Guidelines

Dino. The final. 6 lesson

Basic concepts discussed in the lesson

- Other blocks
- Logical operator
- Graphic editor
- Further development of the game

Features of the presentation of the contents of the lesson topics

This course was developed by the methodological department ROBBO specially for distance learning.

A feature of the course is the presentation of material visually in the form of presentation slides. Explanations for each slide follow below.

Greeting.

Hello guys! I welcome you again to our RobboClub. This is our sixth lesson. And the last in this course.

Launch RobboScratch3 and open your Scratch account online. Show on your screens how it works and what doesn't work. Discussion. For whom everything works, send + to the chat.

Reiteration.

We repeat the basic concepts learned in the last lesson.

Slide №2

How to draw a translucent bubble?

We go to the **graphical editor**. Choose a **circle**. Let the **Fill** be transparent (slash red line - lack of color). **Contour** - assign a thickness of **3**. LMB is pressed, pull diagonally down and to the right, try to draw a circle, not an oval. Further, imagining the future ball as volumetric, we choose two colors for gradient color mixing. To do this, select the **gradient far right**, it changes shades from the middle to the edges, simulating a spherical body. Two mixed colors will choose bright blue and ... no, transparent - the lack of color. This will make the ball translucent like a soap bubble and justify its ability to fly. We do everything with thetool **Fill** (tilted bucket). Hover over the ball - this will be the brightest highlight. Let it be not in the center, but something like this, because the light falls from above.

Slide №3

What are Other Blocks for?

I would like to unite performers of the same type in one group.

The Command block **Other blocks are** located next to theblock **Variables**. We click. Already familiar proposal **Create**. Click,again **Create**. You have already guessed what to create means to name. We print the name. We simply say: **height**. OK. You made your first **Other block**! Each of the other blocks corresponds to the team of the same name, which appeared automatically with the block. We extract the group of commands responsible for the height from the script. It is enclosed in a **cyclic repeat operator**. We put on a new cap on it, and insert the same command in place of the removed group. We check. What is the next step? Correct - define a return group.

Slide №4

How to place clones in the places we need? Let's increase the Gobo company using the cloning method too. First we place it at a point with these coordinates. **Go to x: -155 at: 22**. She is selected taking into account his further movements. **We clone the**Gobo **5 times** with a delay of **1 second**. Each subsequent clone will change its coordinates, adding x to the starting point of placement 77, and to y 22. In this sense, the commands **change x to 77** and **change y to 22**. Click on the flag and get the location of the clones on the scene such as we see at the bottom of the slide. The gobos lined up as if they were about to fly to distant lands. Soon they will fly.

Recall our game. Here it is ready-made. In the finished, but not in the final. The speed can be adjusted if you do not like it.

https://scratch.mit.edu/projects/401793549 **Dino on the balls**

Remember? And now:

Moving on to new topics.

But first:

https://scratch.mit.edu/projects/387341536 Gymnastics for the eyes (45 seconds).

Slide No. 5

Dino: full script. We continue to create adventures for Dino.

Let's change the main Dino script.

Thewill hardly change **initial conditions**. Just add change the costume to 1. Starts Dino in the lower left corner. As experienced programmers, you already realized that it is most rational to applyhere **Other blocks**. What we will do now. Create all 6 routines. But not right away, but along the way. **We create another new block** and compose a script for it. As a result, we get such a turret.

Performing task number 1 in workbooks.

Slide number 6

blocks **toright** theand **to the left** we have done in the last lesson. We will also change them a little. So far, they determine the behavior of the sprite when you press the **right arrows** and **left**, but ... anywhere in the scene. That is, if Dino is floating on a ball, and we press the **right arrow**, he immediately appears on the ground, according to the command **set at -155**. But we need the arrows to control the sprite only on the ground itself, and not higher. Therefore,on the ability to control these two **arrows** we will impose a restriction, which wecall **will the earth**. The next slide.

Physical education. **Slide 13** Select the music.

Slide No. 7

Let us write this restriction with **another block of land**. **Create the** block **earth**. On this **earth,** Dino will walk under the control of the shooter, and soaring above the ground, the sprite should not obey these arrows. This is our basement floor.

