Guidelines

In space. Final. Lesson 6

The main concepts discussed at the lesson

- 3D
- Shooting
- Module

Electronic educational resources used at the lesson

- Internet-enabled computer
- preinstalled RobboScratch3, Zoom or Blue Jeans
- Own account in the community on https://scratch.mit.edu/

Features of the lesson content presentation

This course was developed by the ROBBO Methodology Department specifically for distance education.

A special feature of the course is the visual material presentation in the form of slides. Explanations for each slide are given below.

Greeting.

Hello, guys! Welcome back to our Robboclub.
This is our fifth lesson. Link to a recent game.
https://scratch.mit.edu/projects/399855796 **SpaceShooter.**

We can take a look at it. Show me yours.
Some of you have trained at home and have consolidated your scripting skills.
Each of you shall open your own game. Run it. Make sure everything is OK. Show your options for adventures of the Circle in the unpleasant society of Squares, if possible. Do you have any problems? Everything works OK? Probably only those who trained at home.

Repetition.

Show the way it works and what doesn't work on your screen. Discuss situation. For those who have everything working, send + to the chat.
Let's repeat the basic concepts learned in the last lesson.

Slide №2

What is the method of controlling a Blaster?

There are "**send**" messages in the keyboard scripts. The Blaster itself accepts "**messages**". After receiving the "**message** to the left", the Blaster turns to the left with the command "**turn** counterclockwise by **1 degree**". Limit the sideways turn. The **"conditional operator if**" will help us. **If** the rotation angle exceeds **45 degrees** in one direction or another, the **"turn 1 degree"** command in the opposite direction is enabled. Why **1 degree**? It's easier to aim when you turn slowly.

The "**straight**" direction is given to us by the green help button: it instantly puts the barrel in the up position.

Slide №3

How does the shot happen?

If the mouse is pressed, it means that the command to shoot has been received. First, it's time to show up. Second, the shot should be accompanied by a sound. **Select** a sound in the music library. **Set the volume**. After "**show up**", the plasma is removed from the Blaster by the "**go 22 steps repeat 10 times**" command. But in the repetition process, the command "**change size by -8% (minus! 8)**" also works. In other words, the more the projectile moved away from the Blaster, the smaller it became, each cycle decreasing by 8%. It creates the impression that the plasmoid is not just rising up, but moving away from the viewer (from the first person). So we got a kind of 3D effect.

The cyclic operator "repeat 10 times" has a conditional operator "if it concerns a Meteorite". It makes the plasmoid bounce off a meteorite, changes its trajectory to a random direction in the range from -33 to 33. But the length of the plasmoid path remains the same: 10x22 steps. Along the way, it also decreases. And in any case, when the cycle "**repeat 10 times**" stops, the script proceeds to execute the next command-the "**wait**" subprogram. Let's see what this subprogram is.

Let’s start new topic.

But first of all:

https://scratch.mit.edu/projects/387341536 Eye gymnastics (45 seconds).

Original:
https://scratch.mit.edu/projects/399855796 **SpaceShooter.**

Our modification:
https://scratch.mit.edu/projects/401390602 **In the asteroid belt**

Slide №4

We just need to create two sprites: an Asteroid and a Meteorite. Let's start from a small one. Nothing is clear yet - just subprograms. In addition, if a meteorite collides with a plasmoid, it should "**hide**". That is, it will simply evaporate, which corresponds to the nature of things. Let's start the **subprograms**.

Completion of task № 1 in Activity Books.

Slide №5

Let's create Another block start.

Set the size of the meteorite at the beginning of the path **to 33%**. Scratch is so designed that small objects can't be reduced to, say, 1%. **Turn in the direction of 180**. But so that it does not hit one point of the dome, we will set it a certain spread, the firing sector - **"turn to give a random from - 45 to 45 degrees"**. And before to "**show up**", it will "**wait for a random issue from 1 to 3 seconds**". This is necessary so that it does not walk with an asteroid. These are the initial conditions for a sprite. It will return to them under different circumstances.

Slide №6

Let's create Another block flight.

Who can explain why the meteor is flying continuously to the very edge? After all, there is no **cyclical repetition** of the command "**go 5 steps**". I hear the correct answers. Yes, because the **cyclic operator is always** located in the body of the main script, and a subprogram "**flight**" is already integrated into it. If the subprogram is **always repeated**, then each of its commands is also **always repeated,** i.e. over and over again. And our meteor flies without stopping to the very **edge**, which it makes a new requirement: **hide** and return to the initial conditions.

https://scratch.mit.edu/projects/387341536 **Eye gymnastics (45 seconds)**

Slide № 7

Creation of Another block in the dome.

Meteorite's life is dangerous and difficult. Without understanding, you can fall into the **dome**. That's what we'll call our **Other block**.

The subprogram determines the behavior of a meteorite after a collision with the dome. It is simple: a meteorite does "**hide**" - it splits into dust. And starts all over again. When we collide we hear our favorite **pop** sound.

Question to the chat. Are sounds heard in space? If not, then "**pop**" should be removed. Leave it. In space, as in any other airless area, sounds are not heard. However, we can hear "**pop**" because we are in the cockpit, there is air in it, and the meteorite hit just on its shell.

Question to the chat. Why do we have a command to "**hide**" and have no command to "**show up**"? Does it mean that the sprite will remain invisible after returning to the initial conditions? No! The "**show up**" command exists in the "**start**" subprogram.

Warm up. **Slide 14** Select the music.

