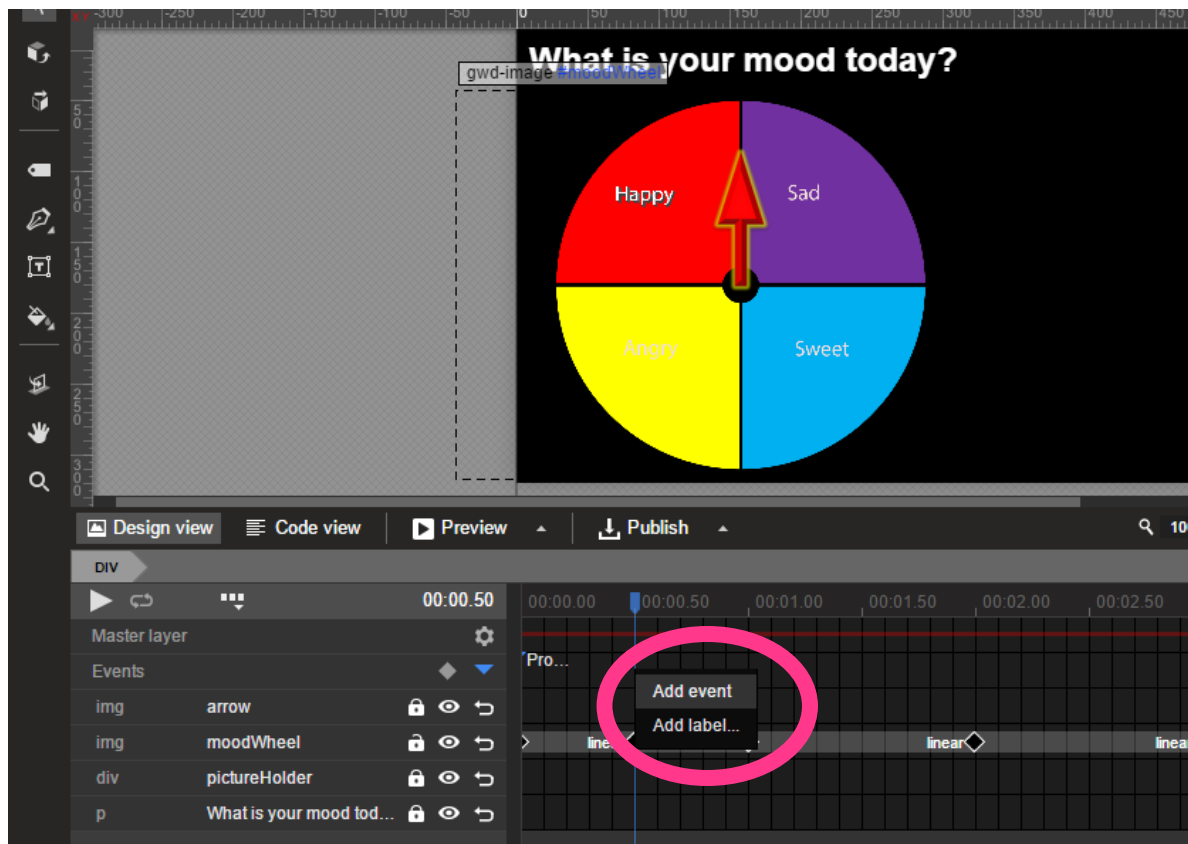
Step 1: Add Label



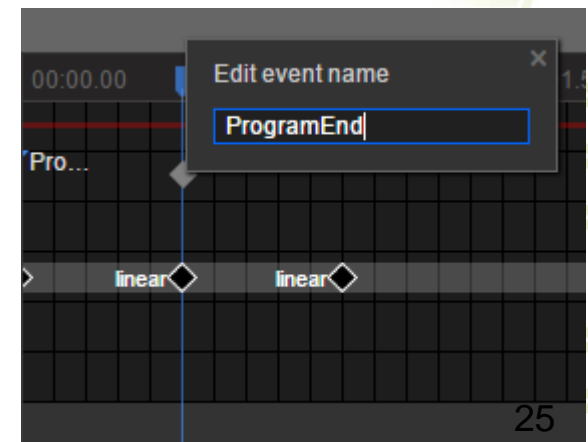
1. Right click on the event layer, and select "Add Label"
2. Name it as "ProgramStart"

Timeline Event

Step 2: Add Event

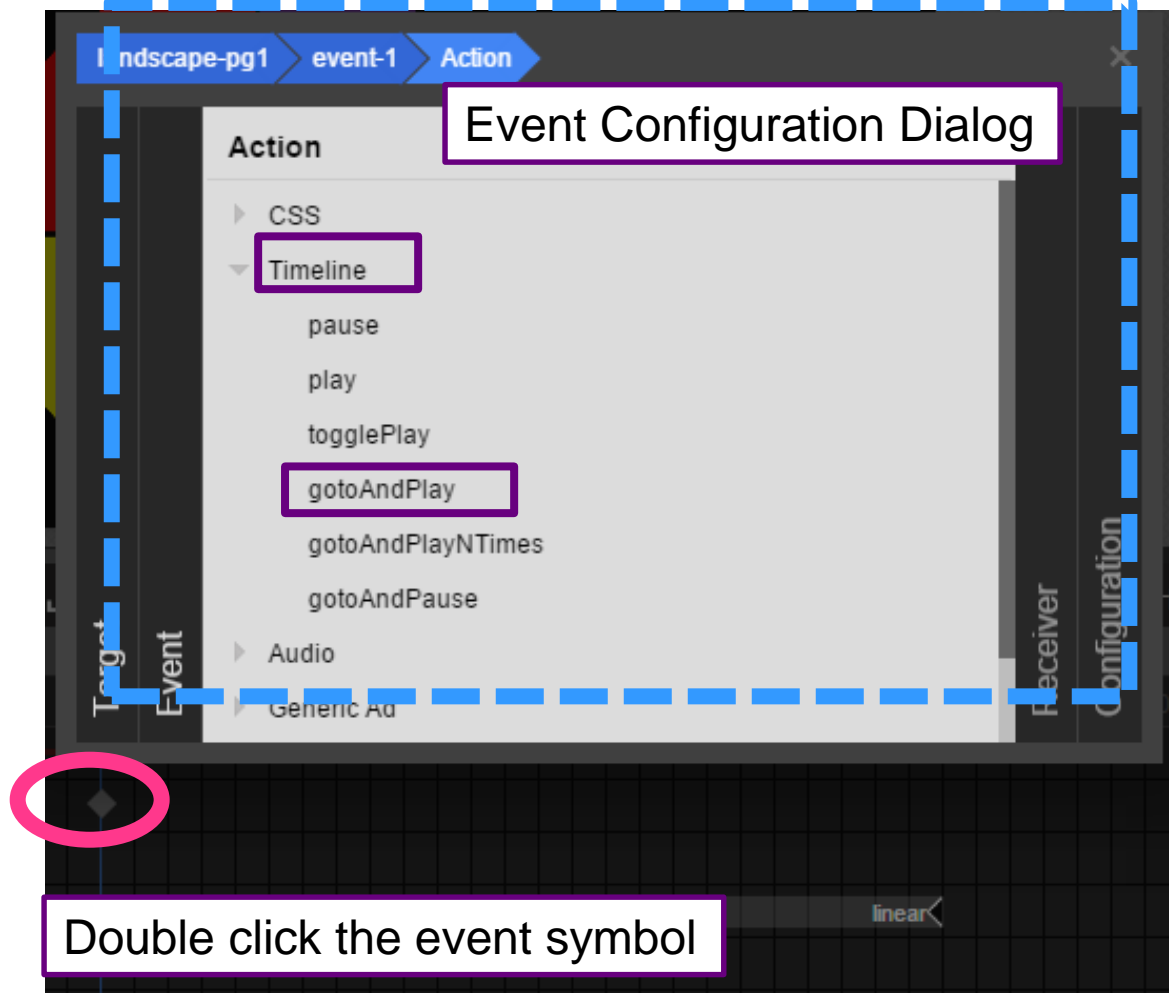


1. Right click on the event layer, and select "Add Event"
2. Right click on the **event symbol**, select "Edit event name" to change the name



