Checking qualifications

Demonstrate your knowledge of:

- → creating classes;
- → the basics of Pygame;
- → lists and their methods.

Checking qualifications

How do you create a game scene ?

How do you set a background color for it?

How can you set a specific color?

Checking qualifications

A game scene with a colored background

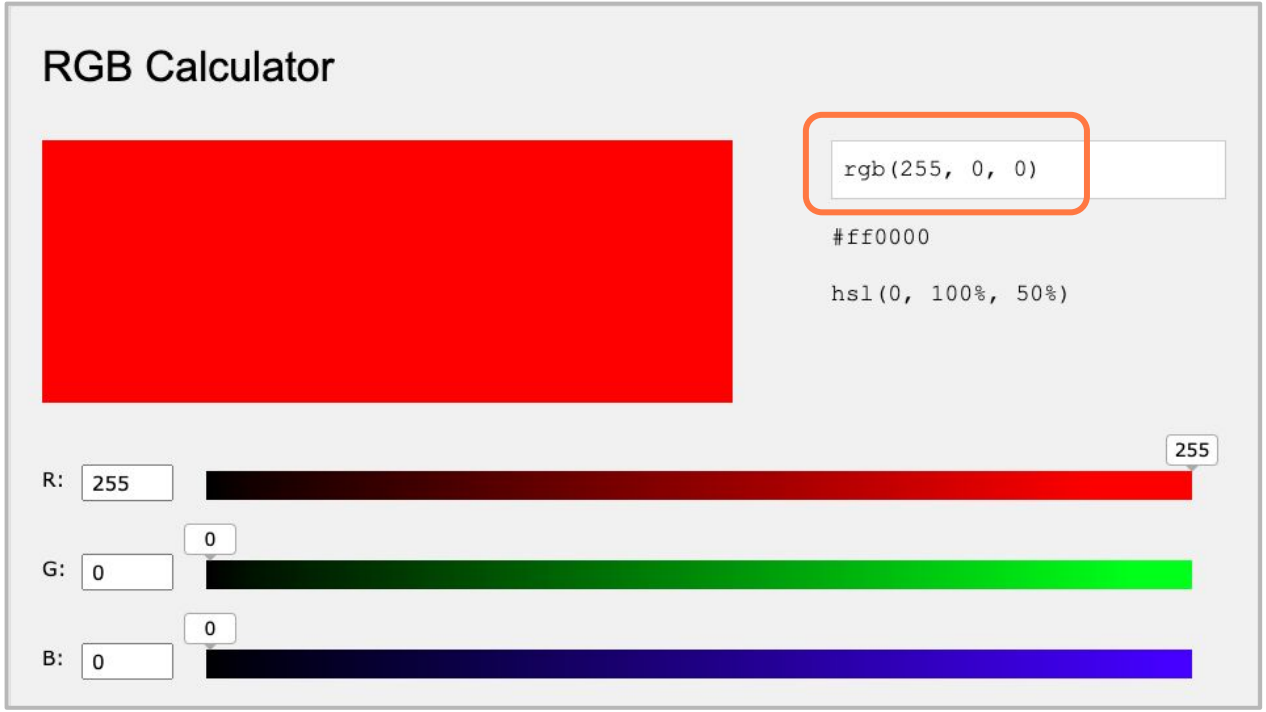
Command	*Purpose*
`window = display.set_mode((500, 500))`	Creates a window with the following size: (width, length).
`window.fill(<color>)`	Fills the background with the specified color
`pygame.display.update()`	Updates the content of the game window
`clock = pygame.time.Clock()`	Creates a game timer.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

<u>Note</u>. The color can be set using the RGB palette.

Checking qualifications

RGB color palette (red, green, blue)

You can set the shade you want by mixing the red, green and blue colors.



Link to the RGB color calculator

Checking qualifications

What is a game loop ?

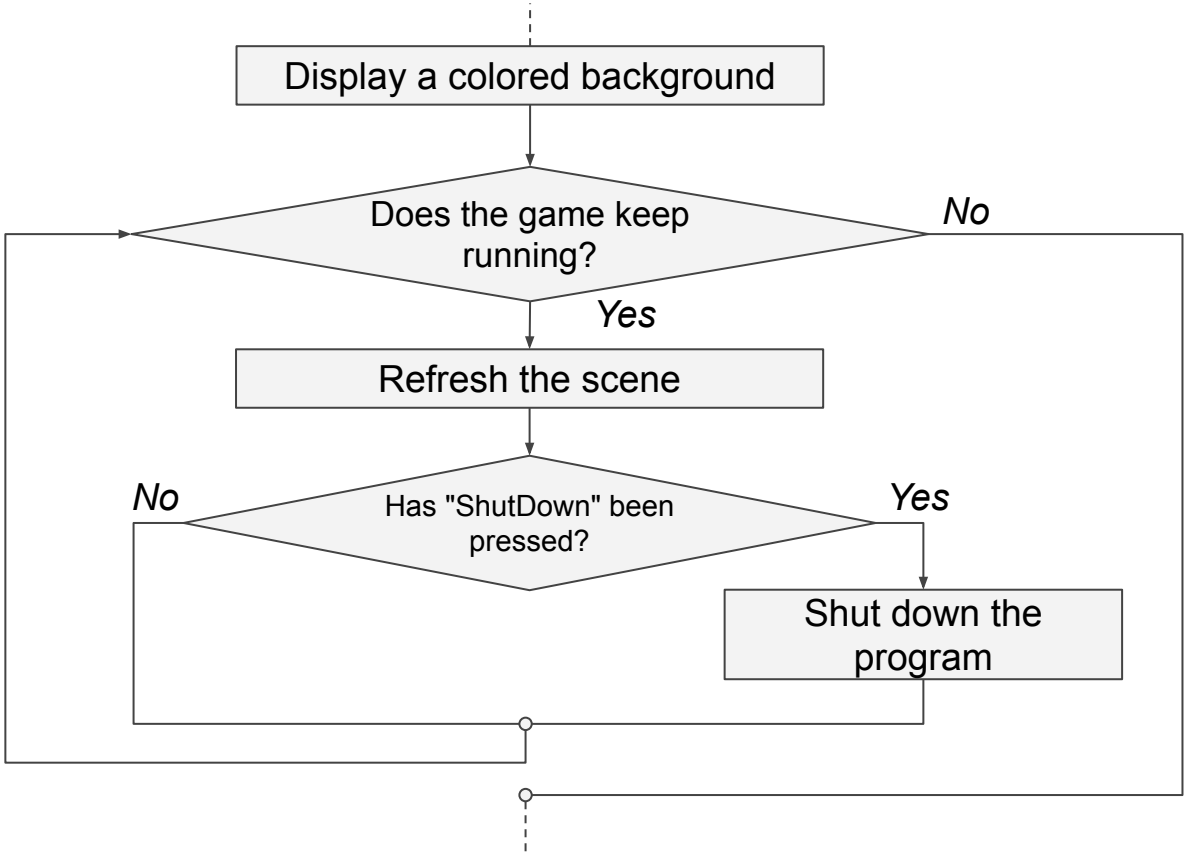
How do we create one?

