



thinkable

Thinkable

Знакомство с
математическими операциями
Урок 2

Создадим новый проект



Create New Project ×

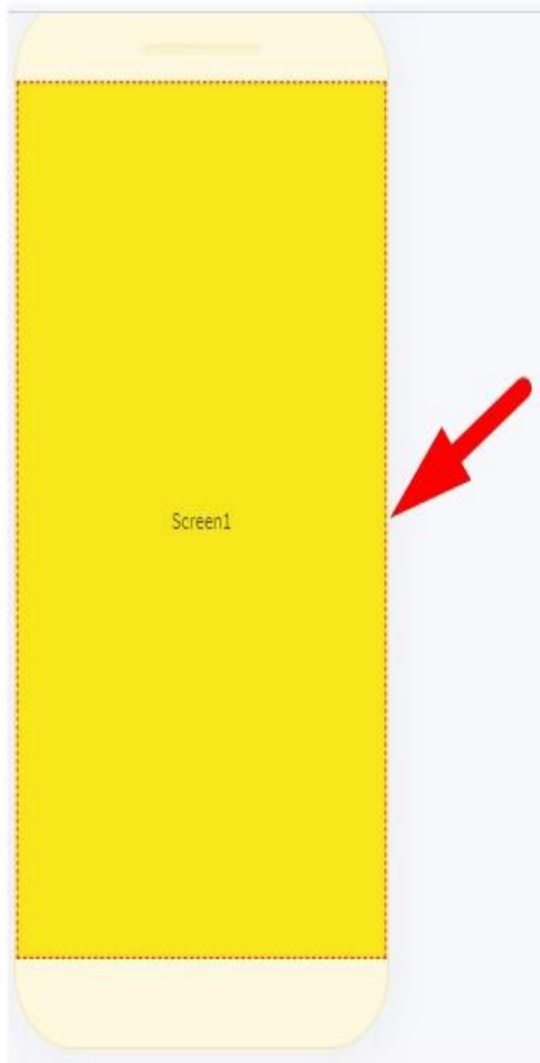
New Project Name:

Private Public

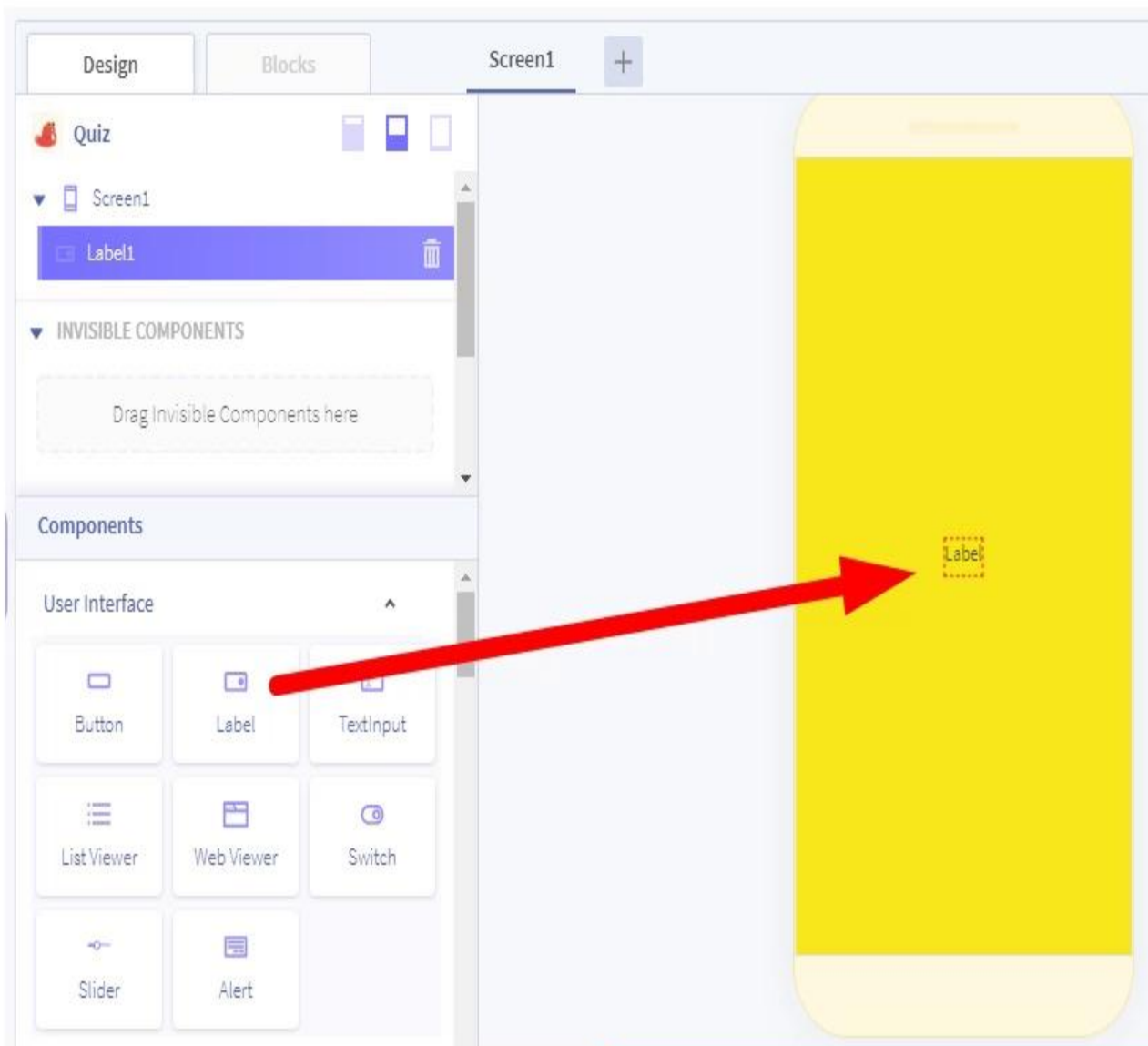
Private projects are only viewable and accessible to their creators.

