


Topic	Mobile Compatible Games	
Class Description	The student learns to adjust their game dimensions to fit all screen sizes. The student will host their game online on GitHub and then build a web wrapper around the application to be able to generate an APK file that can be published on Google Play.	
Class	C18	
Class time	45 mins	
Goal	<ul style="list-style-type: none"> <li>Adjust the game dimension to fit all screen sizes.</li> <li>Make the game touch friendly for smartphones.</li> <li>Host the game online on GitHub.</li> <li>Build a web wrapper around the game to generate an APK file.</li> </ul>	
Resources Required	<ul style="list-style-type: none"> <li>Teacher Resources:               <ul style="list-style-type: none"> <li>Laptop with internet connectivity</li> <li>Earphones with mic</li> <li>Notebook and pen</li> </ul> </li> <li>Student Resources:               <ul style="list-style-type: none"> <li>Laptop with internet connectivity</li> <li>Earphones with mic</li> <li>Notebook and pen</li> </ul> </li> </ul>	
Class structure	<b>Warm-Up Slides</b> <b>Student - led Activity 1</b> <b>Teacher - led Activity</b> <b>Student - led Activity 2</b> <b>Wrap-Up Slides</b>	<b>10 mins</b> <b>10 mins</b> <b>10 mins</b> <b>10 mins</b> <b>5 mins</b>
WARM UP SESSION - 10 mins		
<div>  </div> Teacher starts slideshow from slides 1 to 7		

Refer to speaker notes and follow the instructions on each slide.	
Activity details	Solution/Guidelines
<p><i>Hey &lt;student name&gt;. How are you? It's great to see you! Are you excited to learn something new today?</i></p> <p><b>Run the presentation from slide 1 to slide 4.</b></p> <p><b>Following are the warm up session deliverables:</b></p> <ul style="list-style-type: none"> <li>Connecting students to the previous class.</li> <li>Help student recall Github basics and the difference in hosting a game locally versus globally</li> <li>Adapting games as per mobile devices</li> </ul>	<p><b>ESR:</b> Hi, thanks, yes I am excited about it!</p> <p>Click on the slide show tab and present the slides.</p>
QnA Session	
Question	Answer
<p>Select the correct block of code to set a collision radius for the mainCyclist.</p> <p>A. <code>mainCyclist.setCollission("rectangle",0,0,40,40);</code></p> <p>B. <code>mainCyclist.setCollider("rectangle",0,0,40,40);</code></p> <p>C. <code>mainCyclist.setCollission("rectangle",0,0,40,40,50);</code></p> <p>D. <code>mainCyclist.setCollider("rectangle",0,0,40,40,50);</code></p>	<p><b>B</b></p>

Select the correct block of code to call the reset( ) function when the up arrow key is pressed.



A. 

```
if(keyDown("UP_ARROW")) {  
    reset;  
}
```

B. 

```
if(key("UP_ARROW")) {  
    reset();  
}
```

C. 

```
if(keyDown()) {  
    reset();  
}
```


D. 

```
if(keyDown("UP_ARROW")) {  
    reset();  
}
```

**D**

**Continue the warm up session**

Activity details	Solution/Guidelines
<p><b>Run the presentation from slide 5 to slide 7 to set the problem statement.</b></p> <p><b>Following is the warm up session deliverables:</b></p> <ul style="list-style-type: none"> <li>Introducing the problem statement.</li> </ul>	<p>Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.</p>

<div>  </div> <p>Teacher ends slideshow</p>	
STUDENT-LED ACTIVITY 1 - 10 mins	
<ul style="list-style-type: none"> <li>Ask the student to press the ESC key to come back to the panel.</li> <li>Guide the student to start Screen Share.</li> <li>The teacher gets into Fullscreen.</li> </ul>	
<p><b>ACTIVITY</b></p> <ul style="list-style-type: none"> <li>Create a Student class</li> <li>Create the object for the Student class.</li> </ul>	
Teacher Action	Student Action
<p>In the last 2 classes, we got started with object-oriented programming. In this class, we are going to perform a very small activity just to revise the concepts of OOP.</p> <p>We are going to create a <b>Student</b> class. In this class, we will have our <b>constructor()</b> function which will take three arguments name, age, and grade and at last, we are going to create a method in the class that will display this information on the console.</p> <p><i>Note: The <b>student.js</b> file is already created and added in the <b>html.js</b>.</i></p> <p>Can you recall how we created a class?</p> <p>We use the <b>class</b> keyword and then the name of the class, such as <b>class Student</b></p> <pre>{ }</pre> <ul style="list-style-type: none"> <li>The first letter of the class name is always capitalized.</li> </ul>	<p><i>The student downloads the <a href="#">Student Activity 1</a> template code and opens it in the VS Code Editor.</i></p> <p><i>Guide the student to open the student.js file and write the code.</i></p> <p><b>ESR:</b> Varied</p>

- In the curly brackets, you write the class methods. Methods are just like functions, but we do not use the keyword **function** while creating them. When we create the functions in the class, we call them methods.

The first method we are going to create is the constructor. When we create the object of a class, this is the first function that gets executed.

In the arguments of the constructor, we will pass three parameters:

First is the name, which is going to be the name of the Student, followed by the age and finally the grade.

```
class Student
{
    constructor(name, age, grade)
    {
    }
}
```

When we define our **constructor()** function we are going to pass name, age, grade as parameters. So that when you create the object of the student, you need to pass the details of the actual student.

Now we are going to define these attributes of the student using the **this** keyword such as:

```
this.name = name
this.age = age;
```

**this.grade = grade;**

Here, **this.name** is the attribute, and then we are assigning the value which is going to be passed by you while creating the object.

```
constructor(name,age,grade)
{
    this.name = name;
    this.age = age;
    this.grade = grade;
}
```

The next step is to create a method that will display the details of the Student on the console.

It is very easy to create a **display()** function and in that function, we will use **console.log()** to display the name, age, and grade of the student.

**Note:** You can name your function anything, but to make our code readable, we use the most appropriate name for the function based on what work this function is going to perform.

```
display()
{
    console.log(this.name);
    console.log(this.age);
    console.log(this.grade);
}
```

We have defined our class now, and we are going to create the object for this class in the **sketch.js** file.

Define a variable, we can name this anything.  
Let's say **var s1;**

In the **setup()** function, we are to create the object of the **Student** class and store that in this variable.

Can you tell me how we create an object?

**ESR:**  
Varied

To create the object, we are going to use a **new** keyword followed by the name of the class.

In the brackets of the class name, we will pass the details of the student i.e. (**name, age, and grade**).

At last, we are going to call the **display()** function of the **Student** class and to call a function from the class, we use the **object.function()** which in our case is **s1.display()**.

Now we can run the code, and we can see that all the details are displayed on the console.

```
var s1;

function setup() {
  createCanvas(400,400);

  s1 = new Student("john",12,6);
  s1.display();
}
```

Output:



This concept is used in schools, banks, and offices to store the information related to students, bank customers, and employees.