What will be the condition for its termination ?

Checking qualifications

The flowchart for a game loop

The loop ends when you click on the "Shutdown" button



A game loop for a scene 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

Create a game timer

Game loop:

Set the frame rate to ~40 FPS

Update the scene
(next frame of the game loop)

Checking qualifications

What is a class?

How do we create a class ?

Checking qualifications

Creating classes

To create a class, you need to:

- list the **properties** that define the features of a class instance in the constructor;
- list the **methods** for managing the instance.

```
class [Class name]():
    def __init__(self, [Value]):
        self.[Property name] = [Value]
    def [Method name](self):
        [Action with the object and properties]
        [Action with the object and properties]
```

A special function of the **constructor** that creates an instance of a class with the specified properties.

__init__

Two lower underscores.

Checking qualifications

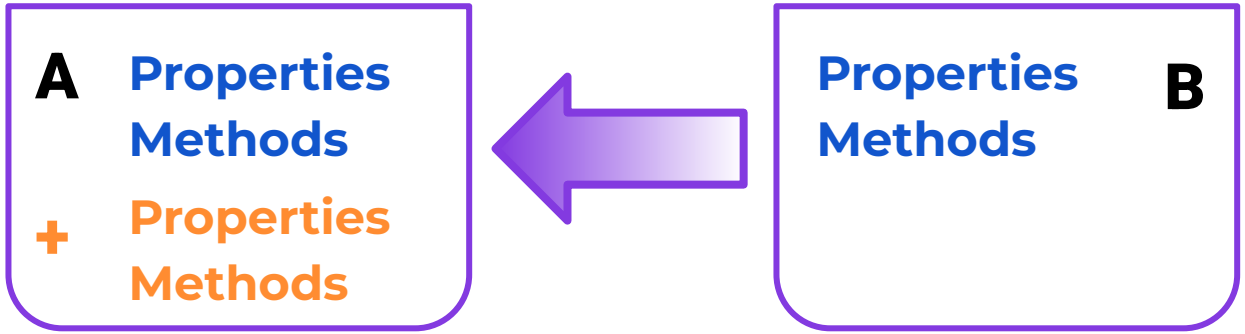
What does inheritance mean?

How can you create a derived class of an existing class?

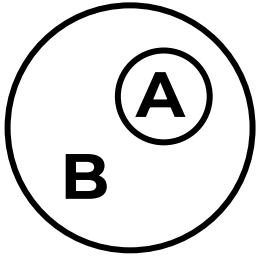
Checking qualifications

Inheritance

Class inheritance helps transfer all the skills previously written for a more general class into another, more private class, the derived class.



Derived class **Superclass**



Class A is nested within class B

Superclass and derived class

Supposing the superclass has already been written, to create a derived class we then need to:

- when creating a derived class, specify the name of the superclass;
- add the required methods to the derived class.

```
class Derived class name( Superclass name ):
    def __init__(self, Value ):
        super().__init__( Value )
    def Method name (self):
        Action with the object and properties
```

*A variant where **new properties are not introduced**.*

*The derived class is only being supplemented with a **new method.***

Checking qualifications

What is a list?

How can you create a list and fill it with elements?

What other methods of working with lists do you know?

Checking qualifications

A list is a structure for storing different types of data in a well-ordered manner .

Example. A list of results from a tournament of the online game "Space Shooter".

```
results = [181, 176, 160, 178, 171, 179, 165]
```

```
print('Best result:',results[0])
```

→ Best result: 181

Get a list item by its number (index).

Checking qualifications

Working with lists

Command	*Purpose*
`participants = list()`	Declare an empty list
`participants.append('Smith')`	Add an item to the end of the list
‘Johnson’ `in participants`	Search for the occurrence of an element in the list (returns True or False)
`participants.sort()`	Sort the list in lexicographic order (by ascending numbers and letters of the alphabet)
`for result in results:` Command1 Command2	Iterate through the `results` list items. “For each item (`result`) of the list (`results`), execute Command1, Command2”
`len(results)`	Determining the length of the `results` list

Checking qualifications

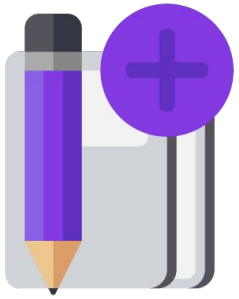
Qualifications confirmed!

Great, you are ready to brainstorm and complete your work task!

Checking qualifications

Brainstorm:

Template for the game Fast Clicker



Let's create a game template

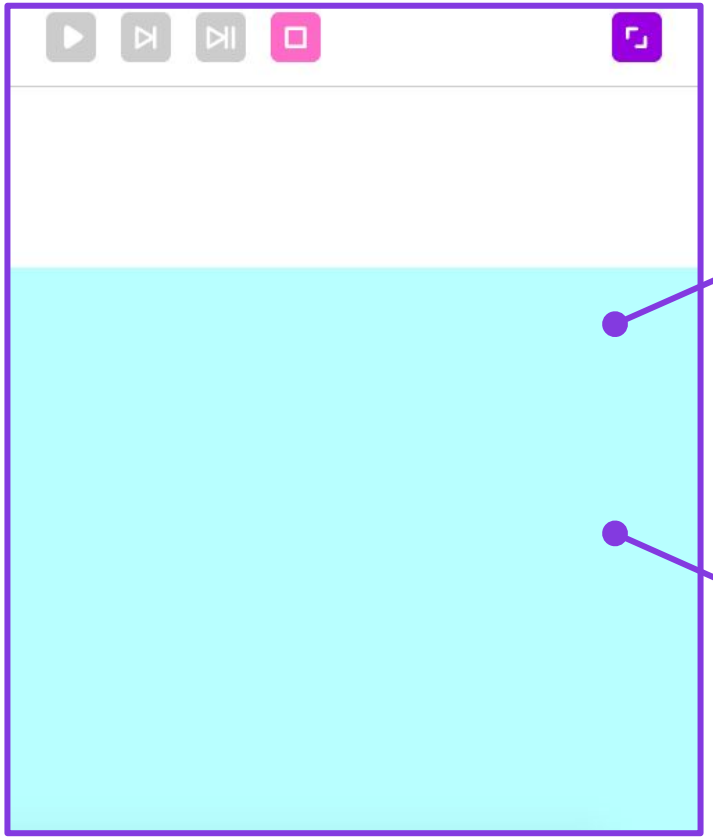
The result will be a game scene with a background and cards.

