



Guide For

Hatch Kid's Blocks-Based Beginner Curriculum

Project 1:

Build your own 3D Story



Build your own 3D Story

Objective

In this guide we are going to learn about the basics of the Hatch Kids workspace, and how it can be used to design and code your own 3D/AR/VR games, all the while building your own 3D story. We will learn how to move objects in a 3D space using block coding, and how make objects interact with each other.

Concepts covered:

Basics of Block coding & Hatch Workspace

Motion in 3D Space

Understanding event blocks

Understanding speech and sound blocks

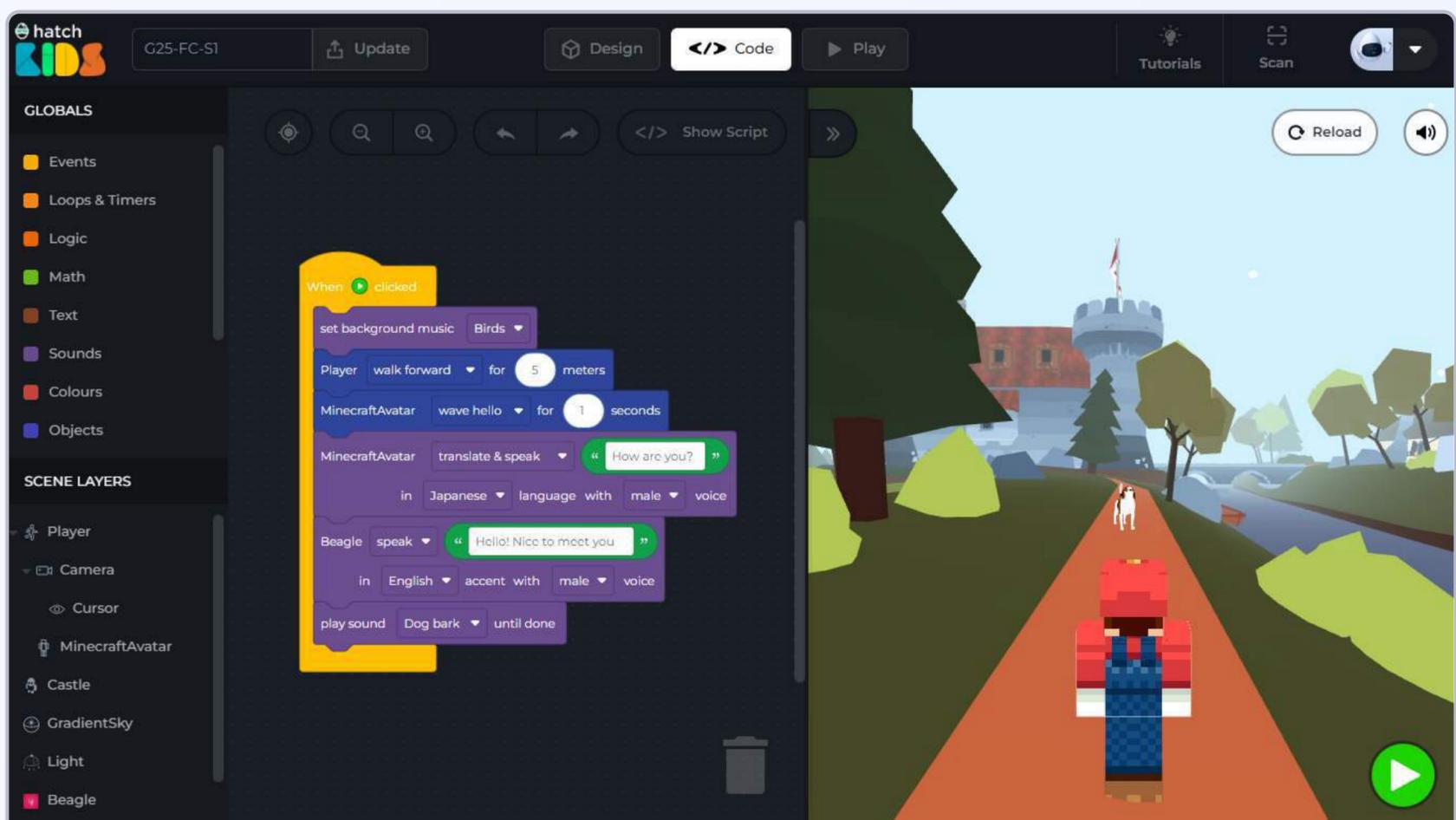
Making characters interact using speech blocks

Final Output Link:

<https://kids.hatchxr.com/@XR4schools/G25-FC-S1>

Student Template Link:

<https://kids.hatchxr.com/@XR4schools/G25-FC-S1-template>

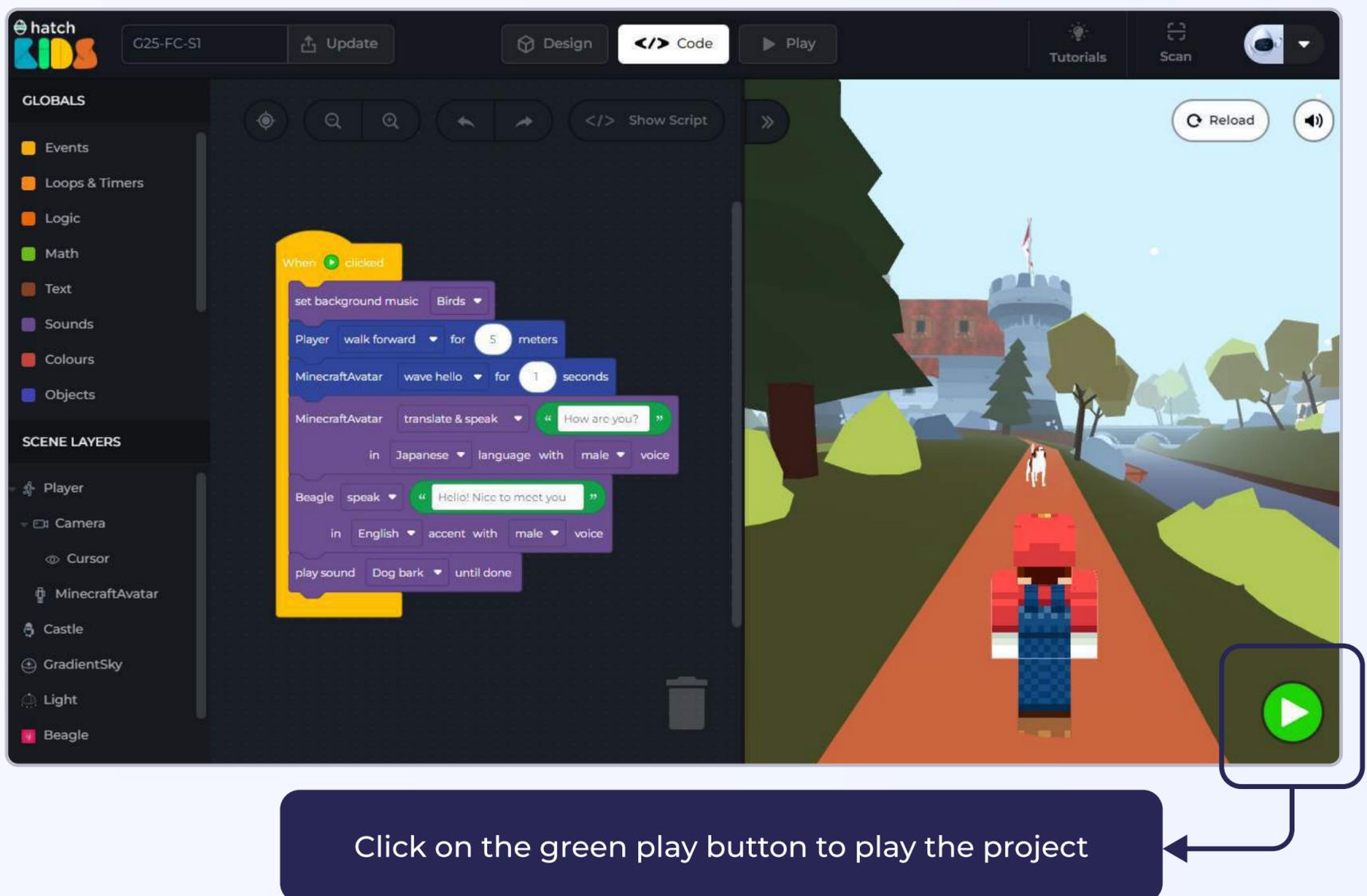


How it works?

Let's first understand what we are going to be building in this session.

Open the completed project link: <https://kids.hatchxr.com/@XR4schools/G25-FC-S1>

You will see a screen as shown below:



The moment you click on the **“Green Play Button”**, you will notice

1. A background music start playing
2. The Mario character starts walking forward and moves closer to the dog and stops.
3. The Mario character waves hello to the dog, and speaks, “How are you?”
4. The dog speaks back and says something, and then there is dog barking sound that plays.

We are going to learn how to code this simple interaction between two 3D objects using code blocks.

So let's get started.

Objective No. 1: Understanding the template

Step 1: We will start by opening the template link of the project mentioned here.

Student Template Link: <https://kids.hatchxr.com/@XR4schools/G25-FC-S1-template>

Step 2: The link above will open up an empty project with no code in the hatch workspace, that looks as shown here.

The screenshot shows the Hatch Kids workspace interface. At the top left, the project name 'G25-FC-S1-template' and a 'Remix' button are highlighted in a blue box. To the right, 'Design', 'Code', and 'Play' tabs are highlighted in a pink box. On the left side, a list of objects under 'SCENE LAYERS' is highlighted in an orange box. The central area is the code workspace, highlighted in a green box. On the right, a 3D game preview window is highlighted in a blue box. Below the screenshot, three callout boxes provide further details: an orange box for the object list, a green box for the code workspace, and a blue box for the game preview.

You can give your project a unique name and publish and share with everyone

These tabs help you switch between the design, code and play windows

On the left panel you will see the list of objects added in the game

This is the code workspace. This is where you can add your code blocks to define the functionality of your game

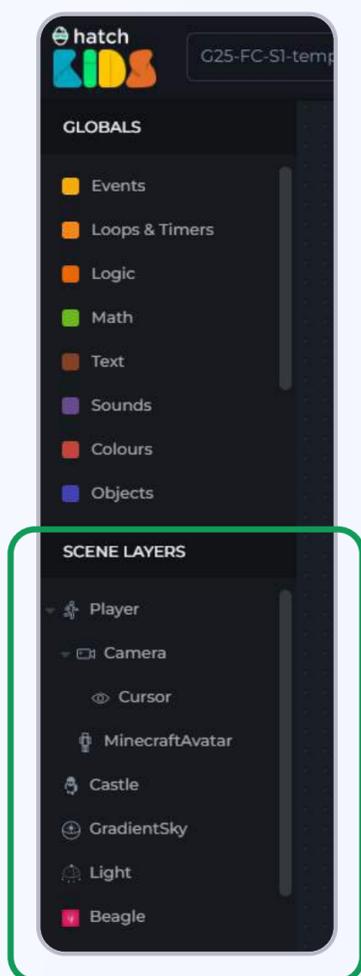
In the right half of your screen you can run the code and view the final look of your game.

As you can see, the code window of the hatch workspace opens up, and this is where you can use blocks to define how your game would work.

Objective No. 2: Understanding events and motion blocks

Let's start some coding. We start with learning how to move objects in 3D space.

You will notice that in the game there are some 3D objects present. There is a Minecraft character that looks like Mario, and there is a dog at some distance away from him. And both of these characters are present in a 3D environment with a castle and village scene around them.



Every 3D object that you see in the game has a unique name, and those names are listed in the left panel of the workspace.

To write a code where you want a certain object to perform an activity, you will need to:

1. Click on the name of the object
2. A list of block will appear, displaying all the activities you can perform with the selected object
3. Click and drag the block that you need in the workspace
4. Modify any input values present inside the block as per your requirements.

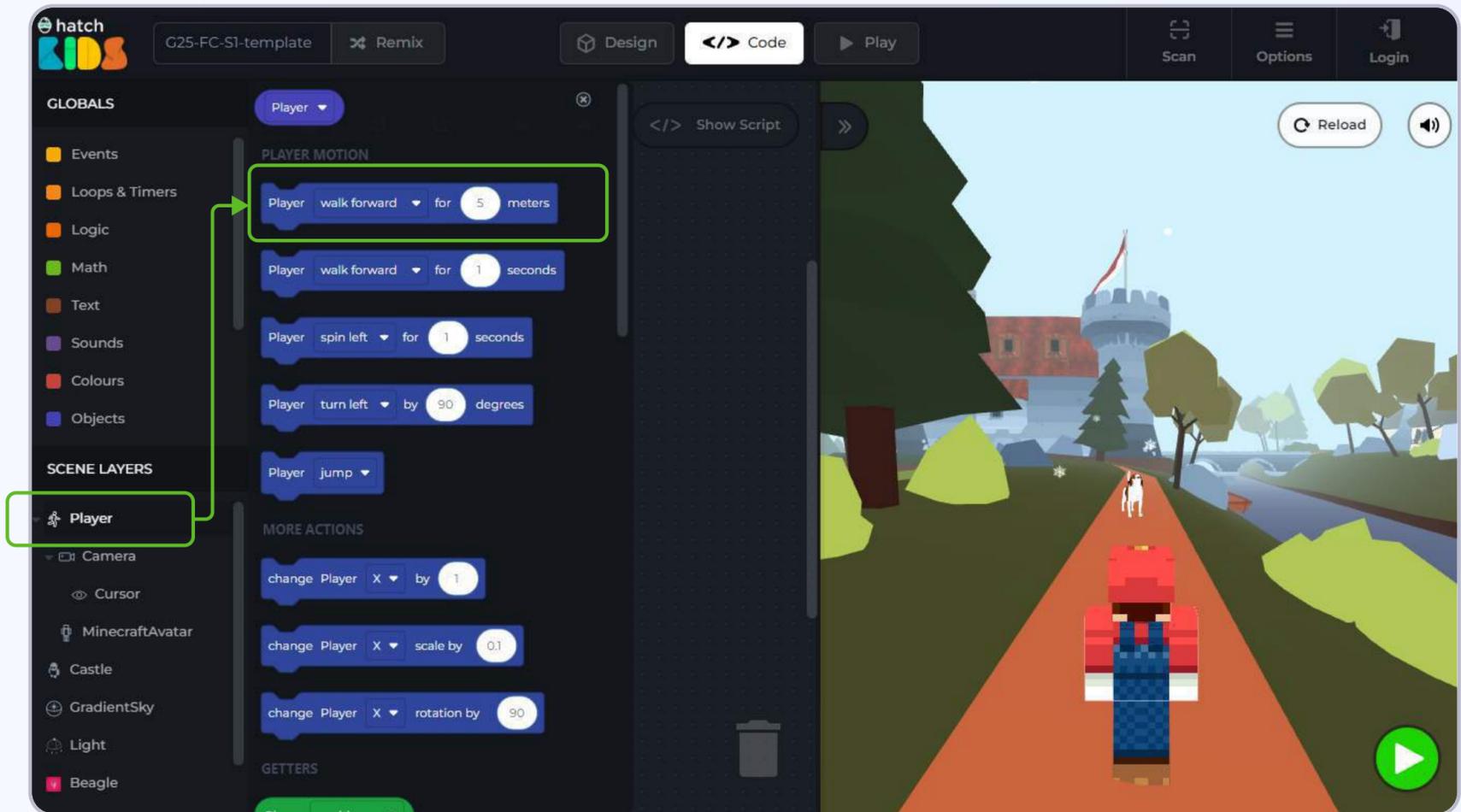
Let's start by making the player move forward in the game.