**Teacher Guides Student to Stop Screen Share**



**TEACHER-LED ACTIVITY - 5mins**

**Teacher Initiates Screen Share**

### CHALLENGE

- **Hosting the game on GitHub.**
- **Adjusting the dimensions of the game so that it can fit all screen sizes.**



<div style="text-align: center;">   <b>Teacher starts slideshow from slides 8 to 17</b>          Refer to speaker notes and follow the instructions on each slide.       </div>	
Activity details	Solution/Guidelines
<p><b>Run the presentation from slide 8 to slide 17 to set the problem statement.</b></p> <p><b>Following is the warm up session deliverables:</b></p> <ul style="list-style-type: none"> <li>Introducing the display size, co-ordinates and arrays.</li> </ul>	<p>Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.</p>
<div style="text-align: center;">   <b>Teacher ends slideshow</b> </div>	
Teacher Action	Student Action
<p><b>Step 3:</b>  <b>Teacher-led Activity</b>  <b>(15 min)</b></p> <p>We have different types of devices such as tablets, mobile phones, computers etc. All of these devices have different screen sizes. But our game is created using a fixed canvas size. We have a problem that if we run the Trex game on different screens it will not look proper.</p> <p>How do you think we can fix the problem of screen sizes?</p>	<p><i>Student thinks about it.</i></p>
<p>In p5, we have predefined variables called - windowWidth and windowHeight.</p> <p>windowWidth and windowHeight automatically take the size of the device in which the game runs.</p>	

<p>So instead of defining the canvas size in absolute numbers, we can put the canvas size as windowWidth and windowHeight.</p> <p>How does the position for the different sprites in the game change if the canvas size is going to change?</p>	<p><b>ESR:</b></p> <p>We will need to adjust the position of all the sprites relative to the width and height.</p>
<p>Awesome! Let's do this.</p>	
<p>We have done exactly that.</p> <p><b>The teacher shows <a href="#">Teacher Activity 1</a>, the canvas size and how coordinates of other sprites are changed.</b></p>	<p><i>The student looks through the changes in the coordinates for different sprites/images.</i></p>



```
function setup() {
  createCanvas(windowWidth, windowHeight);

  sun = createSprite(width-50,100,10,10);
  sun.addAnimation("sun", sunAnimation);
  sun.scale = 0.1

  trex = createSprite(50,height-70,20,50);

  trex.addAnimation("running", trex_running);
  trex.addAnimation("collided", trex_collided);
  trex.setCollider('circle',0,0,350)
  trex.scale = 0.08;

  invisibleGround = createSprite(width/2,height-10,width,125);
  invisibleGround.shapeColor = "#f4cbaa";

  ground = createSprite(width/2,height,width,2);
  ground.addImage("ground",groundImage);
  ground.x = width/2
  ground.velocityX = -(6 + 3*score/100);

  gameOver = createSprite(width/2,height/2- 50);
  gameOver.addImage(gameOverImg);

  restart = createSprite(width/2,height/2);
  restart.addImage(restartImg);
```

**width** and **height** will be the same as canvas width and canvas height.

There is only one more change in our game to make it ready for mobile.

We had to press space to make the Trex jump.

On a mobile phone, we want the Trex to jump when the phone is touched or tapped.

How do we do that?	<i>The student thinks.</i>
<p>Fortunately, in the p5 library we have something called <b>touches</b>.</p> <p>Every time, we touch or tap on the screen, the coordinates of the point where we have tapped get stored inside <b>touches</b>.</p> <p>You have learned about variables and how it can store “numbers” and “strings” data types. Variables can also store a list of items - called an array.</p> <p>The teacher opens and runs <a href="#">Teacher Activity 3</a> to demonstrate the use of arrays.</p> <p>The teacher shows:</p> <ul style="list-style-type: none"> <li>• how to create an array data type</li> <li>• how to access any item in the array</li> <li>• how to find the array length</li> </ul> <p>Touches is an array data type and stores all the touched co-ordinates as a comma separated list.</p> <p>The <b>array</b> object lets you store multiple values in a single variable. It is denoted by [ ].</p> <p><b>Arrays</b> are zero-indexed: the first element of an array is at index 0, and the last element is at the index equal to the value of the array's length property minus 1.</p> <p>How can we use this to detect touch and make the Trex jump when the screen is touched?</p>	<p><i>Student looks at the <a href="#">Student Activity 3</a> - touches reference in p5.</i></p> <p><i>Student thinks and comes up with a response.</i></p>
We can check if the length of the <b>touches array</b> is greater than 0 to detect a tap or touch.	<i>Student listens.</i>

If the array length is greater than 0, the phone screen is tapped. We will again empty the array so that we can detect the next tap.

We are doing exactly that in our code.

*Student looks into the code.*

*The teacher shows the code in 'if condition' to make the Trex jump and to reset the game.*

```
function draw() {
  //trex.debug = true;
  background(backgroundImg);
  textSize(20);
  fill("black")
  text("Score: "+ score,30,50);

  if (gameState===PLAY){
    score = score + Math.round(getFrameRate()/60);
    ground.velocityX = -(6 + 3*score/100);

    if (touches.length > 0 || keyDown("SPACE")) && trex.y
    >= height-120) {
      jumpSound.play( )
      trex.velocityY = -10;
      touches = [];
    }

    trex.velocityY = trex.velocityY + 0.8

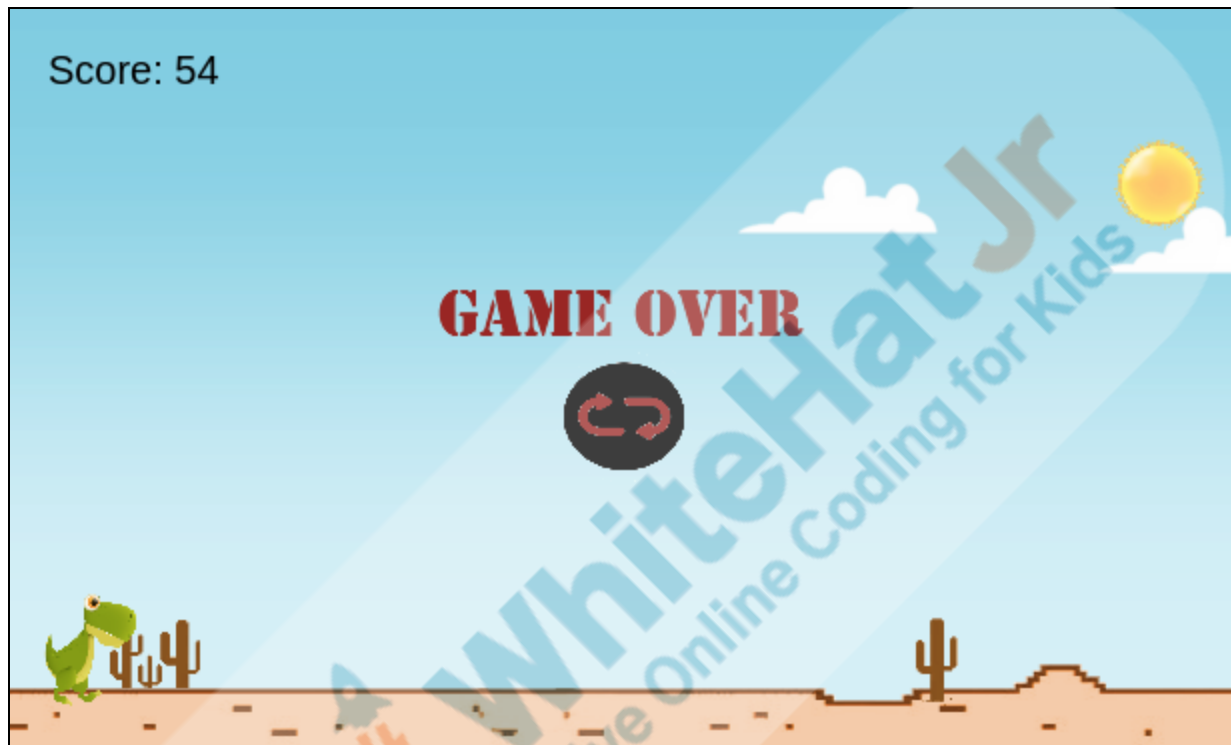
    if (ground.x < 0){
      ground.x = ground.width/2;
    }

    trex.collide(invisibleGround);
    spawnClouds();
    spawnObstacles();
  }
}
```