Timeline Event

Step 3: Configure the Event

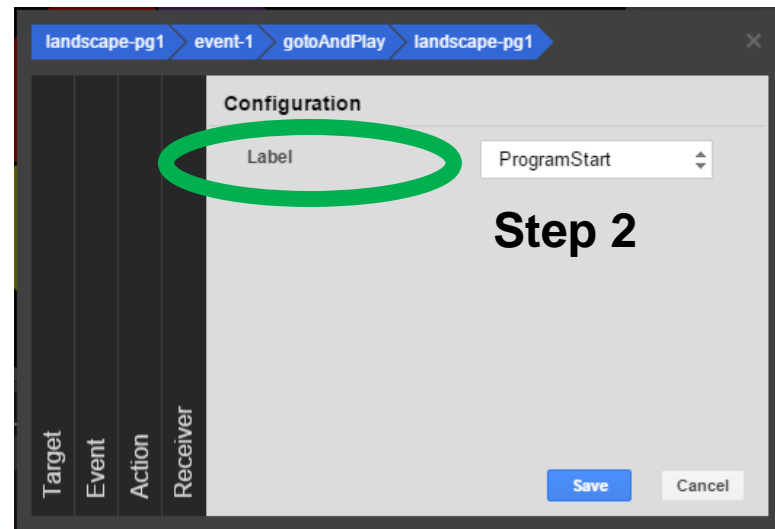


1. Double click on the event symbol to open the event configuration dialog
2. Under **Action**, select Timeline
3. Then select **“gotoAndPlay”**

Timeline Event

Step 4: Configure the Event

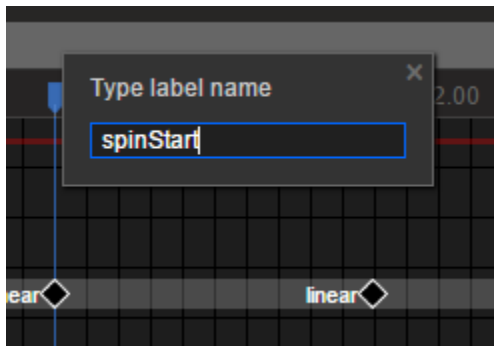
1. For **Receiver**, choose “landscape-pg1”
2. For **Configuration**, choose “ProgramStart”



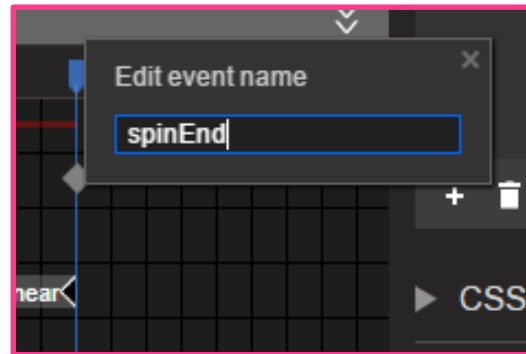
Timeline Event

Step 5: Repeat for spin Event

At 1s, add the label



At 3s, add the event



1. Timeline -> gotoAndPlay

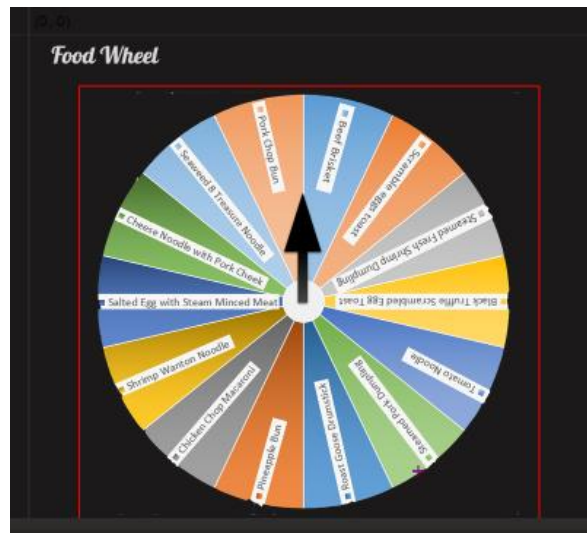
2. landscape-pg1

3. spinStart



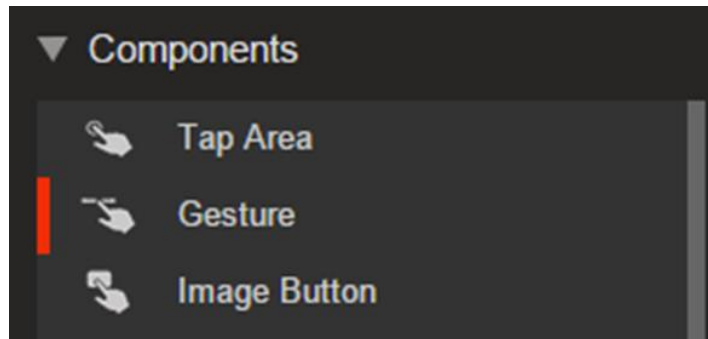
Gesture Interaction

- Interaction is done by “event” in GWD
- 3 main setting of an event
 - Triggered by (e.g., clicking) – *Active Control*
 - Triggering (e.g., multiple objects) – *Synchronicity*
 - After actions (e.g., back to start) – *2-way Communications*

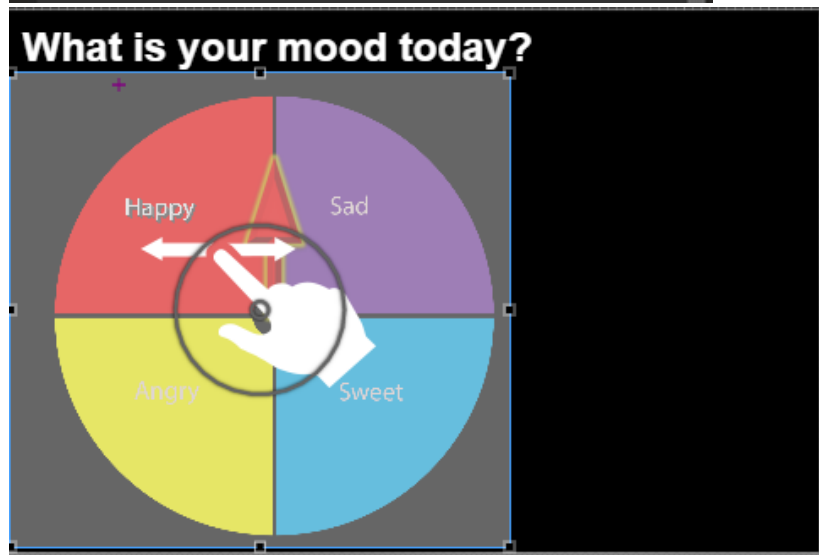


Gesture Interaction

Step 1: Insert Gesture Component



1. Open the **Components Panel**, drag the “*Gesture*” component to the workspace.
2. Rename the component to “*dragGesture*”.