Checklist

- [ ] Create a game scene with a size of 500×500
- [ ] Set the scene's background color using the RGB palette
- [ ] Create a game timer (pygame.time.Clock () object).
- [ ] Create a game loop and set the frame rate to ~ 40 fps

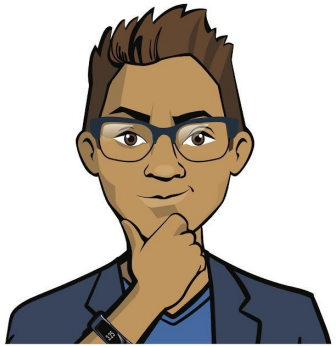
Brain storm

How do you program the template for the game Fast Clicker?



Color blue (200, 255, 255)

Scene size 500x500



Brain storm

Program flowchart:

Result: a game scene filled with color.

Connect Pygame modules

Create background objects

Fill the scene with color

```
back = (200, 255, 255)
mw = pygame.display.set_mode((500, 500))
mw.fill(back)
```

Create a game timer

Game loop:

Refresh rate ~40 FPS

Update the scene

Brain storm

Program flowchart:

Result: a game scene filled with color.

Connect Pygame modules

Create background objects

Fill the scene with color

Create a game timer → `clock = pygame.time.Clock()`

Game loop:

- Refresh rate ~40 FPS
- Update the scene

Brain storm

Program flowchart:

Result: a game scene filled with color.

Connect Pygame modules

Create background objects

Fill the scene with color

Create a game timer

Game loop:

Refresh rate ~40 FPS

Update the scene

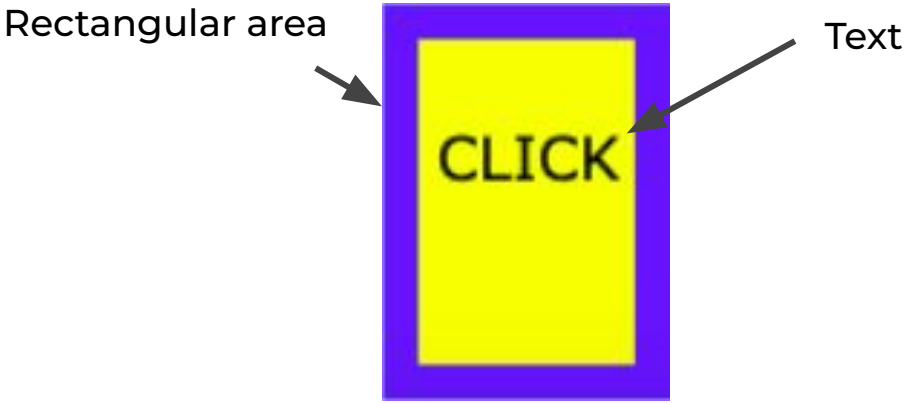
The loop ends when you click on the button

```
while True:
    pygame.display.update()
    clock.tick(40)
```

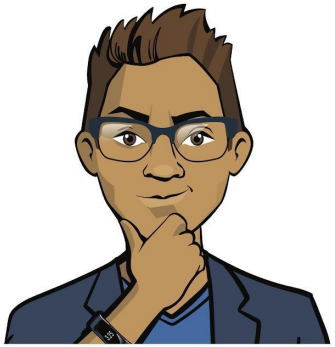
Brain storm

Creating a card sprite

According to the terms of reference, we need to create sprite cards:



Maybe we can use the ready-made TextArea class from the last project?



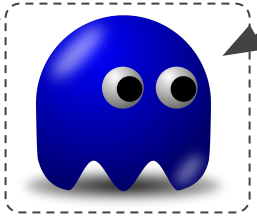
Brain storm

Creating a card sprite

Rectangular areas are the basis for lots of sprites!

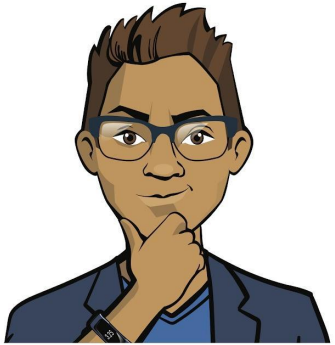
Rectangular area

Picture



Used to track object coordinates

Responsible for external representation



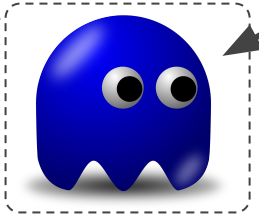
Brain storm

Creating a card sprite

Rectangular areas are the basis for lots of sprites!

Rectangular area

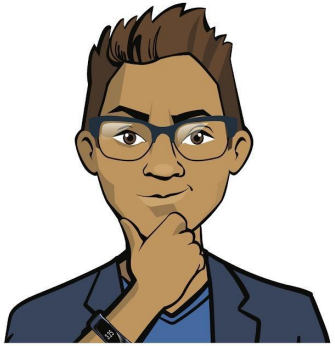
Picture



Used to track object coordinates

Responsible for external representation

A rectangular area can be either colored or without color.



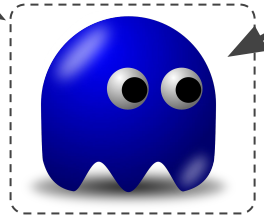
Brain storm

Creating a card sprite

Rectangular areas are the basis for lots of sprites!

