

Guidelines

Dino on the balls. Lesson №5

Guidelines

- Graphic editor
- Vector mode
- Other blocks

Slide №2

The cat splashes into the water if it touches the wall of the corridor. **What cases can you describe?** When the mouse gets to the muffin first. **Which command discards the cat?**

Slide №3

How can you make the mouse fatter? Answer: It is necessary to work in a graphic editor.

Slide №4

What is the difference between the vector and raster modes of the image editor? Answer: It is impossible to edit the image that is drawn in the raster, except by destroying it.

Slide № 5

What is the main thing in animation?

Answer:

1. Change of costumes in order.
2. After changing each costume, you need a little delay

Start new topics.

But, first of all:

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



Slide № 6

Gobo. According to the scenario in our game, the goal of Dino is to catch more than 14 Gobos. Dino can run along the alley and sit on the flying balls. After reaching a certain height, the balls burst and Dino lands on the alley. Control the Dino using the right and left arrows.

You can figure out for yourself what purpose Dino catches Gobo. It is important to direct children's fantasies in a positive direction. For example, on the planet Gobo there will be a volcano eruption soon and it is important to collect as many Gobo as possible and take them to a safe place.

Start from the balls

Draw a ball. Enter the **image editor**. Choose a **circle**. Let the **Fill** be transparent (Note that in the selection of colors in the palette, there is an option to choose a transparent color—a rectangle with a red line diagonally. This option means the absence of color). Let 's set the **contour** thickness to **3**. The left mouse button is pressed, pull diagonally down-to the right, and try to display a circle, not an oval. Next, imagining the future ball as a three-dimensional one, we choose two colors for gradient color mixing. Select the **gradient on the far right**, it changes the colors from the middle to the edges, imitating a spherical body. We will choose two mixed colors: bright blue and transparent (no color). This will make the balloon translucent, like a soap bubble. This pattern will convince the player that the ball can fly. Use the Fill tool (the tipped bucket). Hover the cursor over the ball, this will be the brightest glare. Let it be in the upper right corner. Let's assume that the light falls from above and to the right.

Slide № 7

Ball movement. In the proposed game (<https://scratch.mit.edu/projects/397303106>), balls (or bubbles) are cloned, while **randomly** appearing at the bottom of the scene from edge to edge, rising, increasing, and changing color.

Let's look at how this effect is implemented. We will need two related scripts. Select the initial color of the balls. This is provided in the initial conditions by the **"remove graphic effects"** command. Apply **"hide"**. If this is not done, the original sprite (**itself**) will be visible and stationary. We choose the **size of 11%** (another size is possible).

The clone block determines the spread of balls across the width of the stage (from edge to edge). Clones will be created every second, 11 times. In the script, replace temporarily **"hide"** with **"show"** to make sure that the clones are created the way we wanted.

The next script is "When I start as a clone". The first command is **"show up"**, since it performed **"hide"** Then - **0 direction**, up. Colored balls perform **"change color effect to 1"**. Increase the ball **"change size by 0.2%"** as it rises (**go 2 steps**). Each clone is lifted to a **random** height by repeating **2 steps from 122 to 144 times**. Keep in mind that if we want to change the speed of the ball's rise, for example by 3 steps, then the number of steps will be reduced.

Complete Task 1 in Activity Book.



Slide № 8

Ball movement. Continuation. After completing the climb to a **random** height, the clone **hides**, but continues to climb until it **touches the edge**, then returns back down to a **random** place along the width of the stage, where it accepts the initial conditions and the decision to **show up**. This is followed by a conditional operator that defines what the ball will do in the specified case.

Check how the Ball sprite works.

Complete Task 2 in Activity Book.

Slide № 9

Ball movement. Others blocks. Did you do? Well done. Make improvements!

It is important to remember!

In addition to programs, there are subroutines. This is a separate, independent program. It is used (called) in another program.

Example. Adults teach the children the subroutines:

1. make the bed after sleep,
2. wash the face and brush the teeth,
3. behave carefully at the table during breakfast.

The main program: "get ready for school" will include all these subroutines and some others.

Question to the chat. What other subroutines will help your child get ready for school?

Possible answers:

- collect the school bag,
- put on the school uniform,
- remove the dishes from the table.

