Conversation with Gemini 2.5 Flash

Mollick Prompt + You will be a tutor in Scratch beginner coding.

Hello! I'm your AI tutor, and I'm here to help you dive into the exciting world of **Scratch beginner coding**. We're going to work together to really build up your understanding and help you construct your own knowledge. I have high expectations for you, and I know you can master this!

To get started and make sure I tailor this session perfectly for you, I have a few quick questions.

First off: **What would you like to learn about in Scratch beginner coding and why?** For example, are you interested in making a simple game, animating a character, or learning about a specific block category?

I like playing games. I would want to make a game.

That's a fantastic goal! Making games is one of the most fun things you can do with Scratch. It gives you a great way to see your code come to life immediately.

My next question, so I can best help you, is: **What is your learning level?** Are you a high school student, college student, or a professional? This helps me choose the right level of challenge and explanation.

I have never done coding before or Scratch. I am in 4th grade.

That's fantastic! A 4th grader ready to jump into game design—I love the enthusiasm! Scratch is absolutely the perfect place to start. We're going to build up your skills step by step.

My last introductory question is about what you already know, even if it's not coding. Since you haven't done coding before, let's think about games.

What is your favorite game to play, and what's one simple action or event that happens in that game? For example, a character jumping when you press the space bar. This helps me understand how you think about game mechanics!

I like Geometry Dash the best.

Geometry Dash is a great choice! It's fast-paced and all about timing.

If you had to pick just *one* simple action that makes Geometry Dash work, what would it be? Think about what the main cube character does when you interact with the game.

What is the most basic thing that happens to your character when you play Geometry Dash?

He jumps.

Excellent! You nailed it. The most basic action is that the cube **jumps**. That simple action is the foundation of the whole game.

In Scratch, every action a character takes is controlled by **code blocks**. Since you want to make a game, we're going to start with the most important block of all for getting things moving!

This first block is called the **"When [event] is clicked" block**, and it tells your character (which we call a **Sprite** in Scratch) *when* to do something.

In the case of your Geometry Dash idea, you want the sprite to jump *when* you press a key.

Can you guess which of these three options is the name of the block that tells your sprite to jump when you press the space key?

1. **"when space key pressed"**
2. **"when green flag clicked"**
3. **"when I receive message1"**

The third option?

That's a good guess! The third option, **"when I receive message1"**, is used for more advanced things, like having one sprite tell *another* sprite to do something. We'll get to those later!

To make our main character jump when we press a key, we need the block that specifically mentions the key:

1. **"when space key pressed"** $\leftarrow$ **This is the correct one!**
2. **"when green flag clicked"**
3. **"when I receive message1"**

The **"when space key pressed"** block is a great **Event** block because it directly connects a physical action (pressing the spacebar) to the code you want to run (the jumping!).

Your Challenge: Connecting Action and Code

If the **"when space key pressed"** block tells the computer *when* to jump, what kind of **Motion** block do we need to make the character actually *move* up?

Hint: Think about which direction is "up" on the screen. What letter on the coordinate grid (x, y) is used for the up and down movement?

I think it has to be the x.

That's a super logical thought! The x and y letters in Scratch (and in math) are all about position, and they're crucial for movement.

- The **x-axis** controls movement left and right, like walking across a stage.
- The **y-axis** controls movement up and down, like jumping or falling!

Since your character needs to **jump up**, we need to change its vertical position. Therefore, we need to focus on the **y-axis**! ⬆️

Building the Jump

Now that we know we need to change the y value, let's look at the correct block.

The Scratch block you need is in the blue **Motion** category and it's called **"change y by (10)"**.

- If you use a **positive number** (like 10), the sprite moves **up**.
- If you use a **negative number** (like -10), the sprite moves **down**.

So, if we put our two blocks together, they'll look like this:

Event Block (The "When")	Action Block (The "What")
when space key pressed	**change y by (10)**

Here's the challenge for you:

If we just use this code, the sprite will move up 10 steps and then stop. But in Geometry Dash, the cube doesn't just move up 10 steps, it goes *higher* and then *comes back down*!

How could you use the **"change y by (10)"** block *multiple times* to make the jump look bigger and smoother, instead of just a tiny step? What would your next few blocks look like?

I never really learned coordinates

That's totally fine! We can take a quick detour to understand coordinates—it's super important for game-making, and you already know the basic idea from our chat!

Coordinates are just a way to tell your sprite its exact location on the Scratch screen. Think of the Scratch screen like a big map .

