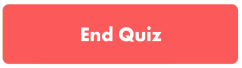
Topic	SUN-EARTH SIMULATION	
Class Description	**Kids learn to simulate the effect of astrophysical concepts like Astronomical Unit(Solar Distance) and orbital speed on Sun-Earth System through coding.**	
Class	**ADV-C53**	
Class time	**50 mins**	
Goal	• Simulate the Sun-Earth system.	

Resources Required	• Teacher Resources
	  ○ Use gmail login credentials
	  ○ Earphone with mic
	  ○ Notepad and Pen
	• Student Resources
	  ○ Use gmail login credentials
	  ○ Earphone with mic (optional)
	  ○ Notepad and Pen	

Class structure	**Warm Up** **Teacher-Led Activity** **Student-Led Activity** **Wrap Up**	**2 Mins** **8 Mins** **30 Mins** **5 Mins**

Class Steps	Say	Do
Step 1: Warm up (2 mins)	In the previous class we learnt a new collection variable called Array. Do you remember what we accomplished with Array in the previous lesson? We learned to declare Array and store more than 1 values in it. This is unlike variables which are used to store only a single value.	Ask the student to get into **Fullscreen** mode. Encourage the student to answer.

	Great! I have an exciting quiz question for you! Are you ready to answer this question? Arrays are also called as Data Structures which means that they store more than one values in them in a sequence. We created 2 arrays in the previous class. String array reasons: which stored 4 reasons to save trees. String array images: which stored 4 URLs of the images of the trees, one for each reason.	Please click on the  button on the bottom right corner of your screen to start the In-Class Quiz. A quiz will be visible to both you and the student. Encourage the student to answer the quiz question. The student may choose the wrong option, help the student to think correctly about the question and then answer again. After the student selects the correct option, the  button will start appearing on your screen. Click the End quiz to close the quiz pop-up and continue the class.

	Can you give some examples of types of data which can be stored in an Array? • Numbers • Alphabets/Characters We will be learning more about arrays along the course. So today we are going to start a new topic called Space Simulation. Simulations are computer programs that mimic real world phenomenon with accuracy. For example: You must have seen weather reports on the news. They usually show the simulation of how the air, clouds, winds work over a particular city. Let me show you a weather simulation over mumbai city.	
Teacher Initiates Screen Share		
Step 2: Teacher-Led Activity (5 mins)	We are going to use Accuweather website. You see if you play the simulation it shows you the formation of clouds and wind directions etc. This simulation is like a movie clip but the computer has done the calculation on real data provided by the satellites orbiting over the earth us and thus it shows the formations of clouds etc. accurately.	Teacher Activity 1-WEATHER  Click on Play button to play the simulation

	Simulation is an accurate computer model of a real phenomenon. Today we are going to create a simulation of the Sun-Earth System while learning amazing facts about space. Let me tell you a few facts about the Sun-Earth System. ● The Earth revolves or orbits around the Sun and the time it takes to complete one revolution is 365 days. That is what we call a year. For example: If you are 9 years old that means you have orbited the sun 9 times with the earth. ● The Sun is 150 million kilometers away from the earth. This distance is perfect for earth to support life. If the earth was closer to the sun, the earth would have become very hot. If it was further away, the earth will become ice cold. In both cases the earth would have been inhabitable, not able to support any life. ● The distance of the earth from the sun and the days it takes to revolve around the sun is directly proportional. This	

	means that if the earth moves further away from the sun the year would be longer than 365 days and if the earth moves closer, year would be shorter than 365 days. So the simulation will tell us the number of days it would take for the earth to complete one revolution around the sun based on its distance from the sun Let me show you the simulation first and how it works. Now let me show you how I made it. I will show you the design and code and you will have to make it once you fully understand it so pay attention and ask questions if you have any.	Teacher Activity 2-SUN-EARTH SYSTEM Click on Edit Project then do a live test and showcase the app.  **Explain in detail Each Design Component and its properties (especially the Sprite properties).**

NOTE - IF LIVE TEST DOES NOT WORK PLEASE -

1. **UNINSTALL THE THUNKABLE ALIVE APP FROM YOUR PHONE**
2. **THEN COPY THE LINK OF STUDENT ACTIVITY - 2 AND SEND ON YOUR MOBILE(USING WHATSAPP OR EMAIL)**
3. **THEN OPEN THE LINK ON YOUR MOBILE AND DOWNLOAD THE THUNKABLE ALIVE APP**
4. **NOW DO THE LIVE TEST**
5.