As mentioned above:

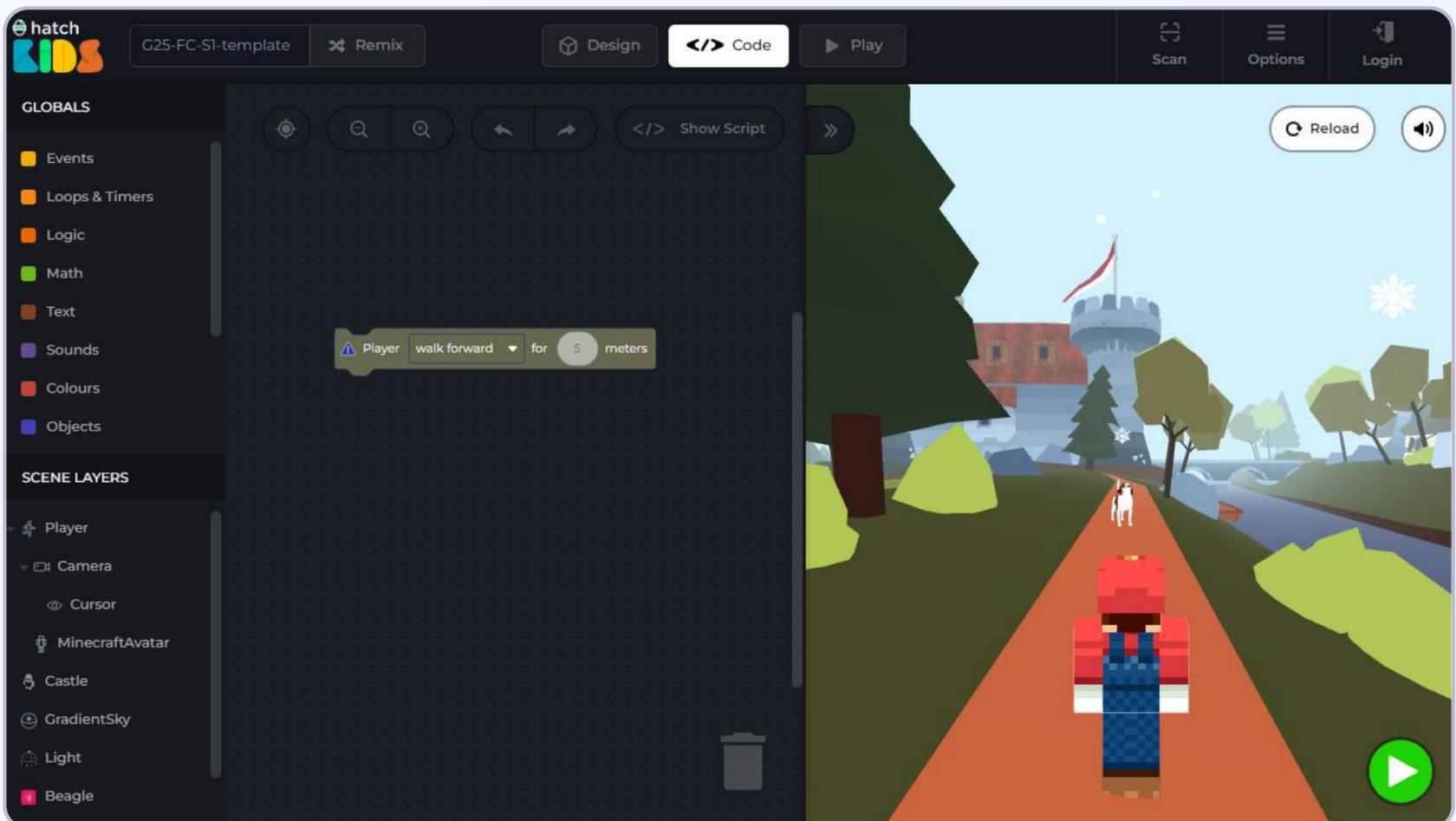
Step 1: Click on name **"Player"** in the left panel. You will see a list of blocks appear, that shows all the activity you can perform on the player object

Step 2: There is a block named **"Play move forward for 5 meters"**. Click on the block and drag it outside into the coding workspace.

Tidbit #1 : Player is a representation of your character in the workspace. Players have an avatar inside of them, like the Mario character you see here, and as you move the player and make it interact with other 3D objects, the Mario character will move and do the activities you want



Click and drag the **“Player move forward for 5 meters”** block inside the workspace



The moment you bring the “move forward” block inside the workspace, you will see it becomes gray. It means that the block is not active anymore.

Let's understand why is that.

Anytime you play a game, you are required to perform an activity first, like pressing a button, or tapping somewhere on the screen, or something similar; and based on what you did, different activities happen in the game.

There needs to be a way to communicate to the computer that the user is performing some activity.

This is where the **“Events”** block come - events tell the computer that the user has performed an activity (eg.: pressing a keyboard button).

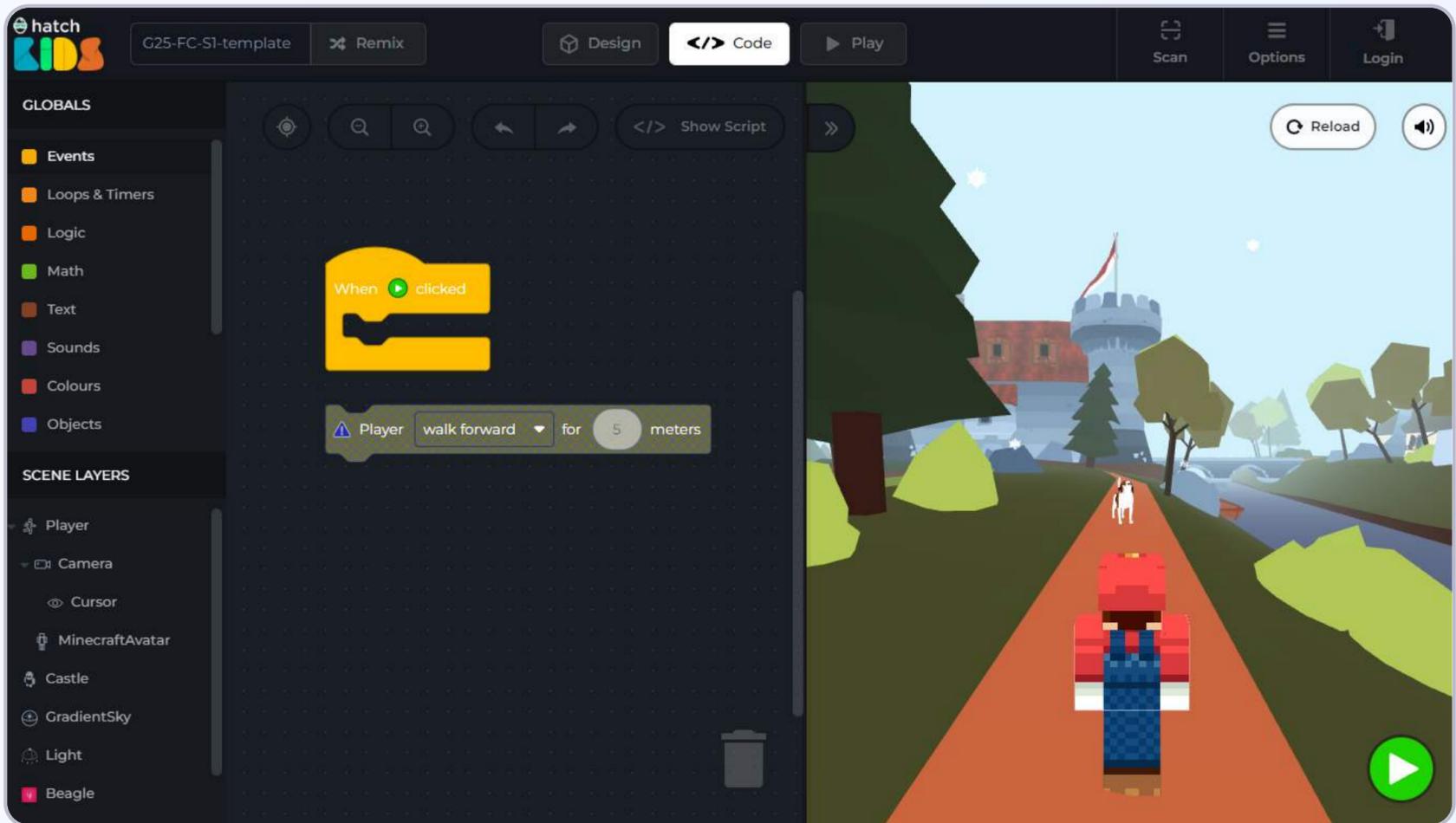
So, the player character in your game is not going to move on its own, it will be able to move when you press a button, or perform a similar activity.

Let's try and implement a similar functionality in our game.

Step 3. Click on the Events section in the top section of the left panel on your screen.

Step 4: You will see a list of yellow colored blocks appear. Lets click and drag out the very first yellow block from the list.

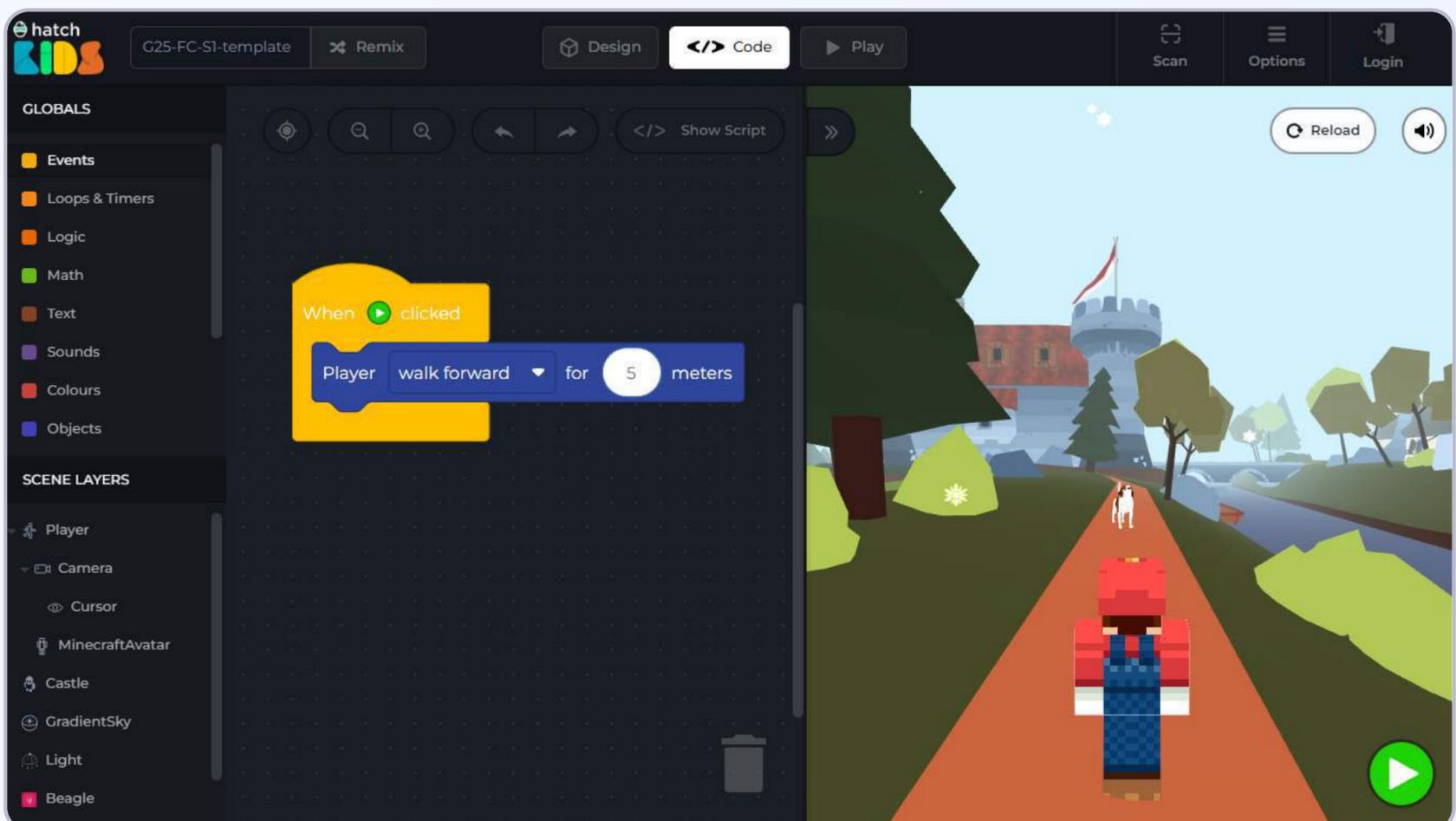




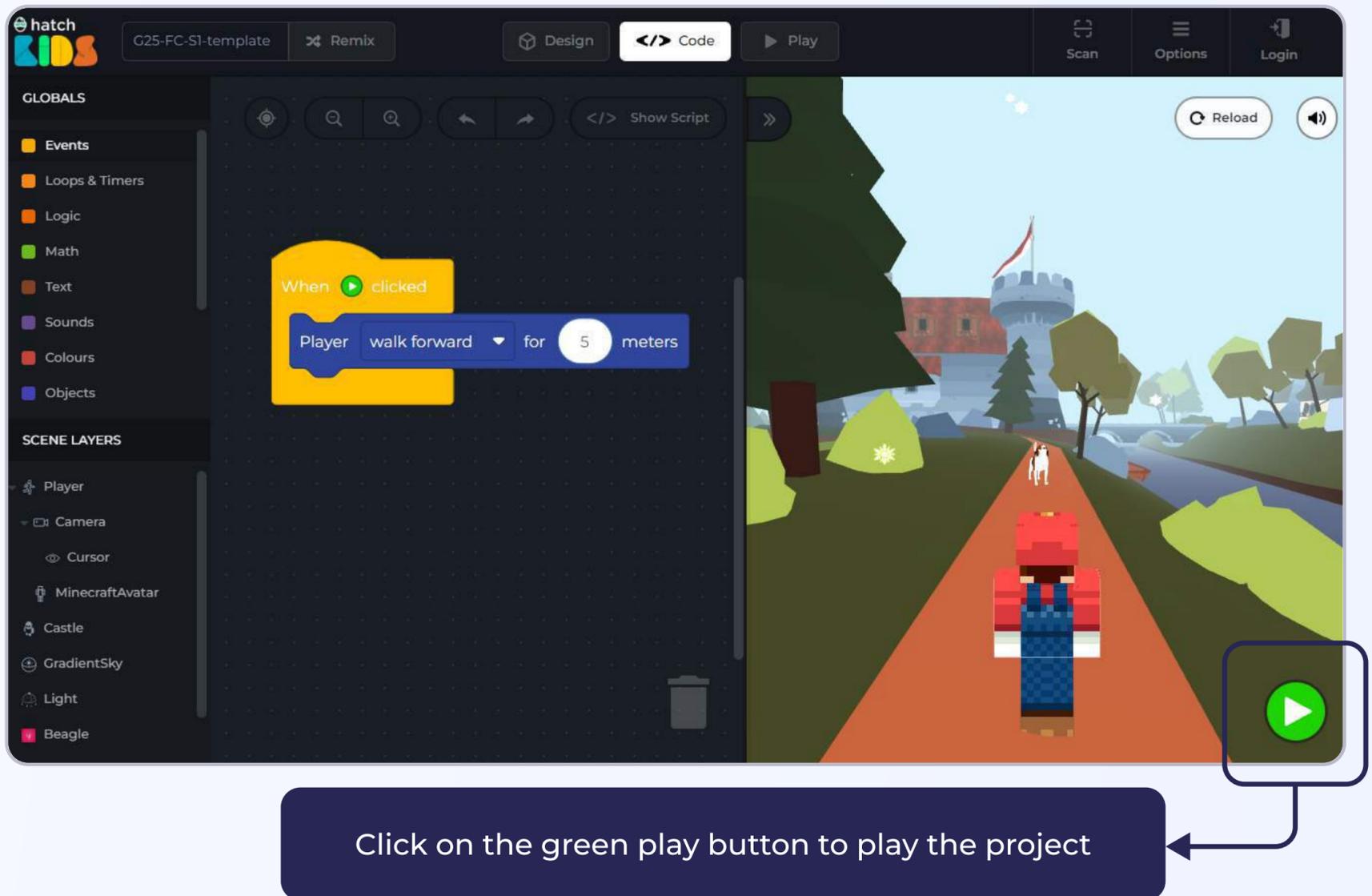
The yellow block that you just added in the workspace reads. “When green play button is clicked”.