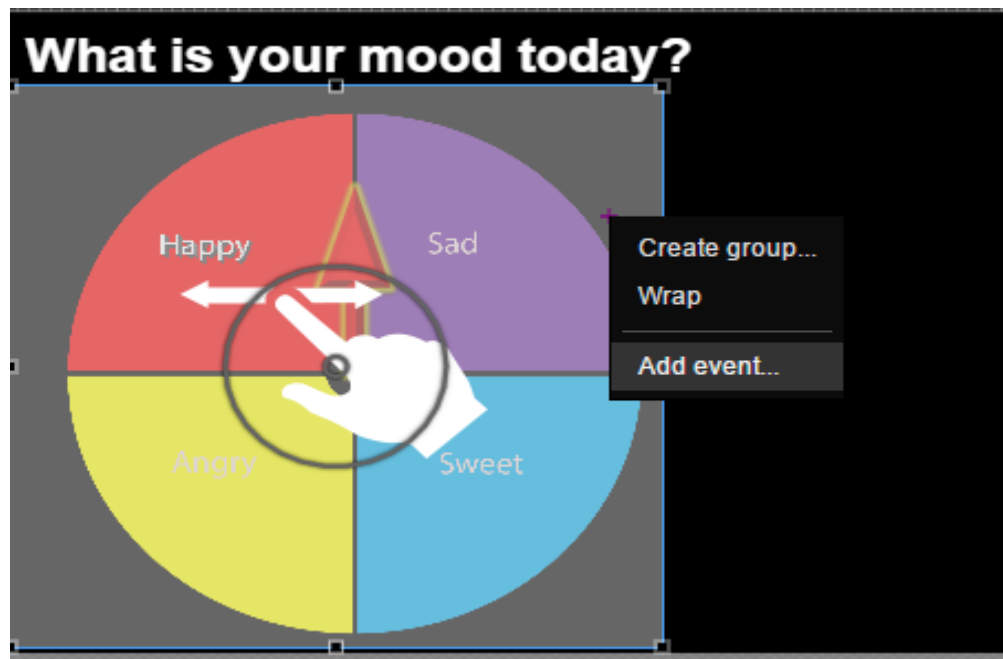
component	dragGesture	🔒	👁	↶
img	arrow	🔒	👁	↶
img	moodWheel	🔒	👁	↶
div	pictureHolder	🔒	👁	↶

3. Enlarge the “*Gesture*” component to fully cover the whole Mood Wheel.

Gesture Interaction

Step 2: Add Event to the Gesture

Right click on the “Gesture” component, and select “Add event”



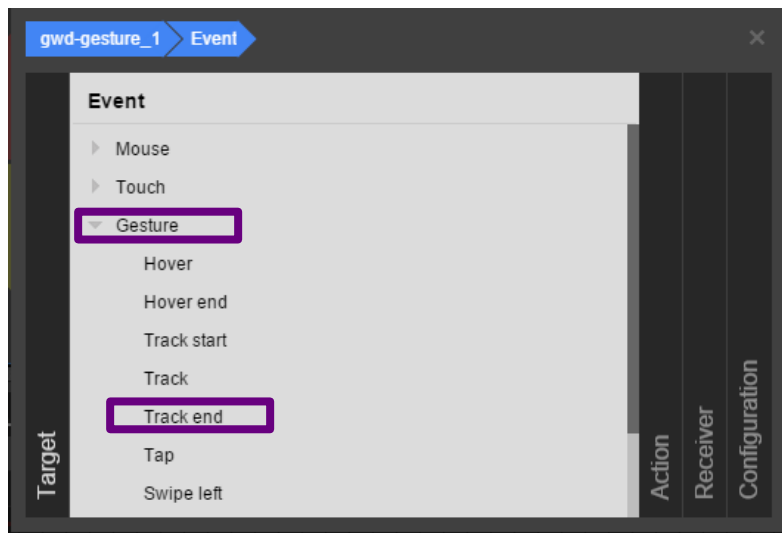
Gesture Interaction

Step 3: Configure the Gesture Event

1. Select Gesture

- Track End (mean: it will track your mouse movement and only trigger the event when your mouse movement end)

2. For Receiver, choose **landscape-pg1** (because the moodWheel is in the landscape-pg1)

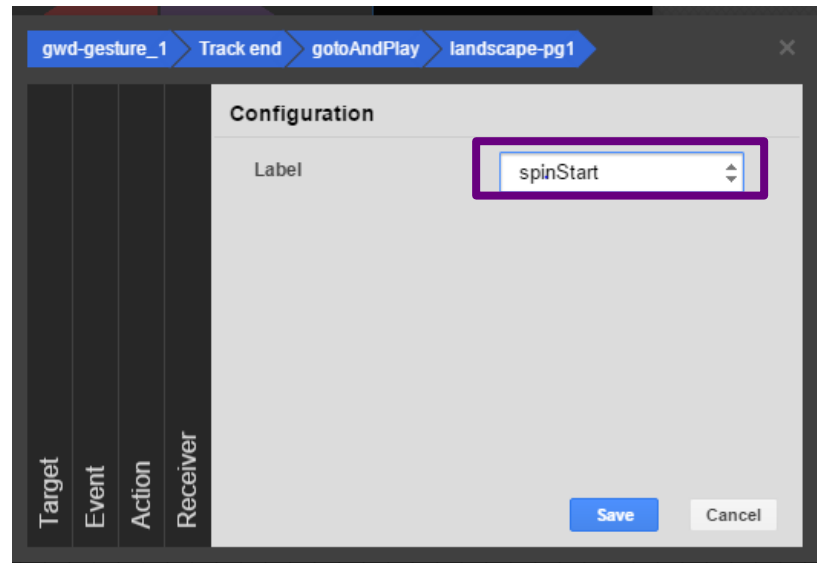
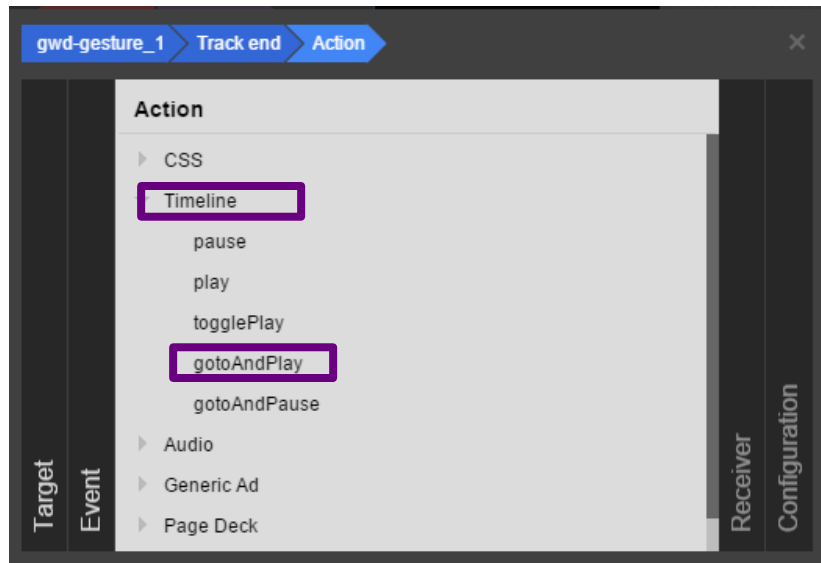