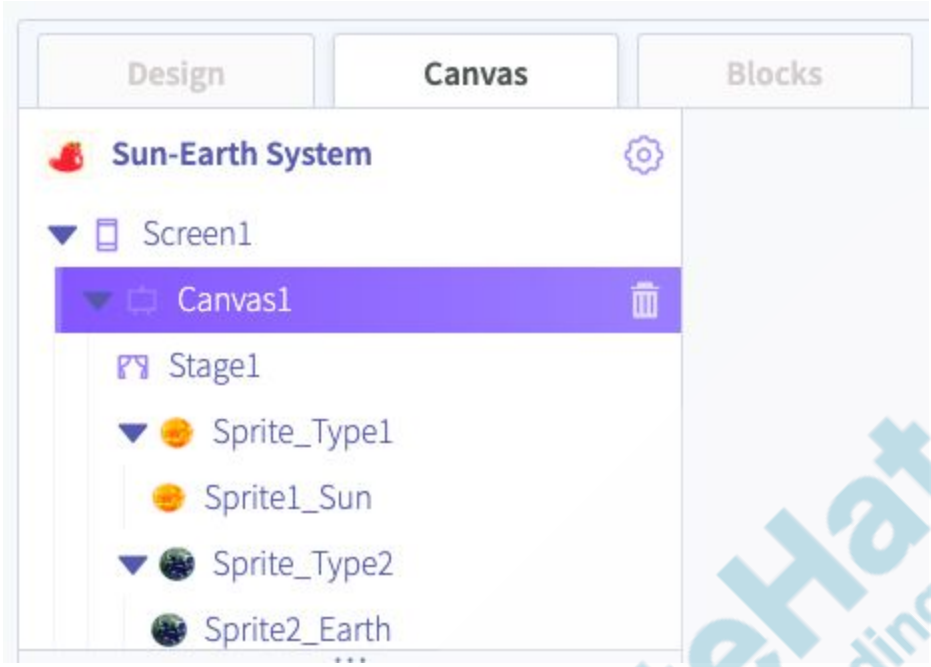
SUN-EARTH SYSTEM SIMULATION APP

The app simulated the Sun-Earth system. The user can drag the earth on the screen to change the Astronomical Unit (Distance between Sun-Earth) and the app will calculate the year in days or time required for one revolution according to the solar distance.

Design

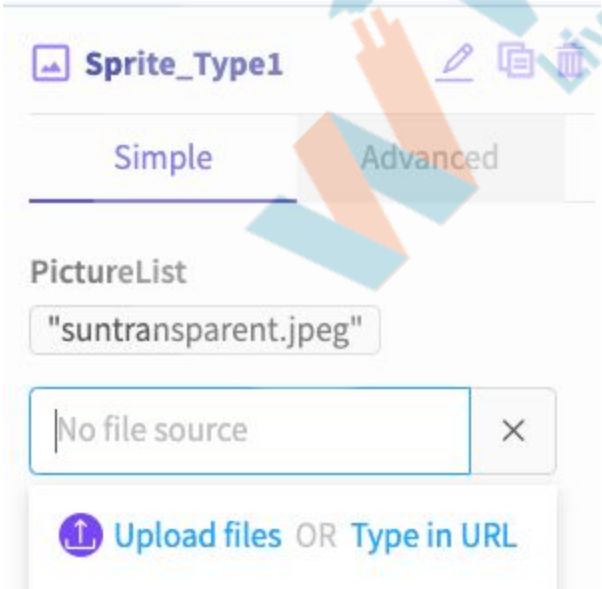
On Screen1 there is Canvas1, where we are going to simulate the Sun and Earth. The Screen1 also has Column1 with the Labels to display Solar Distance or Astronomical Unit in Million Kms and Revolution in days. There is a button to calculate the Revolution in days depending on the solar distance.

Canvas1



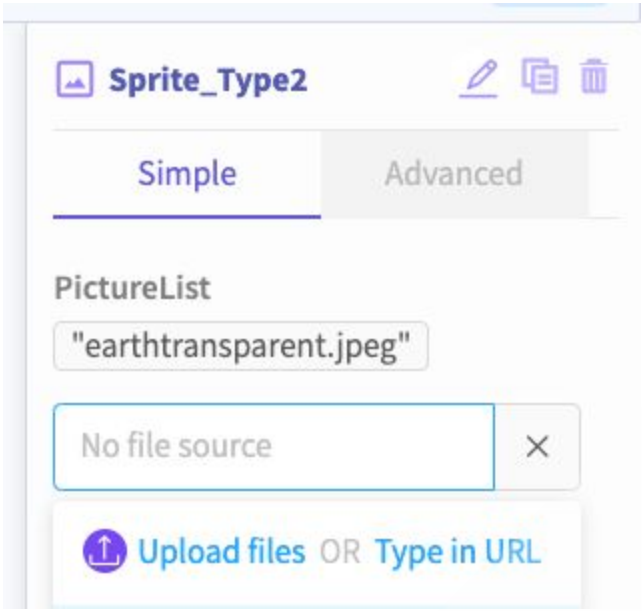
A Canvas is a component that allows animation. So here I have added two Sprite types with Sun and Earth images I have downloaded from the internet.

