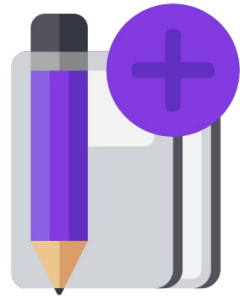


Brainstorming:

Creating games using Pygame



Pygame is a library used to create games

We used to work only with the standard Python library.

Pygame has **modules** with ready-made tools for:

- handling in-game events;
- handling external events;
- configuring game timers;
- configuring game interfaces and sound effects,
and more.

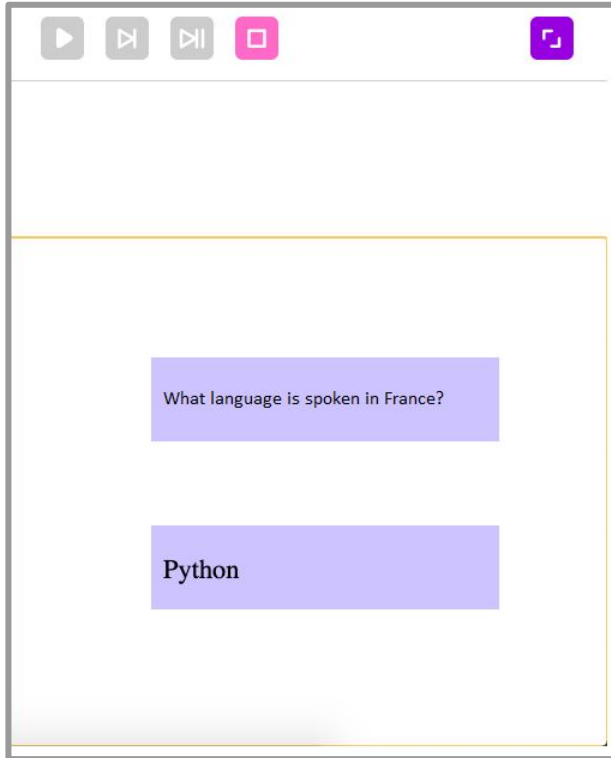
<i>Command</i>	<i>Purpose</i>
<code>import pygame</code>	Enables all features of the Pygame library.



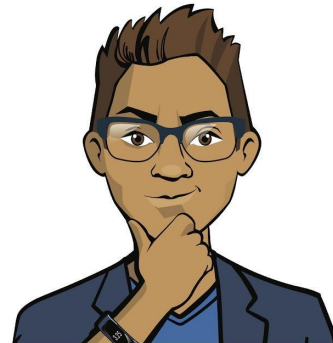
Brain
storm

Creating games using Pygame

You already know some concepts related to games, such as **sprites**, **scenes** and **game loops**.



Show the sprites and the game scene in the screenshot from the game.

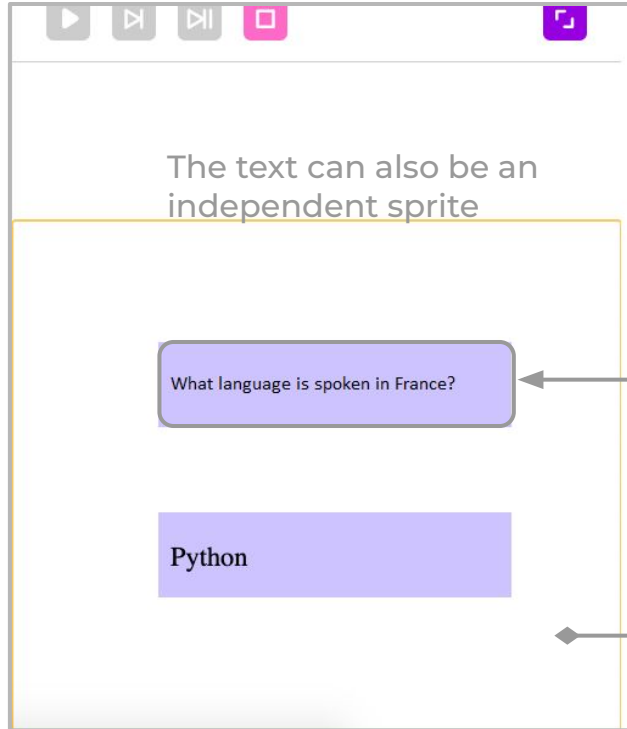


Brain
storm

Creating games using Pygame

The scene is the "background" of the game. It's what all the objects are placed on.

Sprites are any game objects separate from the scene.