Cancel Create

Изменим цвет фона



Добавим надпись



Настроим надпись



Label1

< Simple Adv

Text

Игра "Угадай логотип" X

FontSize

24

A screenshot of a design tool's label editor. It shows a 'Label1' component with two tabs: 'Simple' (selected) and 'Adv'. Below the tabs is a 'Text' field containing the text "Игра 'Угадай логотип'" and a close button 'X'. Below that is a 'FontSize' field with the value '24'. Three red arrows point to the 'Adv' tab, the text input field, and the 'FontSize' input field.

FontStyle

normal

Height

Absolute Size

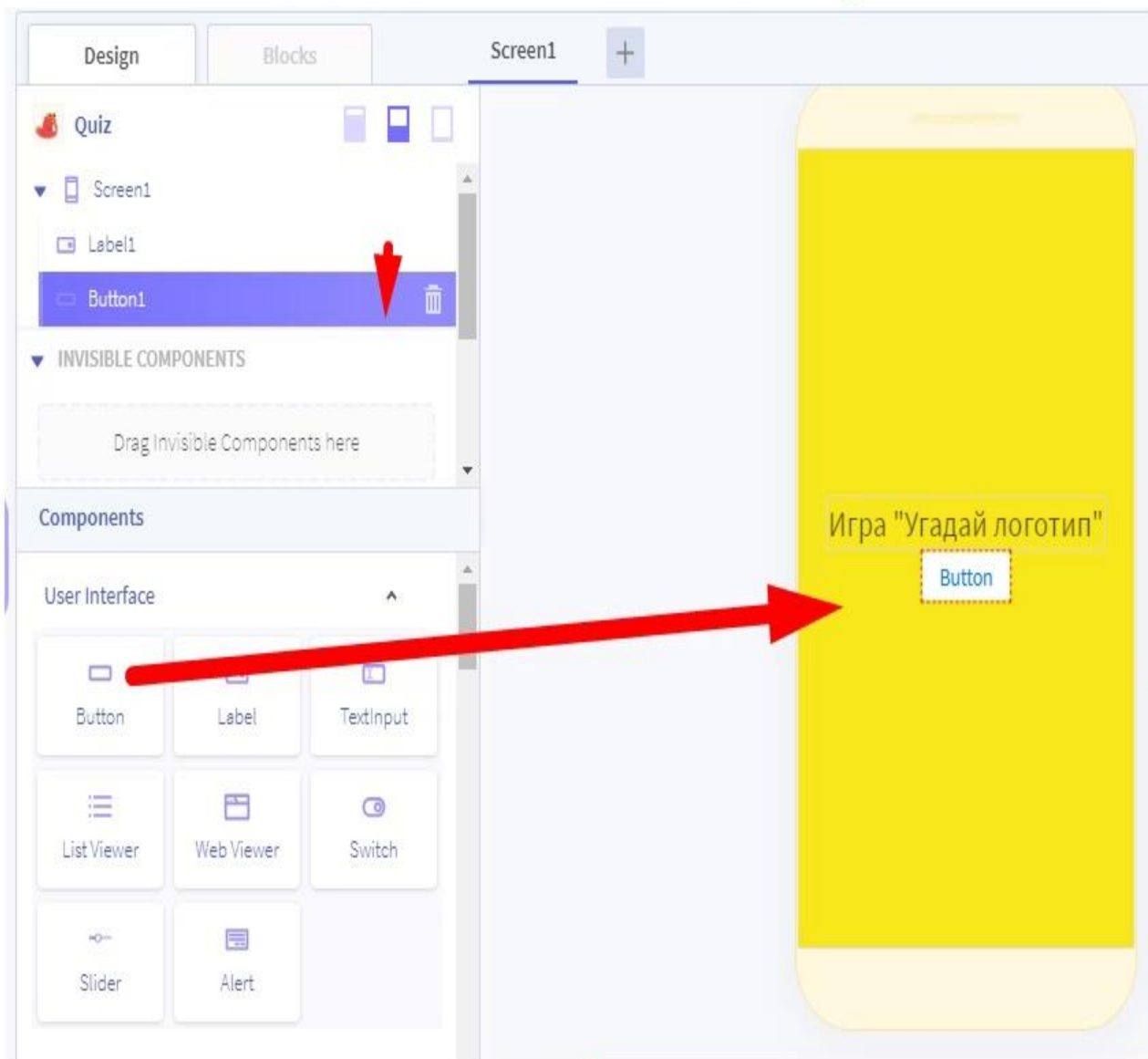
35

Width

Pick One: Fit contents, Fill container

A screenshot of a design tool's label editor showing styling options. It includes a 'FontStyle' dropdown set to 'normal', a 'Height' dropdown set to 'Absolute Size', a numeric input field with the value '35', and a 'Width' dropdown set to 'Pick One: Fit contents, Fill container'. A red arrow points to the '35' value in the Height field.