You may also notice that there is an empty space inside the yellow block that is of the same shape as the “move forward” block.

Sep 5: Drag the “move forward” block and attach it inside the yellow block like they are puzzle pieces.



The combined block in your workspace would now reads, **“When green play button is clicked, player move forward for 5 meters.”**



You will notice that there is a “Green Play Button” on the bottom right corner of your screen.

Click on the green play play button.

You will notice that the moment you click on the green play button, the player character moves forward and stops after moving 5 meters.

Click on the green play button multiple times. Every time that you click on the green play button the player will end up moving 5 meters in it forward direction.

Congratulations. You just coded your very first action in a 3D game

Let's understand what you just coded.



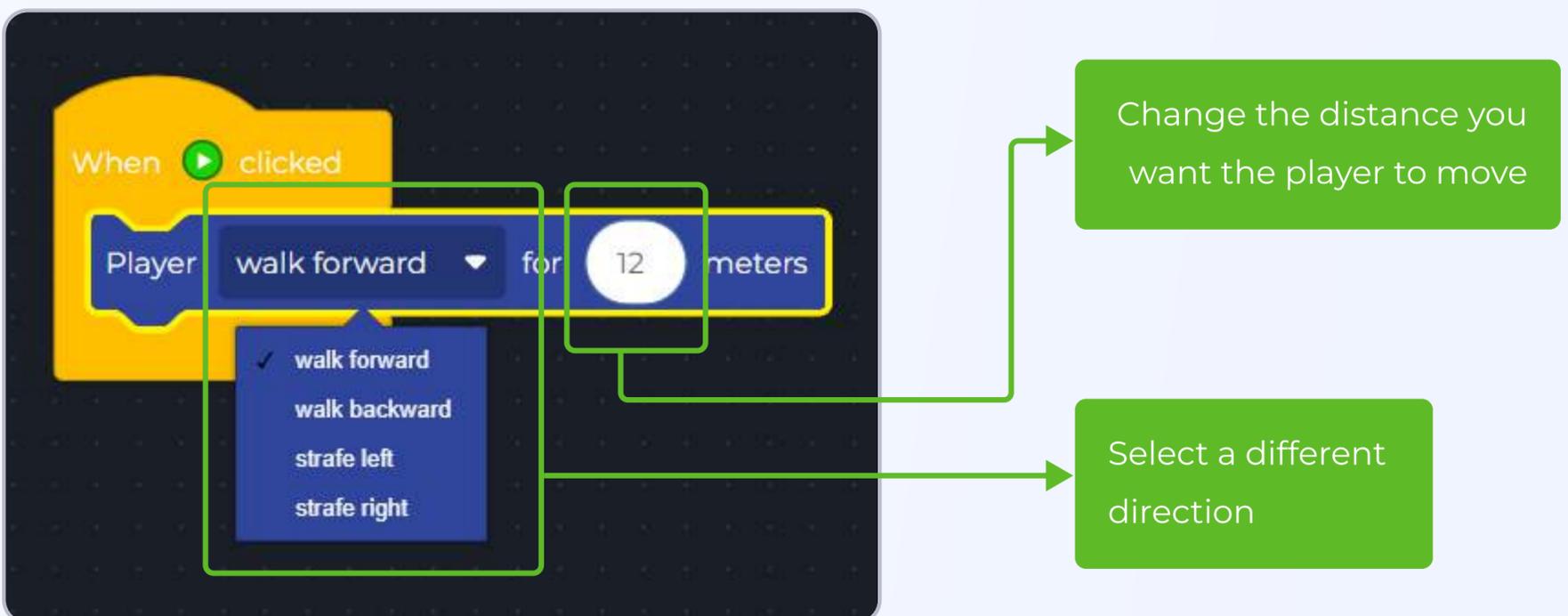
The yellow block is an **event block** that tells your computer to perform an activity (of moving the player 5 meters forward), **anytime you click on the green play button on your screen.**

The blue block is an action block that makes your player move 5 meters forward.

Together, they tell your computer that anytime you click on the green play button in the game, the computer should move the player character 5 meters forward.

You can **click on the number 5, and change it** to any number you want, and then you see the player move that many steps on every click of the green play button.

Also, notice that the **“walk forward”** is inside a box, if you **click on the box**, you will see there are **different options to make your player character move in different directions.** You can **select any direction** you want and **click on the green play button to see the changes take effect** in the game.



Tidbit #2 : After running a set of code, if you want to reset the entire game to how it was at the beginning, click on the **RELOAD** button to the top left corner in the game window

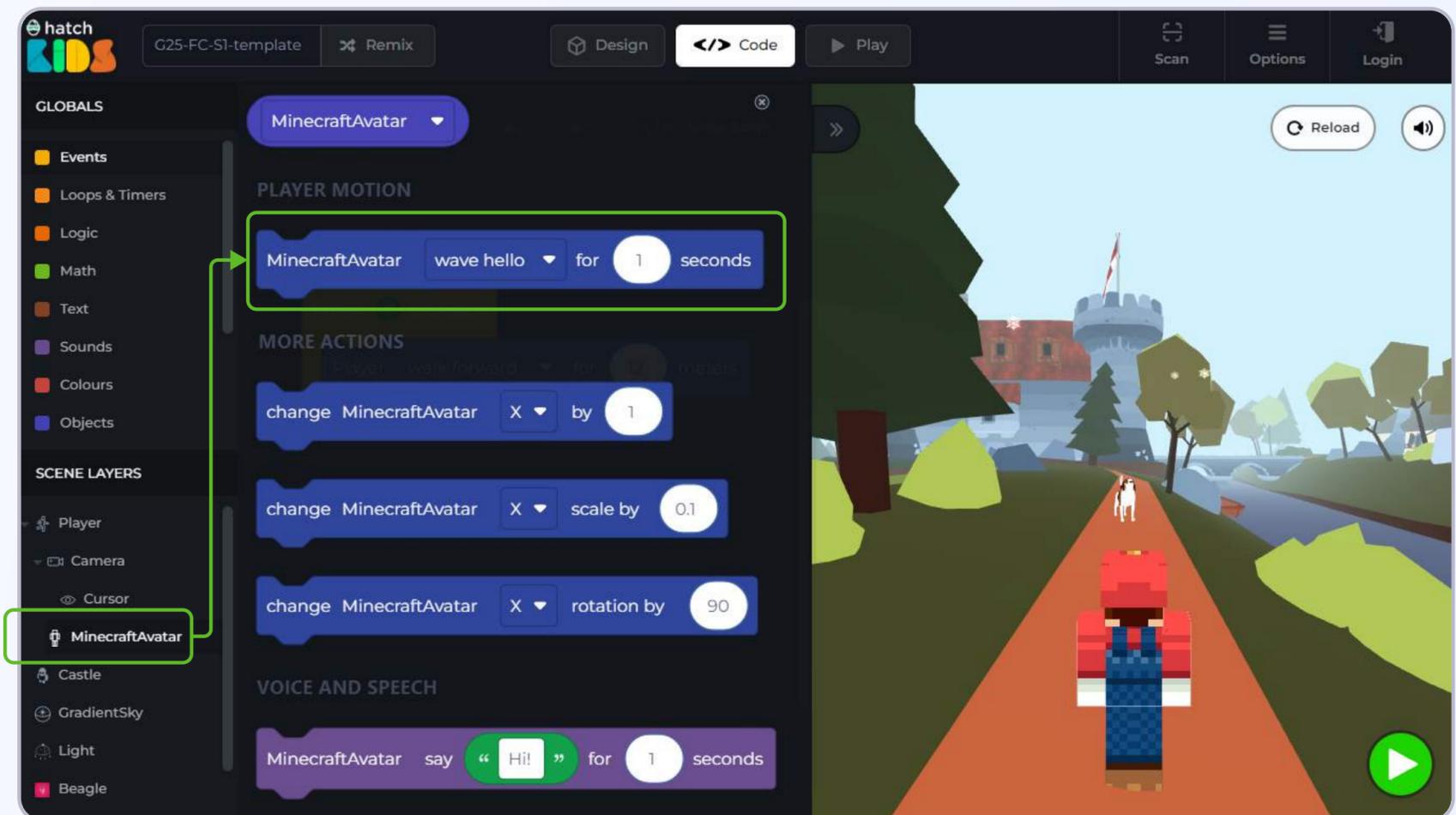
Objective No. 3: Make 3D objects interact with each other

Let's make the Mario character wave to the dog, once it moves close to the dog.

As mentioned earlier, if you want an object to perform an activity, you will have to click on its name and then select from the available list of blocks.

So, to make the minecraft character wave hello to the dog:

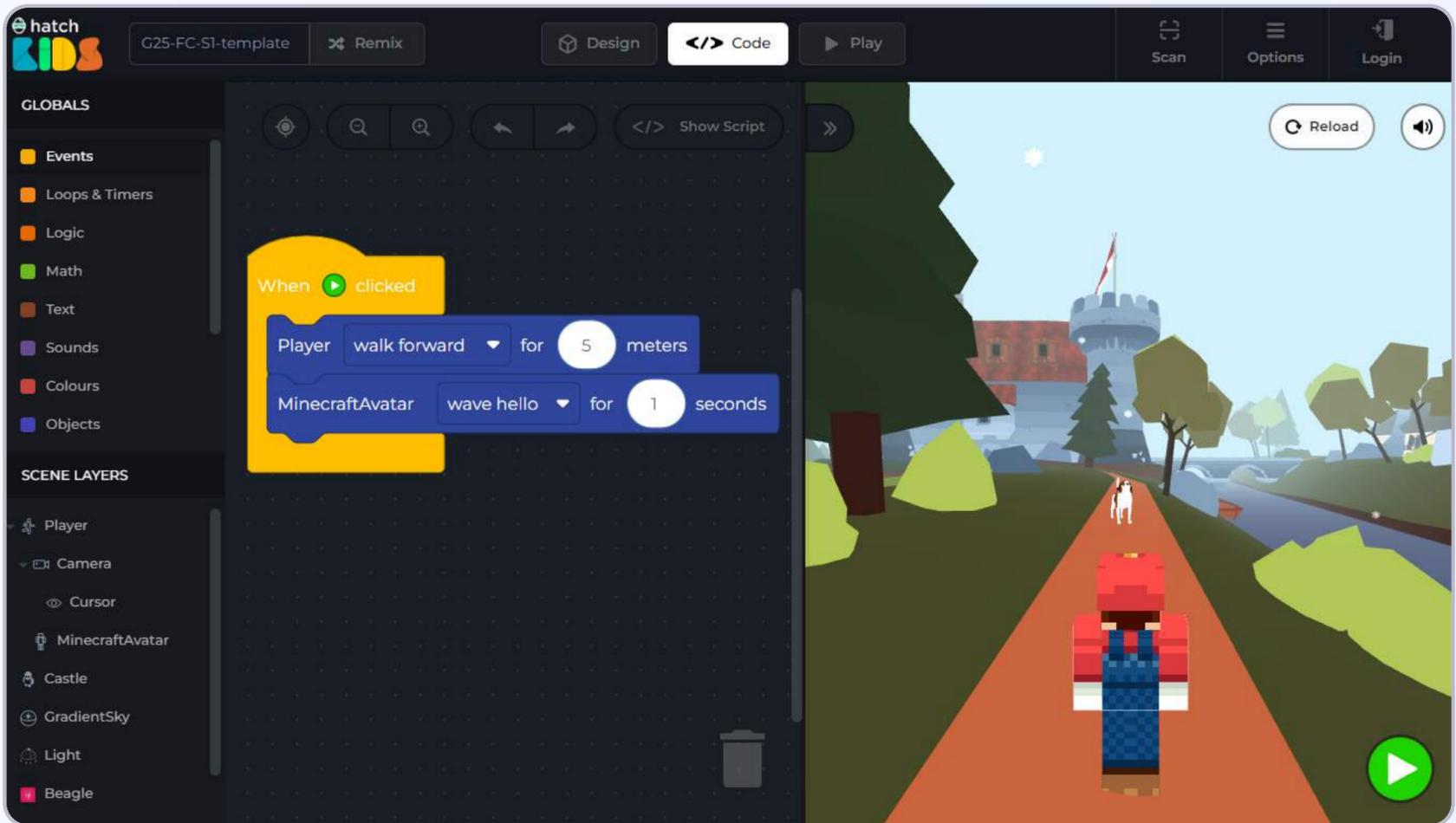
Step 1: On the left panel, click on the name **"MinecraftAvatar"** and you will notice that there is a block that says **"Minecraft Avatar wave hello for 1 second"**



Step 2: Drag out **"wave hello"** block and attach it below the **"move forward"** block, **inside the when green play button clicked block.**

You will notice, as soon as you bring the wave hello block near the when green play button click block, the yellow block's size increases to accommodate the new blue block.

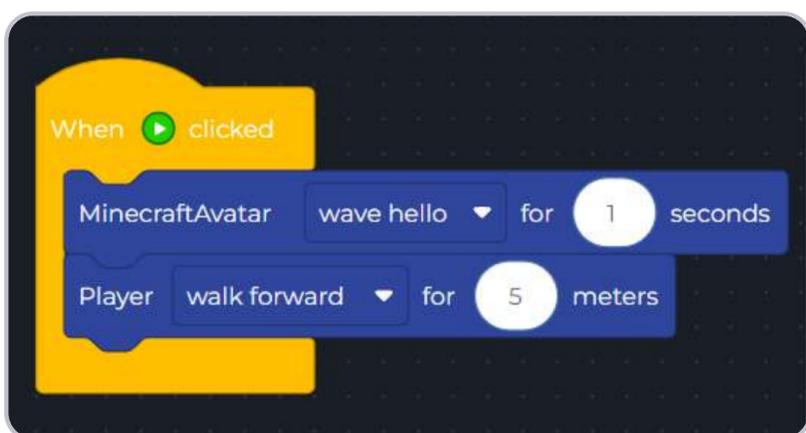
Once attached, your block will now tell your computer that, anytime you click on the green play button, player will move forward for 5 meters and then the minecraft character will wave hello to the dog.



Here as well, you can click on the number 1 inside the wave hello block and change it to any number you want.

After you have made all the changes, you can click on the green play button and you will see your character perform all the tasks in mentioned in the blocks.

Tidbit #3 : Once you have run your code, try changing the order in which the blue blocks are attached and then click on the green play button, and see how changing the order of blocks affect the way your game works.



Here, you will notice that the character first waves hello, and then moves forward.

Indicating that our computers run the blocks one after another in the order in which they are attached.