That's all. We just had to make our game:

- suitable for all screen sizes and
- responsive to touch.

Now our game is ready to be published on Play Store!



Teacher starts slideshow  :Slide 18-20

**Run the presentation for slides from 18 to 20 to set the student activity context.**


Here's a challenge for you. You need to make your game mobile compatible.

Can you tell me what are the various steps you would need to perform to achieve this challenge?

**ESR:**

- change the canvas dimensions to windowWidth and windowHeight.

*Student may have additional responses specific to their*



	<i>game; understand and respond accordingly.</i>
Great! Why don't you share your screen and show me how you are going to code this challenge?	<i>The student shares his/her screen, opens the Student Activity and adds more code to it.</i>
<div>  <b>Teacher ends slideshow</b> </div>	
<b>Teacher Stops Screen Share</b>	
<b>STUDENT-LED ACTIVITY - 10mins</b>	
<ul style="list-style-type: none"> <li>• Ask the student to press the ESC key to come back to the panel.</li> <li>• Guide the student to start Screen Share.</li> <li>• The teacher gets into Fullscreen.</li> </ul>	
<p align="center"><u><b>ACTIVITY</b></u></p> <ul style="list-style-type: none"> <li>• Creating a web wrapper around the game and generating the APK file.</li> <li>• Running and testing the web wrapper on the phone.</li> </ul>	
<b>Step 3:</b> <b>Student-led Activity</b>  Our current game is an application on our computer. Let's quickly learn how to host our game online so that you can share them with your friends.	<i>Student thinks and tries to answer.</i>
GitHub hosts several open-source projects and projects from other developers.  <i>Guide the student to create a new project repository.</i>	

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner

Repository name \*

 whitehatjr / trex 

Great repository names are short and memorable. Need inspiration? How about [super-umbrella](#)?

Description (optional)

Chrome's trex game created using p5.play



**Public**

Anyone can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **None**

Add a license: **None**



Create repository

*Guide the student to upload the Trex project files on GitHub.*

*The student uploads the files to their GitHub project repository.*

*The student has to commit the change to save the files.*

Chrome's trex game created using p5.play

Edit

Manage topics

1 commit

1 branch

0 releases

1 contributor


Branch: master [New pull request](#)

[Create new file](#)

[Upload files](#)

[Find File](#)

[Clone or download](#)

 whitehatjr Initial commit

Latest commit 1b55a9a now

[README.md](#)

Initial commit

now

[README.md](#)

**trex**

Chrome's trex game created using p5.play



 <p>The screenshot shows the GitHub repository upload interface. At the top, it says 'trex /'. Below that is a large light gray box with icons for various file types (code, image, document, etc.) and the text 'Drag files here to add them to your repository'. Below this text is a red-bordered button that says 'Or choose your files'.</p>	
 <p>The screenshot shows the 'Commit changes' dialog in GitHub. It has a title 'Commit changes'. Below the title is a text input field containing 'Update Slingshot.js'. Below that is a larger text area with the placeholder 'Add an optional extended description...'. At the bottom, there are two radio button options: the first is selected and says 'Commit directly to the main branch.', and the second says 'Create a new branch for this commit and start a pull request. Learn more about pull requests.' At the very bottom are two buttons: a green 'Commit changes' button and a red 'Cancel' button.</p>	
<p>We have uploaded the “<b>Trex</b>” Files on GitHub.</p> <p>We can now ask GitHub to host them for us. For this we need to activate the GitHub pages for our project.</p> <p><i>Guide the student to go to <b>Settings of the repository</b> and scroll down to “GitHub Pages” to activate.</i></p> <p><i>Note:- While activating GitHub pages make sure that the Branch is the <b>main branch</b> which contains your files. The folder is the root folder.</i></p> <p><i>It takes some time for the GitHub pages to be activated.</i></p>	<p><i>The student activates the GitHub pages.</i></p> <p><i>The student visits the link to view the project on the browser.</i></p>

whitehatjr / trex

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights **Settings**

Chrome's trex game created using p5.play Edit

Manage topics

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request

Create new file Upload files Find File Clone or download

whitehatjr Upload files from created using p5 editor Latest commit ff928ed 5 minutes ago

README.md	Initial commit	11 minutes ago
cloud.png	Upload files from created using p5 editor	5 minutes ago
gameOver.png	Upload files from created using p5 editor	5 minutes ago
ground2.png	Upload files from created using p5 editor	5 minutes ago
index.html	Upload files from created using p5 editor	5 minutes ago
obstacle1.png	Upload files from created using p5 editor	5 minutes ago
obstacle2.png	Upload files from created using p5 editor	5 minutes ago
obstacle3.png	Upload files from created using p5 editor	5 minutes ago
obstacle4.png	Upload files from created using p5 editor	5 minutes ago

## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

### Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

Branch: main

/ (root)

Save

Select branch

Select branch

✓ main

None

home using the gh-pages branch. [Learn more.](#)

<div> <h2>GitHub Pages</h2> <p>GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.</p> <div> <span>✓</span> Your site is published at <a href="https://whitehatjr.github.io/trex/">https://whitehatjr.github.io/trex/</a> </div> <div> <p><b>Source</b></p> <p>Your GitHub Pages site is currently being built from the master branch. <a href="#">Learn more.</a></p> <div> <span>Branch: master ▾</span> <span>/ (root) ▾</span> <span>Save</span> </div> <p><b>Theme Chooser</b></p> <p>Select a theme to publish your site with a Jekyll theme. <a href="#">Learn more.</a></p> <div> <span>Choose a theme</span> </div> </div> </div>	
<p>Wow! Now you can share the link to your GitHub project with your friends and family so that they can play this game.</p> <p>Our code is now hosted on GitHub's server - it can be accessed from anywhere.</p> <p>We are using the GitHub web address (web domain) right now - but we can also use our own web address.</p> <p>Isn't this amazing?</p>	<p><i>The student visits the <a href="#">GitHub link</a> to see the project in action on the browser.</i></p> <p><b>ESR:</b> Varied.</p>
<p>But how do we make it into a mobile app?</p>	<p><b>ESR:</b> Varied.</p>
<p>What we can do is we can create an application which wraps around a browser showing this game. This is called a web wrapper app.</p> <p>It is one of the easiest and fastest ways to create a mobile application.</p> <p>You create a browser based application and then put it into a web wrapper.</p>	<p><i>Student listens.</i></p>

We will look at simple ways to do this.  
There is a platform called Thunkable which allows us to use drag and drop features to build mobile apps.