**Rectangular area
with text**

Scene

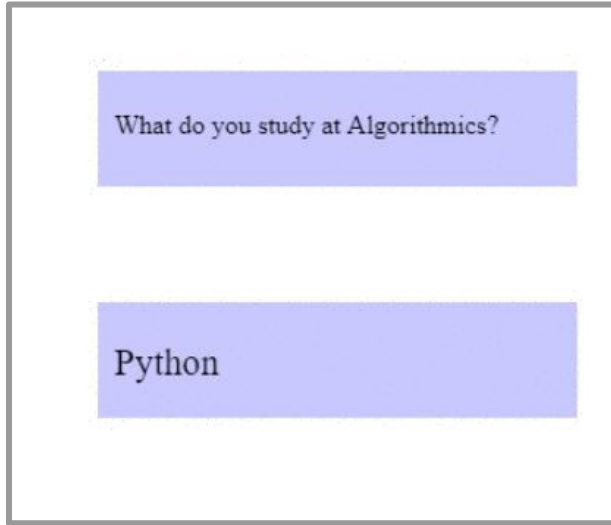


**Brain
storm**

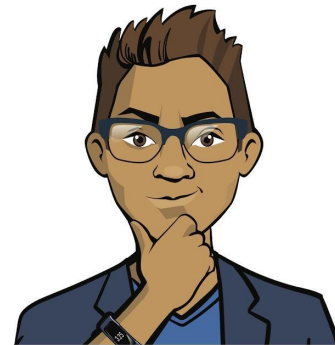
Creating games using Pygame

A **game loop** is a cycle where the following things happen at each stage:

- event analysis and handling;
- rendering of the background and sprites;
- time is counted.



What happens in the "Questions and Answers" game loop?

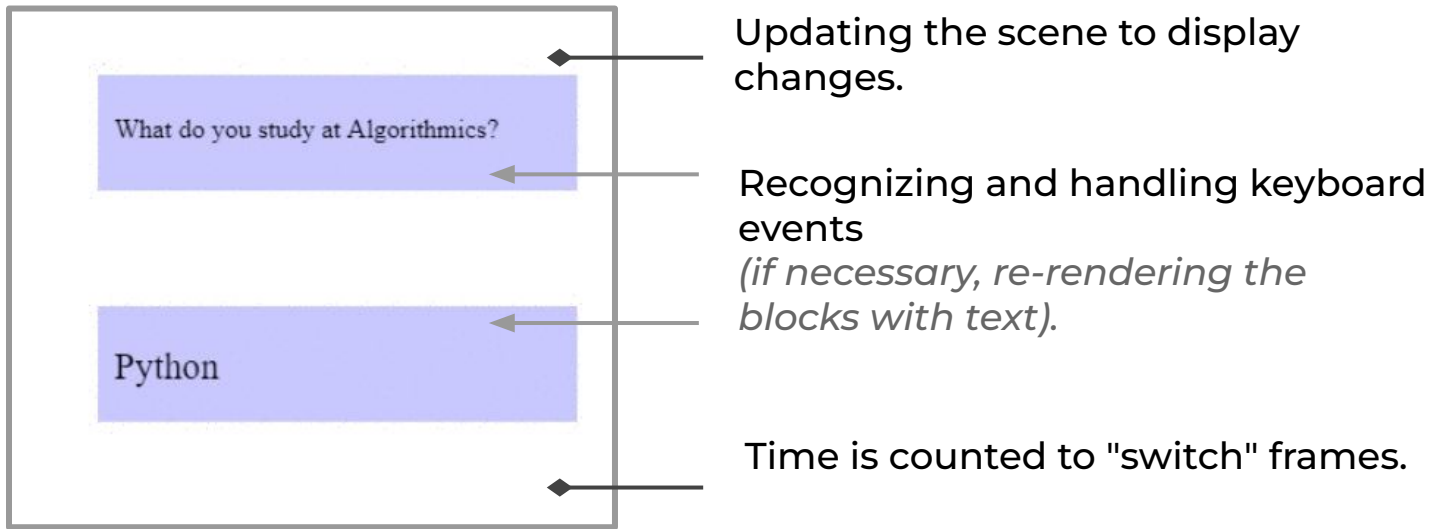


Brain
storm

Creating games using Pygame

A **game loop** is a cycle where the following things happen at each stage:

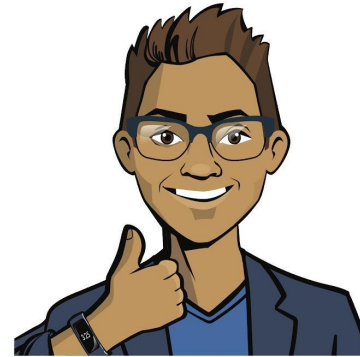
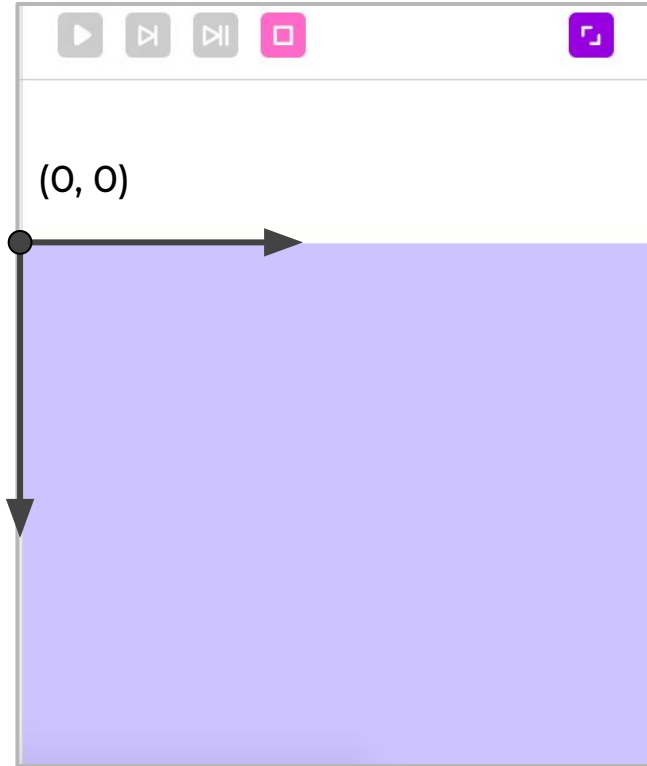
- event analysis and handling;
- rendering of the background and sprites;
- time is counted.



Brain
storm

1. Creating a template with a colored background

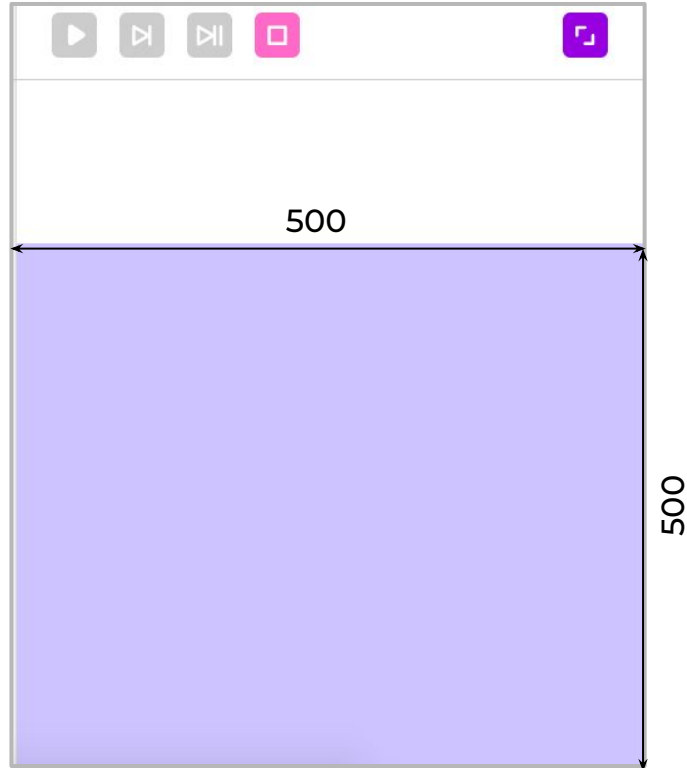
In Pygame, the starting point is located in the upper-left corner of the window.
The developer determines the size of the window.



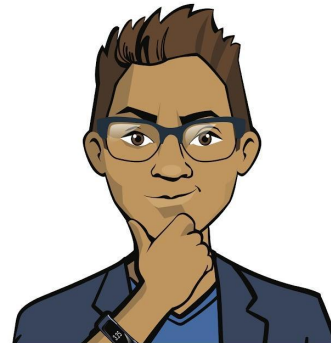
Brain
storm

1. Creating a template with a colored background

We will program a game template with a colored background.



What knowledge do you need to create this kind of scene?



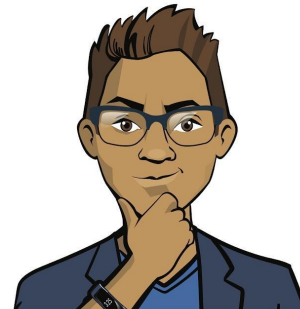
Brain
storm



1. Creating a template with a colored background

<i>Command</i>	<i>Purpose</i>
<code>pygame.init()</code>	Enables the ability to use commands for Pygame objects.
<code>window = pygame.display.set_mode((500, 500))</code>	Creates a window with the following size: (width, length).
<code>window.fill(<color>)</code>	Fills the background with the specified color.

How do we set the color we want?