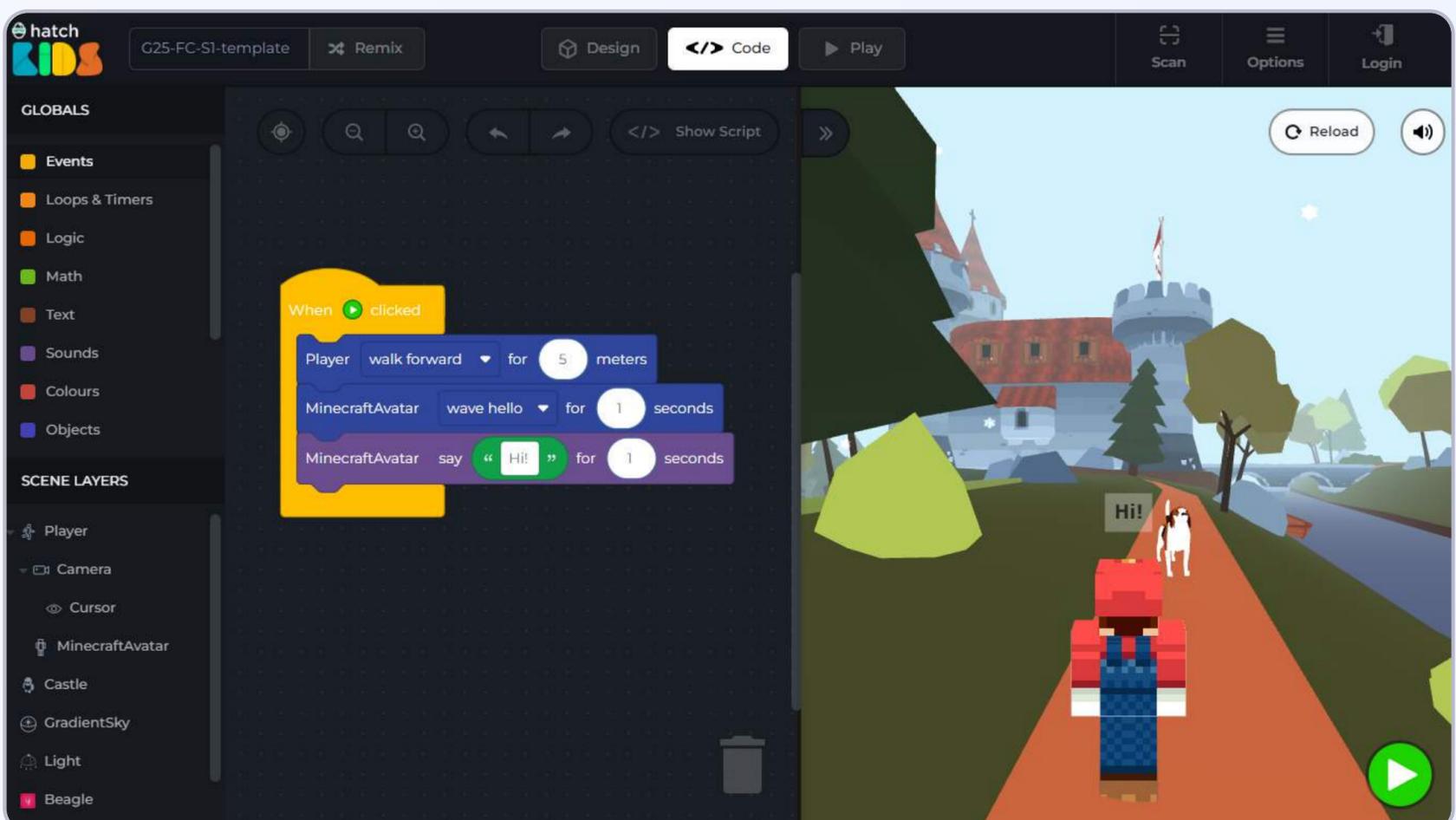
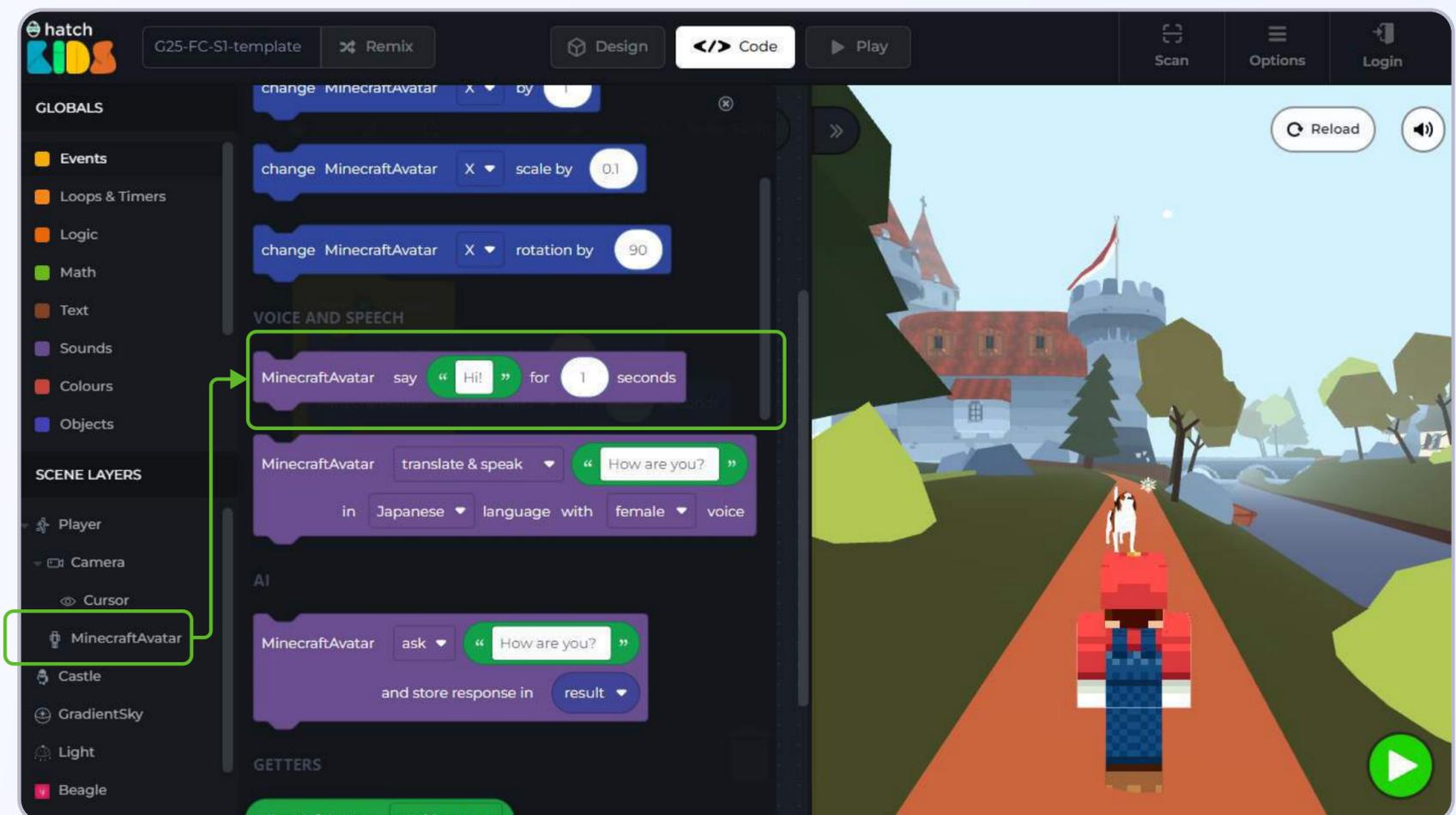
So, be mindful of the order in which you are attaching the blocks, its going to become a very important lesson for the games that you build.

Let's now make the Mario character speak to the dog.

Step 3: As mentioned earlier, click on MinecraftAvatar name in the left panel, and you will see a list of blocks appear.

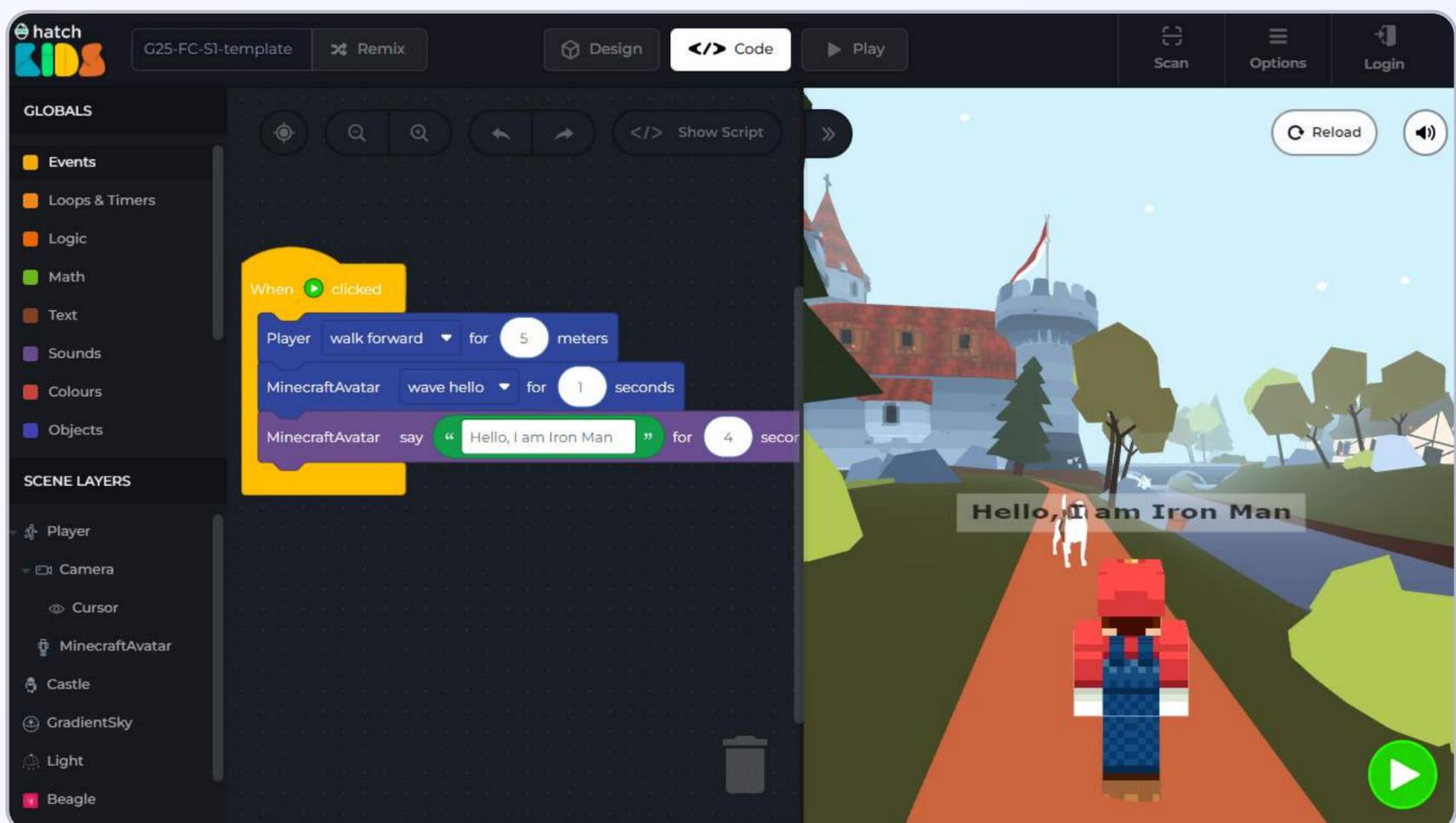
Scroll down the list of blocks and you will find a block that says, **“Minecraft say Hi for 1 second”**

Step 4: Click and drag the **“say hi”** block and attach it as shown.



When you click on the green play button, you will notice that the player walks forward, then the character waves hello and then a speech bubble appears above the character with the words “Hi” displayed on it for a duration of 1 second.

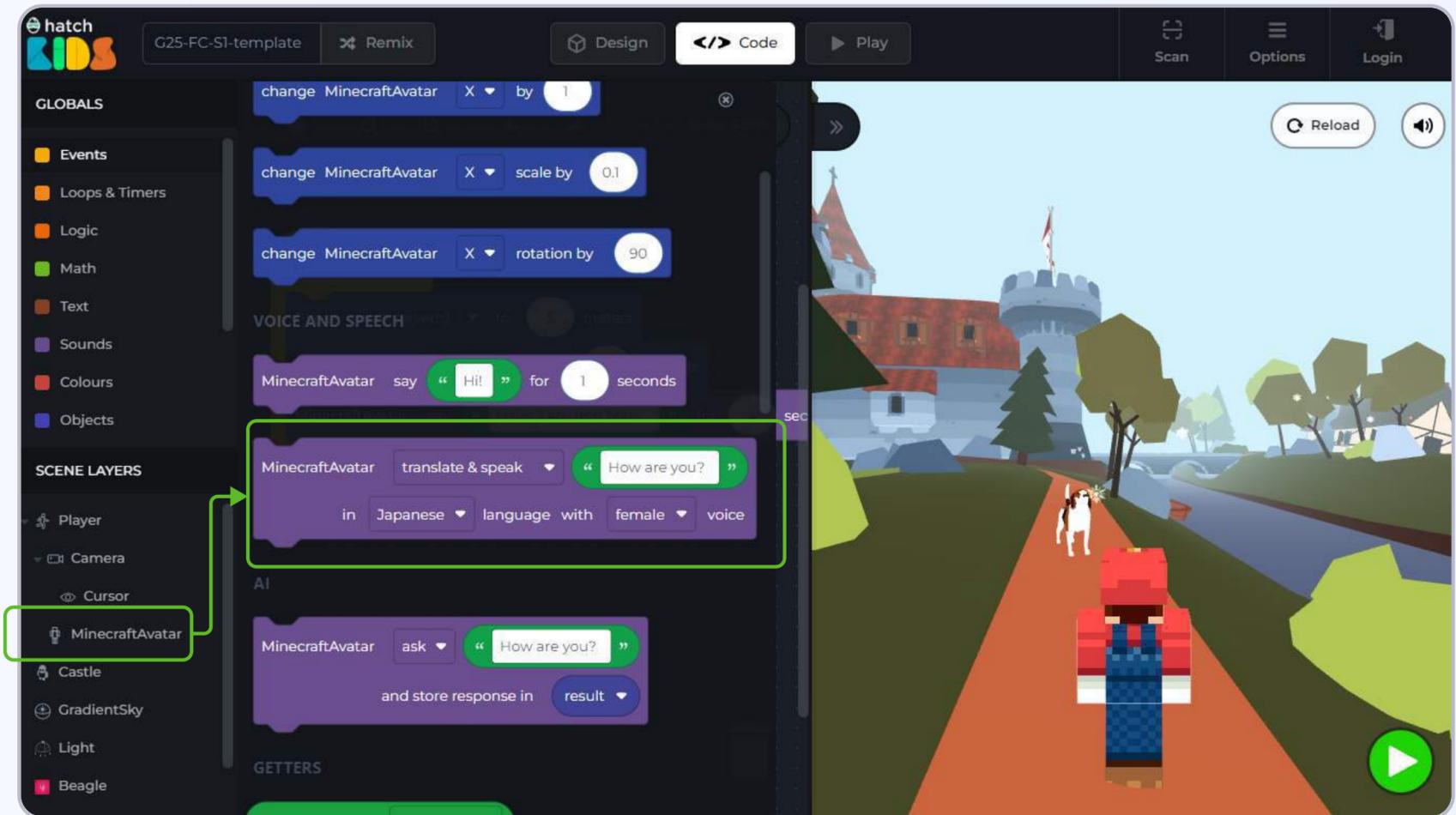
You can click inside the text box and change the text that says “Hi” to make the character say anything that you want and you can also change the time duration for which you want that text to appear.



You would have noticed that the “say” blocks only generate the text bubble, but doesn't generate any sound. Lets make the characters speak the text as well now.