On Thunkable we can easily create mobile apps without writing code. It has the drag and drop blocks for different components of mobile apps such as database, camera and layout.

Once we feel our app is complete we can export it to run on Android as well as IOS.

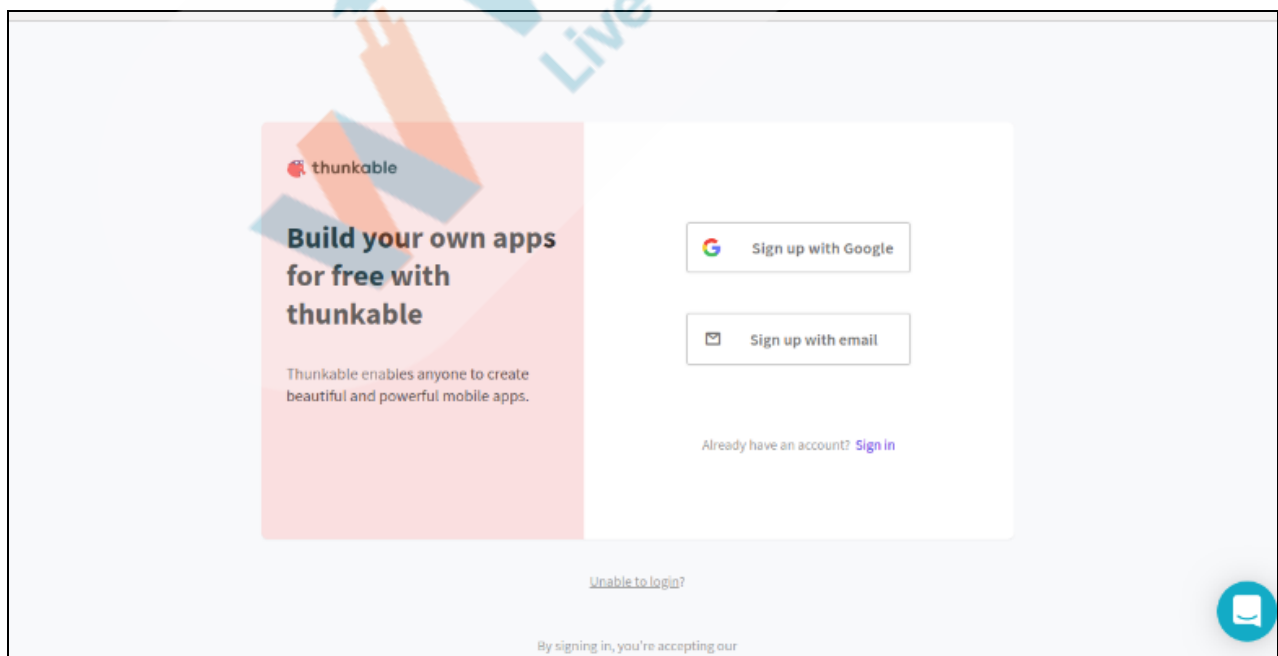
We will be using that for now.

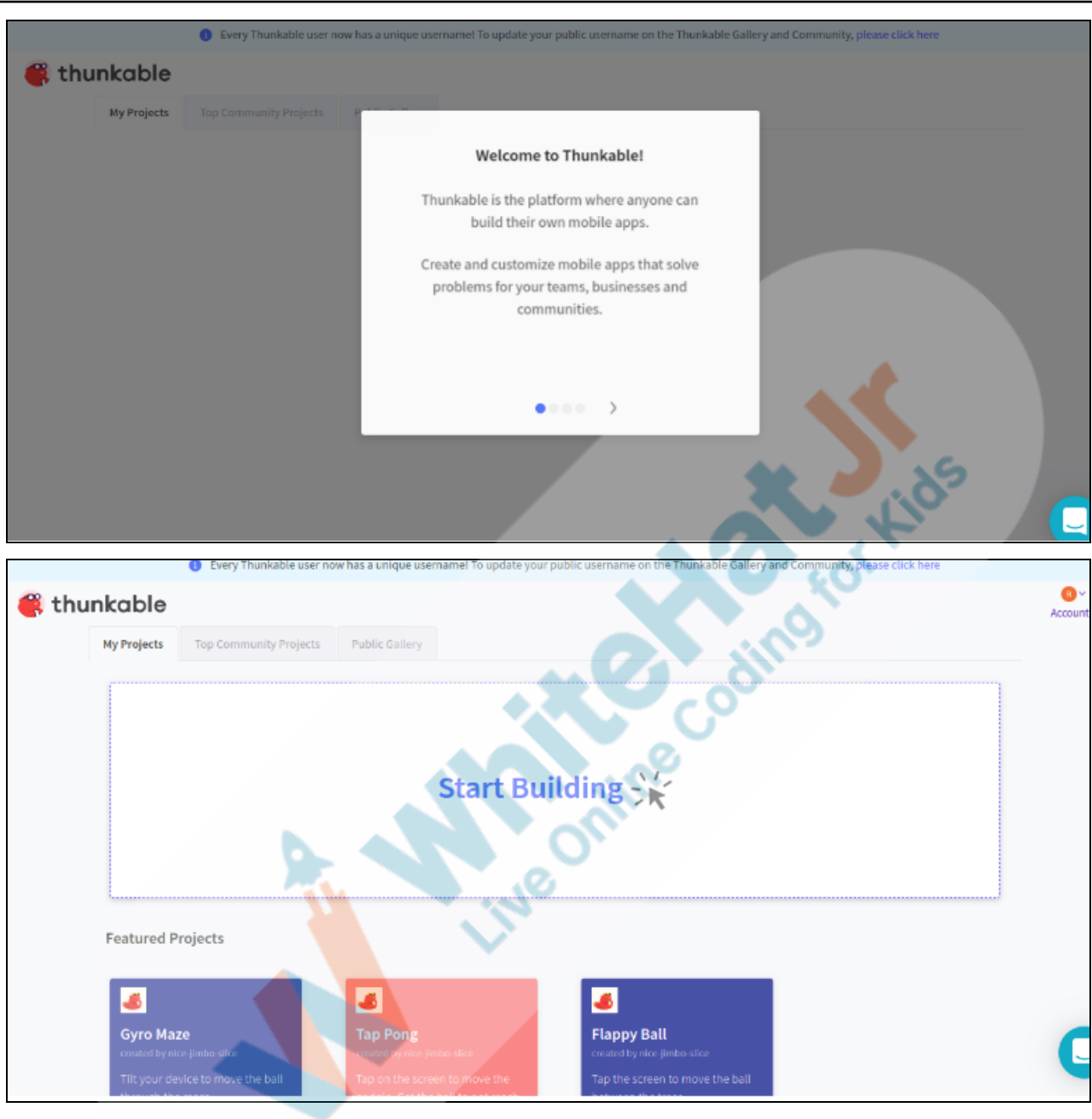
Later, in the second module we will learn how to build apps by writing code.