Sprite Type1:Upload the image of the sun with transparent background.

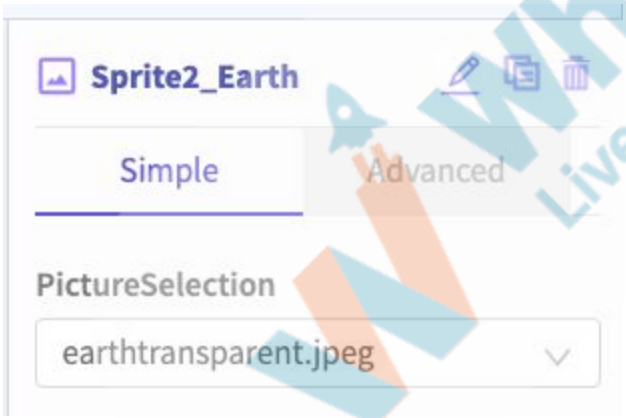


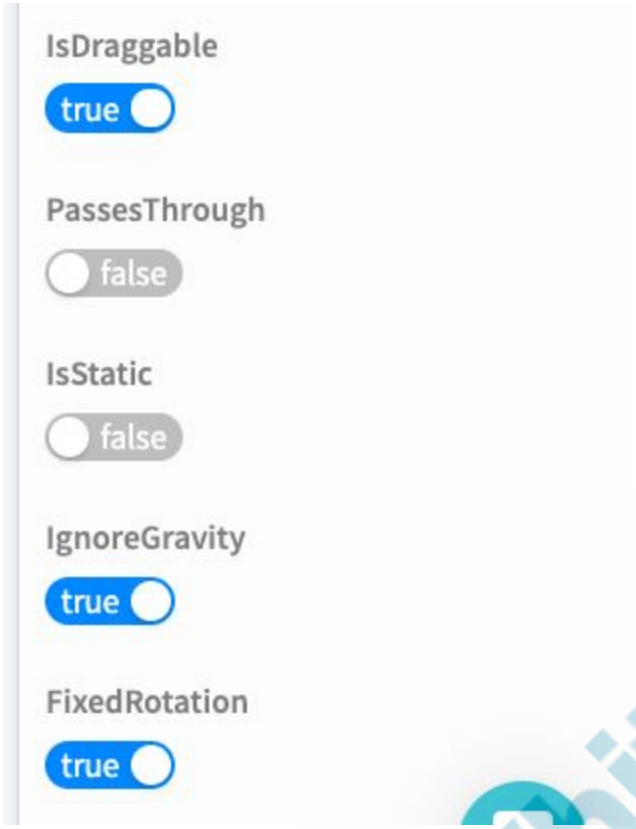
Sprite1: Sun: Set the image to sun.

Sprite Type2:Upload image of the earth with transparent background.



Sprite2: Earth: Set the image to earth.





You can add as many Sprites in Sprite Type 1 itself. I could have added both the sun and earth image to SpriteType1 but I want the Sun and Earth to have different behaviours so I have added them as separate Sprite Types.