The following happens there. **If the right arrow key is pressed?** then the sprite first checks to see if it is on the ground. If it has fallen below (**position at <-155**), then it will rise (**set at -155**). Does nothing sprite sprite check above the earth? Therefore, Dino will obey the arrow only if it is on the ground. In any case, when the ground subroutine appears in the script, the sprite will be on it.

In the team **go ... steps, the** number of steps can be changed for different computers in different ways. It is only necessary that these numbers coincide with the balls in Dino - as far as the ball rises in the cycle, the same amount of Dino. Otherwise, Dino will jump on the ball.

It is possible that here you will notice some roughness of the program. Again, this can

manifest itself in different ways on different resources. On my computer, Dino hovering on a ball doesn't really jump from the ball when pressing the right arrow key. But in the account this happens. Yes, you can strengthen the restriction, but we will not overload the script. One must be prepared for this and not be nervous. And in our case, remember that if you need to jump, do it with the key ... see the next slide.

Slide number 8

Climb and jump. The next **subroutine is the climb**. **Create**. So in order we will play blocks from the main body of the script. **Another lift block**. It is clear that for lifting you need to **turn in the direction 0**, **go 5 steps** - in accordance with the ball. And the very possibility of rising from Dino appears **if** he **touches the Ball**.

Dino is a strategist, he needs to use time rationally. And if he sees that further rise does not make sense, then **the spacebar** will help him **jump**. **Create** this block. The jump is carried out by changing the coordinate x to the right (**change x by 9**) and the coordinates y down (**change y by -6**). Dino jumps: flies diagonally for a while, and then falls steeply down. Why does it fall after a flight? To this, the followingobliges it **subroutine**.

Slide number 9

Fall. Create another block is falling, it looks very simple. Dino is not a pterodactyl, so if he stands above the ground without holding on to anything, he will fall. Earth for Dino is that (below) **<-155** along theaxis **y**. **If the position of y** for Dino is **> -155** and at the same time it **does not touch Ball**, **then** it will fall, i.e. decreasevalue **y** (**change y to -11**). So that Dino does not fall underground, we will put him the familiar limitation of thesubroutine **land**. Dino will land on it. We remember that the **ground** makes the sprite be at the level of **-155**, not lower and not higher.

Here we combine two **logical operators** and one **numerical**.

What are **logical operators**, consider in more detail. Dino has two circumstances: there is a ball and its position in relation to the earth. In which case will it fall? The formula seems complicated, but anyone can answer.

Question to the chat. Who will answer?

The correct answer: Dino will fall if it is **ABOVE the** earth **and** touch the ball **NOT** will. It is important to understand that it will fall when ***BOTH*** conditions are met. If it is on the ground, but DOES NOT touch the ball, it will not fall, because there is nowhere to fall. If it is above the ground and touches the ball, then it also will not fall, because the ball holds it. And **both** conditions are combined by the word **And**, in Scratch, **by the And operator**. It is called **logical multiplication**. It multiplies - that is, adds - it conditions. Conditions, as already found out now 2. If you reduce one, one of two things will also remain: either it does not concern, or above the earth. But we logically multiplied, added and we have 2 conditions. The word is **NOT** called logical negation. And when you meet a logicaloperator **OR**, know that this is a logical addition. If you use **OR**, then your idea will work if at least one of the conditions is met. For example: a car cannot drive if the redis lit. **OR** yellow

The operators are combined in this order (shown on the slide). You need to understand the assembly like this: if the sprite **does NOT apply to** another sprite **AND** (besides!) **ABOVE the** earth, **then** ... the result follows. We have this fall, but not below the **ground**.

Performing task number 2 in workbooks.

Physical education. **Slide 13** Select the music.

Slide number 10

Victory. Create a victory. This routine celebrates victory. The slide is also overloaded, but it clearly shows the connection with the **background**. If Dino **caught** Gobo 15 times, (**> 14**), then he dances and shows off. Everything is clear here. Although ... what kind of transfer won? You need to **pass the background.** Upon receiving this message, the **background** will smoothly reduce the sound of music. After all, our game ends with the end of the song.