Rectangular area

Picture



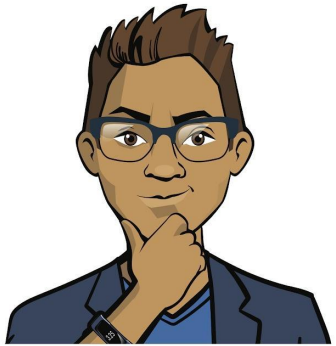
Used to track object coordinates

Responsible for external representation

Rect property

Image property

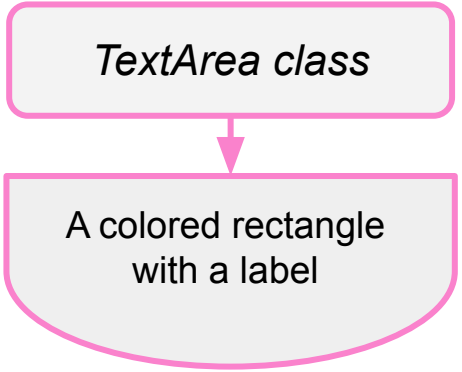
Regardless of what type of sprite it is, it must have these parameters.



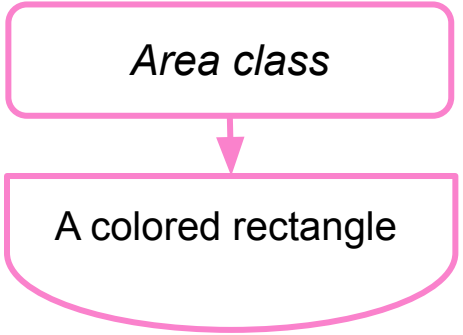
Brain storm

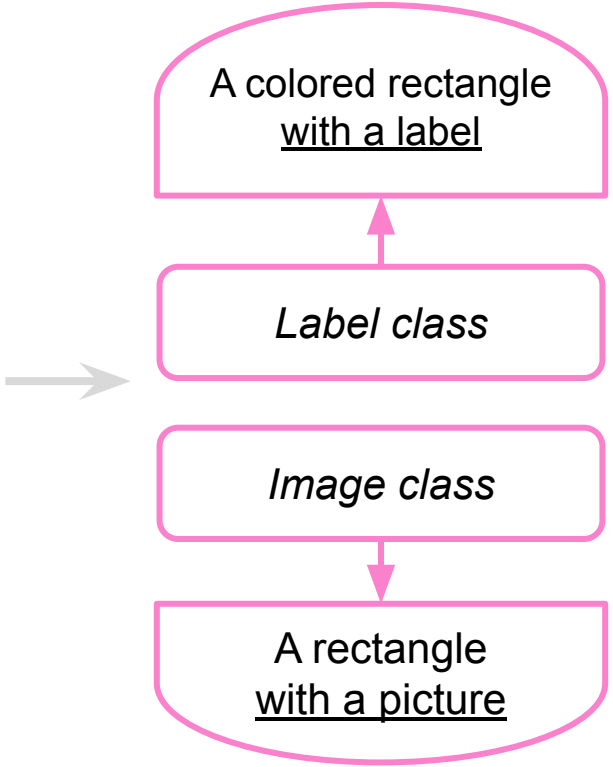
Creating a card sprite

Rectangular areas can also be used in other projects.



It was implemented for the last project

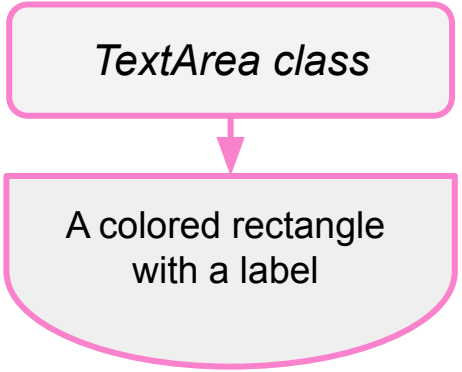




It can be used in other projects as well

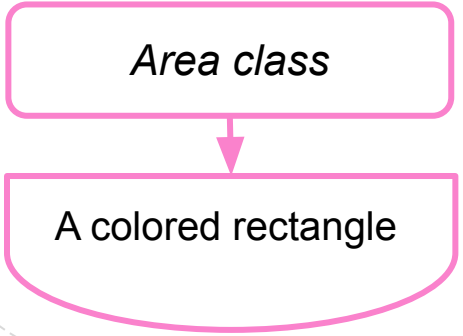
Creating a card sprite

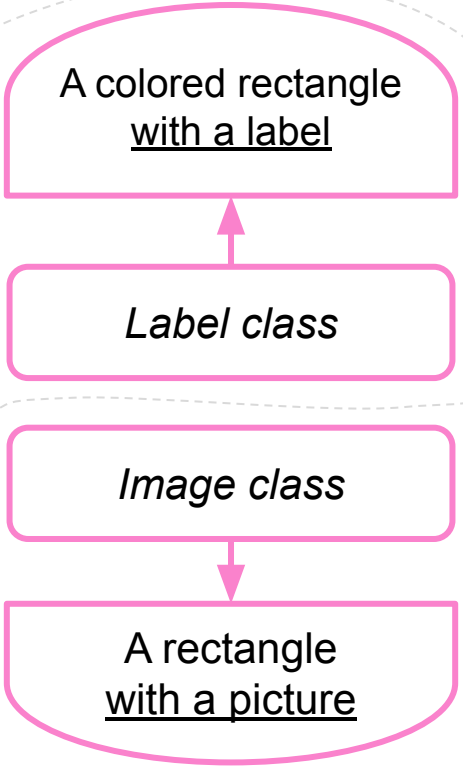
Rectangular areas can be used in other projects as well



It was implemented specifically for the last project

Let's use this idea in Fast Clicker!