Добавим кнопку



Настроим кнопку



Button1

< Simple Adv >

Text

Играть! ×

TextColor

#007aff

BackgroundColor

#ffffff

FontSize

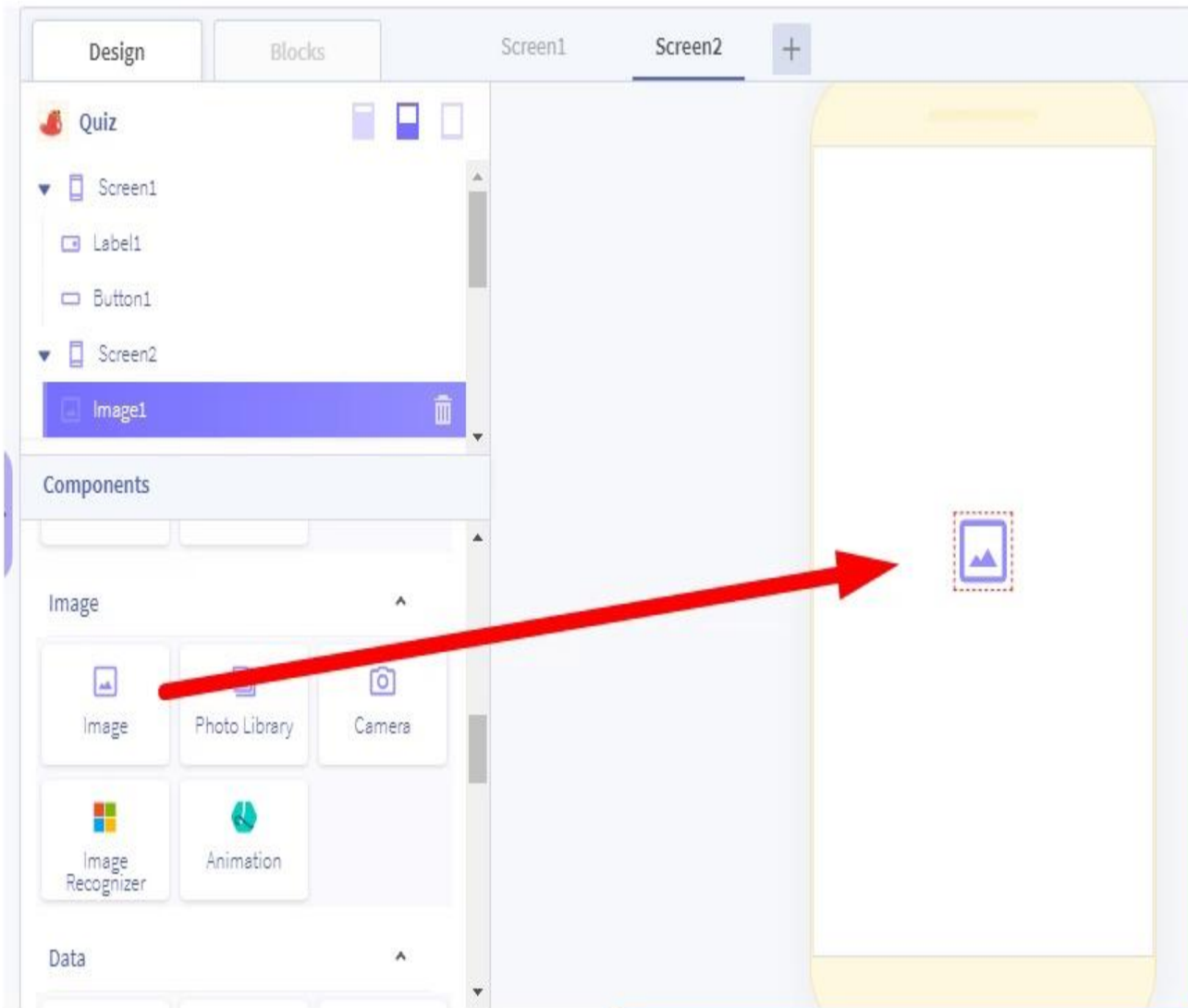
20

A settings panel for a button named "Button1". It has two tabs: "Simple" (selected) and "Adv". Under "Text", there is an input field containing "Играть!" and a close button "×". A red arrow points to this input field. Below that, "TextColor" is set to "#007aff" with a color picker icon. "BackgroundColor" is set to "#ffffff" with a color picker icon. "FontSize" is set to "20" in an input field. A red arrow points to this input field.