Dance of Victory - a simple animation of changing all the costumes in themode **next costume**. We got the costumes from the creators of Scratch, but we also added 2 suits. At the end of the dance, Dino does **change the costume to 3**. A pose of joy. And after **talking Dino won 1 second to change the suit to 4** - a favorite pose of boasting - **to say Dino well done)) 2 seconds**.

Slide No. 11

6 = 8 + 9. This slide is for reference. We added two more to the 7 costumes. I wanted to show the pose is tired and sleep. If anyone has a desire to do this, you can try. But it is not so simple. Need a desire and love for drawing. Both costumes - 8 and 9 - are made from suit 6, which is why such a strange name for the slide.

Suit 6 first **duplicate** PrKM. Get costume **8** (since **7** already exists). Circle it with the arrow **to select**. It is necessary that all its elements are grouped into one. Then turn the suit **to the right**so that Dino "lies" from a standing position. And the hardest thing is tail surgery. Agree, tired to lie with his tail up, for a dinosaur goes beyond the bounds of decency. We click on the tailtool **to change the shape of the**. Many dots appear. Some can simply be deleted by double-clicking on a point. This will simplify the shape of the tail. Stretching the black contour, then the green color, give the tail a form of decency. Yes, contour and color are two different elements, this is another difficulty in changing the picture. So we got a **suit 8**. We do the same to get the **costume 9**. But you still have to use the option to **flip vertically**to lay Dino on his back. Of course, all the costumes are in the Materials folder like a sprite dinosaur.

Performing task number 3 in workbooks.

https://scratch.mit.edu/projects/387341536 Gymnastics for the eyes (45 seconds).

Slide No. 12

Lost ... See you at the Olympics! In the initial conditions **play the sound of dino dino to the end**. The song lasts 168 seconds. This is our kind of timer. Without music, acould be applied; **conditional statementif the timer is> 167 seconds, then ...** etc. But we marked the limits of the game with music, which is more pleasant and even easier. In addition, applying a **timer**, using another **conditional operator**, we would have to use and

repeat the operator cyclically always. So it's easier to measure time with music.

If before the end of the music Dino does not collect 15 gobos, then he will receive the message he lost and acknowledges his defeat. Dino knows how to lose. On the left is the background script, on the right is Dino. Game over. We are testing. We bring it. are discussing.

That's it.

We made our last Scratch game. I hope that you will make many more games on scratch.

Games on Scratch can be used not only on a laptop, but also on a smartphone. Download the made game to your account, drop the link to yourself, click on it and your game will open on your smartphone. There is some difficulty and inconvenience with the virtual keyboard of a smartphone. but you can do without a keyboard. If you can organize the game in such a way that sprites are controlled not by keys, but in the mode **when the sprite is pressed,** or even **if the mouse pointer touches** , your finger will be 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**

Now you are testing every game. We will understand what did not work out for anyone. And this concludes the lesson and the entirecourse **Scratchin 6 lessons**. The course is over, but the game can be developed.

There is still time until the end of the lesson. Perhaps even a lot of time. You can express your ideas about improving and developing the game.

Continue to study Scratch. This will help you learn other programming languages.

Performing tasks number 3 and number 4 in workbooks.

This is where our lesson ends. You did a good job today.

Slide 13

Physical Fitness. Involved at the discretion of the teacher several times during the lesson.

Of course, you know that you cannot work on a computer for a very long time because your eyes and back are tired. Tell me, did you like the lesson? *(children's answers)*

You helped a lot today, you were attentive and carefully performed tasks, Well done!

Give us a mark for our work, noting in RT a robot that has:

Smile or Grimace or None Nothing.

Lesson is over! Thank!

Fulfillment of tasks in workbooks.

Task number 1

Both.

Task number 2

1) **(**IF the amount is NOT more than 50**)**, then goodbye.

Goodbye if the condition is met in parentheses.

2) **[**IF **(**The sum is NOT more than 50**) OR** IF **(**The difference is 50 And besides, the Product is less than 50**)]**, then You will get everything.

You will get everything if at least one condition in parentheses is true.

Task number 3

3) Timer.

Task number 4

1 - 5 ...