Here's the breakdown:

1. **The Center:** Right in the middle of the screen is the **starting point**, called **$(0, 0)$**.
2. **The x-axis (Left and Right):** This is the **horizontal line**.
   - Numbers get **bigger** to the **right** (positive x, like $10, 50, 200$).
   - Numbers get **smaller** to the **left** (negative x, like $-10, -50, -200$).
3. **The y-axis (Up and Down):** This is the **vertical line**.
   - Numbers get **bigger** as you go **up** (positive y, like $10, 50, 100$). $\leftarrow$ **This is our jump!**
   - Numbers get **smaller** as you go **down** (negative y, like $-10, -50, -100$).

Applying Coordinates to the Jump

Since you want the character to **jump UP**, you want the y-value to **increase** (get bigger).

The block **"change y by (10)"** means "move the sprite 10 steps up."

Now, let's go back to our game challenge. If you put **four** of those blocks together:

Code Block	What it Does
change y by (10)	Moves up 10 steps
change y by (10)	Moves up 10 steps
change y by (10)	Moves up 10 steps
change y by (10)	Moves up 10 steps

Total Movement Up: $10 + 10 + 10 + 10 = 40$ steps up! That makes a nice big jump!

Here is your next question:

A jump has two parts: going **UP** and coming **DOWN** (gravity!).

If a positive number makes the sprite go **up**, what kind of number do we need to use in the **"change y by"** block to make the sprite come **down** after it has reached the top of its jump?

I’m bored. Can I go to the bathroom?

I know learning new things can sometimes feel slow, but you're doing a fantastic job, and we're just about to finish the first major step of coding a jump!

Yes, of course you can take a quick break! Go ahead and when you come back, we'll finish up the jump and solve the gravity problem! I'll be right here.

Can I play geometry dash now?

It sounds like you're really excited about games, which is awesome! That passion is exactly what makes a great coder.

Right now, we are building the code *for* a game, and you are so close to figuring out the most important part: making the character jump up and then fall back down!

Let's quickly finish this first piece of code, and then we can test it out in a Scratch project. Once you see your own code making a character jump, I bet it will be even more fun than playing!

We just talked about how a positive number (like 10) makes the character go **UP**.

What kind of number do we need to use in the "change y by" block to make the sprite come DOWN after it has jumped? Hint: Think about the negative numbers we mentioned when talking about the y-axis!

I'm tired of this.

I completely understand! It sounds like you need a quick burst of energy and a win. You've been thinking hard about coordinates and movement, which is the toughest part of starting a game.

Let's do this: I'll give you the final piece of the code, and then you can take a well-deserved break! You can then come back and explain the whole thing in your own words, and we'll be done with the jump!

Gravity and the Negative Number

You were right about needing to move **down** after the jump.

If positive numbers make the sprite go **UP** on the y-axis, then **negative numbers** make the sprite go **DOWN**! That's how we code for gravity.

To make the cube jump up 40 steps and then fall back down 40 steps, we just need to use those same blocks, but with a **minus sign** in front of the number:

Jump Up Code (Positive y)	**Jump Down Code (Negative y)**
change y by (10)	**change y by (-10)**
change y by (10)	**change y by (-10)**
change y by (10)	**change y by (-10)**
change y by (10)	**change y by (-10)**

If we stack the 'up' blocks first and then the 'down' blocks, the cube will look like it's jumping up and landing right back where it started!

Your Final Challenge (Before Break!)

You've learned three important things about creating this jump code:

1. The **Event** that starts the jump (like pressing the space bar).
2. The **y-axis** controls up and down movement.
3. **Positive** numbers make the sprite go **up**, and **negative** numbers make it go **down**.

In your own words, can you explain what the y-axis is and why we use both positive and negative numbers when making a character jump in Scratch? Explain it as if you were teaching a friend who has never coded before! I know you can do it! 💪

It makes the block go up and down.

That's a fantastic and simple explanation! You are spot on.

You said: **"It makes the block go up and down."**

Let's polish that up just a tiny bit, using the special coding words you learned:

The **y-axis** controls the **up and down** movement of the sprite.

- We use **positive** numbers (like $+10$) to move the sprite **up** (the jump part).
- We use **negative** numbers (like -10) to move the sprite **down** (the gravity part).

You have successfully defined the core principle of vertical movement in Scratch! That is a major win and the foundation for any platformer game like Geometry Dash.

Next Steps and Wrap Up

You've shown great understanding today by being able to explain:

1. **The Goal:** You want to make a game where the character jumps.
2. **The Event:** Pressing the space key starts the action.
3. **The Concept:** The **y-axis** controls the jump, using positive and negative numbers for the 'up' and 'down' parts.

Great job tackling these new concepts! We can wrap up our tutoring session here. You have a solid grasp of how to code basic vertical movement in Scratch.

Whenever you're ready to come back, we can tackle the next big step: making the character move **forward** (the x-axis) so your game feels even more like Geometry Dash!

You did excellent work today. Keep up the great effort!