Gesture Interaction

Step 4: Configure the Gesture Event

1. For Action, select timeline -> gotoAndPlay
2. Configuration: spinStart

(**Intuition:** when user drag the wheel, we would like to start the spin Event)



Button Control

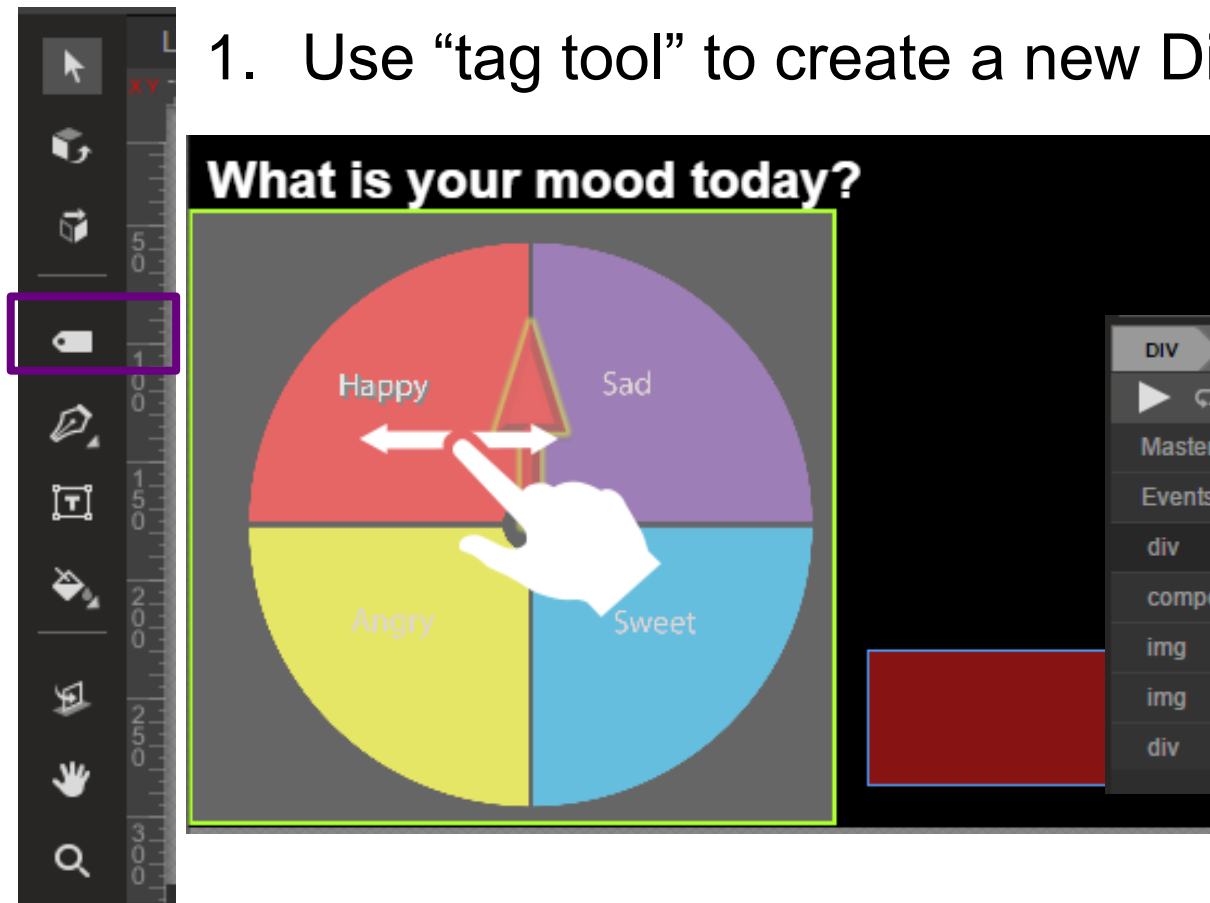
- Allow user to click it to perform an action
- 3 main events with Button control
 - onClick (e.g., clicking) – *Active Control*
 - submit (e.g., submit form) – *2-way communications*
 - Play/pause (e.g., start animation) – *Synchronicity*



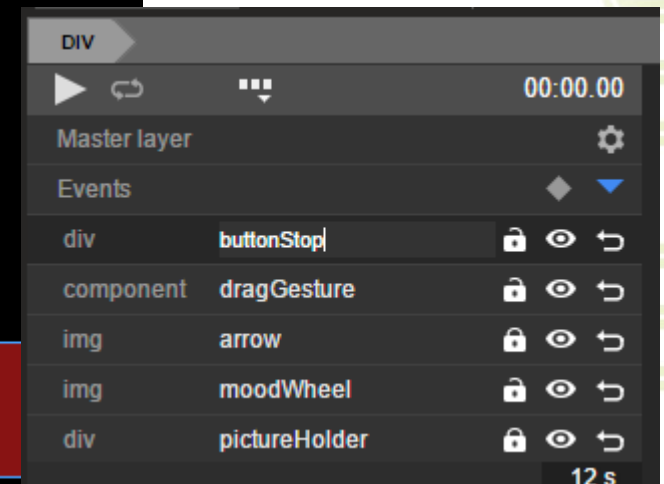
Button Control

Step 1: Add a Button

1. Use “tag tool” to create a new Div



2. Name the Div as “buttonStop”



Button Control

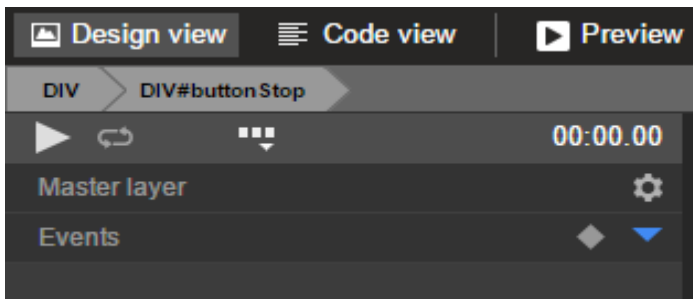
Step 2: Go inside the Div



1. Double click the “button” to go inside the Div Element



2. Once you are inside the Div Element, the button border color change to red



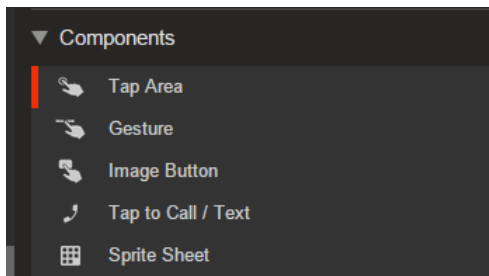
3. Or you can see it from the viewbar here, we are inside the DIV#buttonStop