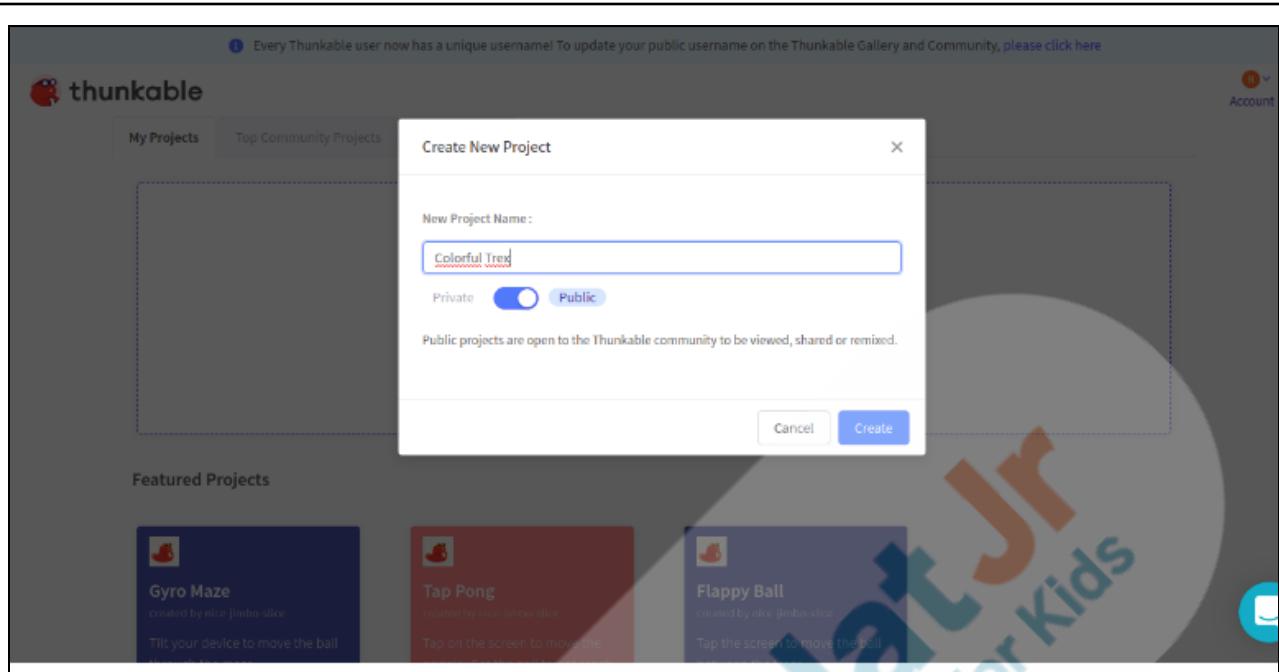
Let's open [Student Activity 4](http://www.thunkable.com) (www.thunkable.com)  
Quickly sign up and then login.

*Guide the student to sign up and login to thunkable.com.*

*The student signs up and logs in to thunkable (using Google log in).*







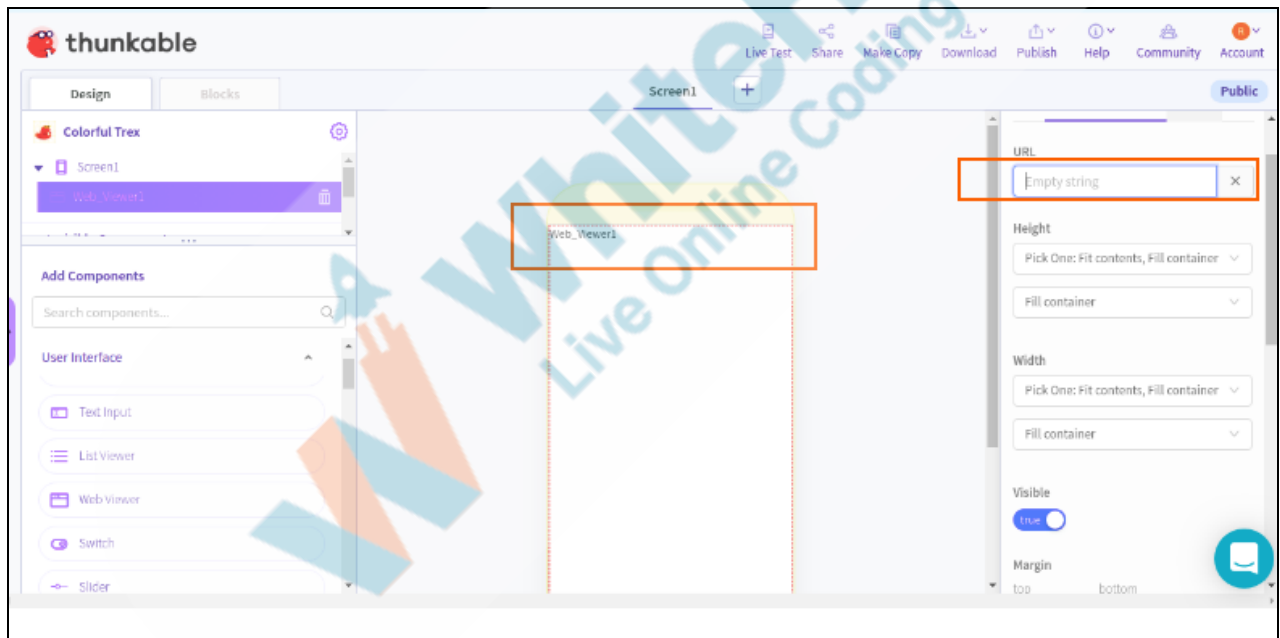
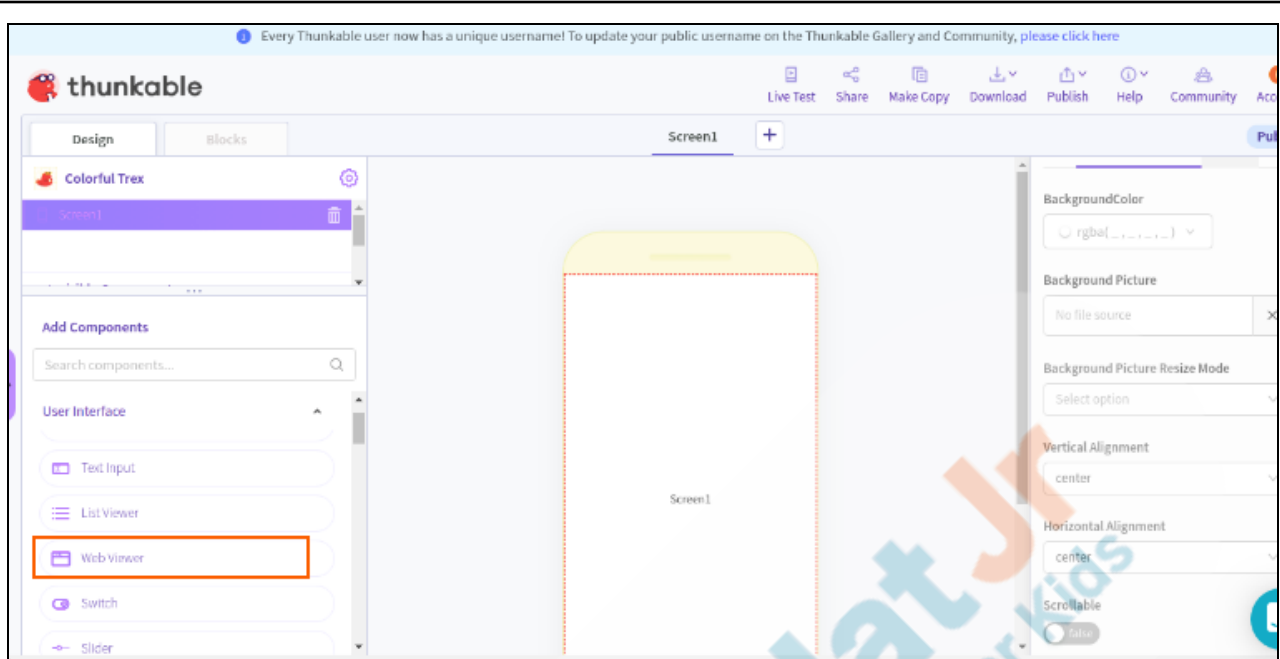
You will see two tabs - design and block.