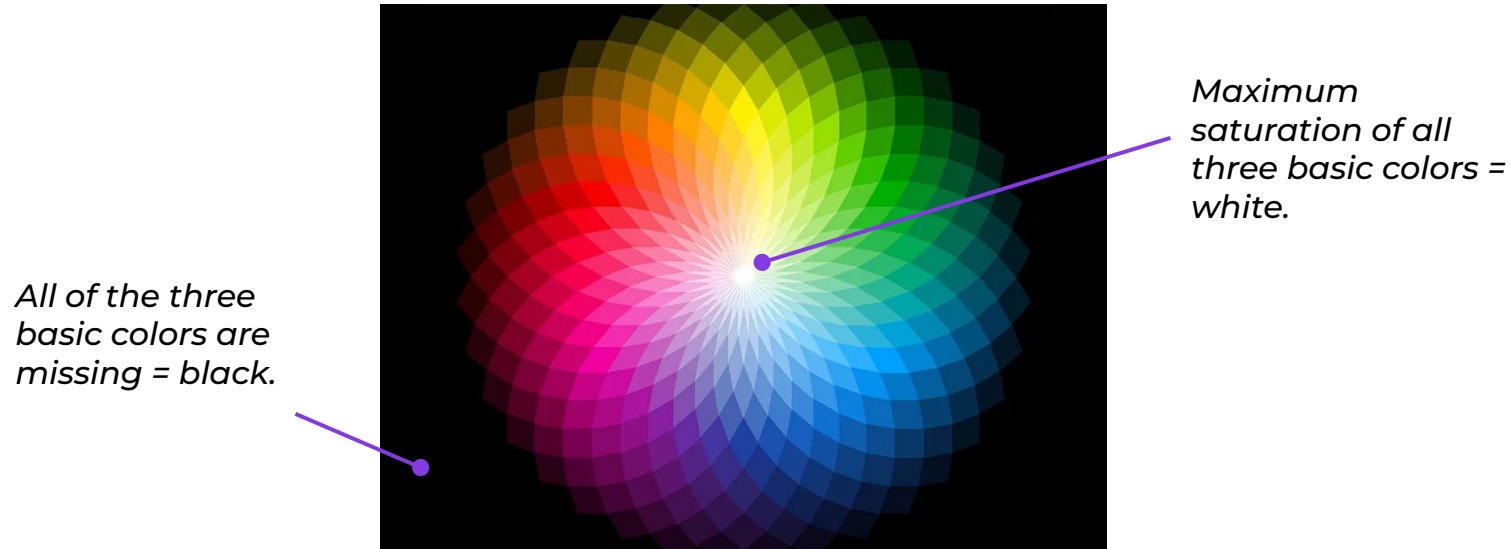


Brain
storm

RGB color palette (red, green, blue)

We can use the RGB color palette.

We get the palette's colors by mixing red, green and blue. The absence of color appears as black.



[Link to the RGB color calculator](#)

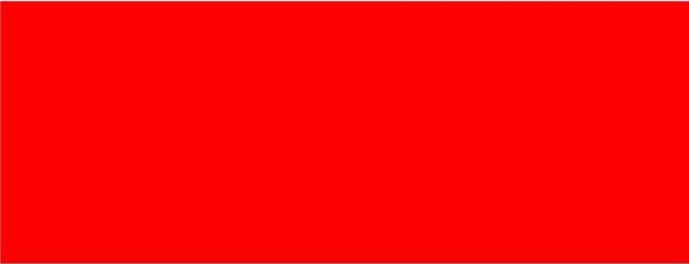


Brain
storm

RGB color palette (red, green, blue)

Example: red (255, 0, 0).


RGB Calculator





`rgb(255, 0, 0)`

`#ff0000`

`hsl(0, 100%, 50%)`

R: 

G: 

B: 

[Link to the RGB color calculator](#)



Brain
storm

1. Creating a template with a colored background

A very simple game template for displaying the scene without sprites:

```
import Pygame
```

```
pygame.init()
```

```
back = (255, 255, 255)
```

```
mw = pygame.display.set_mode((500, 500))
```

```
mw.fill(back)
```

If you create and launch this kind of program, it will start and then immediately stop working! Why?



Brain
storm

1. Creating a template with a colored background

A very simple game template for displaying the scene without sprites:

```
import Pygame
```

```
pygame.init()
```

```
back = (255, 255, 255)
```

```
mw = pygame.display.set_mode((500, 500))
```

```
mw.fill(back)
```

The program **doesn't have a game loop!**

The game runs for just a moment and then stops straight away.