Button Control

Step 3: Add Text & Insert Tap Area



1. Add the “Stop!” text to the button



2. Next, go to the components, and drag the “Tap Area” into the workspace

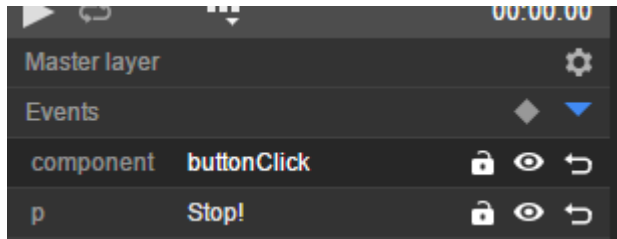


3. Enlarge the “Tap Area” to cover the button

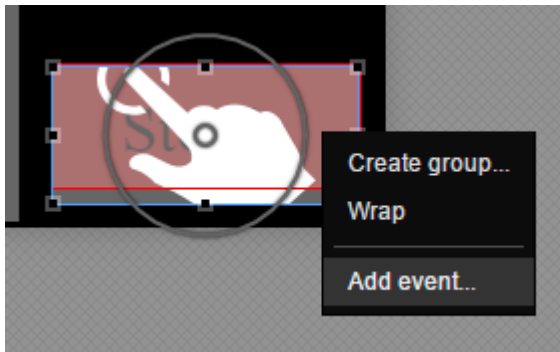
** “Tap Area” allows users to click with their mouse or touch with their finger*

Button Control

Step 4: Add Event



1. Name the “Tap Area” component as **buttonClick**



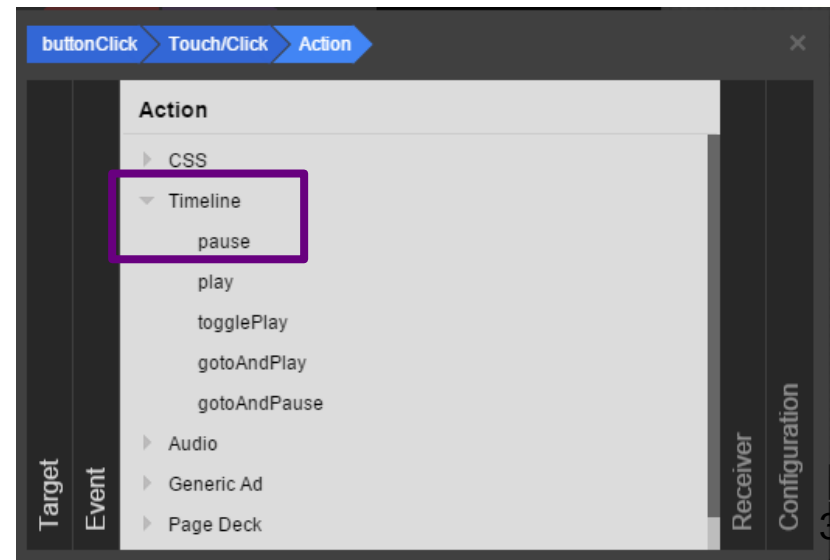
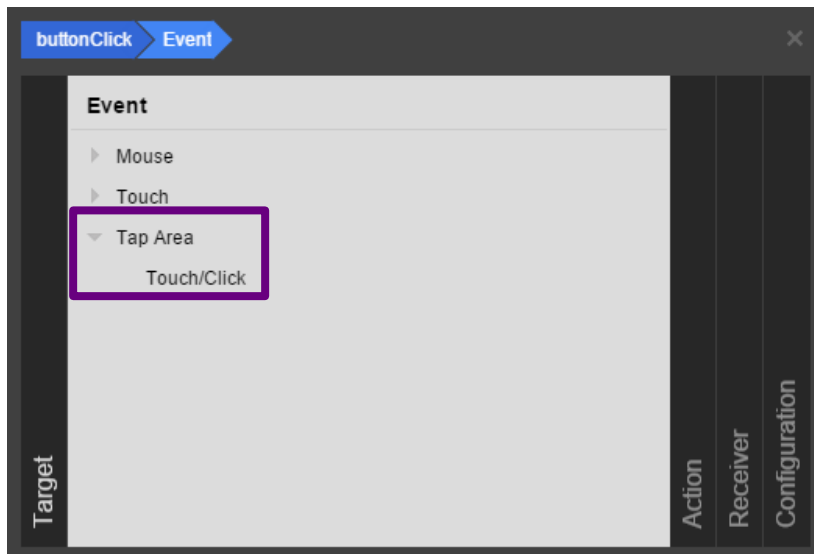
2. Right click on the “Tap Area” to **Add event.**

Button Control

Step 5: Configure the Event

1. Select Tap Area -> Touch/Click
2. For **Action**, choose Timeline -> **Pause**

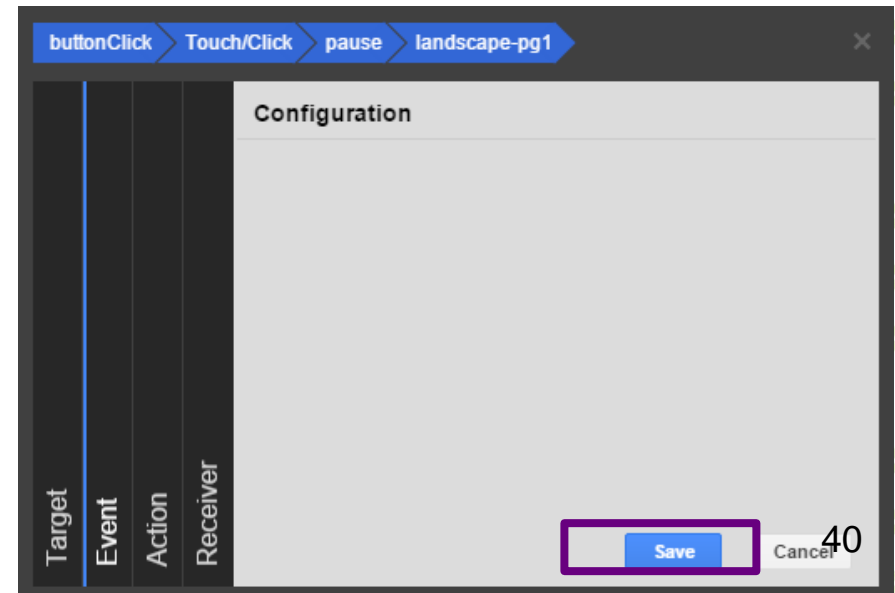
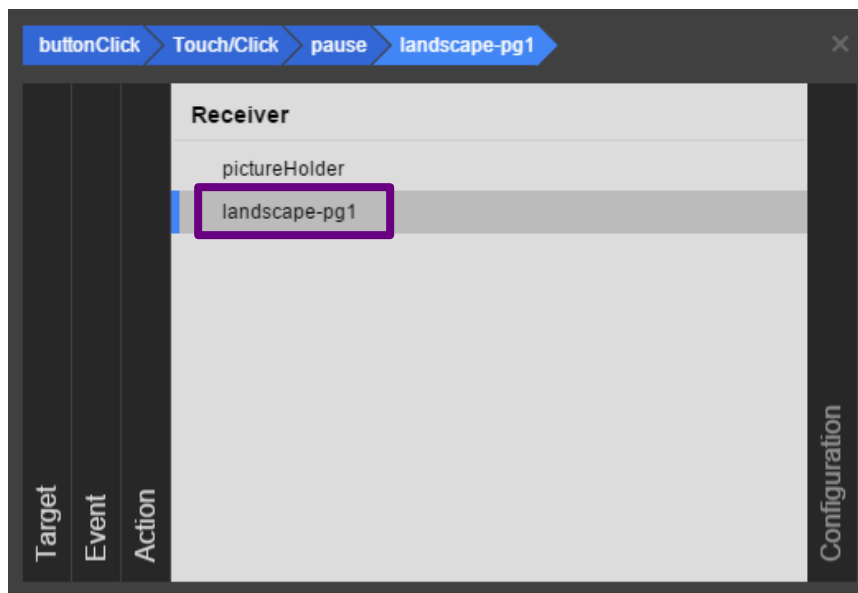
(Intuition: we would like to pause the timeline event when user click the button)