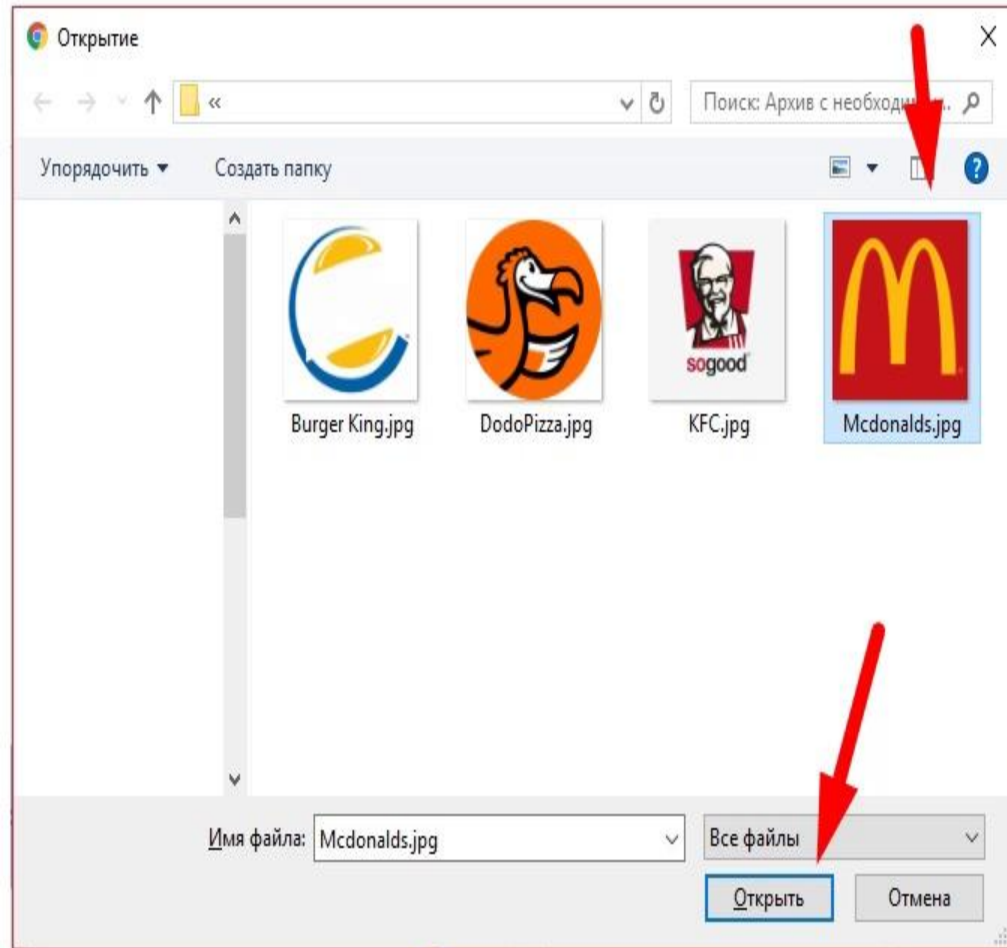
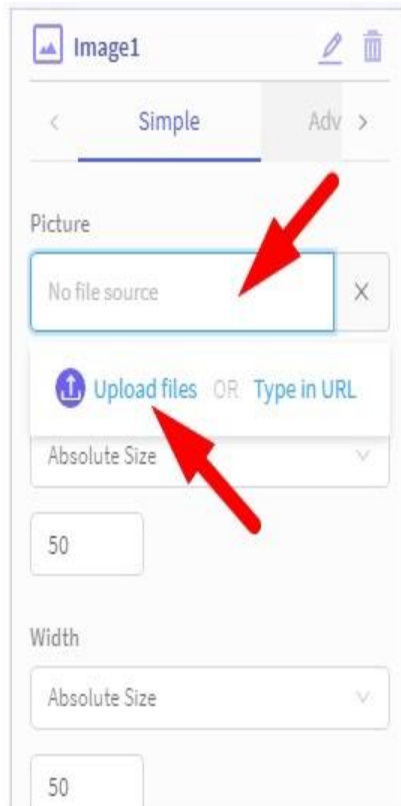
Создадим второй экран