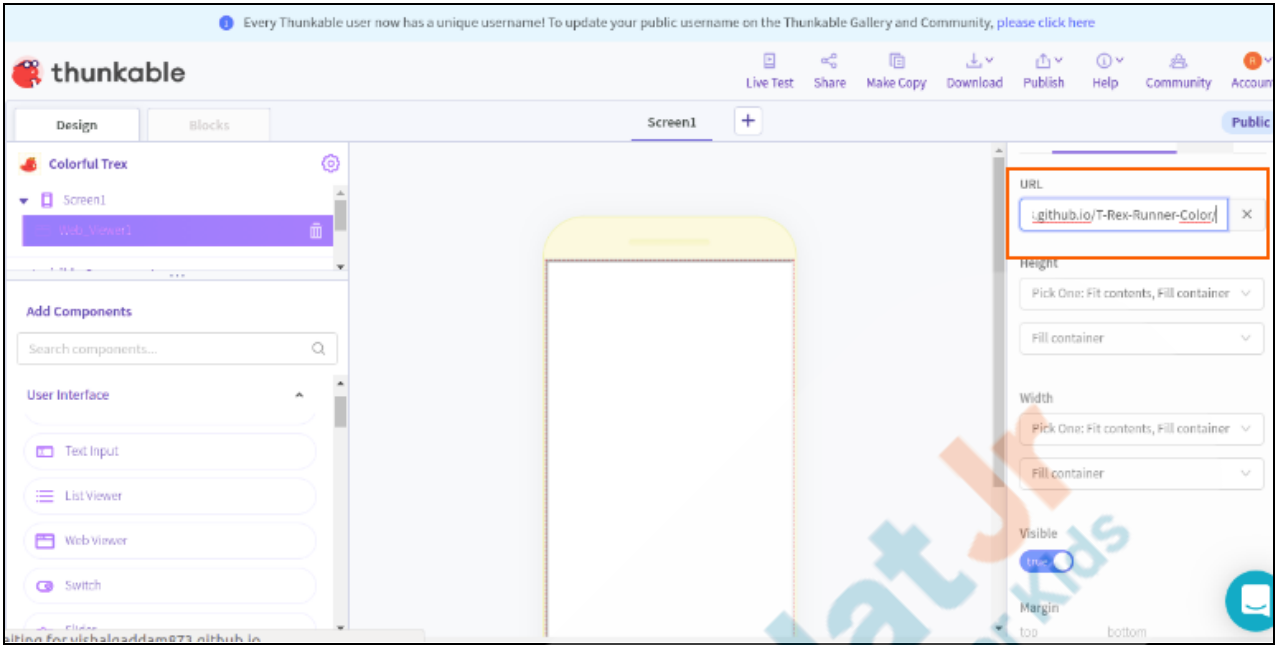
Under the 'design' tab, locate the "Web View" option and drag it to the screen which you see on the right.

You will see an additional web view option to the extreme right.

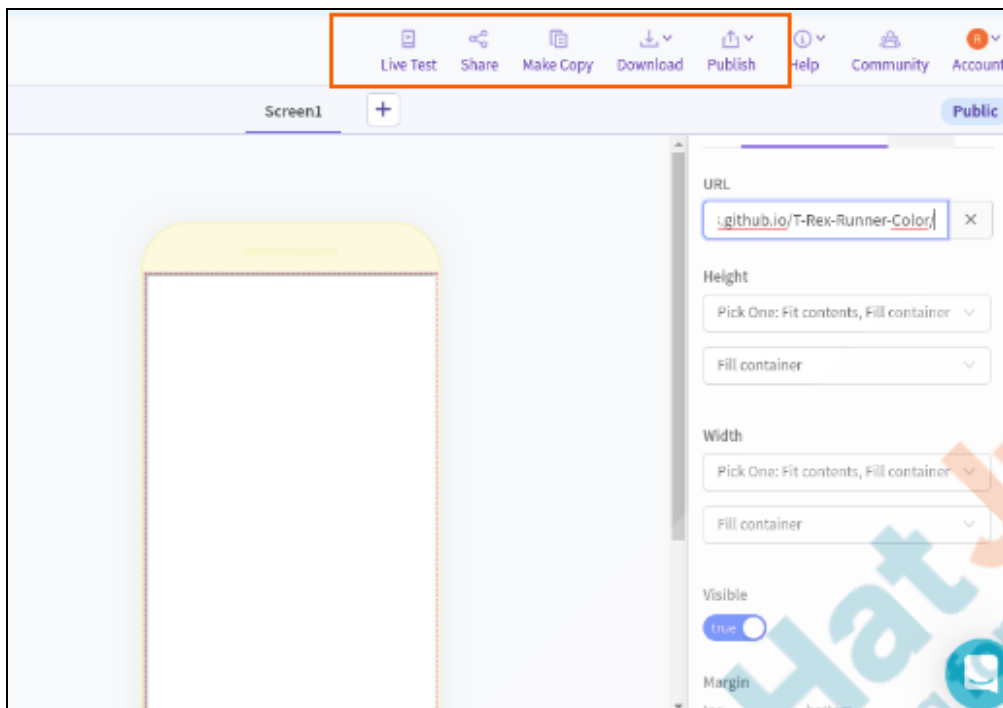
Add the game URL there ([Student Activity 5](#)) which can be hosted on Github for example.