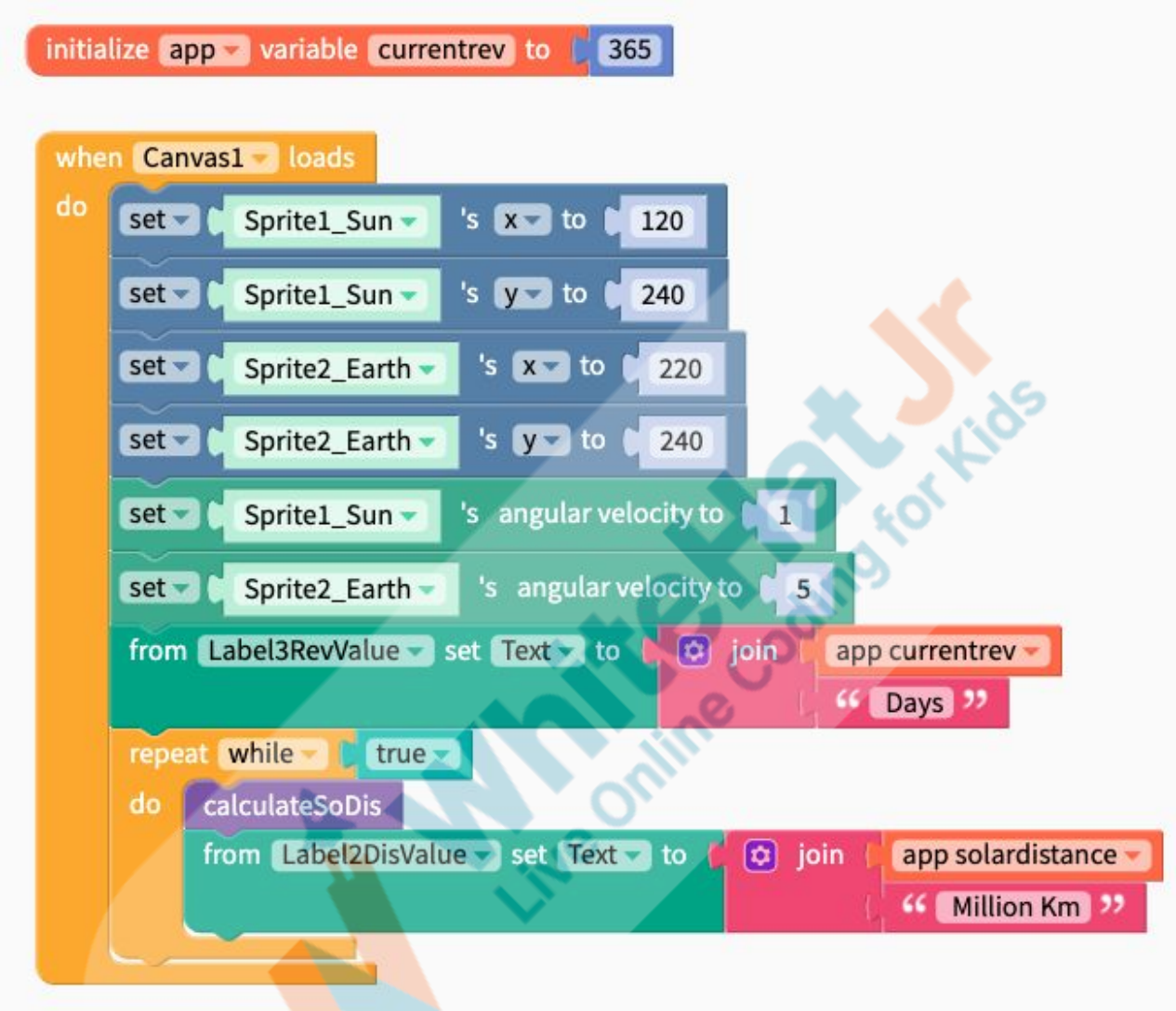
NOTE: Download small images. Large images would take longer for live test.

The SpriteType2 is draggable which means the user can drag the earth on screen unlike the SpriteType1:Sun which is static.

To add SpriteType2:

and set the sprite picture as earth.

Code:



We initiate a variable called current_rev to 365 which is the number of days in a years for the current solar distance between earth and sun. display it to the label when the canvas loads event under  section.

The angular velocity of the earth and sun is set to any number to simulate the effect of rotation. We will simulate the rotation speed with greater accuracy in the next class.

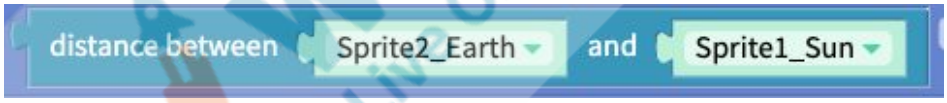
In the simulation When the user drags the earth on the screen, we want the distance to be displayed on the screen proportional to the actual distance in the solar system like when the user drags the earth 10 px away from the sun the label shoul show the distance

change of 15 million km, I can take any proporational distance here. we have used infinite while-loop using while true condition here. This ensures that the while loop executes infinitely or forever as the while condition is always true and keeps displaying the current position of the earth as the user drags it on the screen.

A new function called calculateSoDis which calculates solar distance according to the distance set on screen is executed infinitely in the while loop and this calculated distance is set to the label. eg. If the user move the earth by 10 pixels, this function will multiply it by 1 Million Km and say display as 10 Million kms, its just a ways of scaling. you can choose your own scale here.

Simulation should be accurate to predict what will happen in the real world if the distance between the earth and sun changes. So the distance on screen should be proportional to the actual solar distance.

As you can see, the surface to surface 10 pixel distance between the earth and sun is proportional to the 150 million km of actual solar distance.