Question to the chat. What other programs can use these subroutines?

Possible answers: the subroutine "clean up the dishes from the table" can be used in the program "Feed the family with dinner".

The more we write code, the more often there is a situation when some code fragments are repeated, perform the same work. To save computer memory, design beauty, and readability, we will use subroutines. Let's look at how to do this.

The "**Other blocks**" command block is located next to the "**Variables**" block. Click. The familiar "**Create**" offer. Click "**Create**" again. To create means to name. Printing the name. Let's call the subroutine: height. Confirm-OK. Pull the command group responsible for the height from the script. These 4 commands are enclosed in the "**repeat loop operator**". Add the red header "**Height**", and this command Height is inserted in the place where we deleted the extracted group. The main script has become shorter and clearer in meaning. Check. What's the next step? That is right, define the return group.

Warm up. **Slide 21** Select the music.



Slide № 10

Ball movement. Other blocks 2. Taking out of the script a group of 5 commands in the **conditional operator** responsible for the random return of balls "**if touches the edge**". Creating **Other block**-the **return** subroutine. Return the **return** command to the script. Check

Slide № 11

According to the scenario in our game, the goal of Dino is to catch more than 14 Gobos.

Select the Gobo in the library. He has three costumes. Removing two costumes. Let's increase the number of Gobos by cloning. First, we will place it at a point with these coordinates. **Go to x: -155 y: 22**. It is selected taking into account its further movements. **Clone Gobo 5 times** with a **delay of 1 second**. Each next clone will change its coordinates, adding +77 to the x-axis coordinate of the initial placement point, and +22 to the Y-axis coordinate. This is the meaning of the commands **change x to 77 and change y to 22**. Click on the checkbox and get the location of the clones on the scene as seen at the bottom of the slide. The gobos lined up in a straight line

Complete Task 3 in Activity Book.

Slide № 12

Gobo. Other blocks. Clones accept initial conditions. They move in a circle. This is provided by simultaneously executing the commands "**to go 10 steps and turn 11 degrees**". **Waiting 0.1 seconds** slows down the movement, so it's better.

In a scoring game, you need variables. Creating the **caught variable**.

We will also create several subroutines. Details are shown on the next slide.

Slide № 13

Gobo. Other blocks 2. Clones accept initial conditions. They move in a circle. This is provided by simultaneously executing the commands "**to go 10 steps and turn 11 degrees**". **Waiting 0.1 seconds** slows down the movement, so it's better.

Let's make several copies of the ready-made scripts - the first command **to set caught the value 0**.

In a scoring game, you need variables. Creating the **caught variable**.

We will program the clones to execute the command "**hide and wait for 11 seconds**" before "**showing up**" every time they "**touch Dino**". Check. We see that Dino is still late. We focus on the sample that we played at the beginning of the lesson. Add it from the library, code its behavior in a subroutine, which means in **Other block**. Since the goal of Dino is to catch as many Gobos as possible, each clone is responsible for counting the captured Gobos. Commands from **Other blocks** are inserted in the script "**When I start as a clone**". Under the condition "**if it touches Dino**", the commands "**hide and turn on the pop sound**" are executed simultaneously. You can check by substituting Dino for some clone.



Slide № 14

Gobo. Palm tree. Let's start decorating the stage. Plant palm trees. Selecting a sprite from the library. The algorithm is already familiar. The initial conditions, then cloning, in this case, **change x and y**. We get three palm trees.

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

Slide № 15

Background. Landscape. Enter the background, in the image editor (backgrounds). Select vector mode. Selecting a Rectangle. Fill a is missing. Black outline, 2 units. Stretch it over the entire window, holding the cursor in the lower-right corner. Choose the blue color. Click the gradient "from top to bottom". The second color is white. Take a bucket. Pour on the sky. If it is dark at the top, change the gradient.

Slide № 16

Background. Landscape 2. Stretch the sky over the entire window, you can do a little more. Then use the **circle** tool to make a large oval. Click the **Shape Change** tool (upper-right arrow). We train to create a polyline by adding new points and moving them. Fine lines radiate from each point. By pulling and rotating the line you can achieve different results.