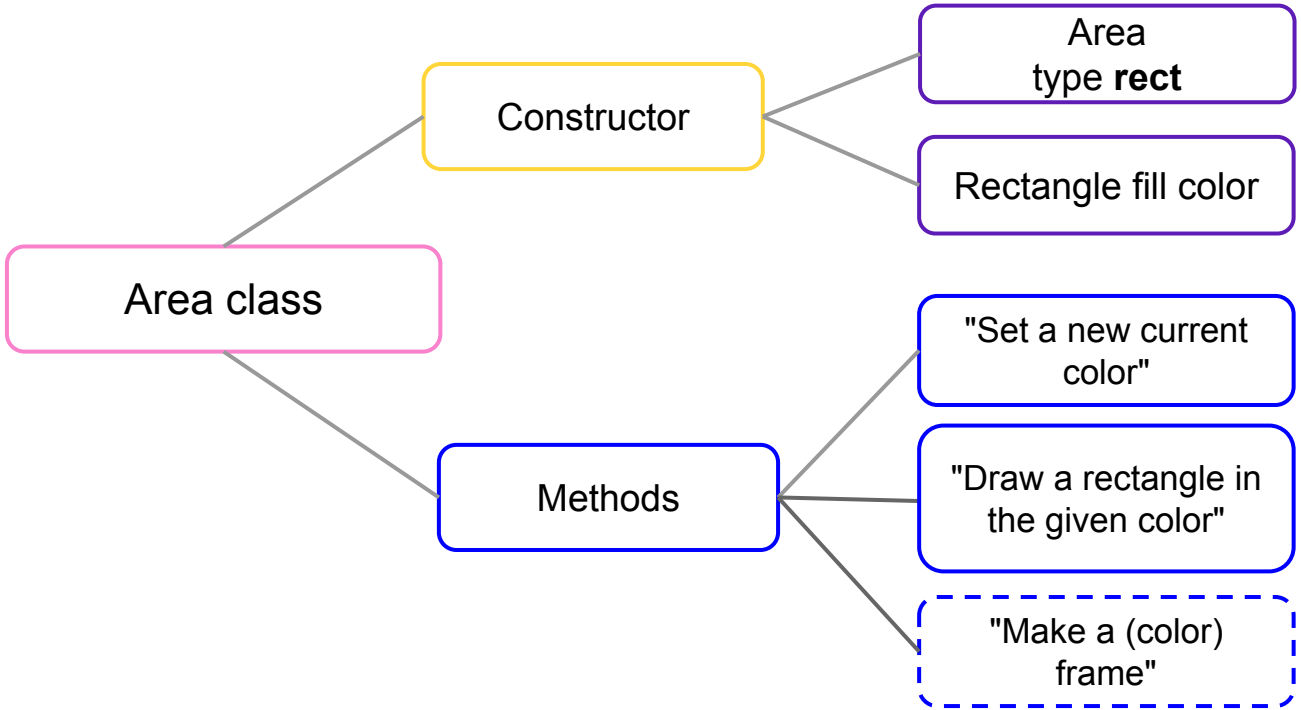


It can be used in other projects as well

Area class

In Area, we describe the rectangular area.

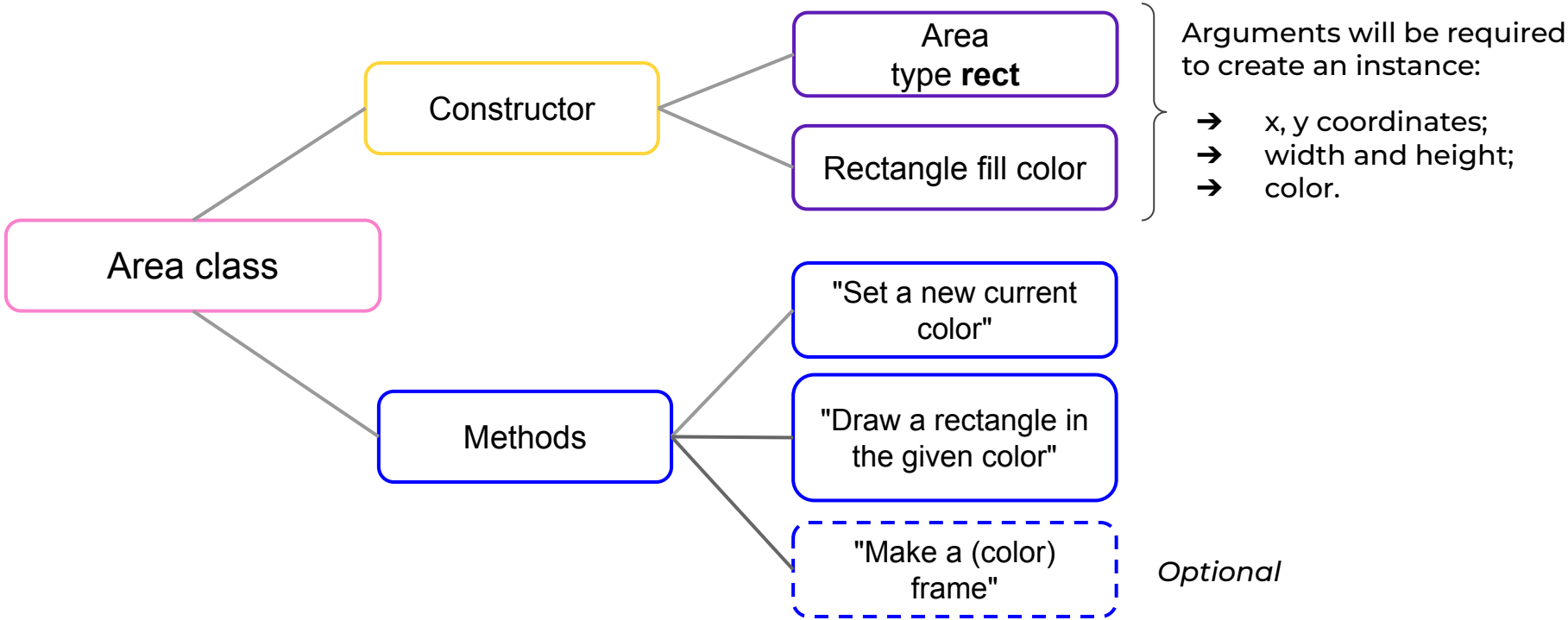
It can be an independent sprite or the frame of a more complex object.



Area class

In Area, we describe the rectangular area.

It can be an independent sprite or the frame of a more complex object.



Area class

In Area, we describe the rectangular area.

It can be an independent sprite or the frame of a more complex object.

```
class Area():
  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 color(self, new_color):
      self.fill_color = new_color
  def fill(self):
      pygame.draw.rect(mw, self.fill_color, self.rect)
  def outline(self, frame_color, thickness):
      pygame.draw.rect(mw, frame_color, self.rect, thickness)
```

The method will be called in the derived Label class to draw both the rectangle and the label on it.