Button Control

Step 6: Configure the Event

1. For Receiver, stop landscape-pg1 (*because we would like to pause the spinEvent, which is defined in landscape-pg1*)
2. For **Configuration**, save



CSS Event

- CSS defines the style of the elements in properties, e.g.,
 - width
 - opacity
 - color
- It can be **triggered by external input** (e.g., Button clicked by user)
- It can be according to the keyframe in the timeline **animated automatically**. That's what you had learnt during the animation.

CSS Event

Step 1: Add Event on Stop Button

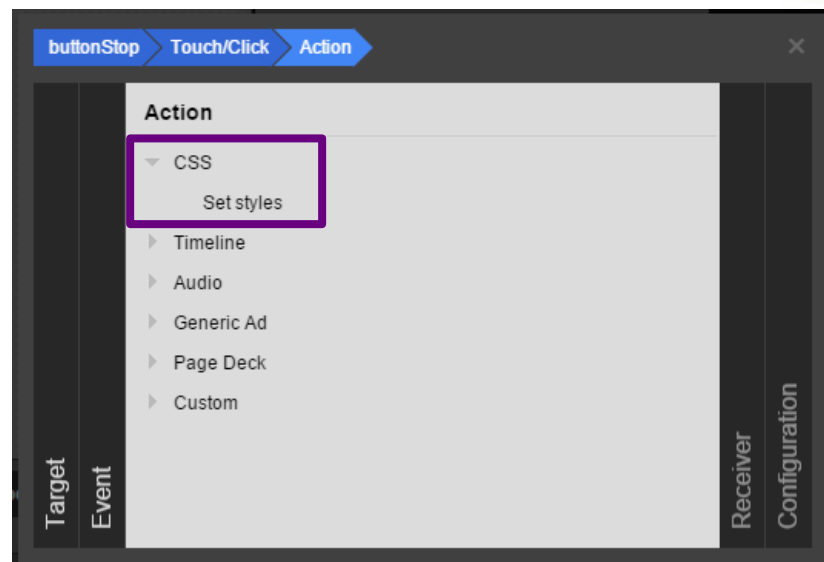
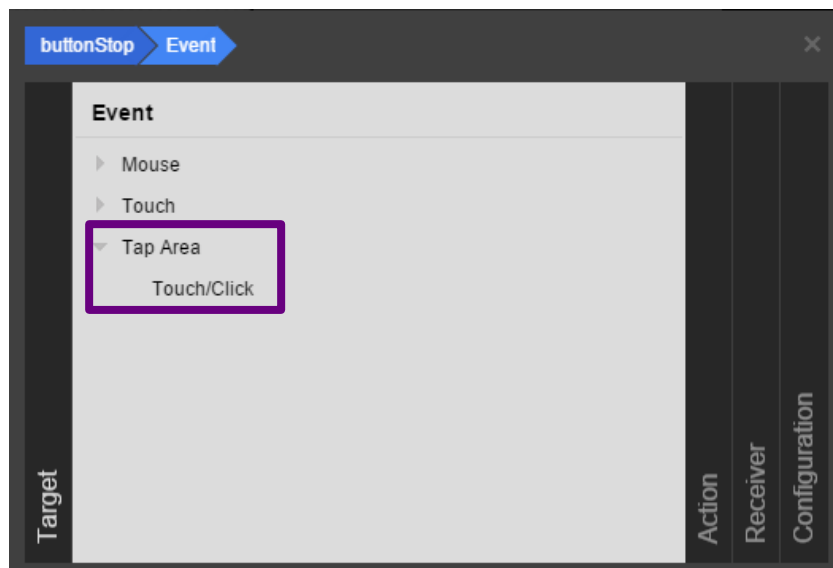


Goal: change Stop button's color from red to Green

CSS Event

Step 2: Configure the Event

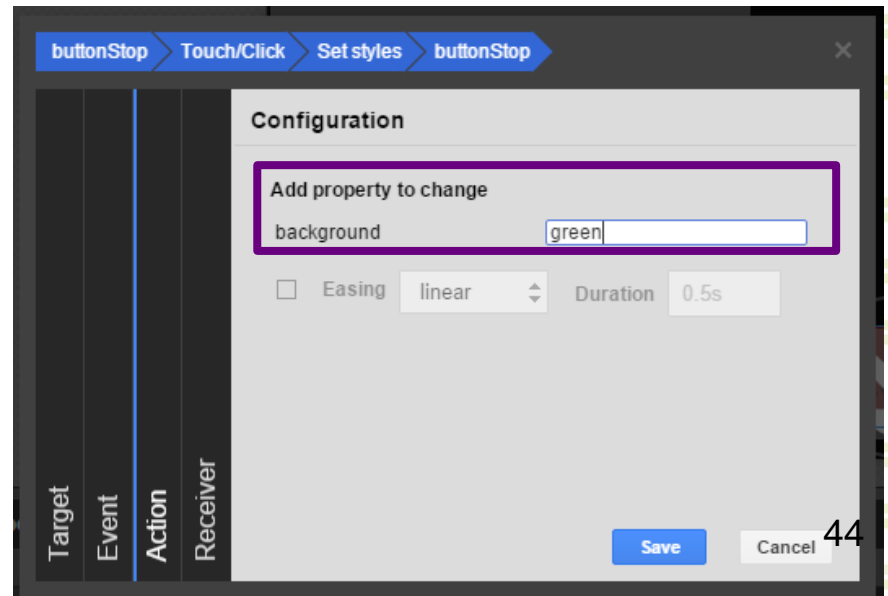
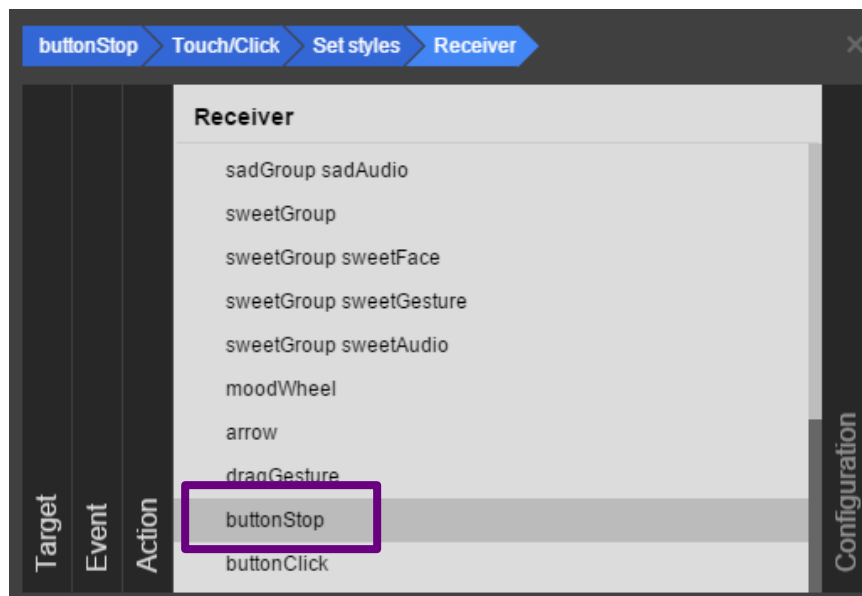
1. **For Event**, choose “Tap Area” -> Touch/Click
2. **For Action**, choose CSS -> SetStyles