*The student follows the instructions to add a web view component in Thunkable.*



	<p>Amazing! You can see the live preview version of it on the site itself.</p> <p>You can download the Thunkable app on your Android/iOS smartphone, sign up and live test it.</p>
<p><i>Student live tests the app on the phone, if the student has access to a smartphone.</i></p>	





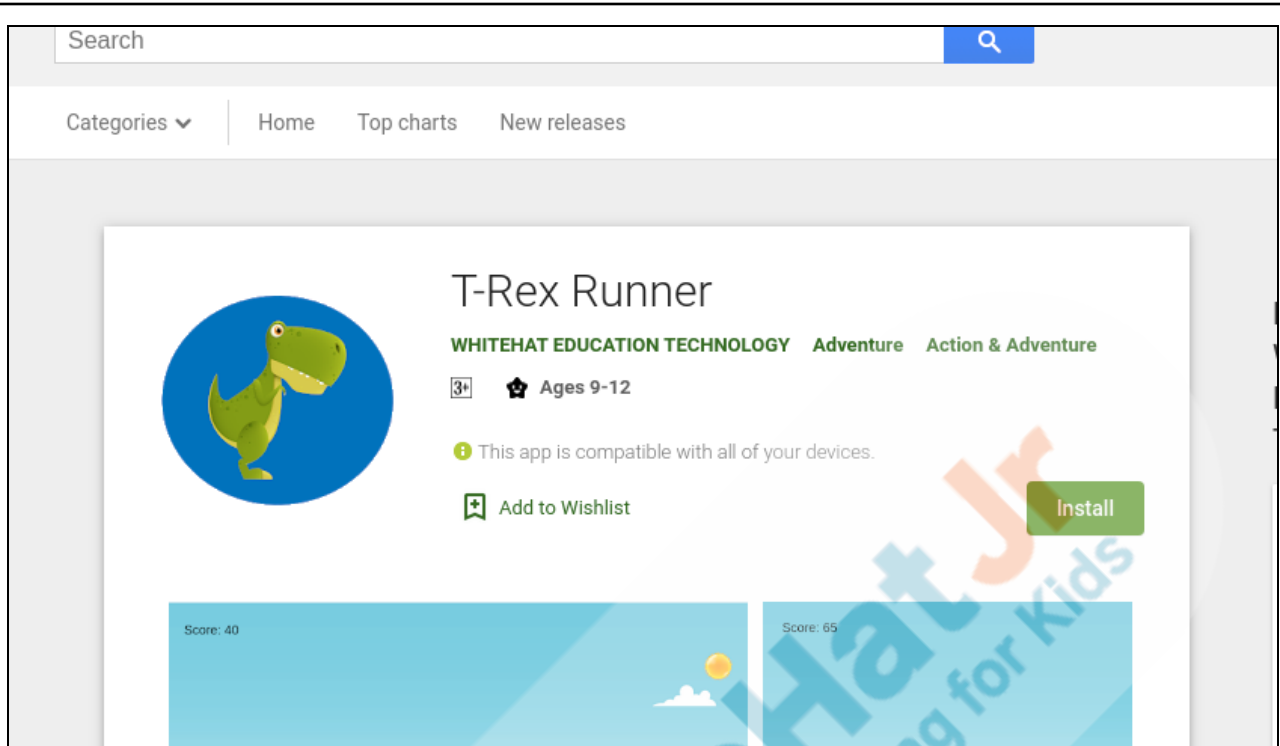
You can use the download button to download the apk file which can be installed on Android or ipa file which can be installed on iOS.

*(Student has to wait for some time after pressing the download button.)*

The APK file needs to be transferred and installed on your system (mobile phones).

*The student also has to allow the installation of the file from unauthorized sources in the file setting.*

*The student uses the download option to test the download and installation of the file.*

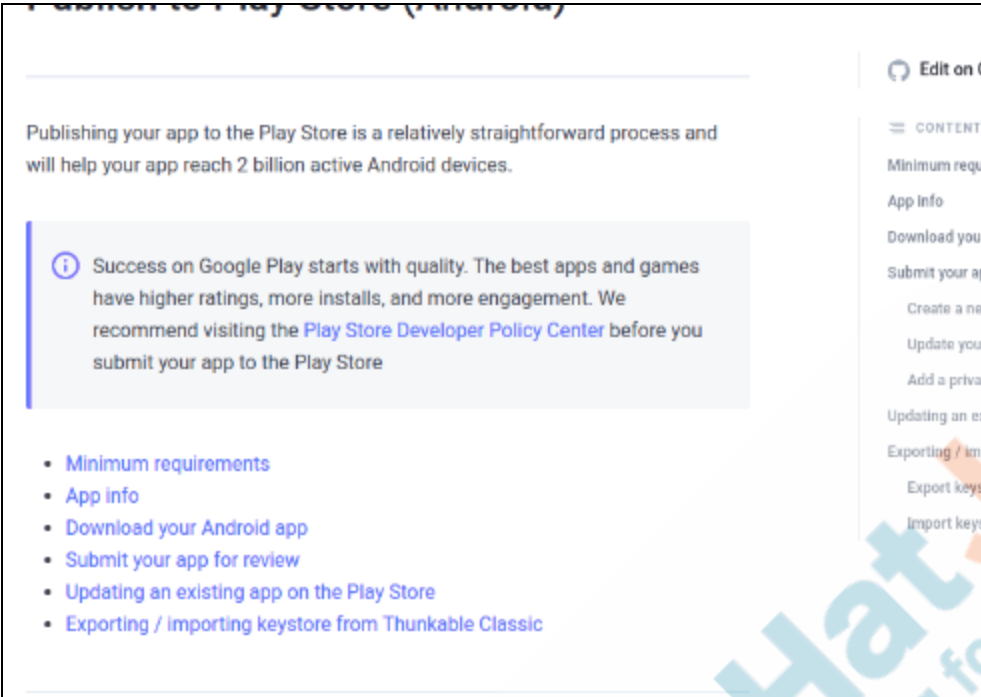







Amazing!

Thunkable is a simple drag and drop tool built to design mobile apps (similar to code.org).

We have our app now. You can choose to download it and publish it on the playstore. Publishing instructions are also on the Thunkable.