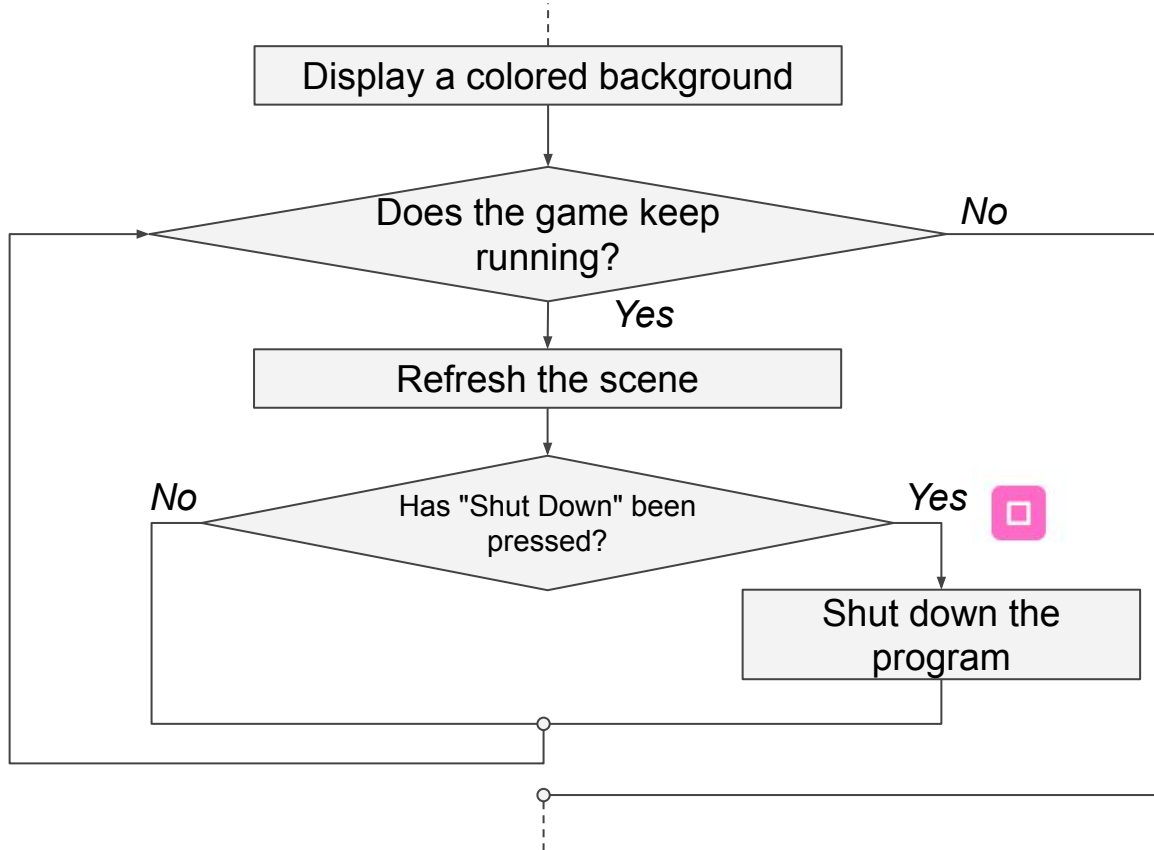
What do we need to add to make sure the game keeps running?



Brain
storm

1. Creating a template with a colored background

A very simple game template for displaying the scene without sprites:



Brain
storm

1. Creating a template with a colored background

Note. The color can be set using the RGB palette.

<i>Command</i>	<i>Purpose</i>
<code>window = pygame.display.set_mode((500, 500))</code>	Creates a window with the following size: (width, length).
<code>window.fill(<color>)</code>	Fills the background with the specified color.
<code>pygame.display.update()</code>	Updates the content of the game window.
<code>clock = pygame.time.Clock()</code>	Creates a game timer.
<code>clock.tick(40)</code>	Sets the scene refresh rate to ~40 FPS.



Brain
storm

1. Creating a template with a background

A very simple game template for displaying the scene without sprites:

Connect Pygame modules

Create a scene object

Fill the scene with color

← The background can be white

Create a game timer

Game loop:

Set the frame rate to ~40 FPS

Update the scene
(next frame of the game loop)



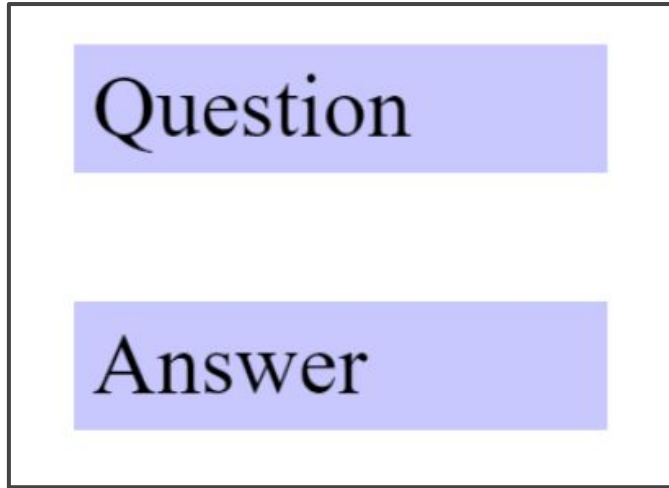
Brain
storm



2. Creating and placing sprites

There are different ways to create sprites.