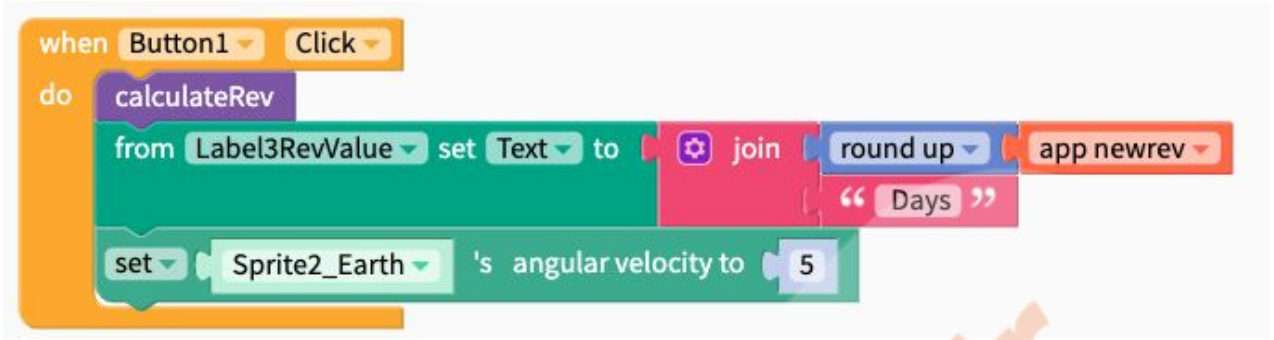
But the block 

in the function calculateSoDis() calculates distance from centre to centre distance.

So we subtract the thickness of earth and sun image from it. We subtract 90 to always get the surface to surface distance and multiply it by 15.
so 10pixels x 15 = 150. This is displayed as the solar distance.
We calculate the Time taken for earth to orbit around the sun using kepler's third law

$$T^2 \propto r^3$$

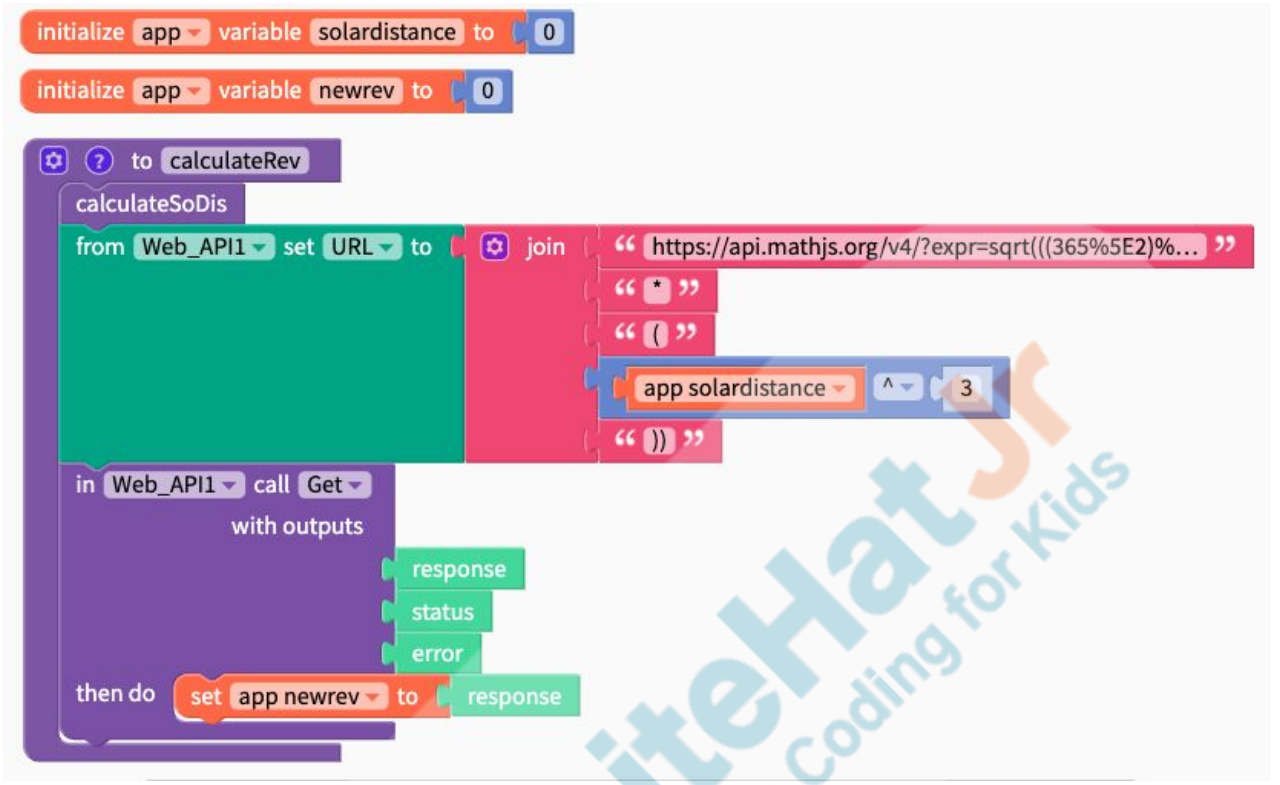
Kepler was a scientist before like Isaac Newton. Kepler's laws inspired Newton a lot.
This means Time of orbit is proportional to the distance. more precisely. The Square of time is proportional to the Cube of distance.
Dont worry if you dont understand it at this point we will use the same equation in other simulations as well. For now you just need to know that Distance from the sun is proportional to the time it takes to orbit. Now lets code this equation and give it a function name **calulaterev** for calculate revolution