Program flowchart:

Result: a game scene filled with color.

Connect Pygame modules

Background design

Create a game timer

*Create the **Area** class*

Game loop:

Refresh rate ~40 FPS

Update the scene

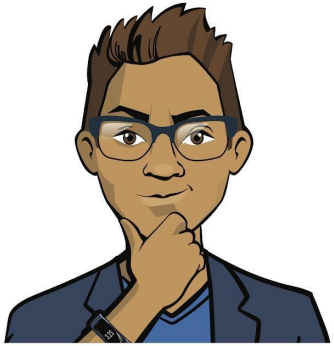
Displaying the card will be done during the second half of the working day.

Brain storm

Your tasks:

1. Program the game template with a colored background.
1. Implement the Area class.

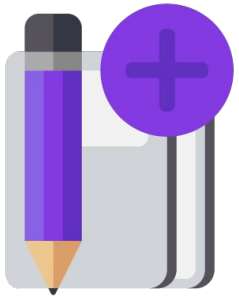
If there is time left over, choose a place to put the card sprites in the scene.



Brain storm

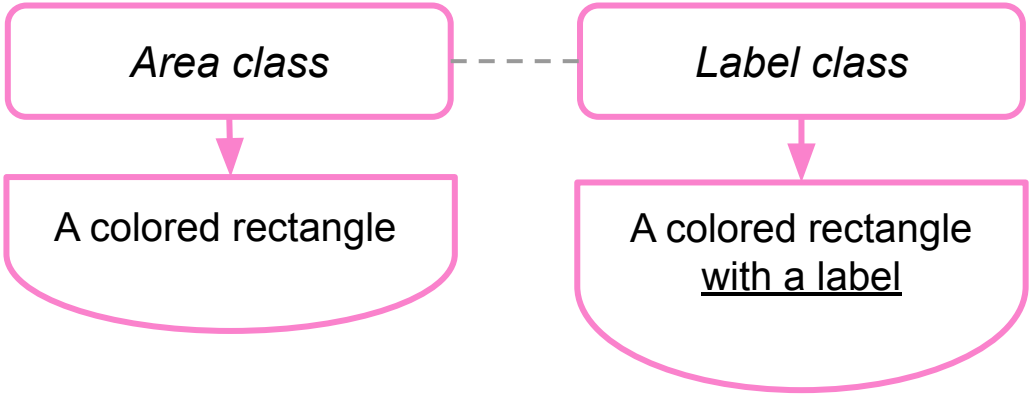
Brainstorm:

Card sprites



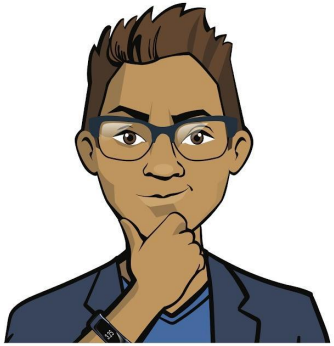
Creating a card sprite

Let's continue creating the card sprite:



The Area class has already been implemented.

Let's program the derived Label class.

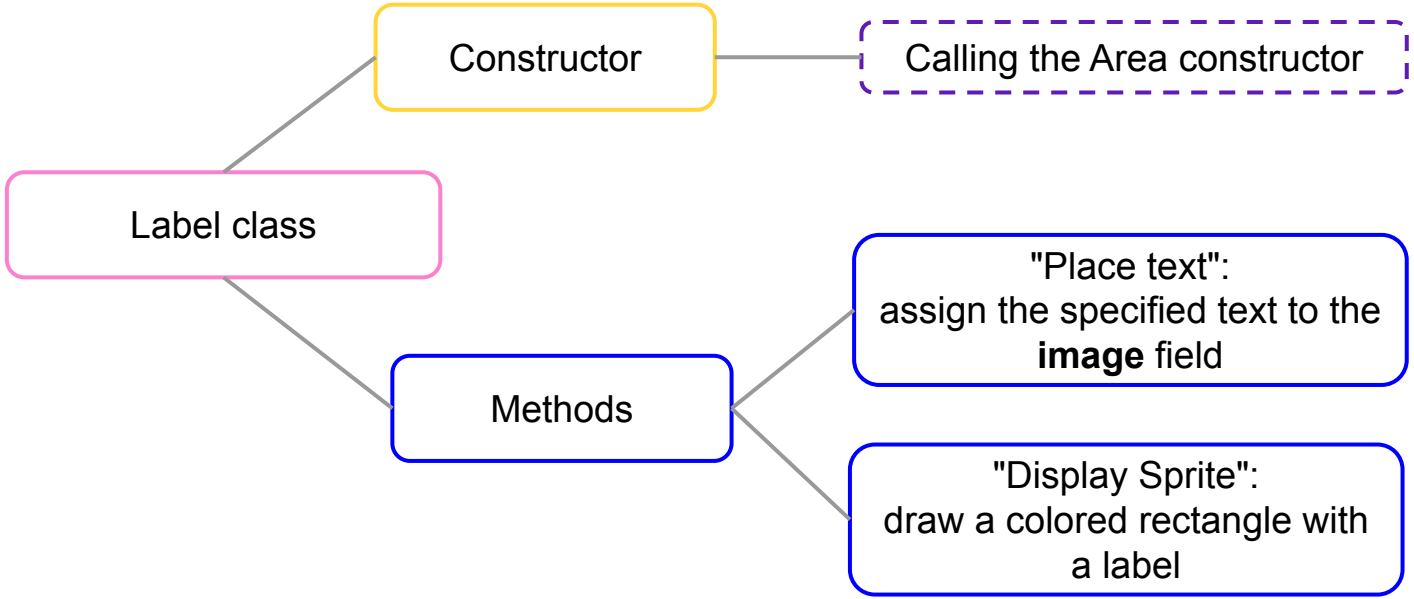


Brain storm

The Label class is the derived class of the Area class

The Label class is derived from the Area class and supplemented with text:

New properties are introduced in the methods if necessary.



Brain storm

Working with text

You already know how to work with text. But a different command will be required for non-standard fonts.

Command	*Purpose*
`font1 = pygame.font.Font(None, 70)`	Set the font / Create a font object with the parameters: font — default, size — 70.
`font2 = pygame.font.SysFont('verdana', 70)`	Set the font / Create a font object with the parameters: font — Verdana, size — 70.