Slide № 8

The Asteroid script represents a complete set of subroutines. Let's get started. We can immediately see that the Asteroid and the Meteor are from the same pack. They come from the same point, and the spread of "**directions**" is the same. We decided to set the "**rotation method - do not rotate**". The meteorite does not need this, because it is round. And the asteroid is lit up on the right side, just like the Blaster. It is necessary to "**Change the costume in photo1**": the asteroid has a lot of costumes.

Completion of task № 2 in Activity Books.

Slide № 9

Let's create **Another block flight**, which also doesn't differ from the meteorite block. To save space, we also inserted "**when I will get everything**" event here. It hides the sprite at the end of the game, so that it does not remain in the middle of the stage. It gets this "**send all**" message from the Blaster. We sent it in the last class.

Slide № 10

Let's create **Another block for the dome**. "**If it touches the dome**" - then starts "**to change the costume to...**", but in fact it falls to pieces, and then returns to **initial conditions**. The command "**wait for 0.01 seconds**" helps to fix the moment of collision. Without this pause, the dome does not always notice that it has been hit. Well, after hitting the dome, the asteroid not only falls apart, but also bounces – "**change y to 22**".

Warm up. **Slide 17** Select the music.

Slide № 11

One more **Other block** - twin. We called it "**came**". The conditional operator "**if it touches the plasma**" also forces you to "**change the costume to ...**" and return to the **initial conditions**. The **costumes** are different. There are pyrotechnic one.

Slide № 12

We haven't created training sprites for a long time. **Let's choose** a bug. We put it on the **space bar**, because, you know, you should not touch the **flag** during training, so as not to be in the thick of a meteor swarm.
The beetle will be ready, but for what? The Scratch has a reporter module in **operators**, in numeric data. There are even two of them. We don't need the lowest one with a single window - this one will show you a number without a minus sign if you enter a negative number. Our **module** compares two numbers, so it has two windows. This operator returns a number equal to the remainder of dividing the entire first number by the second. 27/6=4,5. But also [27/6=3 and 3 in the remainder]. It showed us the remainder of 3. Try different options.
What does this operation give us? Create a script. Click on the beetle many times and observe how every 6th click the beetle changes direction. That it will turn on the 6th click is obvious. After all, 6 is divisible by 6 without a remainder, i.e. the remainder of the division = 0. But 12 is also divided by 6 without remainder, as well as 18!! Should I continue?
How does this new skill relate to the projected game? Let's figure it out.

Completion of task № 3 in Activity Books.

Slide № 13

Specifically for the game, we invented a self-healing dome. This way, the game can become endless. Provided that the pilot does not allow the Blaster to hit either a meteorite, or even an asteroid. As we remember, this is the only **condition** for sending a Blaster (after your own death!) messages "**send everything**" followed by "**stop everything**".
But while the Blaster is intact, the dome is undergoing its metamorphosis. Look at them attentively.
The **cycle** of asteroid impacts on the dome consists of 6 **touches**, each of which changes the variable **bang** to **1**. The result of dividing a variable by 6 can also have 6 residuals from this division: from 0 to 5. Before any hits, when **bang** = 0, the remainder of the division is 0/6=0, so for the dome, this means changing the **costume to costume1**. The first hit prescribes "**change bang to 1**". The program checks, searches for the value of the remainder of the division 1 by 6. And what does it find? What do you think is the remainder of dividing 1 by 6? Divide 1 by 6. What you'll get? Not that we needed. But our **module** will give us exactly what we need. It returns the remainder of a division of 1 by 6 as 1. As if it mentally added 1 to 6. That's the way it is. This function is so convenient. If the dividend is less than divisor, then it is divisible by itself and put it in the remainder. Verify it. 5 **module** 6 and 11 **module** 6 gives the answer 5. Not to mention 17 **module** 6, 23... etc.

Completion of task № 4 in Activity Books.

After "**if bang module 6 = 3**", the restoration of the dome begins.
That's it. We made a 3D application in Scratch. However, the graphics are very simple.
Here's a link to a game like ours where the pilot gamer is inside the cockpit - something we talked about recently. This brings us one step closer to 3D.

https://scratch.mit.edu/projects/238290026 **Space**

Scratch games can be used not only on a laptop, but also on a smartphone. Upload the game to your account, send yourself a link, click on it and your smartphone will open your own game. There is some difficulty and inconvenience with a virtual smartphone keyboard. However, you can do it without the keyboard. If you can arrange the game so that sprites are controlled not by keys, but in the "**when the sprite is pressed**" or even "**if the mouse pointer touches**" mode, your finger will be serve as the mouse.
For example, try to play this simple game first on your computer, then on your smartphone.

https://scratch.mit.edu/projects/364193703 **Virus**

But our space game is fully adapted for your smartphone. We made it that way ourselves. Make sure.
Now you will test your own games. Let's figure out what didn't work out for anyone.
And this is the end of the lesson and **6 Scratch course lessons**. In this game, there is no point chase, no time limit. You can add all this by yourself, as well as additional effects if you come up with them. The course is over, but the game can be developed.
We've made something like this
https://scratch.mit.edu/projects/401390602 **In the asteroid belt**.
There is still time until the end of the class. Maybe even a lot of time. You can share your ideas for improving and developing the game.
Continue to study the Scratch. This will help you learn other programming languages.

Completion of task № 4 in Activity Books.

Slide №14 Select the music.
Warm up. Shall be used at the discretion of the teacher several times during the course of the lesson.

The lesson is over! Thank you!

Completion of tasks in Activity Books.

Task № 1

2) and 3)

Task № 2

360

Task № 3

2.

Task № 4

24