CSS Event

Step 3: Configure the Event

1. **For Receiver**, choose buttonStop
2. **For Configuration**, click the (+) button to Add property to change
3. Set the **background** to green color



Basic Task

1. Complete a Mood Wheel program using a template as the example:

http://ihome.ust.hk/~pcng/Lab5_BasicTask_Completed/

2. Post your iHome link to the Submission Post in FB, together with your **Name** and **ID**.

Basic Task Explanation

1. When user **drag** the wheel:
 - The wheel should spin
 - The pictures start to change (synchronize with the spinning wheel in the background)
 - The spinAudio should sound
2. When user **click** the stop button:
 - The wheel should stop as well as the spinAudio
 - The picture in the background should appear in the foreground
3. When user **hover over** the picture:
 - The corresponding sound effect should appear
4. When user **spin the wheel again**:
 - The picture should be disappear, but it need to be continue running in the background to synchronize with the spinning

Steps for the Basic Task

1. Download the template (basic task) from <http://ow.ly/ZDpEY>
2. Add 3 events below to the program:
 - stopping spinning the wheel by pressing a button
 - start the “spinAudio” when user drag the wheel
 - set the “pictureHolder” to invisible when users drag the wheel for second time.
3. Add 4 photos of yourself that show the Mood state: “Happy”, “Sad”, “Angry”, “Sweet”
4. Add the sound effects corresponding to each of the mood states.
5. (Optional) Loop continuously “spinAudio” when the wheel is spinning
6. Upload to iHome, and post the link the Facebook

Creative Task (**max 2 marks**)

Describe and sketch your ideas for midterm project using the following template:

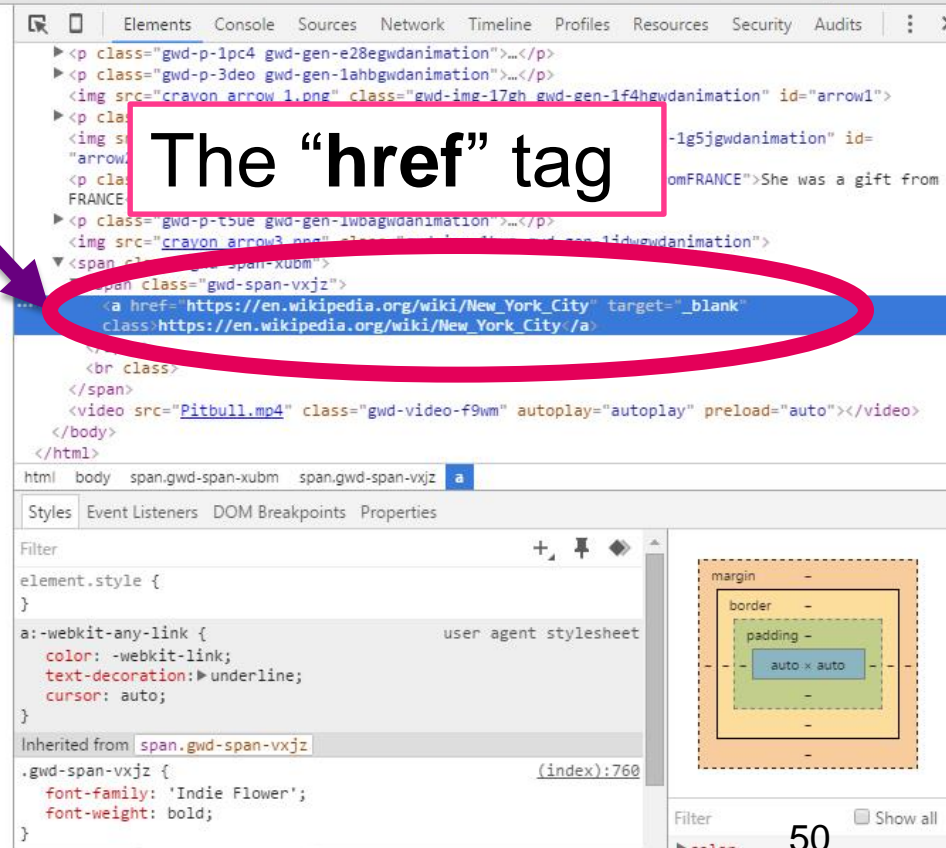
- Template: <http://ow.ly/ZAeLL>
- Sample: <http://ow.ly/ZAeUG>

Submit to the course Facebook with Snapshot or scanned work, with your Name and ID.

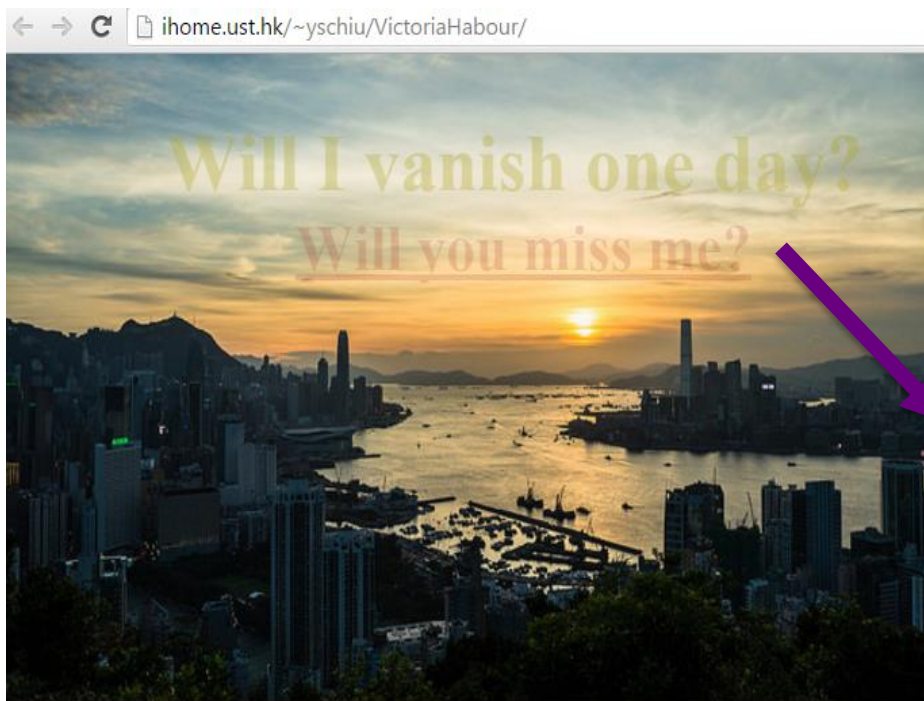
- **Submission Deadline: Tomorrow Noon**
 - **20%** penalty for submitting after the deadline.