Brain storm

Working with text

You already know how to work with text. But a different command will be required for non-standard fonts.

Command	*Purpose*
`font1 = pygame.font.Font(None, 70)`	Set the font / Create a font object with the parameters: font — default, size — 70.
`font2 = pygame.font.SysFont('verdana', 70)`	Set the font / Create a font object with the parameters: font — Verdana, size — 70.

```
image = pygame.font.SysFont('verdana', fsize).render(text, True, text_color)
```

↓

"Create a text label in the text_color color with the Verdana font and size fsize."

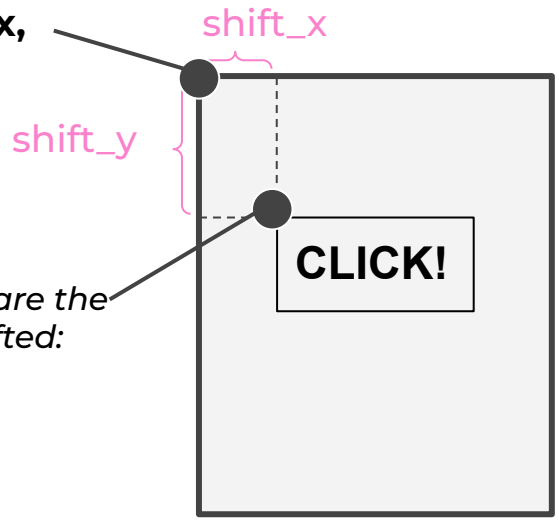
Brain storm

How can you display the text in the center of the card?

When rendering objects, it's important to remember that in Pygame, the upper-left corner is considered the beginning of drawing.

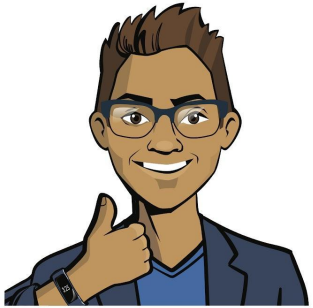
The text rendering point can be calculated if you know the card rendering point:

*The **rect** coordinates are already known:* **(x, y)**



*The **image** coordinates are the rect coordinates but shifted:*

(x + shift_x, y + shift_y)



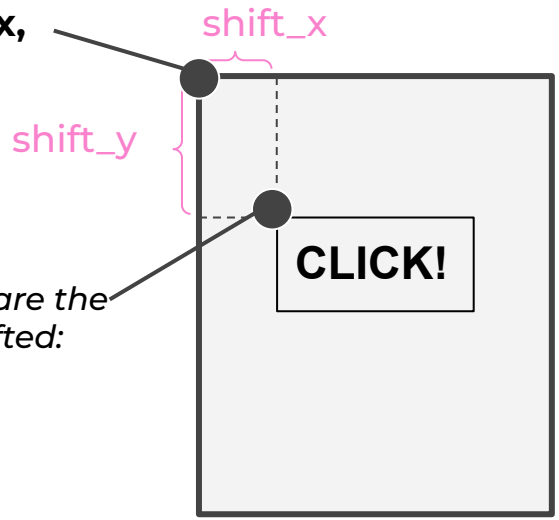
Brain storm

How can you display the text in the center of the card?

When rendering objects, it's important to remember that in Pygame, the upper-left corner is considered the beginning of drawing.

The text rendering point can be calculated if you know the card rendering point:

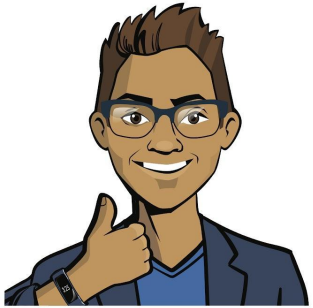
*The **rect** coordinates are already known:* **(x, y)**



*The **image** coordinates are the rect coordinates but shifted:*

(x + shift_x, y + shift_y)

shift_x and shift_y will be passed to the draw() method as arguments.



Brain storm

Creating and displaying a card

A card with text is an instance of the Label class.

The card is drawn in several stages:

1. Creating the color base of the card.
2. Setting the text — this is the set_text() method.
3. Drawing the card (in the game loop) — this is the draw() method:

   ➔ a colored rectangle is drawn — this is the fill() method of the Area class;
   ➔ the finished text is displayed on it.

Brain storm

Creating and displaying a card

There is a set of cards in the scene.
The word "Click" appears for a split second on one of the cards.

*How do I display "Click" on a **random** card?*

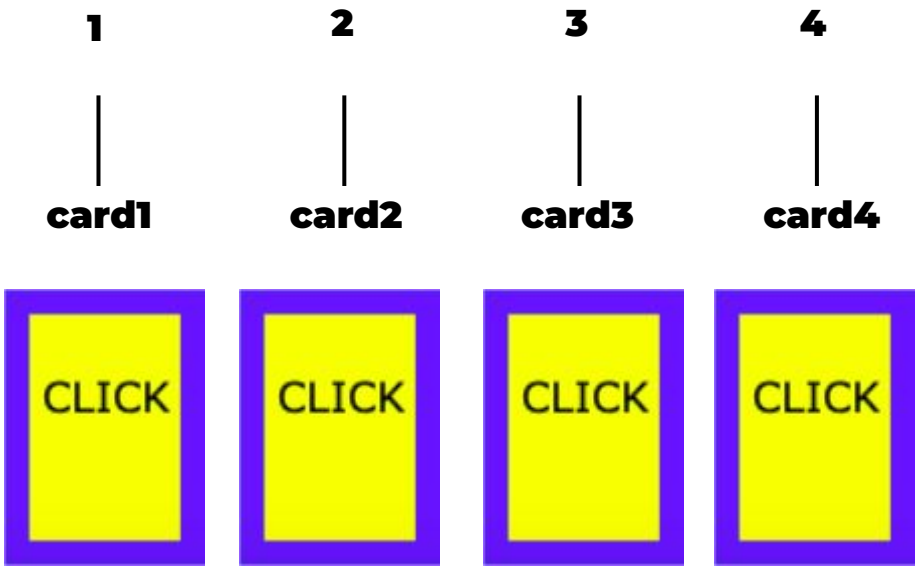
Randomly displaying "Click" is not the current task, but it might be worth thinking about it for the future.