Let's say that in the "Questions and Answers" game, a sprite is **a rectangle with a label**.



In Pygame, there are tools for working with text and rectangular areas. Can you make the sprite you need from them?

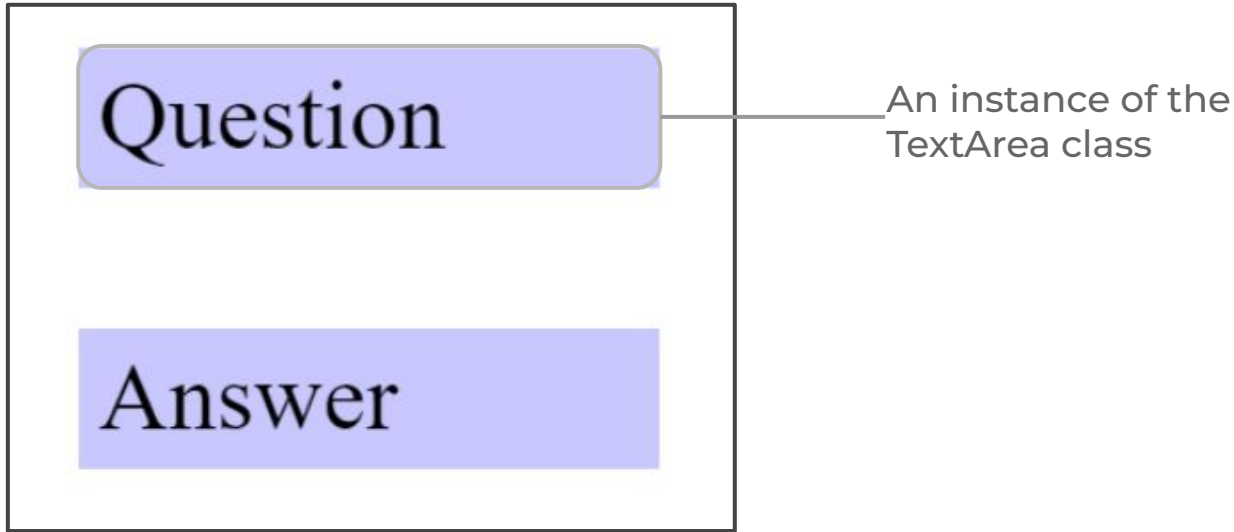


Brain
storm

2. Creating and placing sprites

There are different ways to create sprites.

Let's say that in the "Questions and Answers" game, a sprite is **a rectangle with a label**.



Yes, all of its functionality can be "gathered" in its TextArea class.



Brain
storm

2. Creating and placing sprites

Commands for working with rectangular (Rect) areas.

<i>Command</i>	<i>Purpose</i>
<code>pygame.init()</code>	Enables the ability to use commands for Pygame objects.
<code>rect = pygame.Rect(x, y, width, height)</code>	<p>Creates a rectangle at the point (x, y) with a certain width and height.</p> <p>(x, y) — the upper left corner of the shape.</p>
<code>pygame.draw.rect(mw, fill_color, rect)</code>	Draws a rectangle (rect) in the mw window and fills it with fill_color.



Brain
storm

2. Creating and placing sprites

Commands for working with text.