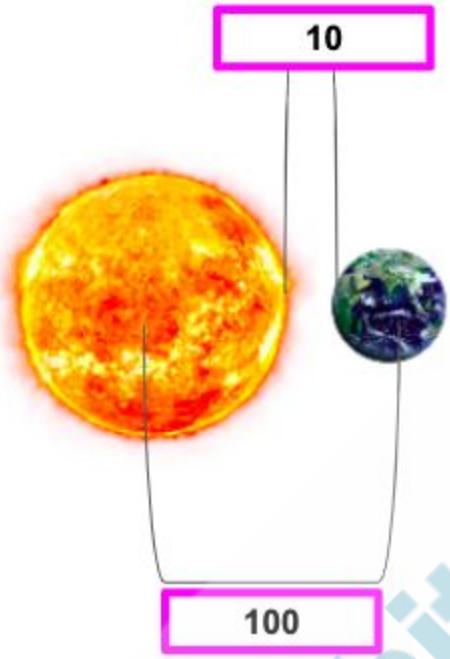
Now when the user moves the earth and clicks the calculate button to calculate the revolution time in days the calculaterev() function is called. It calculates the rev in days and stores this calculated value to newrev variable to be displayed in the label.

Also when the user drags the earth, the sprite stops rotating so we again set the angular velocity here for earth.
We first initialize a solar distance variable to 0, then get the actual surface to surface distance using calculateSoDis.

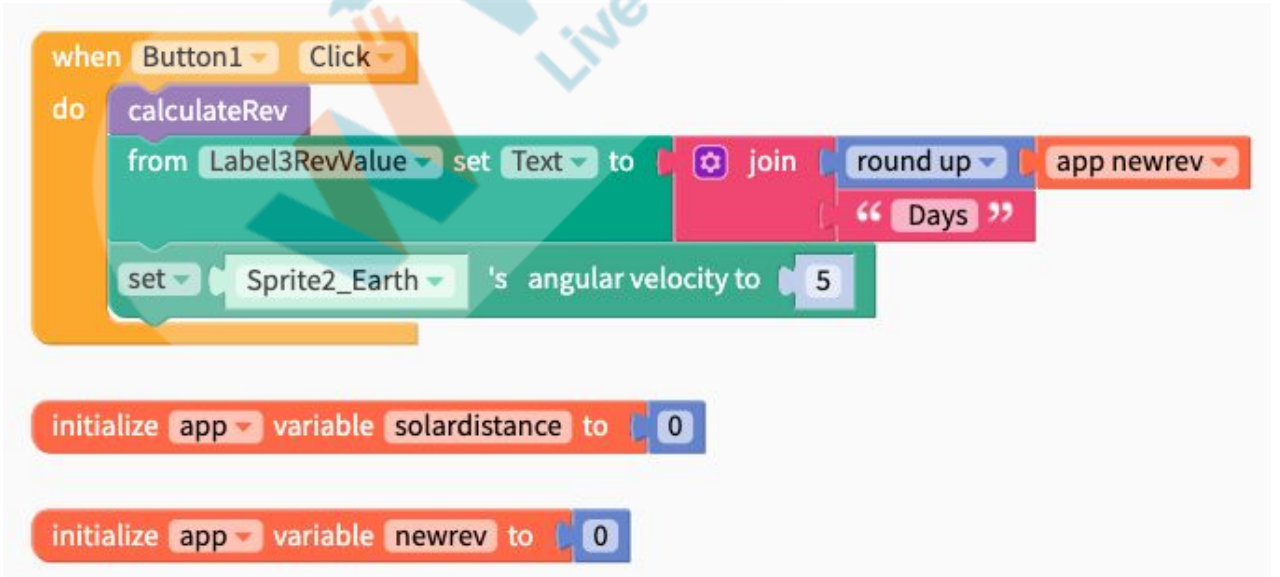
Then the function calculaterev calculates the revolution in days.