Brain storm

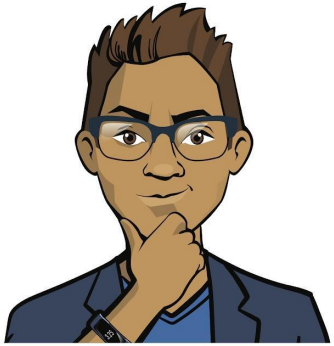
Creating and displaying a card

There is a set of cards in the scene.
The word "Click" appears for a split second on one of the cards.

*How do I display "Click" on a **random** card?*



The cards can be numbered, and then you can "search" for the generated number and the corresponding object in the conditional statement...

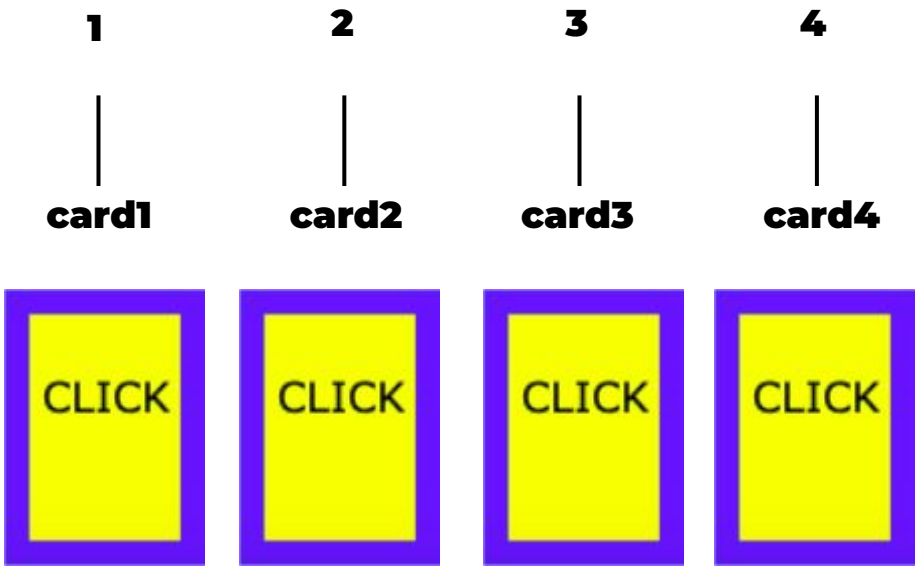


Brain storm

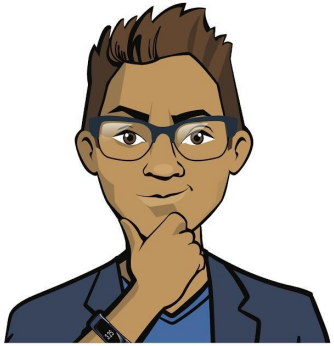
Creating and displaying a card

There is a set of cards in the scene.
The word "Click" appears for a split second on one of the cards.

*How do I display "Click" on a **random** card?*



But we already did this in the "Questions and Answers" game, and iterating took a very long time!



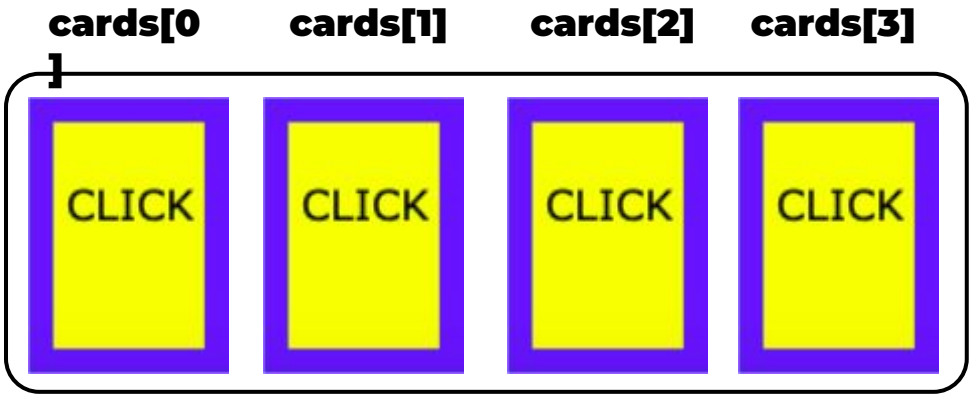
Brain storm

Creating and displaying a card

There is a set of cards in the scene.
The word "Click" appears for a split second on one of the cards.

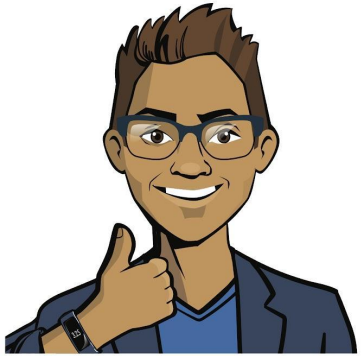
Let's define a set of cards as a list.

num = randint(0, 3) → **cards[num]**



List of cards

In the list, you can access a card or all the cards in turn by index.



Brain storm

Create a set of cards

When you create cards, they are added to the **cards** list.

The main part of the program:

Create an empty cards list

Input the number of cards num_cards = 4

Set the starting value of X

```
for i in range(num_cards):
```

Create a new yellow card at the point (X, Y) with a width of W and a height of H

Set the CLICK text for the card

Add the card to the cards list

Increase the X coordinate by 100

Brain storm

Program flowchart:

The result will be a game scene with a set of cards. All of them will have "Click" written on them.

Connect Pygame modules

Create a background and a timer

*Create the **Area** class*

*Create the **Label class***

Create a set of cards

Game loop:

Display all cards from the set

Frame rate ~40 FPS, scene refresh

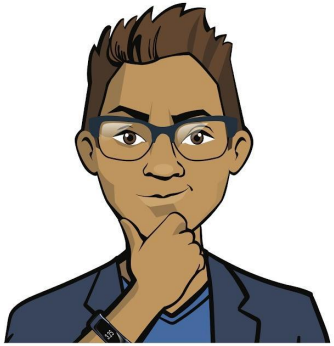
```
while True:
    for card in cards:
        card.draw(10, 30)
```

Brain storm

Your tasks:

1. Implement the Label class.
2. Create a set of 4 cards and display them in the scene.

If you have time left, think about how you can randomly display the CLICK label on one of the cards.



Brain storm