Appendix

Hyperlink in HTML5



Hyperlink in HTML5



Will I vanish one day?
Will you miss me?

Specify the name after “href” tag

```
<!DOCTYPE html>
<html>

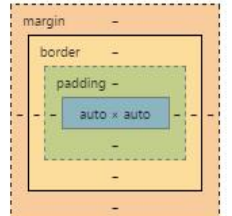

<p class="gwd-p-h43n gwd-gen-1aligwdanimation">Will I vanish one day?</p>
<p class="gwd-p-13o7 gwd-gen-7eevgwdanimation" style>
  <a href="https://en.wikipedia.org/wiki/Victoria_Harbour" target="_blank" class="gwd-a-2ir1 gwd-a-o4tx">Will you miss me?</a>
</body>
</html>
```

html body p.gwd-p-13o7.gwd-gen-7eevgwdanimation a.gwd-a-2ir1.gwd-a-o4tx

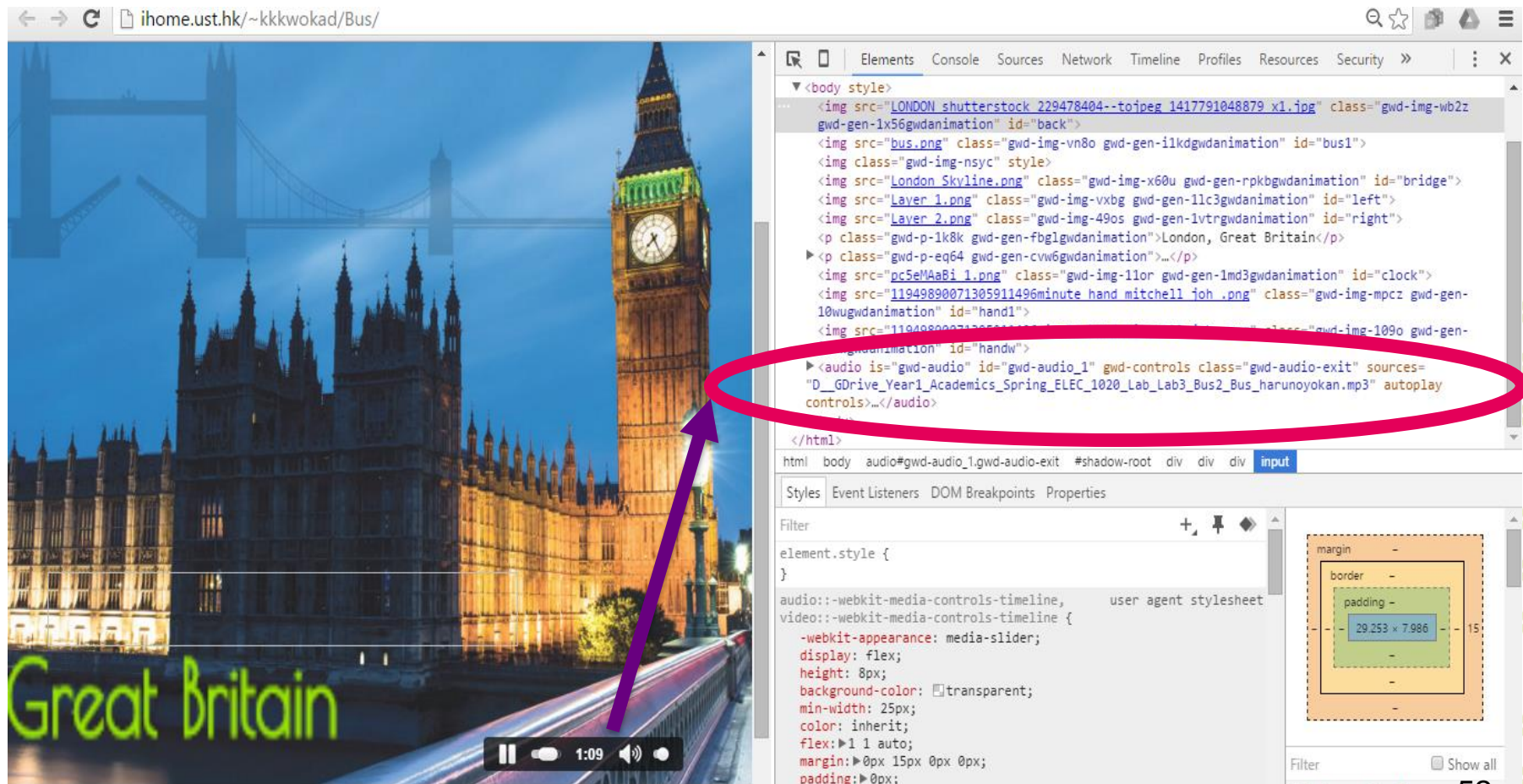
Styles Event Listeners DOM Breakpoints Properties

Filter

```
element.style {
}
.gwd-a-o4tx {
  color: rgb(150, 3, 10);
}
.gwd-a-2ir1 {
  color: rgb(0, 0, 0);
}
```



Audio Element



The screenshot shows a web browser displaying a video of Big Ben and the London skyline. The video player has a progress bar at 1:09. A red circle highlights the audio element in the browser's developer tools, and a purple arrow points from the video player to it.

The HTML code in the developer tools shows the following structure:

```
<body style>  
    
    
  <img class="gwd-img-nsyc" style>  
    
    
    
  <p class="gwd-p-1k8k gwd-gen-fbglgwdanimation">London, Great Britain</p>  
  <p class="gwd-p-eq64 gwd-gen-cvwb6gwdanimation">...</p>  
    
    
    
  <audio is="gwd-audio" id="gwd-audio_1" gwd-controls class="gwd-audio-exit" sources="D_GDrive_Year1_Academics_Spring_ELEC_1020_Lab_Lab3_Bus2_Bus_harunoyokan.mp3" autoplay controls>...</audio>  
</html>
```

The CSS code in the developer tools shows the following styles for the audio element:

```
audio::-webkit-media-controls-timeline,  
video::-webkit-media-controls-timeline {  
  -webkit-appearance: media-slider;  
  display: flex;  
  height: 8px;  
  background-color: transparent;  
  min-width: 25px;  
  color: inherit;  
  flex: 1 1 auto;  
  margin: 0px 15px 0px 0px;  
  padding: 0px;
```

Video Element



```

"first">



<p class="gwd-p-16d5 gwd-gen-1mcogwdanimation">Let's go sightseeing in Shiratani Cloud
Water Valley!</p>
<p class="gwd-p-1293 gwd-gen-1bqagwdanimation">Oh My God, I'll go with you!</p>
<video src="Nature music - .mp4" class="gwd-video-1t6k" loop="loop" autoplay=
"autoplay"></video>
</html>
```