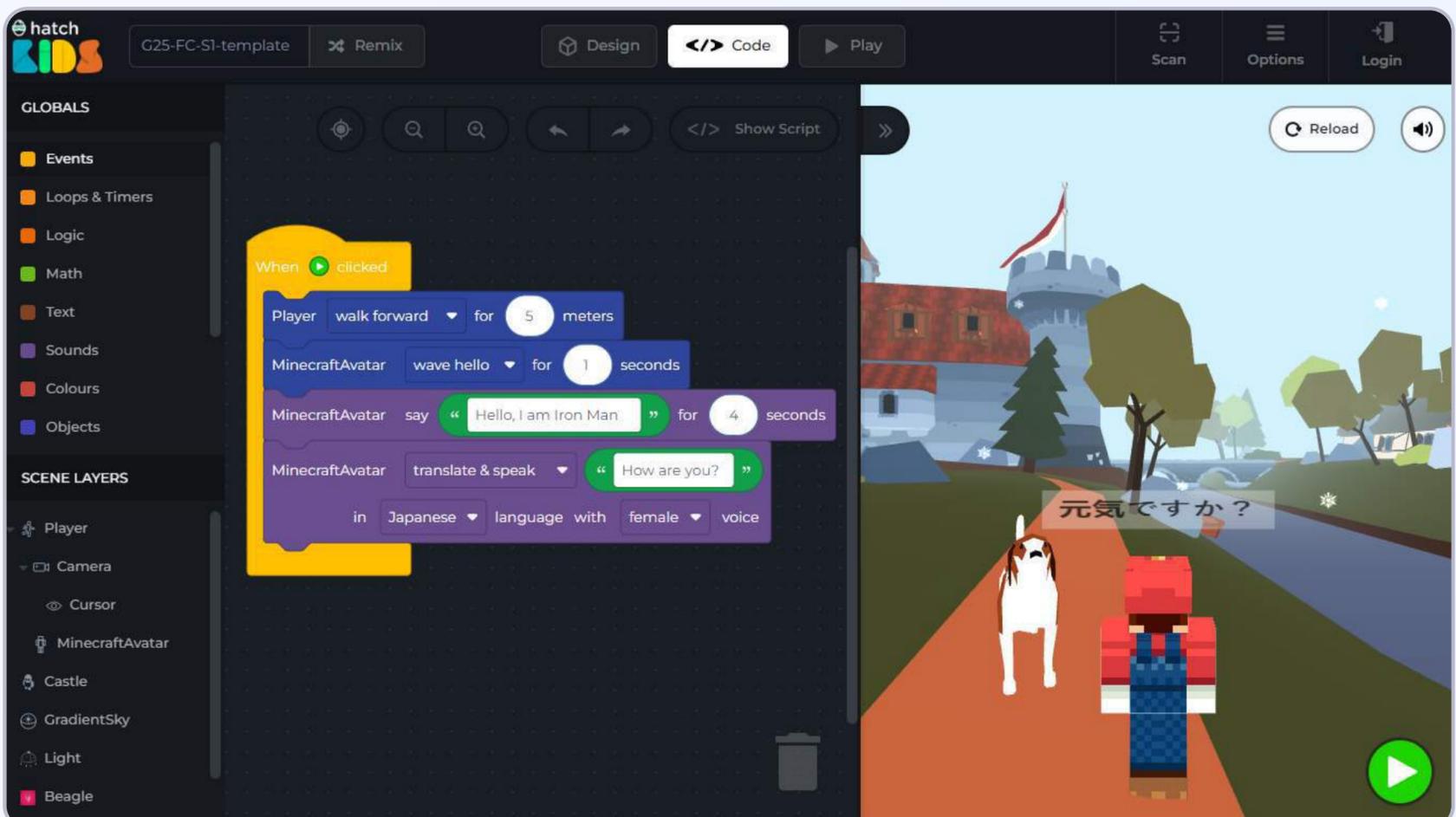
Step 5: Click on the name MinecraftAvatar on the left panel, and in the list of blocks that appear, scroll down and you will see a block that says “MinecraftAvatar translate & speak”

Click and drag the block into the workspace and attach it inside the “when green play button clicked” block.



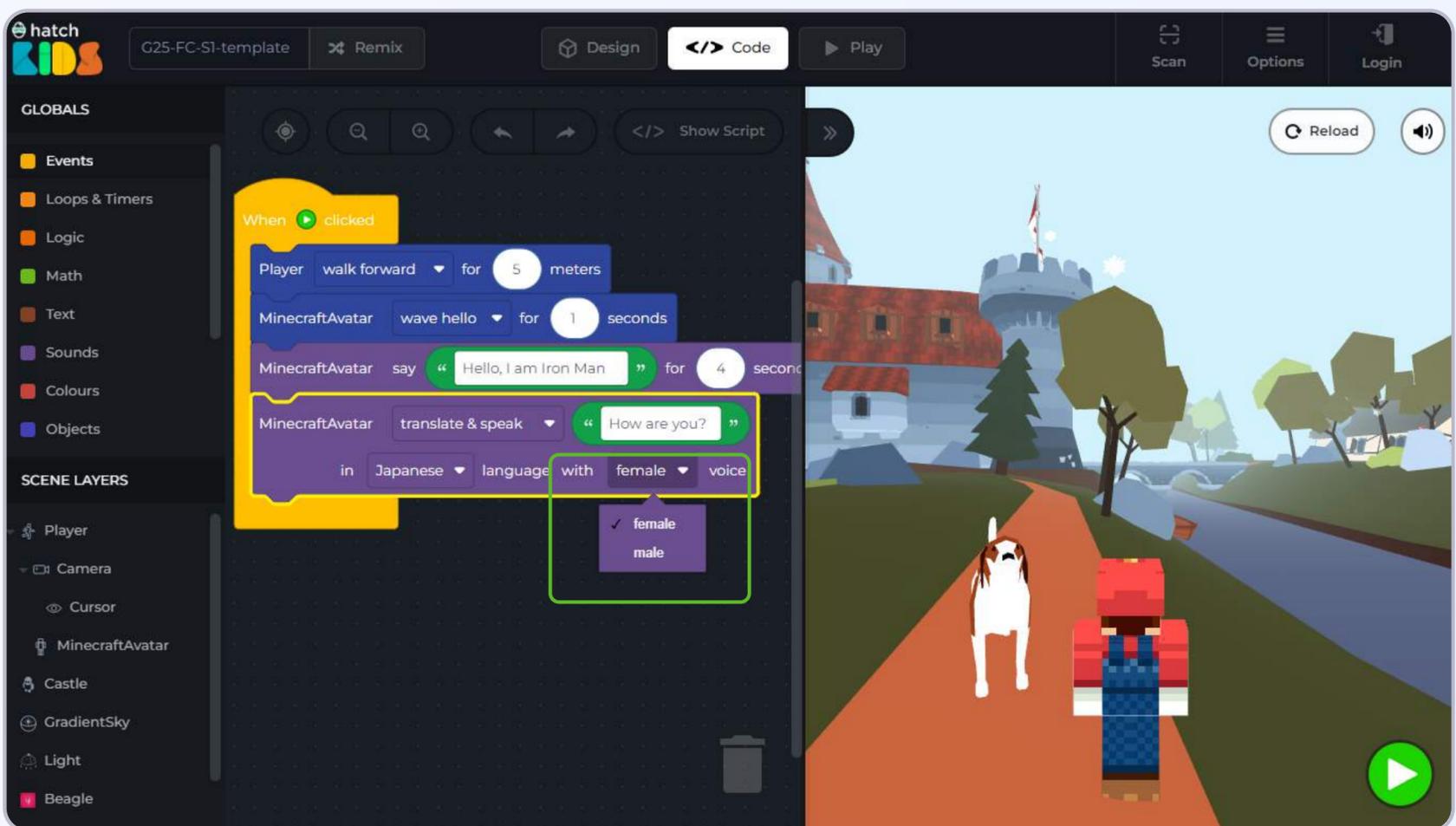
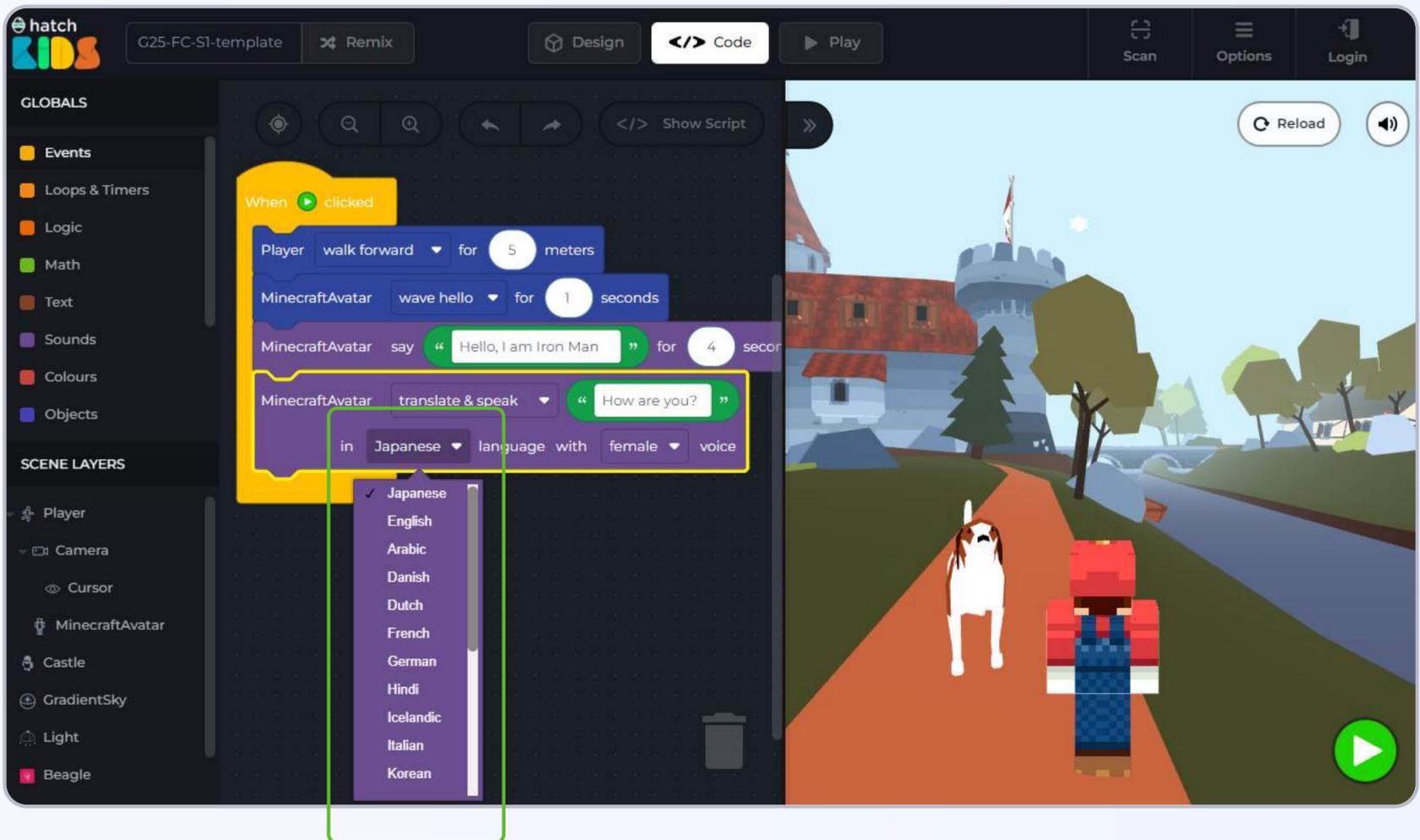
Click on the green play button and this time, you will notice, the player walk forward, then wave hello, and then the speech bubble for the “say” block appear. After that you will see the speech bubble for the translate & speak block appear and you will hear the character speak the words written in the text bubble, in the language selected and in the male/female voice that’s selected.

Even inside the text bubble of translate & speak block, the text would appear in the language of your selection.



In the translate & speak block, you can change the language to your choice, and you can even select the gender between male or female, and change the text input to anything you want the character to say.

Tidbit #4 : There is a volume button at the top right corner of your game window that you can use to increase or decrease the volume of the translate and speak blocks.

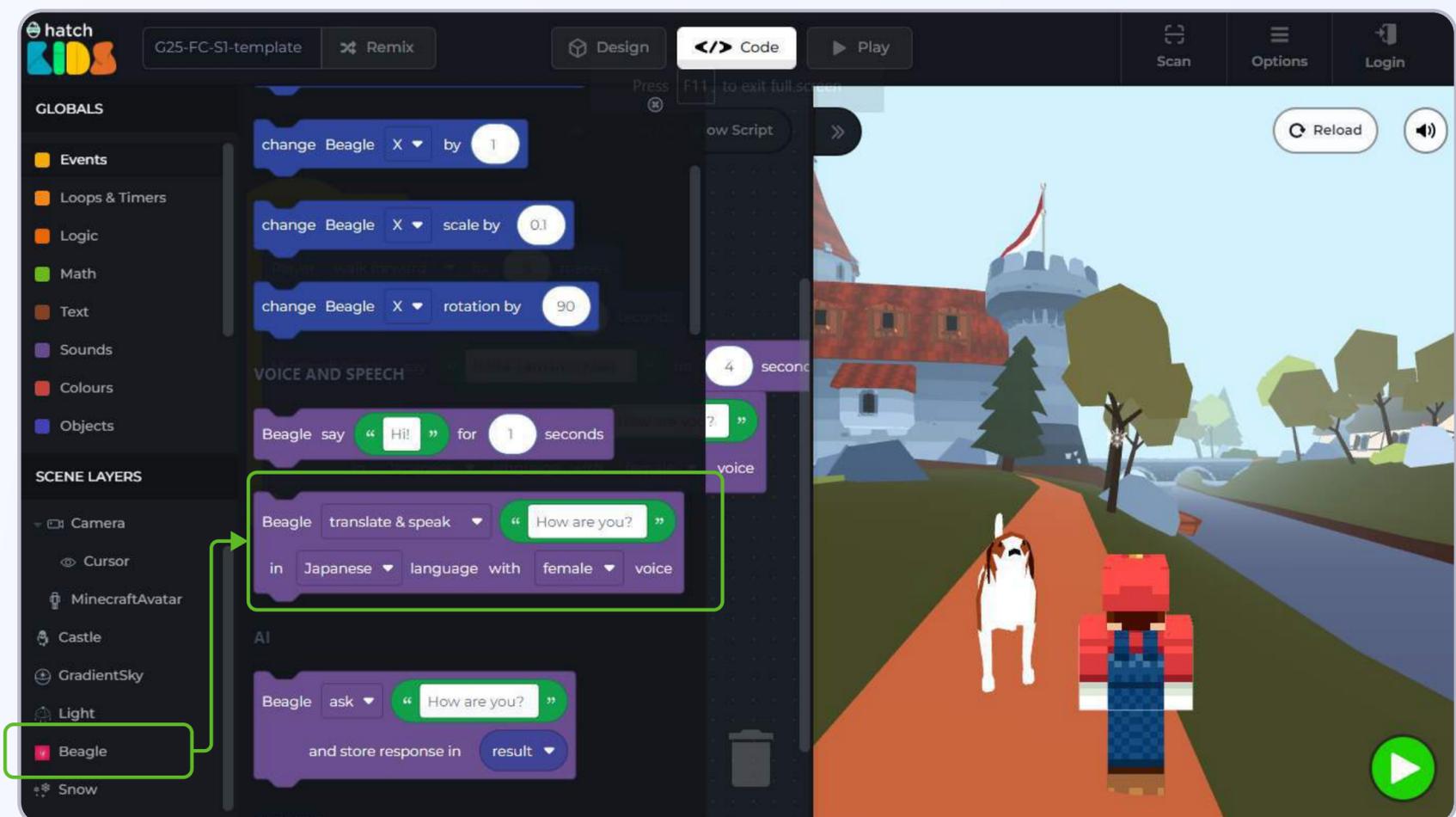


Let's now make the dog reply back to the minecraft character. Just like we use the translate & speak block to make the minecraft character speak, we can also use it to make the dog speak as well.

Step 6: Since we want the dog to speak this time, on the left panel, we first have to find the name of the dog object, which is **Beagle** in this case.