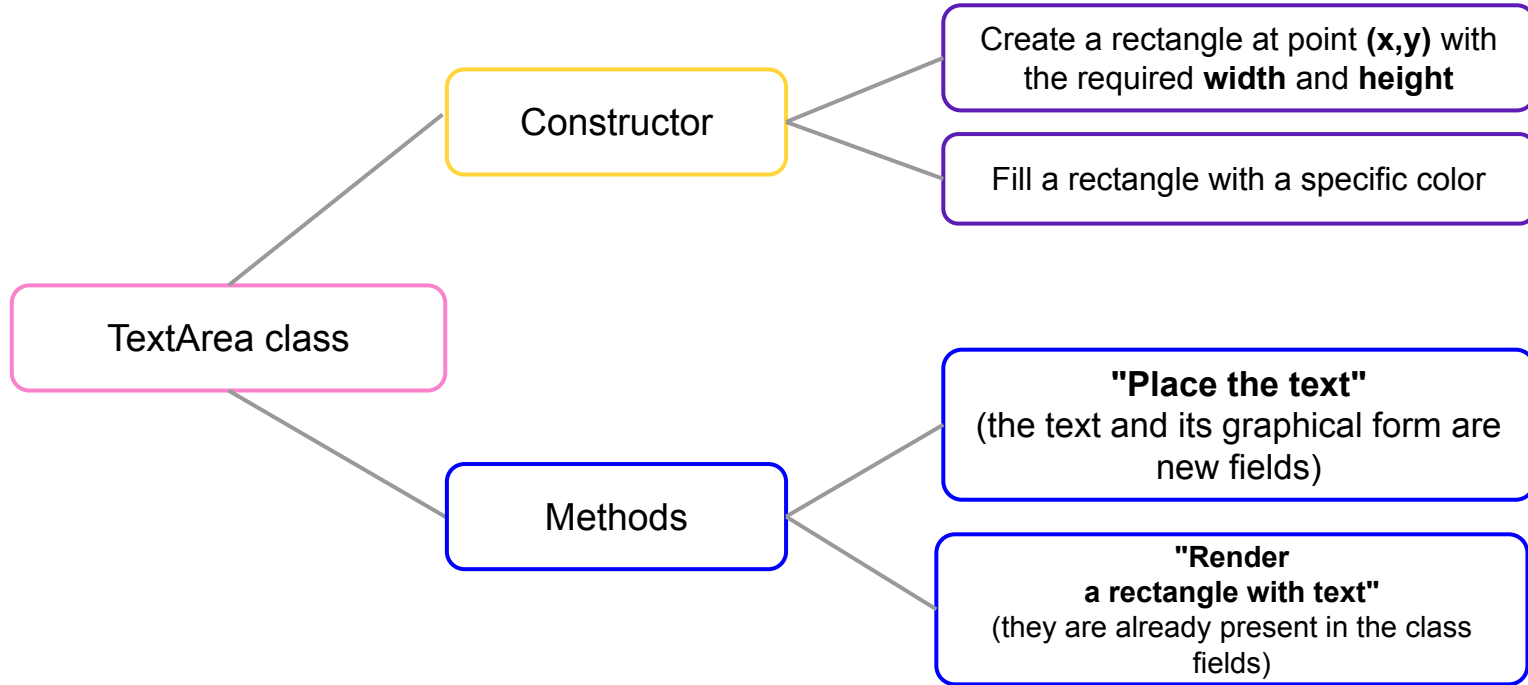
<i>Command</i>	<i>Purpose</i>
<code>pygame.init()</code>	Enables the ability to use commands for Pygame objects.
<code>font1 = pygame.font.Font(None, 70)</code>	Sets the font / Creates a font object with the parameters: font — default, size — 70.
<code>question = font1.render(text, True, (255, 215, 0))</code>	Creates a question with text, drawn using font1, color (255, 215, 0).
<code>mw.blit(question, (x, y))</code>	Displays the text at point (x, y) in the mw window.



Brain
storm

TextArea class

The class for the sprite question and the sprite answer:



Brain
storm

TextArea class

```
class TextArea():  
    You can specify default parameter values  
    def __init__(self, x=0, y=0, width=10, height=10, color=None):  
        self.rect = pygame.Rect(x, y, width, height)  
        self.fill_color = color  
  
    def set_text(self, text, fsize=12, text_color=BLACK):  
        self.text = text  
        self.image = pygame.font.Font(None, fsize).render(text, True, text_color)  
  
    def draw(self, shift_x=0, shift_y=0):  
        pygame.draw.rect(mw, self.fill_color, self.rect)  
        mw.blit(self.image, (self.rect.x + shift_x, self.rect.y + shift_y))
```



Brain
storm

TextArea class

```
class TextArea():
```

```
    def __init__(self, x=0, y=0, width=10, height=10, color=None):
```

```
        self.rect = pygame.Rect(x, y, width, height)
```

```
        self.fill_color = color
```

```
    def set_text(self, text, fsize=12, text_color=BLACK):
```

```
        self.text = text
```

```
        self.image = pygame.font.Font(None, fsize).render(text, True, text_color)
```

You can set the font right

...and render the text you want

```
    def draw(self, shift_x=0, shift_y=0):
```

```
        pygame.draw.rect(mw, self.fill_color, self.rect)
```

```
        mw.blit(self.image, (self.rect.x + shift_x, self.rect.y + shift_y))
```



Brain
storm

TextArea class

```
class TextArea():  
    def __init__(self, x=0, y=0, width=10, height=10, color=None):  
        self.rect = pygame.Rect(x, y, width, height)  
        self.fill_color = color  
  
    def set_text(self, text, fsize=12, text_color=BLACK):  
        self.text = text  
        self.image = pygame.font.Font(None, fsize).render(text, True, text_color)  
  
    def draw(self, shift_x=0, shift_y=0):  
        pygame.draw.rect(mw, self.fill_color, self.rect)  
        mw.blit(self.image, (self.rect.x + shift_x, self.rect.y + shift_y))
```

*The coordinates of the
rectangle are already known*

*The text is displayed according to the coordinates
of the rectangle, but shifted slightly.*

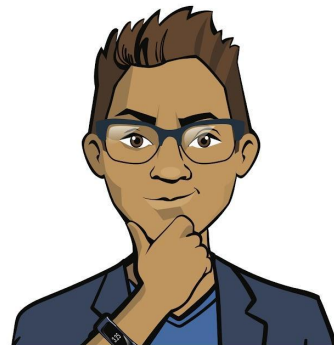


Brain
storm

Your tasks:

1. Create a game template with a scene, a color fill, and a game loop.
2. Create a class for rectangle sprites with labels.