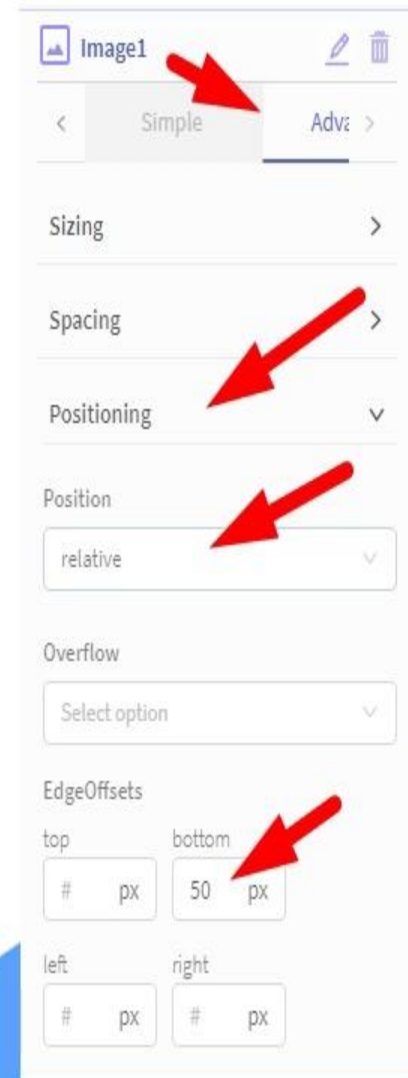
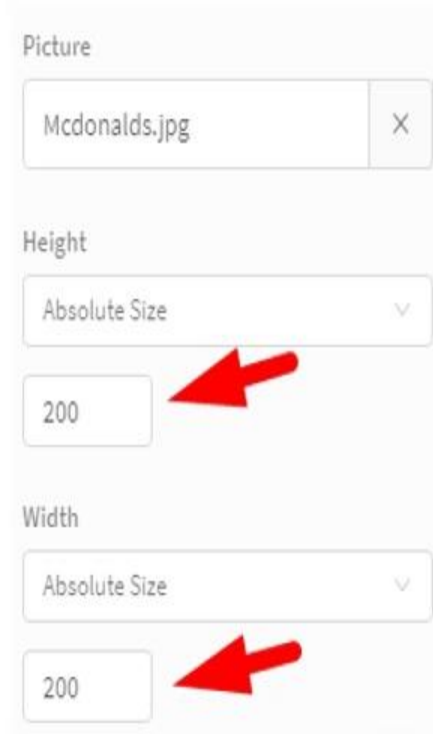
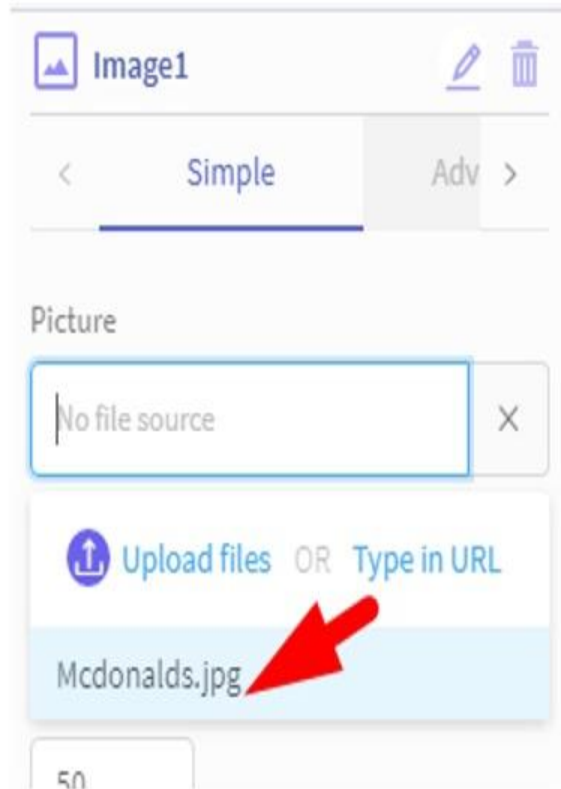
Добавим изображение



Добавим изображение



Добавим изображение



Добавим кнопки

The screenshot shows a design tool interface with two tabs: "Design" and "Blocks". The "Design" tab is active, showing a project named "Quiz" with a sub-screen "Screen2". The "Blocks" panel on the left lists components: "Image1", "Button4" (highlighted in blue), "Button3", and "Button2". Below this is a section for "INVISIBLE COMPONENTS" and a "Components" panel. The "Components" panel is expanded to "User Interface" and shows a grid of UI elements: "Button", "Label", "TextInput", "List Viewer", "Web Viewer", "Switch", "Slider", and "Alert". A red arrow points from "Button4" in the "Blocks" panel to the "Button" component in the "User Interface" panel. Another red arrow points from the "Button" component in the "User Interface" panel to the three buttons on the mobile screen. The mobile screen itself features a large red square with the yellow McDonald's "M" logo at the top, and three smaller, stacked buttons labeled "Button" at the bottom.

Добавим изображение



Width

Absolute Size



120

Добавим действие

The screenshot displays a visual programming environment with a 'Design' view on the left and a 'Blocks' view on the right. The 'Design' view shows a list of categories: Control, Logic, Math, Text, Lists, Color, Objects, Variables, Functions, Timer1, Label1, Button1, and Screen1. The 'Button1' category is highlighted with a red arrow. The 'Blocks' view shows a list of blocks: 'when Button1 Click', 'when Button1 LongClick', 'when Button1 TouchDown', and 'when Button1 TouchUp'. A red arrow points from the 'when Button1 Click' block in the 'Blocks' view to the 'when Button1 Click' block in the 'Design' view. Another red arrow points from the 'when Button1 Click' block in the 'Blocks' view to the 'when Button1 Click' block in the 'Design' view. The 'Design' view shows a screen with a 'when Button1 Click' block added to it. The 'Screen1' tab is selected in the top bar.

Добавим действие

The image shows a visual programming environment with a 'Blocks' palette on the left and a workspace on the right. The 'Control' category is selected in the palette. The workspace shows a 'when Button1 Click' block with a 'navigate to Screen2' block attached to its 'do' slot. A red arrow points from the 'Control' category to the 'when Button1 Click' block, and another red arrow points from the 'navigate to Screen1' block in the palette to the 'do' slot of the 'when Button1 Click' block. A third red arrow points to the 'navigate to Screen2' block within the 'do' slot.

Design Blocks Screen1 Screen2 +

Control

- Logic
- Math
- Text
- Lists
- Color
- Objects
- Variables
- Functions
- Label1
- Button1
- Screen1

if do

navigate to Screen1

wait 1 seconds

repeat 10 times do

when Button1 Click do navigate to Screen2

Добавим действие

The screenshot displays a mobile application development environment. At the top, there is a navigation bar with three tabs: "Screen1", "Screen2", and "Screen3". "Screen2" is currently selected and underlined. To the right of the tabs is a plus sign icon. Two red arrows point to the "Screen2" tab and the plus sign icon. Below the navigation bar, there are three orange script blocks, each representing a click event handler:

- when Button4 Click do navigate to Screen3
- when Button3 Click do navigate to Screen3
- when Button2 Click do navigate to Screen3

Each block consists of a "when" clause (e.g., "when Button4 Click") and a "do" clause (e.g., "do navigate to Screen3").