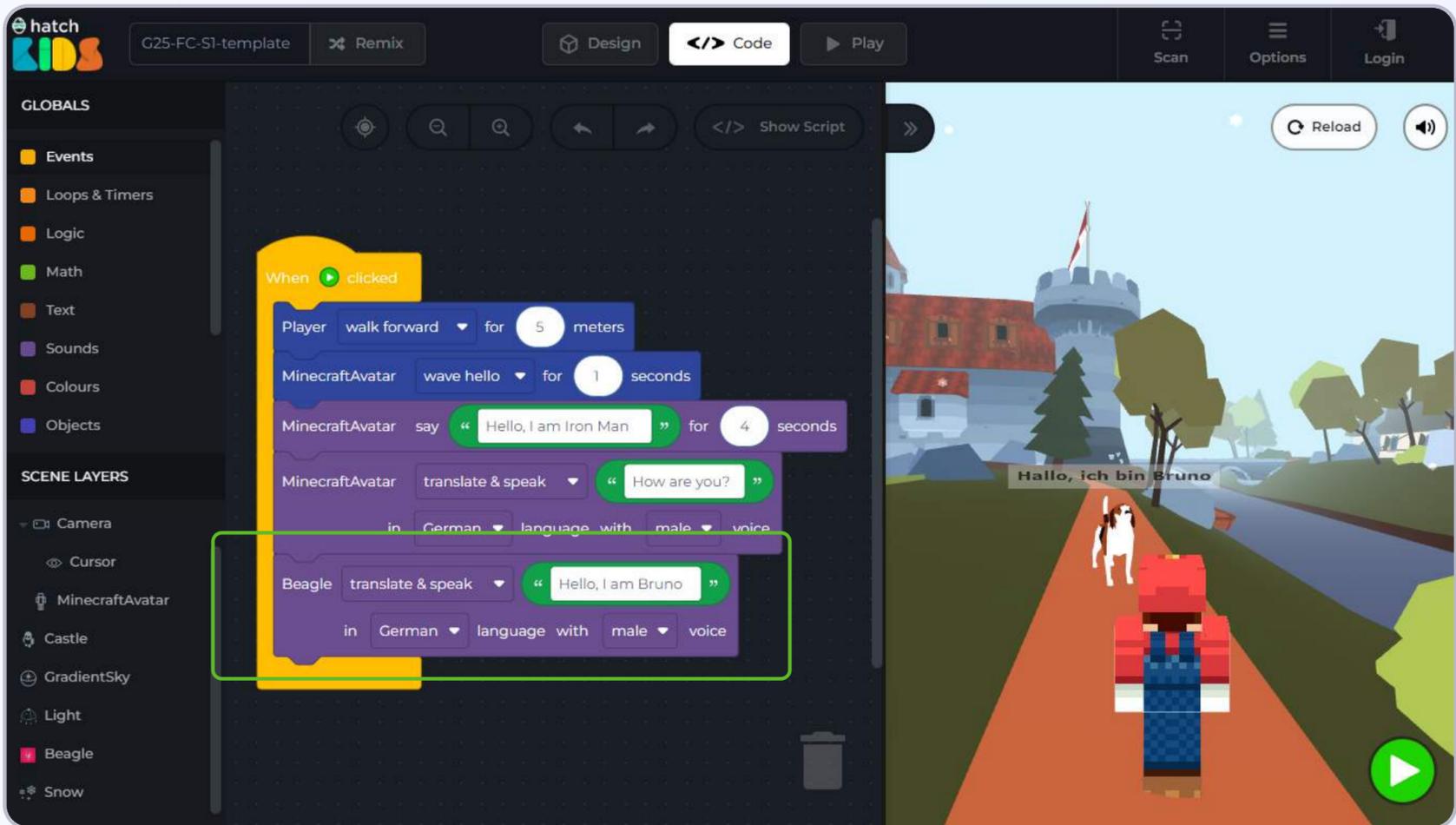
Step 7: Click on the name **"Beagle"** on the left panel, and in the list of blocks that appear, scroll down and you will find a similar **"translate & speak"** block.

Click and drag out the block and attach it below the minecraft's translate & speak block and change the text and language and gender to your choice.



Remember to click on the reload button before running the new changes that you have made in the code.

Additional Activity: Once you have attached translate and speak blocks for both the dog and your character and run the code, you can attach multiple such translate and speak blocks and make a full 3D conversation happen between the two objects

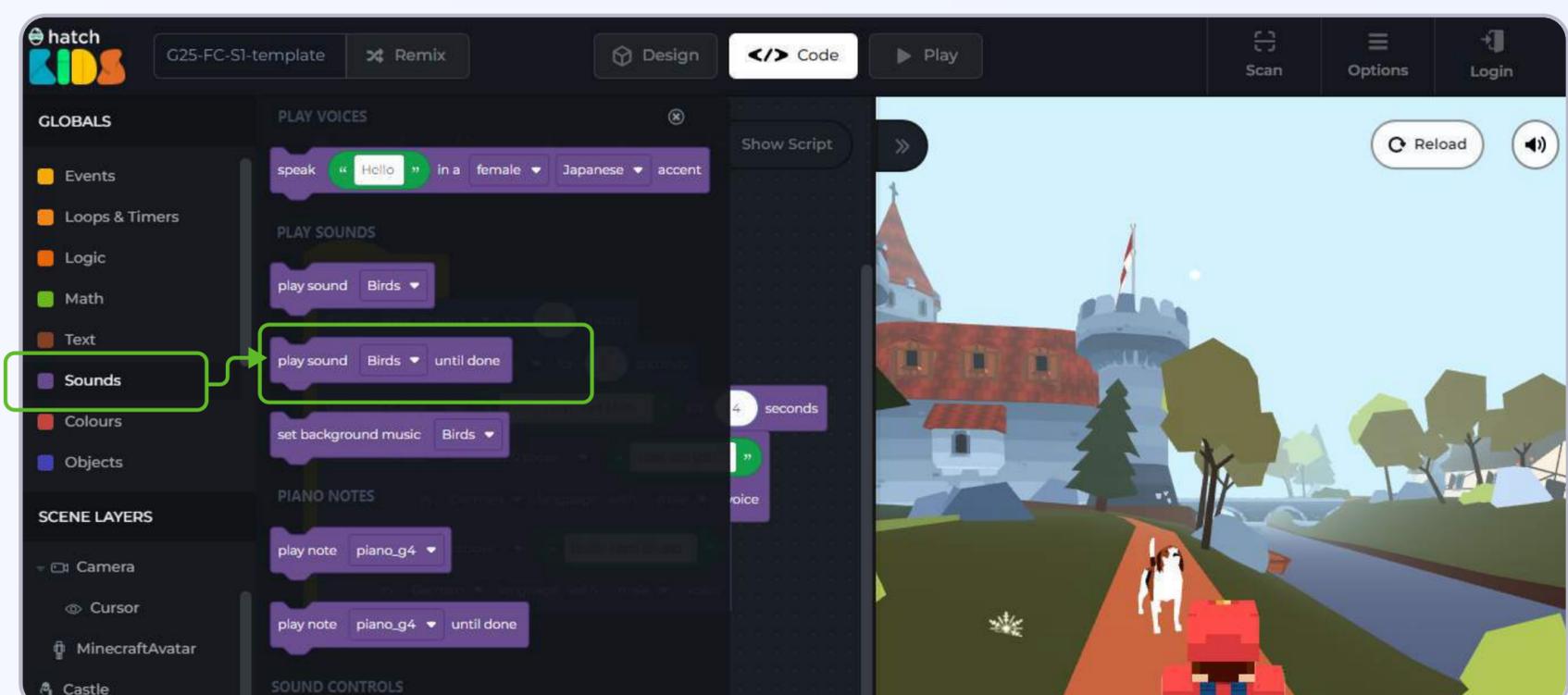


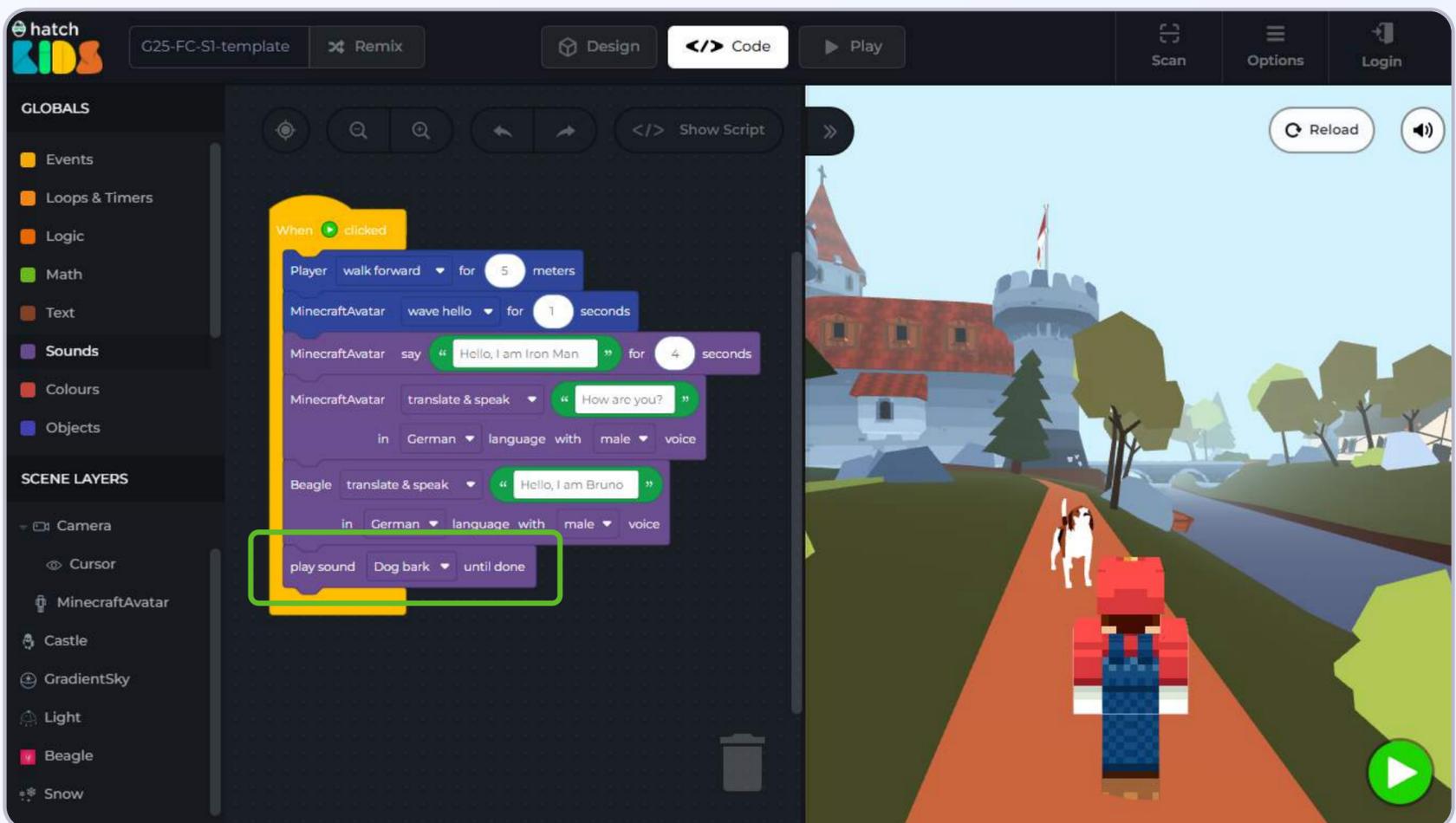
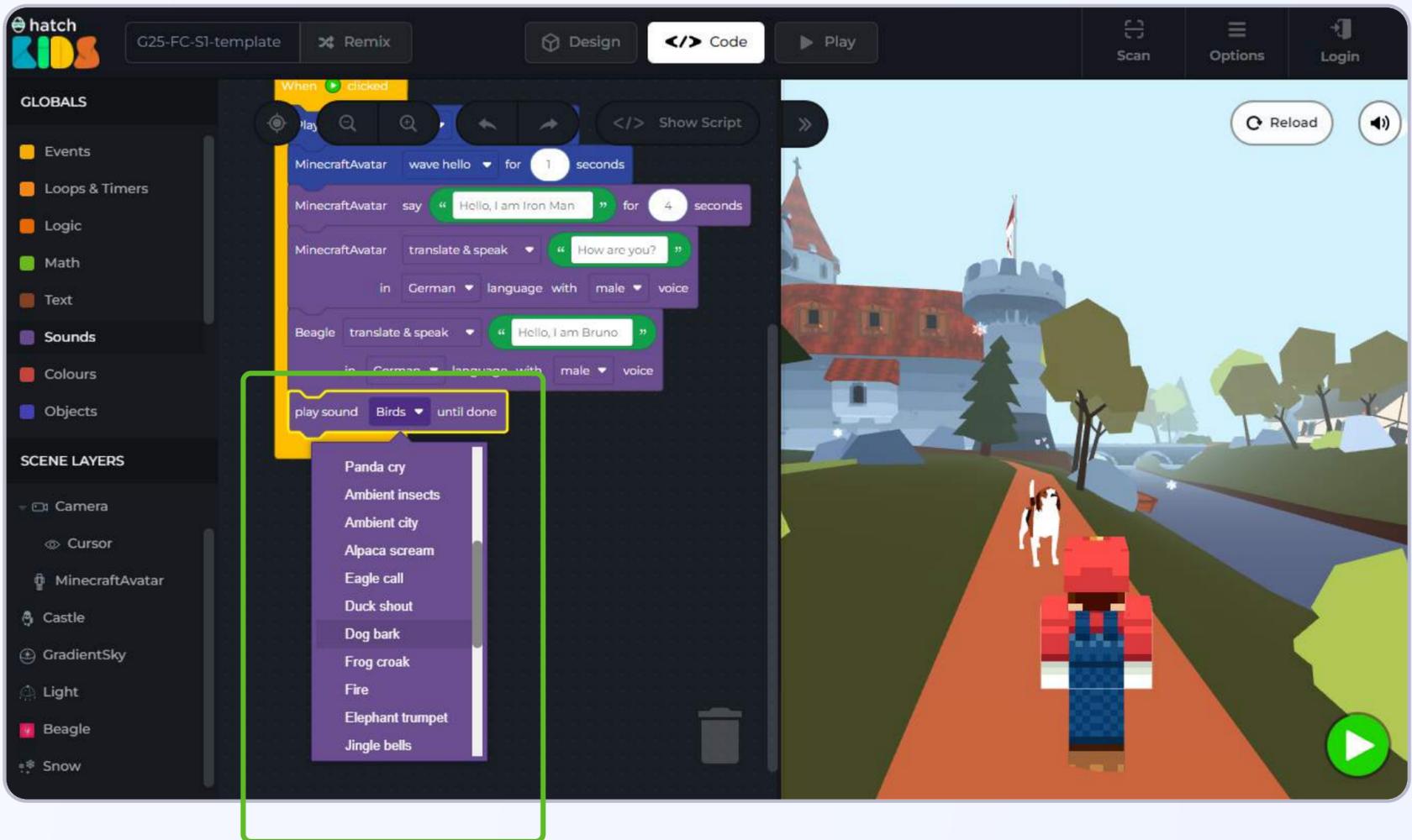
Step 8: Let's add some sounds in the game. In the top left panel, click on the **"Sounds"** option and you will see some sound and music blocks appear.

Step 9: Click and drag the block that says **"Play sound birds until done"** and attach it at the end of your code.

Step 10: Click on the **bird option** and from the drop down menu that appears select the **"dog bark"** option.