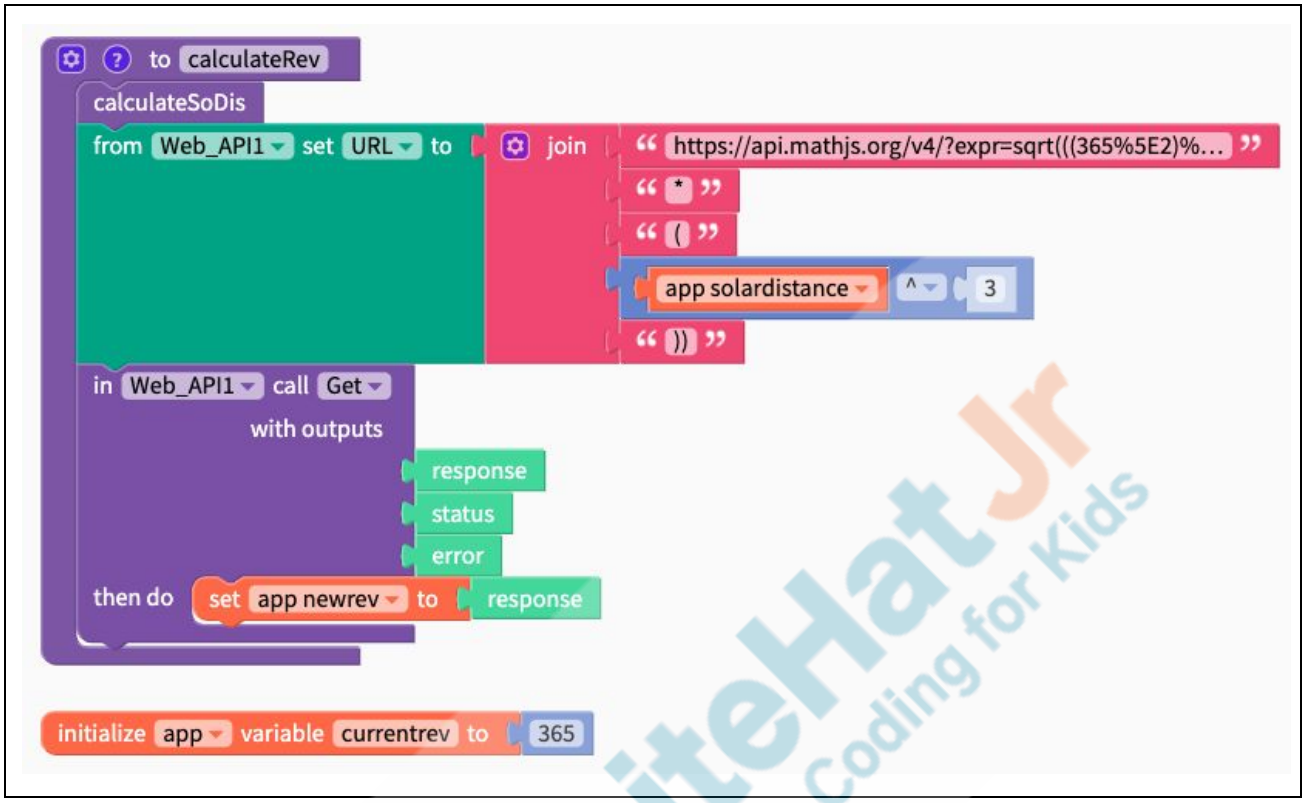


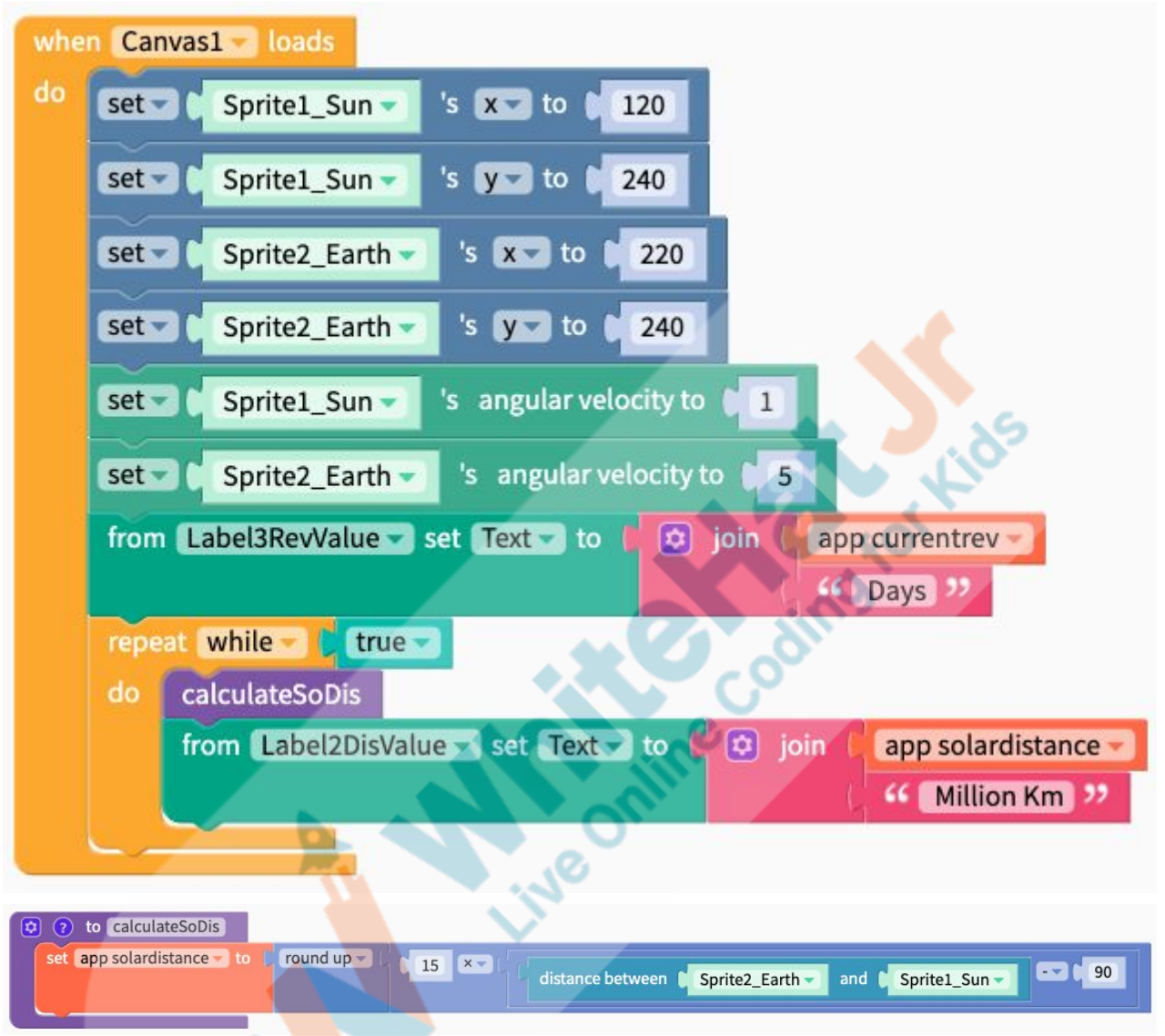
CalculateRev Functions just plugs in the the value of time and distance of earth which is set by nature that is 365 days and 150Million Kms and compares it with distance set by the user on the screen and thus calculates time taken for revolution or orbit the new distance set by the user.



