

## What shape is it?

This program presents a quiz to preschoolers to recognize simple geometric shapes.

It basically draws a shape on the screen and the user has to identify it by clicking the correct button. This then repeats until the timer expires.

### Explore the program:

If you want to play with my final program to get a feel for this quiz, click the link given at the end of the article. Try not to peek at the scripts yet, since we want to design them ourselves below.

1. Click on the "Green flag".
2. Press SPACE to start the timer and the quiz.
3. Your score is presented when the timer expires.

## Scratch and CS Concepts Used

When we design this program, we will make use of the following Scratch and CS concepts. Learn these concepts if you don't know them before proceeding further.

- Algorithms
  - o Designing new algorithms
- Backdrops – multiple
- Concurrency
  - o Synchronization using broadcasting
- Conditional statements:
  - o Conditions: YES/NO questions
  - o Relational operators (=, <, >)
  - o Conditionals (IF)
  - o Conditionals (If-Else)
  - o Conditionals (Wait until)
- Data types – basic
  - o Integers
- Events
- Geometry - square, circle, triangle, spiral
- Looping (iteration)

- Looping - simple (repeat, forever)
- Motion
  - Motion - relative
- OOP
  - Clones
- Pen commands
- Program output
  - Text
- Random numbers
- Sequence
- Stopping scripts
- User input
  - Click buttons
- Variables
  - Simple
  - As timer
- XY Geometry

## High Level Design

As usual, there is some backend work and frontend work. Backend work consists of these actions:

- Pick at random a shape from the available collection
- Ask it to draw itself
- Check if user clicks correctly and update score variables
- Stop everything when timer expires and present score

Frontend work consists of a collection of shapes. Each shape should know how to draw itself. It should also present a click-button for the user to click. When clicked the shape should check the click was correct and send a message to the backend.

## Objects:

Using object-oriented approach, we will use the following objects:

- Backend (any sprite and/or stage)
- Click button for each shape

## Feature Idea # 1: Backend with just one shape

*Create the backend and have it work with one shape (say square).*

### Design:

Step 1: *Design the backend.*

Here is a simple algorithm for the backend:

```
Forever:  
    Send "draw" message to the "shape" object  
    Wait till it is done  
End forever
```

Another "timer" script can terminate the above loop.

```
When "correct" received  
    Increment the variable "correct"
```

Step 2: *Design a square shape object.*

The shape object (in this case, a "square" click button) will draw a square when it receives the "draw" message. In order to remember that the "current" shape is a square, it will set the "shape" variable. There is a slight problem here. We cannot let the click button do the drawing (since it will look weird to have a click button moving around). So, we will create a clone and make it hide and do the drawing.

When the click button is clicked we should check if the "shape" variable is set to "square" and if so, send a "correct" message to the backend logic.

### Save as Version 1

Save your program as version 1. Compare your program with my program at the link below.

Version 1: [shape-quiz-1.sb2](#)

## Feature Idea # 2: Add more shape objects

*Using identical logic add more shapes, such as, circle, pentagon, hexagon, etc.*

## Design:

One challenge for the "backend" now is how to pick from among the collection of shapes. It could use the random operator to pick a number randomly and just broadcast that number. Each shape could identify itself by a unique serial number. Here is the modified algorithm:

```
Assumption: There are N shape objects
Forever:
    Pick a random number R between 1 and N
    Send R as a message to all
    Wait till it is done
End forever
```

Every shape object will then listen for its own number and react accordingly.

## Feature Idea # 3: Niceties

*Add means to vary the color, pen thickness, and size of each shape.*

## Design:

We will have a "size" variable to control the size of the shape. Then it is a matter of using the random operator to pick a color, pen thickness and size.

## Save as Program Version "Final"

Congratulations! You have completed all the main features of the program. Compare your program with my program at the link below.

<https://scratch.mit.edu/projects/359342908/>

## How to run the program:

1. Click on the "Green flag".
2. Press SPACE to start the timer and the quiz.
3. Your score is presented when the timer expires.

*Author: Abhay B. Joshi (abjoshi@yahoo.com)*

*Last updated: 14 January 2020*