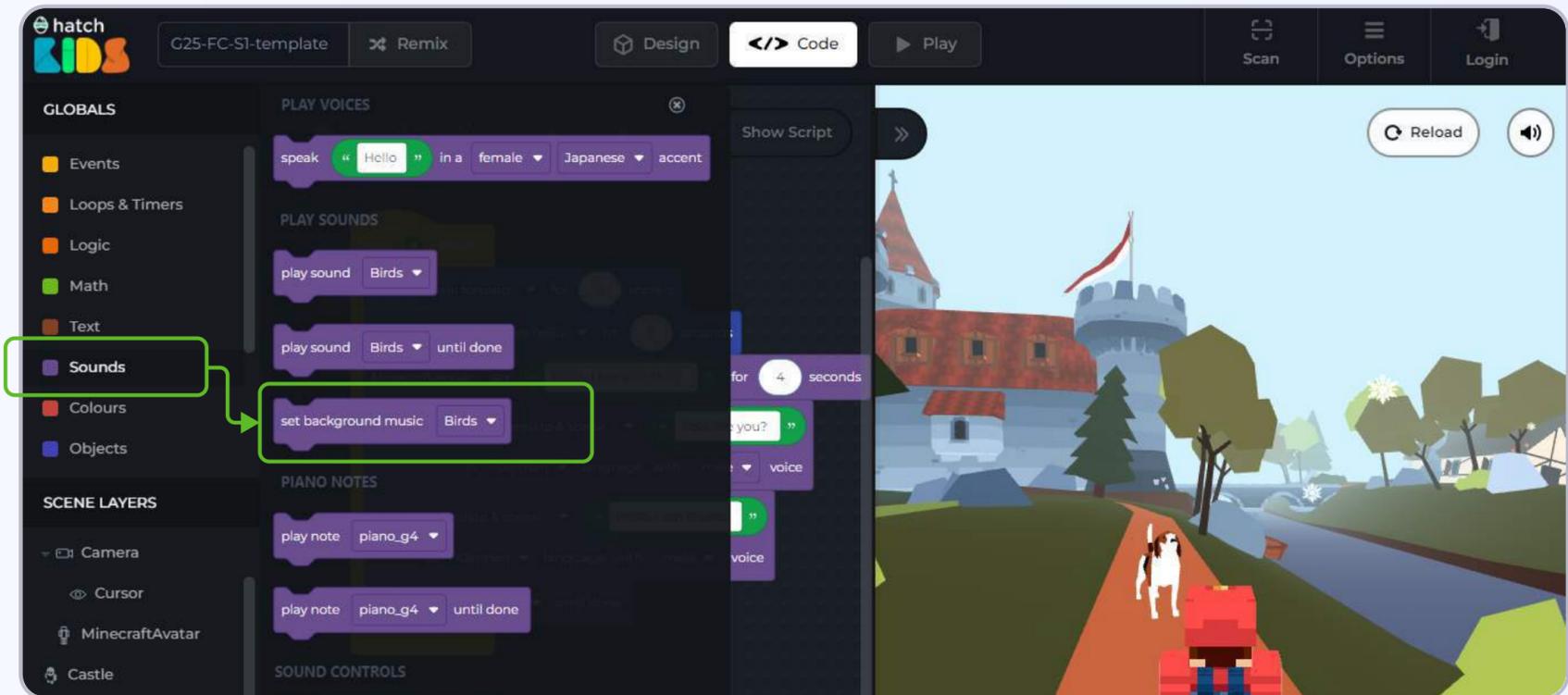
Now when you run the code, at the end, after the dog stops speaking, you will hear a dog barking sound,





Step 11: Let's add some background music to our game and then we can publish and share our game with everyone.

Click on the **Sound** option in the left panel and this time, click and drag out the block that says **"Set background music"**

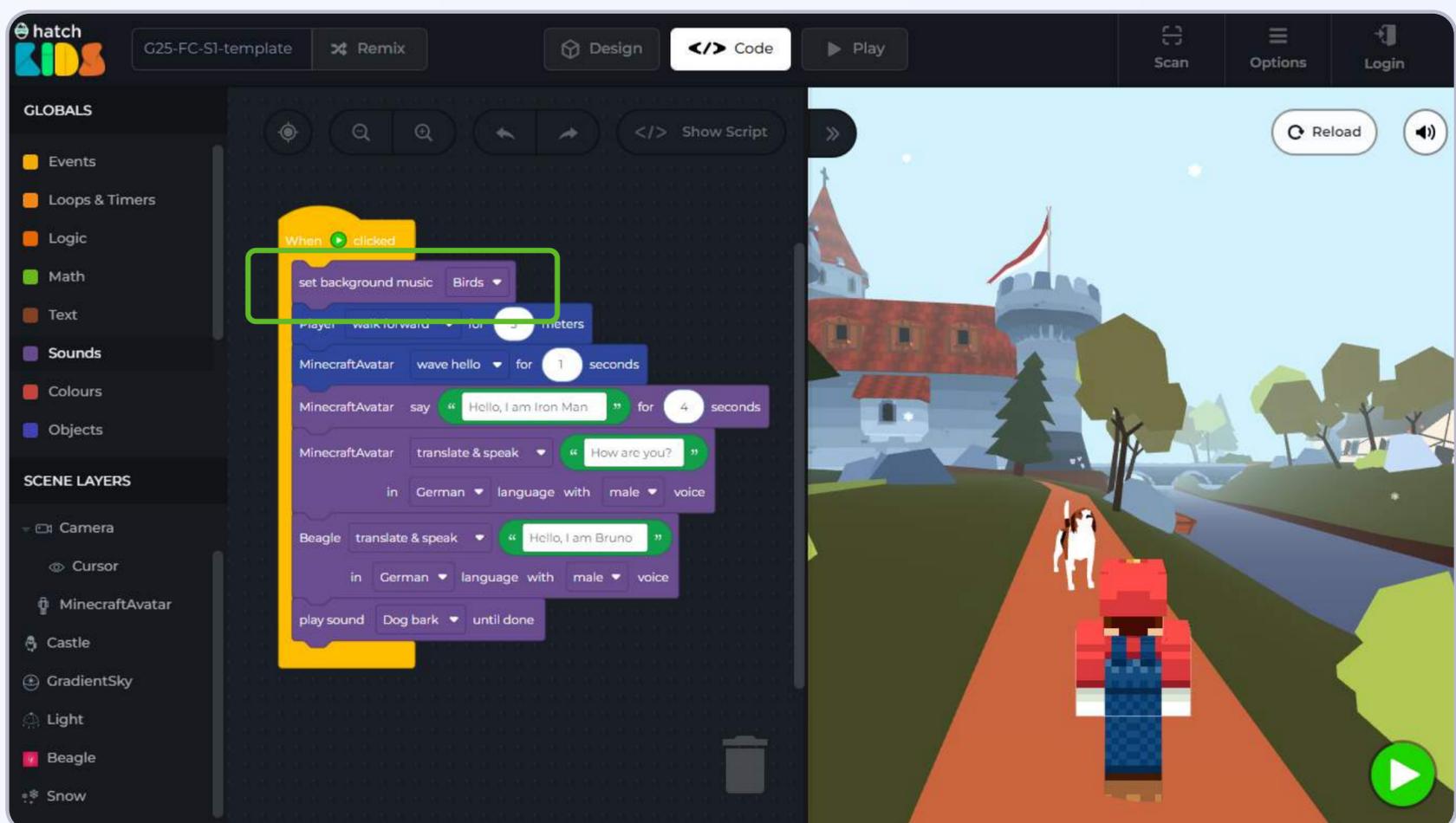


Step 12: Attach the “Set background music” block at the top, above the move forward block.

We learnt that the code is being run one block after the other in the order in which the blocks are connected. You would want the background music to start playing immediately as the game starts running, hence the need to attach the background music block at the top.

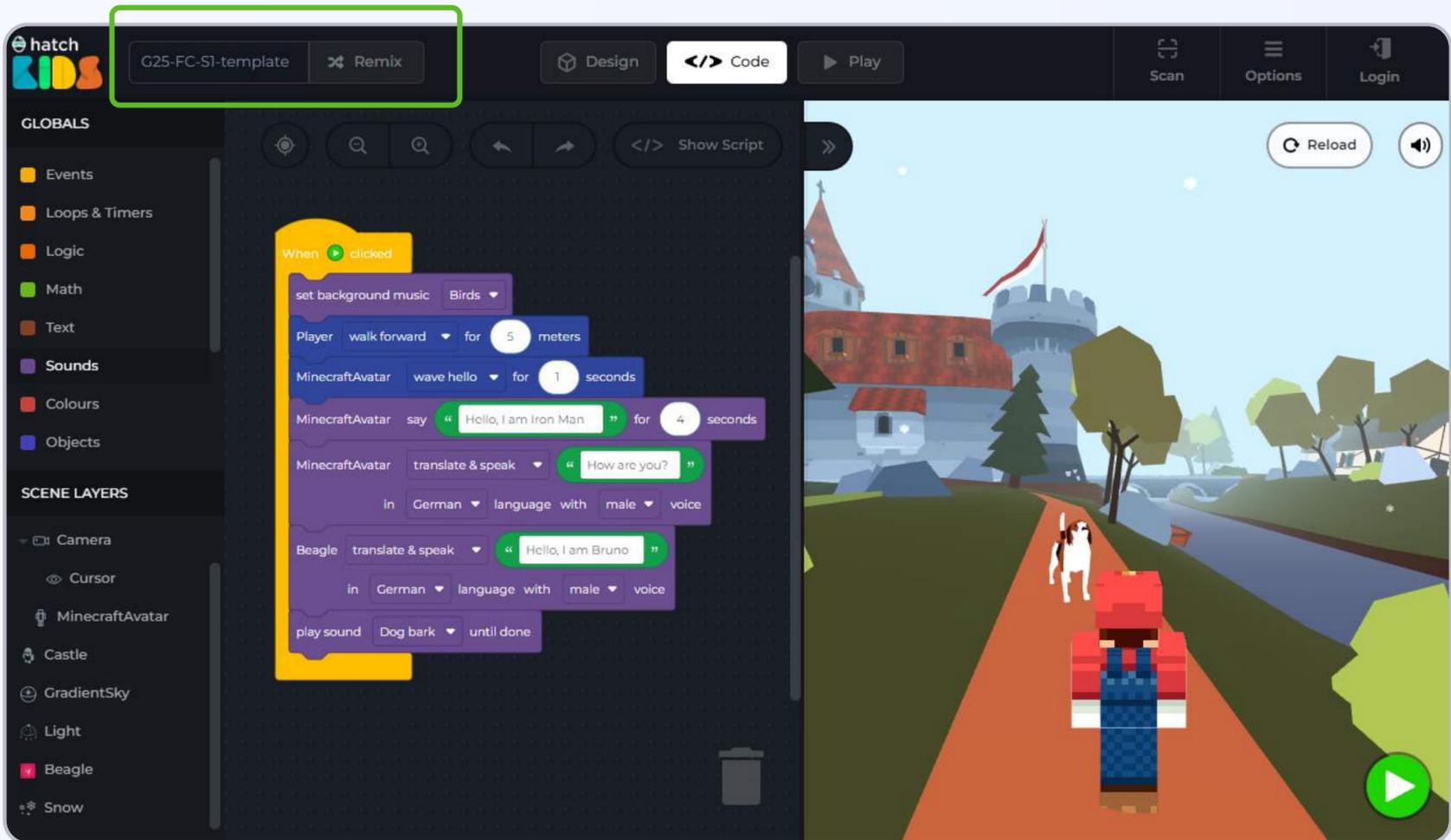
Now when you click on the green play button to run your game, you will have music playing in the background, with your character and the dog talking to each other in multiple languages .

And so your project it complete. 🎉🎉🎉🎉



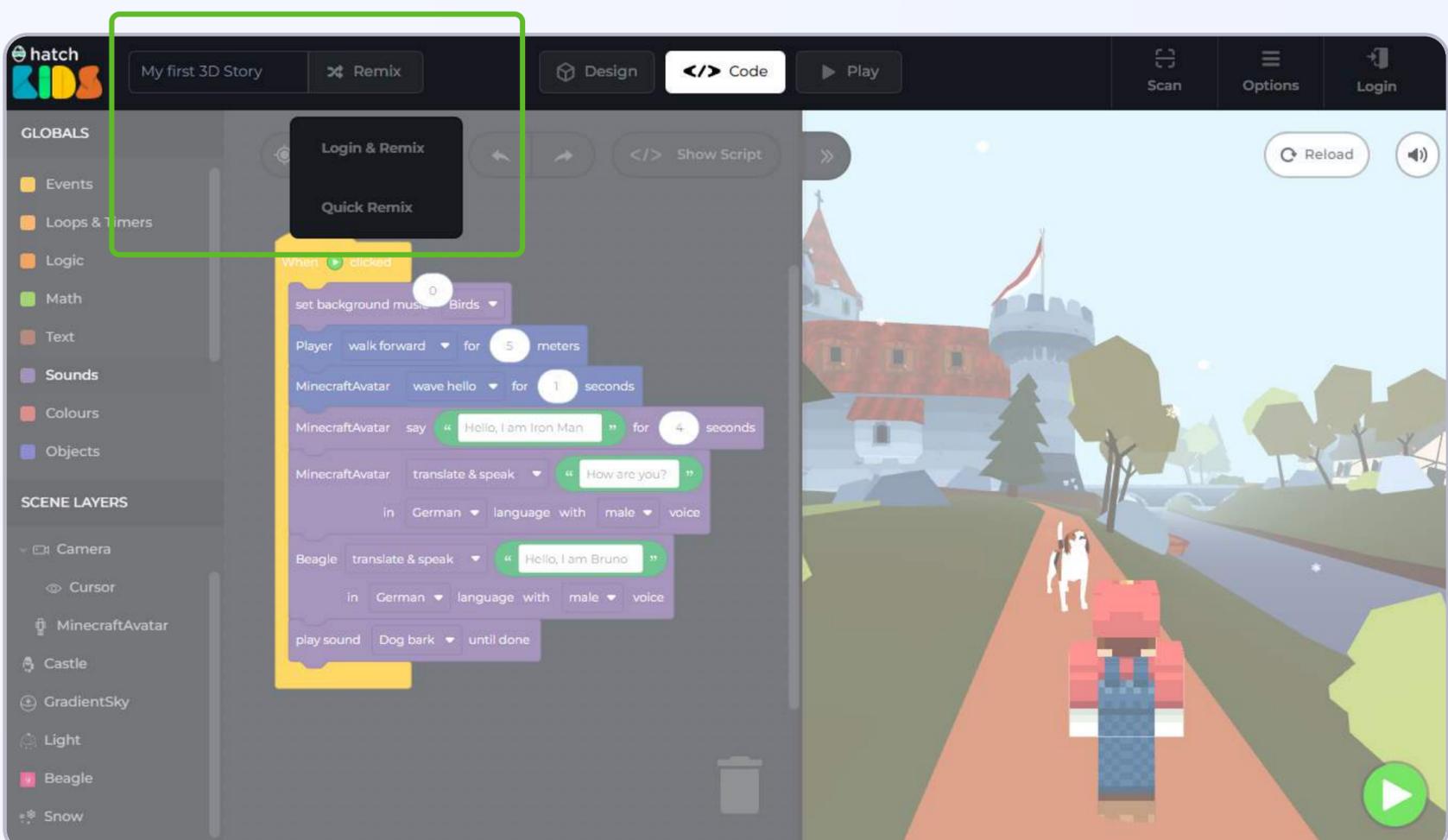
Objective No. 4: Publish and share your project

Step 1: Once your code is ready, you can now click at the top left corner of the screen, and give your project a name, and then click on the “Remix” button to publish it.