Complete Code:







	Great! Now it is your turn. Let's Start.	

Teacher Stops Screen Share

- **Ask Student to press ESC key to come back to panel**
- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

Step 3: Student-Led Activity (10 mins)	Open Activity 1. The functions  are very mathematical so I have already coded them for you. You just have to use them wherever required.	[Student Activity 1- SOLAR SIMULATION] Make sure the student follows all the design and code steps in detail. Encourage the student to code freely if there is time
Teacher Initiates Screen Share		
Step 5: Project Pointers and Cues (5 min)	**Project Name: SUN EARTH WEBSITE** **Problem Statement:** As in the class you have learned a lot of new concepts regarding sun, earth and how the earth revolves around the sun. Your project is to create an informative website that will be having all the concepts that you have learned now. I am very excited to see your website and I know you will do really well. Good Luck!	
Teacher Guides Student to Stop Screen Share		
Step 4: Wrap-Up (3 mins)	You did great today as well. You have bagged two hats off. Let's revise What is a simulation?	*(Give at least 2 hats off)* Press the Hats Off Icon for ***Creatively Solving Activities*** 

	Simulation is an accurate computer model of a real phenomenon. What is the difference between rotation and revolution? Rotation is circling around itself like the earth completes 1 rotation in 24 hours and 1 revolution in 365 days. What is the solar distance? Distance between the earth and sun this is also called as astronomical unit. What is Kepler's Third Law? Kellers third Law says that Orbital Time and Solar Distance is proportional, More precisely. Time squared is proportional to Distance Cubed.	Press the Hats Off Icon for ***Great Question***  Press the Hats Off Icon for "***Strong Concentration..***" 
Teacher Clicks		
Additional Activities	Student can insert a moon sprite to make the simulation look attractive.	

Activity	Activity Name	Links
Teacher Activity 1	WEATHER	https://www.accuweather.com/en/in/mumbai/204842/satellite/204842
Teacher Activity 2	SUN-EARTH SIMULATION	https://x.thunkable.com/projectPage/5ec7b45a08a8c7231e888b1e

Student Activity 1	SIMULATION APP	https://x.thunkable.com/projectPage/5f0595cf2e58f204425eb420
Student Activity 2	THUNKABLE OLD VERSION	https://drive.google.com/drive/folders/1YNoVVy18MqEq07Gonbat-4A1aBvVzP9X?usp=sharing

NOTE - IF LIVE TEST DOES NOT WORK PLEASE -

1. **UNINSTALL THE THUNKABLE ALIVE APP FROM YOUR PHONE**
2. **THEN COPY THE LINK OF STUDENT ACTIVITY - 2 AND SEND ON YOUR MOBILE(USING WHATSAPP OR EMAIL)**
3. **THEN OPEN THE LINK ON YOUR MOBILE AND DOWNLOAD THE THUNKABLE ALIVE APP**
4. **NOW DO THE LIVE TEST**