After creating colorless mountains using a gradient, let's give them the appearance of remote peaks. Then make an oval with **Circle** and give it the colors and shape of the future coast. The resulting image can be selected by clicking, then **copy-paste** (duplicate) and **reflect**. Or duplicate the coast a couple of times, then stretch, flip (reflect) or flatten. Choose a different pair of colors for each one. Make the coast on the first plan. We have a condition for the palm sprite to "**Move to the front layer**", so it will be on top of all of them. If it fails, click on the sprite flag. Admiring the resulting landscape. We are proud of the acquired ability to use the graphic editor in vector mode.

Warm up. **Slide 21** Select music.

Slide № 17

Background. Palm grove. We already have the palm sprite. Enter it. There should be a shadow under the palm tree. Draw a shadow. Tool **circle**, make an oval, **fill** with greenish, dark. Use the "**select**" arrow to circle the palm tree and shadow together. Make sure that the blue frame captures both the palm tree and its shadow. Click "**copy**". Go to "**backgrounds**". Click "**insert**". The background will be a copy of a palm tree with a shadow. Customize the size. Again, click "**insert**" several times. We distribute the trees according to your choice, changing the size, slopes, and thickness. The palm grove is ready. It will be a row of trees on the edge of the Bank. And three palm trees from the sprite palm will appear in the foreground. According to the game scenario, Dino runs along the alley. We see it between the palm trees that are far and near to the viewer.



Complete Task 4 in Activity Book

Slide № 18

Background. Scripts. There is always a lot of work concerning the background in the image editor. Sometimes it takes longer than writing a program. When Dino wins, you can just turn off the music (**stop everything**), but it is better if it will gradually decrease. If you lose, the music stops smoothly, this is done in the recording itself. The value "**to set the volume ... %**" can be selected. The song about Dino will play until the end. This will be the limit of our game in time. **If** the song is over and the Gobo is **caught less than 15**, then the **background** should pass **lost**. Otherwise, the background will get the message won and start to smoothly change the **volume to-0.25**. The change is made with a minus sign, which means that the volume is reduced. The number 0.25 is selected. You can change it, play with the sound.

Slide № 19

Dino. Add Dino from the Materials. We acquainted with him in the last lesson. We will choose it from the Jurassic Scratch period. As you remember, it has **7 costumes**. However, for most of the game, it will have just one. It will only slide. At the end of the game, it will show us all the costumes. Let's set the **initial conditions** for it. Dino starts in the lower-left corner. Dino can run along the alley and sit on the flying balls. Dino controls - using the arrows to the right and left. Let's write **right** and **left** subroutines.

Slide № 20

As you can see, the blocks are the same, only they direct Dino in opposite directions. This concludes our fifth lesson. In the next lesson, we will finish this game and show each other what happened. Everyone will hold a small presentation of their game. Let's see who gets Dino to dance and who gets some sleep.

Warm up. **Slide 21** Select the music

Tasks №3 and №4 in AB.

This is the end of our lesson. You did a good job today.

Slide 21

Warm up. Used at the discretion of the teacher several times during the lesson. For physical training, you can use your own cartoon. Here is the link <https://scratch.mit.edu/projects/397666421> What do you need to make the same one



yourself? Penchant for drawing. Understanding **vector mode**. Patience. Desire. I don't know if you recognized the season's hit of Little Big? If you put this music on, of course you will find out. So the scratchers can start their own Music Show. You can arrange a flash mob - whose performer is better.

- 1) Choose your own voice-over option;
- 2) Scratch music only from our music **library**;
- 3) First, Scratch the music, and then the **popular hit**;
- 4) Only a **popular hit**;
- 5) Simultaneous playing of Scratch music from the music library and a **popular hit**.

Feel free to experiment! To change the options, you need to go **inside the project**. Try to revive something yourself.

You know, of course, that you can't work on a computer for very long, because your eyes and back get tired. Tell me, did you like the lesson? (children's responses)

You were very helpful today, were attentive and carefully completed the tasks, well done!

Give us a rating for our work, noting in Activity Books the robot that has:

A smile or grimace or a neutral expression.

Class is over! Thanks!

Complete tasks in Activity Books.

Task № 1

Equally.

Task № 2

3) Returned unnoticed. This creates the illusion of an infinite number of sprites.

Task № 3

Left and down.

Task № 4

The second from the top.