Тестируем приложение!

Добавим переменную

The screenshot displays a visual programming environment with a 'Blocks' palette on the left and a workspace on the right. The workspace is divided into five screens: Screen1, Screen2, Screen3, Screen4, and Screen5. A red arrow points from the 'initialize app variable name to' block in Screen1 to the 'initialize app variable Success to' block in Screen2. The 'Variables' category in the 'Blocks' palette is highlighted with a red arrow. The workspace contains the following blocks:

- Screen1:
 - initialize app variable name to
 - set app Success to
 - change app Success by 1
 - app Success
- Screen2:
 - initialize app variable Success to
- Screen4:
 - when Button1 Click do navigate to Screen2

Добавим переменную

The screenshot shows a visual programming environment with a 'Design' tab and a 'Blocks' panel. The workspace is titled 'Screen1'. In the 'Control' category of the 'Blocks' panel, a block containing the number '0' is highlighted with a red arrow. Another red arrow points from this block to an 'initialize app variable Success to 0' block in the workspace. Below this, there is a 'when Button1 Click' block with a 'do' block containing 'navigate to Screen2'.

Определим верную кнопку

The image shows a mobile application design interface. The top navigation bar includes tabs for Design, Blocks, and five screens (Screen1 to Screen5). The Design tab is active, showing a tree view of the app's structure. Under Screen2, a button labeled 'Button4' is highlighted in blue, with a red arrow pointing to it. Below the tree view is a 'Components' panel with a 'User Interface' section containing various UI elements like Button, Label, TextInput, List Viewer, Web Viewer, Switch, Slider, and Alert. The main canvas displays a mobile app mockup with a yellow header, a red square containing the McDonald's logo, and a white list box containing three items: 'McDonald's', 'Burger King', and 'KFC'. The 'McDonald's' item is highlighted with a dashed red border.

Изменим значение переменной

The screenshot displays a visual programming environment with a 'Blocks' palette on the left and a workspace on the right. The workspace is divided into five screens: Screen1, Screen2, Screen3, Screen4, and Screen5. A red arrow points from the 'set app Success to' block in Screen1 to the 'set app Success to' block in Screen3. Another red arrow points to the 'Variables' category in the 'Blocks' palette.

Blocks Palette:

- Control
- Logic
- Math
- Text
- Lists
- Color
- Objects
- Variables**
- Functions
- Image1
- Button4
- Button3
- Button2
- Screen2

Screen1:

- initialize app variable name to
- set app Success to
- change app Success by 1
- app Success
- set app "name" to
- change app "name" by 1
- app "name"

Screen2:

- when Button4 Click do navigate to Screen3
- when Button3 Click do navigate to Screen3
- when Button2 Click do navigate to Screen3

Screen3:

- set app Success to

Изменим значение переменной

The image shows the Scratch interface. On the left is the 'Blocks' palette with categories: Control, Logic, Math, Text, Lists, Color, Objects, Variables, Functions, Timer1, and Image1. A red arrow points to the 'Math' category, which contains a '1 + 1' block. A second red arrow points from this block to a script area on the right. The script area is for 'Screen2' and contains three event-driven scripts:

- when Button4 Click do: navigate to Screen3, set app Success to 1 + 1
- when Button3 Click do: navigate to Screen3
- when Button2 Click do: navigate to Screen3

The 'set app Success to 1 + 1' block is highlighted with a red arrow pointing to it from the 'Math' category block.

Изменим значение переменной

The screenshot displays a visual programming environment with a 'Blocks' panel on the left and a workspace on the right. The workspace is currently on 'Screen2' and shows two event-driven blocks: 'when Button4 Click' and 'when Button3 Click'. The 'when Button4 Click' block contains a 'do' block with 'navigate to Screen3' and 'set app Success to app Success + 1'. The 'when Button3 Click' block contains a 'do' block with 'navigate to Screen3'. In the 'Blocks' panel, the 'Variables' category is selected, showing an 'app Success' variable block. A red arrow points from this variable block to the 'set app Success to' block in the workspace. Another red arrow points from the 'app Success' variable block to the 'set app Success to' block in the workspace.