*Student listens.*

 <p>The screenshot shows the 'Publish to Play Store' page. It includes a heading 'Publishing your app to the Play Store is a relatively straightforward process and will help your app reach 2 billion active Android devices.' Below this is a tip box: 'Success on Google Play starts with quality. The best apps and games have higher ratings, more installs, and more engagement. We recommend visiting the <a href="#">Play Store Developer Policy Center</a> before you submit your app to the Play Store.' A list of steps follows: 'Minimum requirements', 'App info', 'Download your Android app', 'Submit your app for review', 'Updating an existing app on the Play Store', and 'Exporting / importing keystore from Thunkable Classic'. On the right, a sidebar menu lists options like 'Edit on Google Play', 'CONTENTS', 'Minimum requirements', 'App info', 'Download your Android app', 'Submit your app for review', 'Create a new app', 'Update your app', 'Add a privacy policy', 'Updating an existing app', 'Exporting / importing keystore', 'Export keystores', and 'Import keystores'.</p>	
<p>Guide the student to generate the apk file.</p> <p>Well done!</p> <p>You have done a great job.</p> <p>How are you feeling after running the game on the mobile?</p> <p>Now you can create any game and make it mobile compatible and install on the mobile using Thunkable.</p> <p>Let's wrap-up our class for now.</p>	<p><i>The student generates the apk files.</i></p> <p><i>The student can also install it on their phones and run the app.</i></p> <p><b>ESR:</b> Varied.</p>

<b>Teacher Guides Student to Stop Screen Share</b>	
<b>WRAP UP SESSION - 5 Mins</b>	
<b>Wrap-Up</b> <b>15 min</b> In this class, you also learned how to create a web wrapper app to convert your game published over the web into an android app.	<i>Student asks questions.</i>
Are you afraid of ghosts? Well, if you're, then it's time to face your fears!  In the upcoming class, we will not just create the Ghost Runner game, but also upload it on GitHub and host it via GitHub server to share it online.  Please invite your parents to join us in the gameplay.  How are you feeling?	<b>ESR: Varied.</b>          <b>ESR: Excited!</b>
You get a hats off.    Alright, I will see you in the next class then!	<i>Make sure you have given at least 2 Hats Off during the class for:</i> <div> <div>Creatively Solved Activities  +10</div> <div>Great Question  +10</div> <div>Strong Concentration  +10</div> </div>
<div>  <b>Teacher starts slideshow</b>  <b>from slide 21 to slide 29</b> </div>	

Activity details	Solution/Guidelines
<b>Run the presentation from slide 21 to slide 29</b>  <b>Following are the wrap-up session deliverables:</b> <ul style="list-style-type: none"> <li>● Explain the facts and trivias</li> <li>● Next class challenge</li> <li>● Project for the day</li> </ul>	Guide the student to develop the project and share with us.
<b>Quiz time - Click on in-class quiz</b>	
Question	Answer
Which of the following is used in p5 to identify a touched position on a touchscreen device?  A. touched B. touchable C. touches D. touch	C.
Which of the following is used in p5 to automatically capture the displaySize of the device?  A. (displayWidth, displayHeight) B. fullScreen(); C. (windowWidth, windowHeight) D. WindowFullScreen;	C.
What is to be created in order to convert the game published over the web into an Android app?  A. packager B. web packager C. wrap D. web wrapper	D.

## End the quiz panel

### FEEDBACK

- Encourage the student to make reflection notes in markdown format.
- Compliment the student for her/his effort in the class.
- Review the content of the lesson.

**\* This Project will take only 30 mins to complete.  
Motivate students to try and finish it immediately after the class.**

### Project Overview

### DEVICE COMPATIBLE GAME

### Goal of the Project:

In Class 18, you learned how to make your Trex Runner game compatible across devices by changing the window size and object dimensions to publish your app on Play Store.

In this project, you will revise the concepts by making the Collecting Treasure game device compatible, which you created for Project 15.

### Story:

You shared the game created by you, "Collecting Treasure", with your friends to play. But when they try to play it on mobile, half the game is not visible. They have given you feedback to make changes in the game to fit into all the screens.

I am very excited to see your project solution and I know you will do really well.

Bye Bye!

**Note: You can assign the project to the student in class itself by clicking on the Assign Project button which is available under the projects tab.**

*Students engage with the teacher over the project.*

Teacher ends slideshow



Teacher Clicks

✕ End Class



**Note to teachers [Only Applicable for C19]:**

**Next class C19 is a CHECKPOINT REVISION CLASS** meant for revising concepts learned so far. Teachers should guide students to complete pending/expired projects. Please check on the dashboard if the student has pending projects less than 3, then you can continue to do the class activity.

**IF STUDENT HAS > 3 PENDING PROJECT  
SKIP CLASS ACTIVITY( INCLUDING VA & QUIZ) & HELP THEM COMPLETE PROJECT**

**ELSE  
CONTINUE CLASS ACTIVITY**

### Additional Activities

*Encourage the student to write reflection notes in their reflection journal using markdown.*

Use these as guiding questions:

- What happened today?
  - Describe what happened.
  - The code I wrote.
- How did I feel after the class?
- What have I learned about programming and developing games?
- What aspects of the class helped me? What did I find difficult?

*The student uses the markdown editor to write her/his reflections in the reflection journal.*

Activity	Activity Name	Links
Teacher Activity 1	Trex Color	<a href="https://github.com/procodingclass/PRO-C18-v3-Trex-color.github.io">https://github.com/procodingclass/PRO-C18-v3-Trex-color.github.io</a>
Teacher Activity 2	Trex Color Playstore	<a href="https://play.google.com/store/apps/details?id=com.gmail.totalgamingaman.trex">https://play.google.com/store/apps/details?id=com.gmail.totalgamingaman.trex</a>
Teacher Activity 3	Array Code Snippet	<a href="https://github.com/pro-whitehatjr/PRO_C18_LP_TA3">https://github.com/pro-whitehatjr/PRO_C18_LP_TA3</a>
Teacher Reference	Student Activity 1 reference code	<a href="https://github.com/pro-whitehatjr/pro-c18-sa1-reference">https://github.com/pro-whitehatjr/pro-c18-sa1-reference</a>
Student Activity 1	Template Code	<a href="https://github.com/pro-whitehatjr/Pro-c18-sa1-template_code">https://github.com/pro-whitehatjr/Pro-c18-sa1-template_code</a>
Student Activity 2	Trex Old	<a href="https://github.com/pro-whitehatjr/PRO_C18_LP_SA1">https://github.com/pro-whitehatjr/PRO_C18_LP_SA1</a>
Student Activity 3	Touches Reference	<a href="https://p5js.org/reference/#/p5/touches">https://p5js.org/reference/#/p5/touches</a>
Student Activity 4	Thunkable	<a href="https://thinkable.com/#/">https://thinkable.com/#/</a>
Student Activity 5	Trex Color GitHub	<a href="https://procodingclass.github.io/PRO-C18-v3-Trex-color.github.io/">https://procodingclass.github.io/PRO-C18-v3-Trex-color.github.io/</a>
Student Activity 6	Trex Color Playstore	<a href="https://play.google.com/store/apps/details?id=com.gmail.totalgamingaman.trex">https://play.google.com/store/apps/details?id=com.gmail.totalgamingaman.trex</a>



Visual Aid	Visual Aid	<a href="https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/BJFC-PRO-V3-C18-withcues.html">https://curriculum.whitehatjr.com/Visual+Project+Asset/PRO_VD/BJFC-PRO-V3-C18-withcues.html</a>
In -Class	In -Class	<a href="https://s3-whjr-curriculum-uploads.whjr.online/b402b206-7377-4851-a19c-4e3e589394f9.pdf">https://s3-whjr-curriculum-uploads.whjr.online/b402b206-7377-4851-a19c-4e3e589394f9.pdf</a>
Project Solution	Device Compatible Game	<a href="https://github.com/pro-whitehatjr/Project_C18_Device_Compatible_Game">https://github.com/pro-whitehatjr/Project_C18_Device_Compatible_Game</a>