If there is time left over, create two sprites for the question and the answer and place them in the scene!

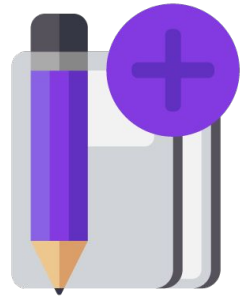


Brain
storm



Brainstorming:

Handling keyboard events



Handling keyboard events

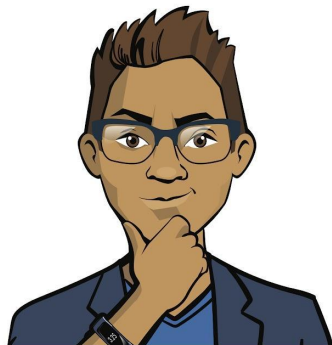
Instances of the `TextArea` class are controlled using the keyboard.

According to the terms of reference:

1. **When you press** the **Q** key, a random question is displayed in the first block.
2. **When you press** the **A** key, a random answer is displayed in the second block.

Previously, event handling was implemented using methods of the `Screen` class.

Pygame has its own methods for handling keystrokes.



Brain
storm

Handling keyboard events

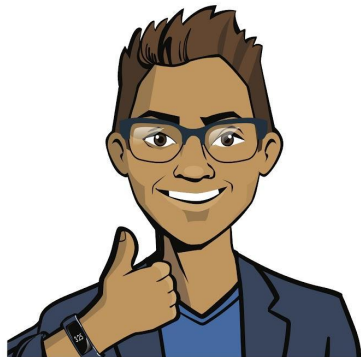
Let's take a look at the commands for handling keyboard events.

<i>Command</i>	<i>Purpose</i>
<code>pygame.event.get()</code>	A set of events that occur during a given frame of the loop.
<code>event.type</code> / <code>event.key</code>	Event type / Event name (they have highlighted names).

Examples of event types:

KEYDOWN — "the key is pressed down".

KEYUP — "the key is raised" (not pressed).



Brain
storm

Handling keyboard events

Let's take a look at the commands for handling keyboard events.

<i>Command</i>	<i>Purpose</i>
<code>pygame.event.get()</code>	A set of events that occur during a given frame of the loop.
<code>event.type</code> / <code>event.key</code>	Event type / Event name (they have highlighted names).

```
for event in pygame.event.get():  
    if event.type == pygame.KEYDOWN:  
        if event.key == pygame.K_q:
```

Action

"If there is an **event** with a **key** pressed down among the **current events** and this **key** is **Q**, then perform the action."

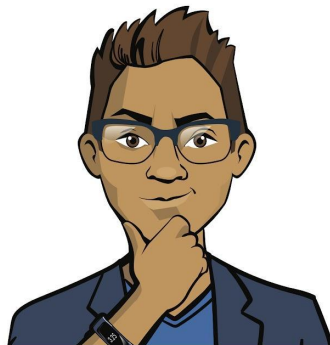


Brain
storm

Random questions and answers

It's required that a random question should be displayed when you press Q, and a random answer should be displayed when you press A.

1. How do we save the wording for questions and answers?
2. How do we select and display a random formulation?



Brain
storm



Random questions and answers

A possible solution to the problem for the questions block:

- When pressing **Q**, you can generate a random number from 1 to 3:
 - ◆ if you get the number 1, set it to 1 in the block;
 - ◆ if you get 2, set it to 2;
 - ◆ if you get 3, set it to 3.
- Re-render the block with a different question.

Come up with a solution for the answers block yourself.



Brain
storm

Diagram for the "Questions and

Connect Pygame modules

Create a scene with the color you want

Create a game timer

Description of the TextArea class

Create blocks for questions and answers

Game loop:

Handling keyboard events:
label changes when pressing Q and A.

Set the frame rate to ~40 FPS

Update the scene
(next frame of the game loop)



Brain
storm

Your tasks:

- Supplement the game loop with handling for Q and A keystrokes.
- Add at least 3 formulated questions and answers to the game.
- Launch and test the game.



Brain
storm