Design Blocks Screen1 Screen2 Screen3 Screen4 Screen5 +

Control
Logic
Math
Text
Lists
Color
Objects
Variables

initialize app variable name to

set app Success to

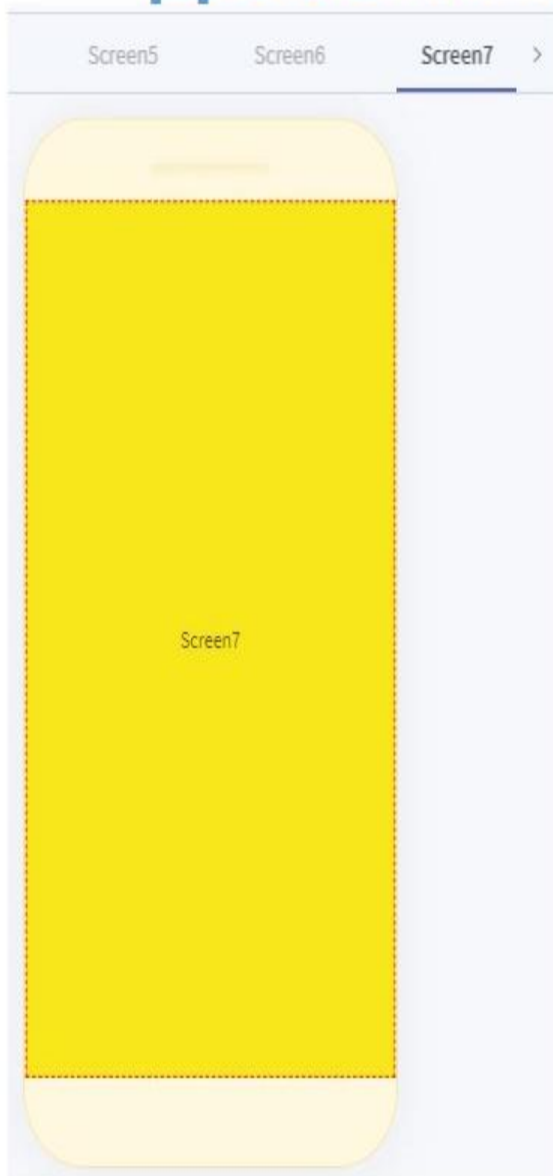
change app Success by 1

app Success

when Button4 Click
do
navigate to Screen3
set app Success to app Success + 1

when Button3 Click
do
navigate to Screen3

Добавим последний экран



Добавим надписи



Добавим действие

The screenshot shows the Scratch interface with the 'Blocks' palette open for 'Screen7'. The left sidebar has 'Screen7' selected. The palette contains several blocks:

- Control: when Screen7 Opens (do)
- Logic: when Screen7 Starts (do)
- Variables: from Screen7 set BackgroundColor to [black]
- Functions: from Screen7 get BackgroundColor
- Screen7: from Screen7 set BackgroundPicture to []
- Screen7: from Screen7 get BackgroundPicture

A red arrow points from the 'when Screen7 Opens' block in the palette to the same block on the stage. Another red arrow points from the 'Screen7' category in the sidebar to the 'from Screen7 get BackgroundPicture' block.

Изменим надпись

The image shows a visual programming environment with a 'Blocks' palette on the left and a workspace on the right. The workspace is titled 'Screen7' and contains a 'when Screen7 Opens' event block. Inside this event, there is a 'do' block containing a 'from Label3 set Text to' block. A red arrow points from the 'set Text to' block in the palette to the 'do' block in the workspace. Another red arrow points from the 'set Text to' block in the palette to the 'do' block in the workspace.

The 'Blocks' palette contains the following blocks for 'Label3':

- from Label3 set Text to " " (highlighted)
- from Label3 get Text
- from Label3 set FontSize to 0
- from Label3 get FontSize
- from Label3 set Color to (black)
- from Label3 get Color
- from Label3 set BackgroundColor to (black)
- from Label3 get BackgroundColor

The workspace contains the following blocks:

- when Screen7 Opens
- do from Label3 set Text to " "

Изменим надпись

The image shows a programming interface with a 'Design' tab and a 'Blocks' tab. The 'Blocks' tab is active, displaying a list of block categories on the left: Control, Logic, Math, Text, Lists, Color, Objects, Variables, Functions, Label2, Label3, and Screen7. A red arrow points from the 'Text' category to a 'join' block in the 'Text' category. Another red arrow points from this 'join' block to a script area on the right. The script area shows a 'when Screen7 Opens' block followed by a 'from Label3 set Text to' block, which is being dragged from the 'Text' category. The 'set Text to' block is currently empty, and the 'join' block is being attached to its end. The 'join' block contains the text 'hello' and 'world'.

Design Blocks < Screen3 Screen4 Screen5 Screen6 Screen7 > +

Control “ ”

Logic

Math

Text

Lists

Color

Objects

Variables

Functions

Label2

Label3

Screen7

length of “ abc ”

does “ abc ” contain “ b ”

in text “ abc ” get substring from letter # 1 to letter # 2

in text “ abc ” get letter # 1

when Screen7 Opens

from Label3 set Text to

join “ hello ” “ world ”

Изменим надпись

The screenshot displays the Scratch IDE interface for editing a script on Screen7. The 'Blocks' palette on the left is open to the 'Variables' category, where a variable named 'app Success' is defined. A red arrow points from this variable block to its use in a script on the main stage. The script begins with a 'when Screen7 Opens' event block, followed by a 'do' block. Inside the 'do' block, there is a 'from Label3 set Text to' block with a 'join' block containing the string 'hello ' and the 'app Success' variable. The 'do' block also includes an 'app Success' variable block. Other visible blocks in the 'Blocks' palette include 'initialize app variable name to', 'set app Success to', 'change app Success by 1', 'set app "name" to', 'change app "name" by 1', and 'app "name"'. The main stage area shows a script for Screen7 with a 'when Screen7 Opens' event block containing a 'do' block with 'from Label3 set Text to' and 'join' blocks, and an 'app Success' variable block.

Изменим надпись

```
when Screen7 Opens  
do  
  from Label3 set Text to join  
    “Правильно отвечено: ”  
    app Success
```

Тестируем приложение!