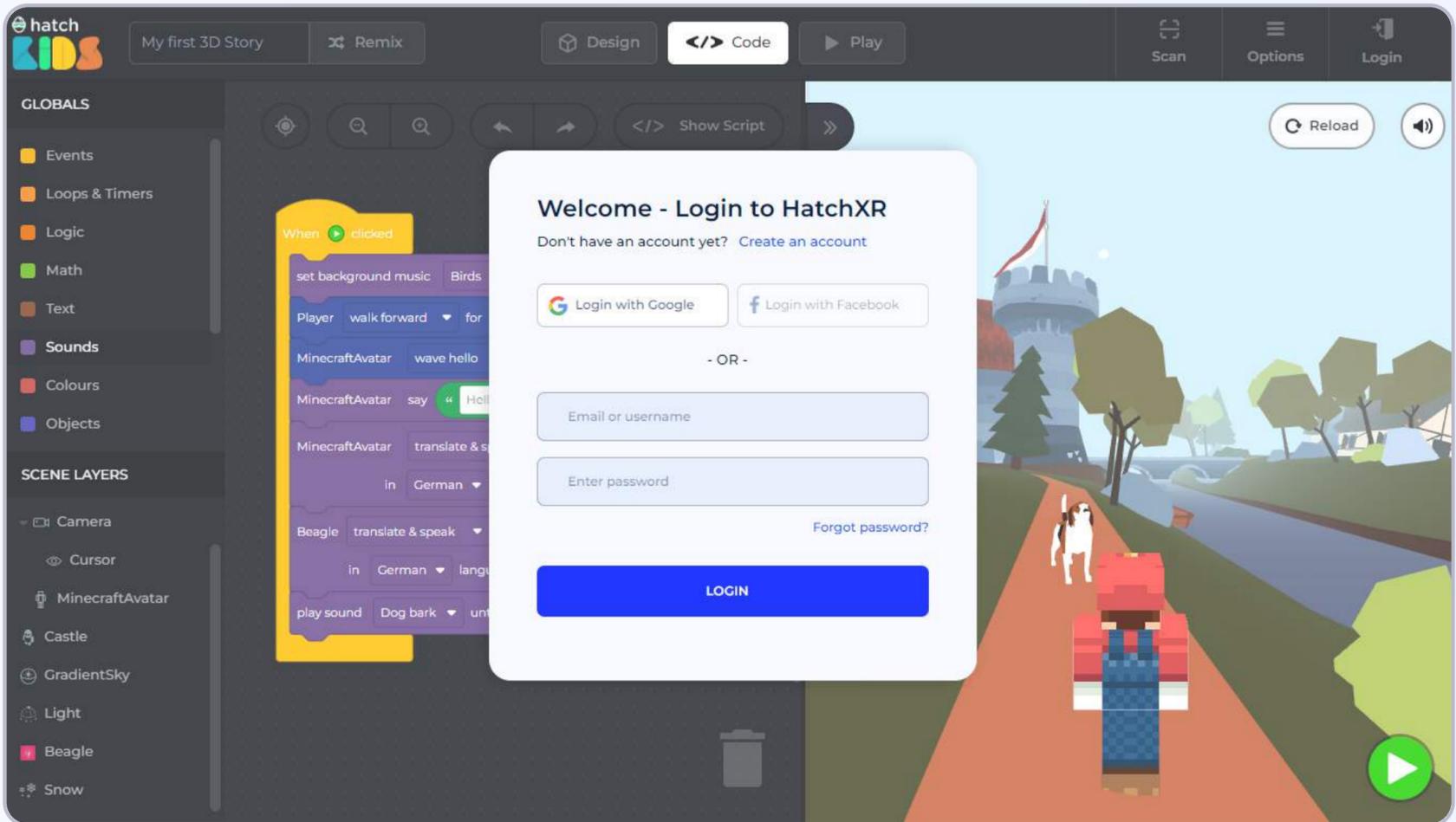


Step 2: If you have not used hatch before, then you will get two options.

- Login & Remix
- Quick Remix



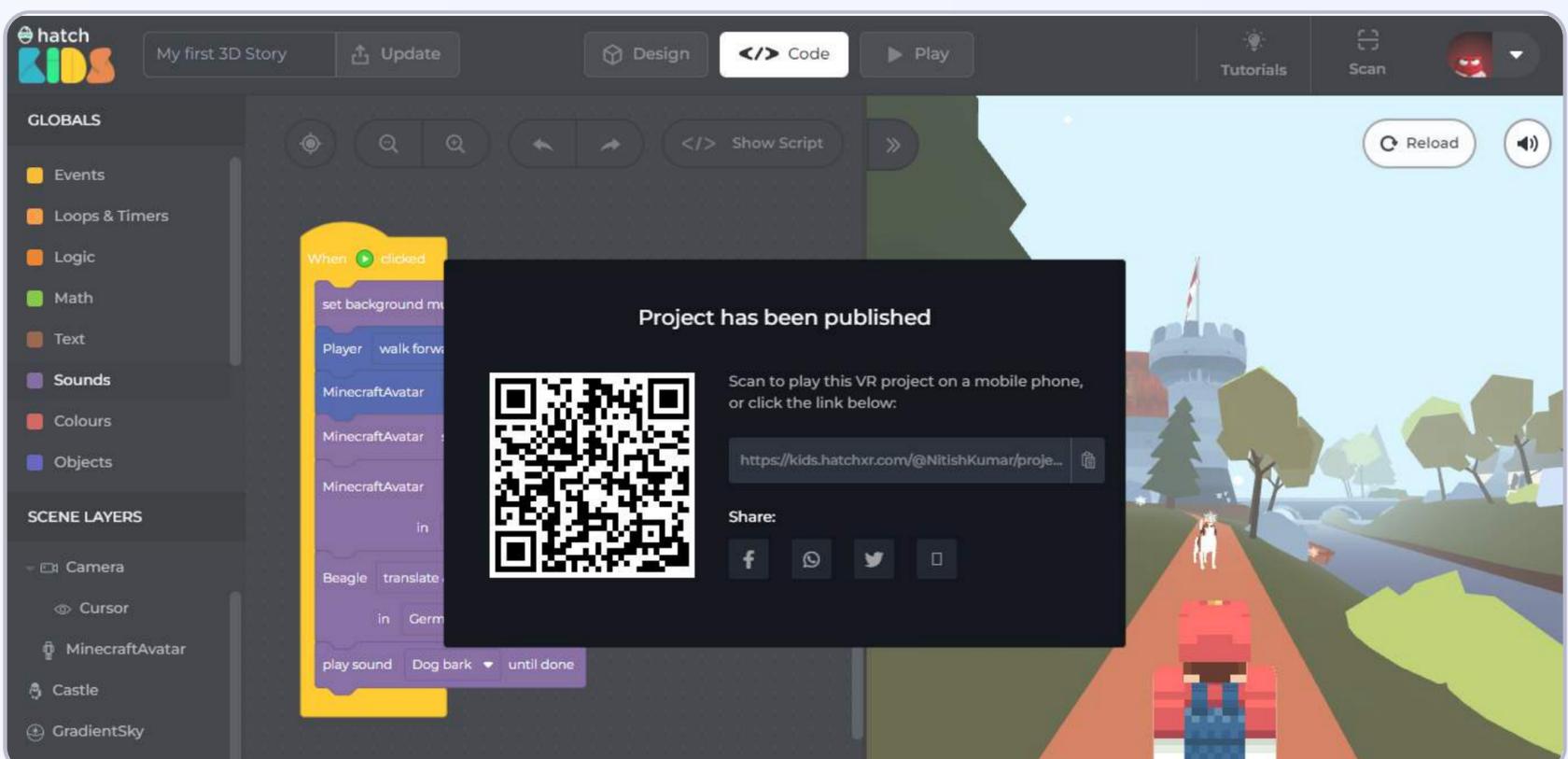
Step 3: Click on “Login & Remix”, and you will get the prompt to Login into your account



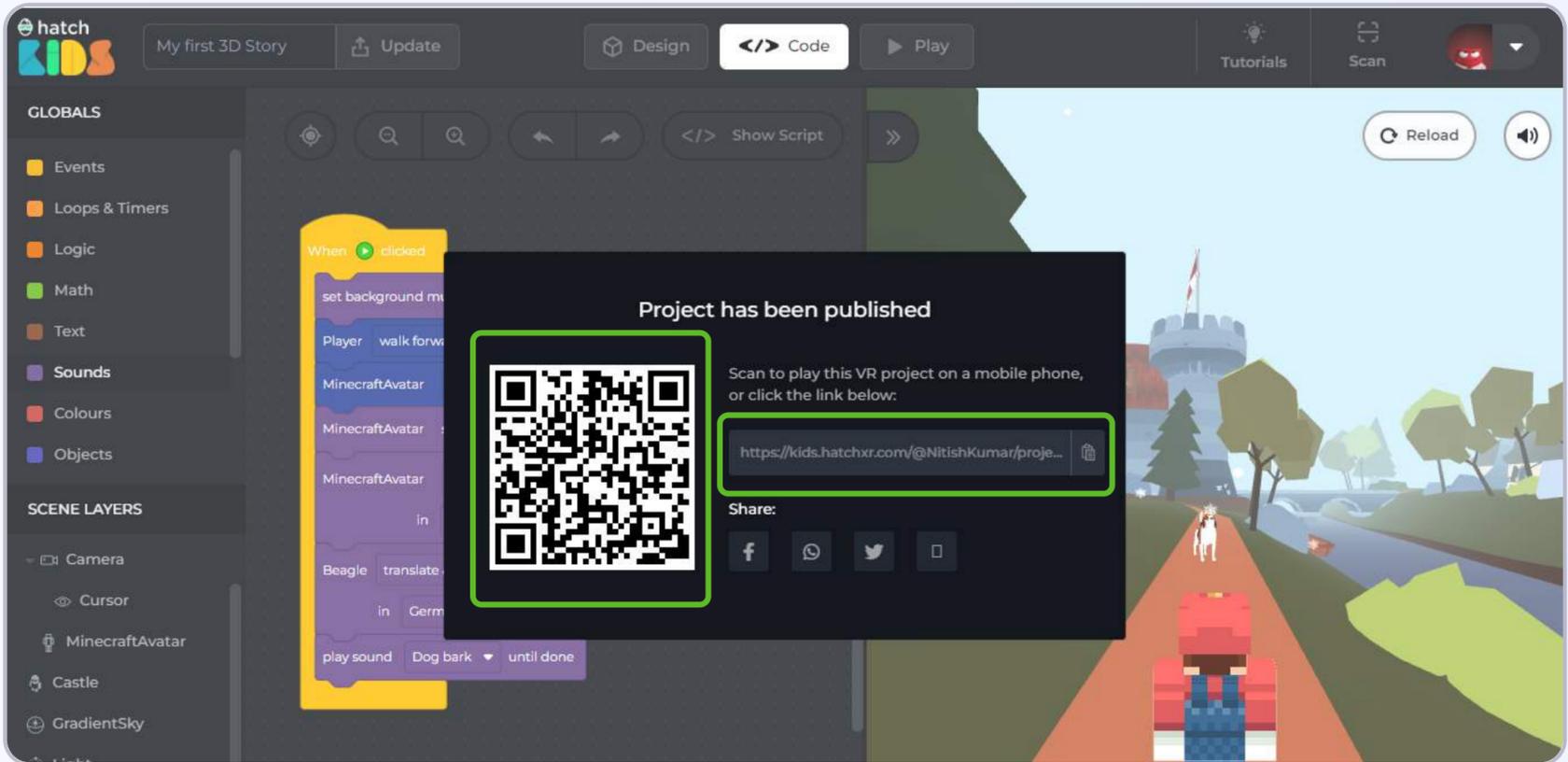
Step 4: If you have previously created a Hatch account then proceed to fill in your login details and click on the login button.

If you **don't have an hatch account**, click on “**Create an account**” and **fill in the form** or **use the Signup with Google** option to create an account on Hatch.

Once your account is created, your project will be published to your account and you will see a window appear on your screen as shown.



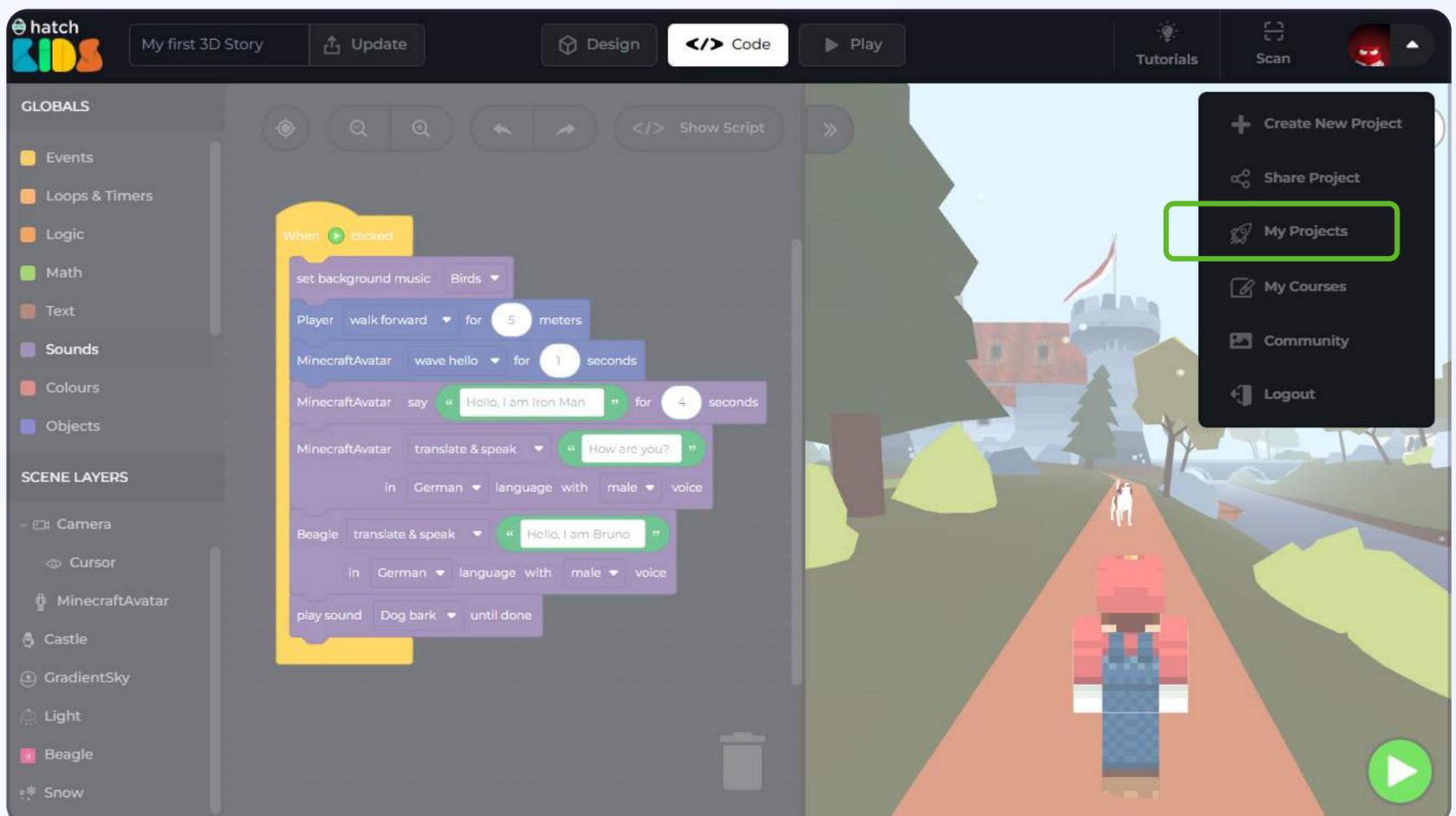
You can scan the QR code to run the project tht you just created on any mobile device. Or there is a link of your project in that window as well, you can copy and share that link with anyone you want.



You can click outside to close the QR code window.

At the top right corner of your screen, you will now be seeing your account details (profile picture).

Click on the picture, and you will see the following menu.



You can click on the **My Projects** option, and it will open up a new window where you can see all the projects created by you.

Any project that you create in future will also be visible in the **My Projects** page.

 **END OF GUIDE** 