Добавим таймер

The screenshot shows the Xcode interface for a quiz application. The main canvas displays a yellow mobile screen with the text "Игра 'Угадай логотип'" and a blue button labeled "Играть!". A red arrow points from the "Timer" component in the "Sensors" section of the Components palette to the bottom of the mobile screen. A dashed box at the bottom of the screen indicates the location of the "Timer1" component.

The screenshot shows the settings for the "Timer1" component. The "Simple" tab is selected. The "Enabled" toggle is set to "false", the "IntervalMilliseconds" is set to "1000", and the "Loops" toggle is set to "true". Red arrows point to each of these settings.

Timer1

Simple Advanced

Enabled
 false

IntervalMilliseconds
1000

Loops
 true

Добавим переменную

The screenshot displays the CodaLab IDE interface with a focus on the 'Blocks' tab for 'Screen1'. The left sidebar shows a category list with 'Variables' highlighted in orange. A red arrow points from the 'Variables' category to the 'initialize app variable Success to 0' block in the code area. Another red arrow points from the 'initialize app variable name to' block to the 'initialize app variable Success to 0' block. The code area contains the following blocks:

- initialize app variable name to
- set app Success to
- change app Success by 1
- app Success
- app Time
- set app " name " to
- change app " name " by 1
- app " name "

On the right side, the code for 'Screen2' is visible, showing:

- initialize app variable Success to 0
- initialize app variable Time to 0
- when Button1 Click
- do navigate to Screen2

Активируем таймер

The screenshot displays the Scratch Blocks editor interface. The top navigation bar shows tabs for Screen1, Screen2, Screen3, Screen4, and Screen5. The left sidebar contains a category list with 'Timer1' selected. The main workspace is divided into two sections: 'Design' (left) and 'Blocks' (right).

Design Section:

- Control: when Timer1 Fires (do block)
- Logic: from Timer1 set Enabled to true
- Logic: from Timer1 get Enabled
- Logic: from Timer1 set IntervalMilliseconds to 3000
- Logic: from Timer1 get IntervalMilliseconds
- Logic: from Timer1 set Loops to true
- Logic: from Timer1 get Loops

Blocks Section:

- Logic: initialize app variable Success to 0
- Logic: initialize app variable Time to 0
- Control: when Button1 Click (do block)
- Logic: navigate to Screen2
- Logic: from Timer1 set Enabled to true

A red arrow points from the 'from Timer1 set Enabled to true' block in the Design section to the 'from Timer1 set Enabled to true' block in the Button1 Click event's do block in the Blocks section. Another red arrow points from the 'Timer1' category in the sidebar to the 'from Timer1 set IntervalMilliseconds to 3000' block.

Считаем время

The screenshot displays a visual programming environment with a 'Blocks' palette on the left and a workspace on the right. The workspace is divided into five screens: Screen1, Screen2, Screen3, Screen4, and Screen5. The 'Timer1' block is selected in the palette, and a red arrow points to it. The workspace shows the following logic:

- Screen1:** A 'when Timer1 Fires' block with a 'do' loop containing:
 - 'from Timer1 set Enabled to true'
 - 'from Timer1 get Enabled'
 - 'from Timer1 set IntervalMilliseconds to 3000'
 - 'from Timer1 get IntervalMilliseconds'
 - 'from Timer1 set Loops to true'
 - 'from Timer1 get Loops'
- Screen2:** A 'when Button4 Click' block with a 'do' loop containing:
 - 'navigate to Screen3'
 - 'change app Success by 1'
- Screen3:** A 'when Button3 Click' block with a 'do' loop containing:
 - 'navigate to Screen3'
- Screen4:** A 'when Button2 Click' block with a 'do' loop containing:
 - 'navigate to Screen3'
- Screen5:** A 'when Timer1 Fires' block with a 'do' loop.

Считаем время



```
when Timer1 Fires  
do set app Time to app Time + 1
```

The image shows a Scratch code block. It is an orange 'when' block with 'Timer1' selected in the dropdown and 'Fires' in the event dropdown. Below it is a 'do' block containing a 'set' block. The 'set' block has 'app Time' in the dropdown and 'to' in the dropdown. The value being set is 'app Time + 1', where 'app Time' is in a dropdown, '+' is an operator, and '1' is a number.

Задание:

Добавьте подсчет времени на все остальные экраны



Добавим надпись



Остановим таймер

The screenshot displays a visual programming environment with a 'Blocks' palette on the left and a workspace on the right. The workspace shows a sequence of events for 'Screen7' opening.

Blocks Palette (Left):

- Control
 - when Timer1 Fires
- Logic
 - do
- Text
 - from Timer1 set Enabled to true
- Lists
 - from Timer1 get Enabled
- Color
 - from Timer1 set IntervalMilliseconds to 3000
- Variables
 - from Timer1 get IntervalMilliseconds
- Function
 - from Timer1 set Loops to true
- Timer1
 - from Timer1 get Loops

Workspace (Right):

- when Screen7 Opens
 - do
 - from Label3 set Text to join " Правильно отвечено: "
 - app Success
 - from Timer1 set Enabled to false

Red arrows point to the 'Timer1' category in the palette, the 'when Screen7 Opens' event block, and the 'from Timer1 set Enabled to false' block.

Изменим надпись

```
when Screen7 Opens  
do  
  from Label3 set Text to join (" Правильно отвечено: ")  
  from Timer1 set Enabled to false  
  from Label4 set Text to join (" Затрачено времени: ")
```


Тестируем приложение!

