

Module 2. Lesson 3.

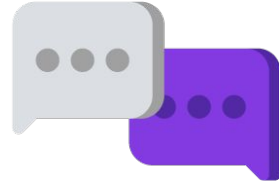
Memory Card Application

Link to the training
manual



Discussion:

Memory Card Application



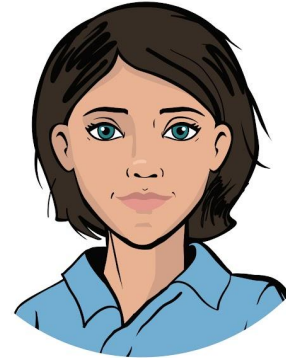
We've got a big order !

The ProTeam developers been contacted by the “Citizen of the World” cultural center.

To help their specialists memorize the cultures and languages of people all over the world, the cultural center has ordered a **Memory Card application**.

The application should ask multiple choice questions and determine whether the user answered correctly.

Ready to start?



*Emily,
Project manager*



Discussing the tasks



Let's examine our task

- Product type:
an application (for the computer)
- Functionality:
asks a random question from a set; checks the answer



Memo Card

Village

Answer options

☒ wilena ☐ woinstva
☐ woiska ☐ wioska

Answer



Memo Card

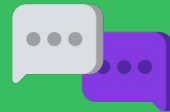
Village

Test result
Incorrect!

wioska

Answer

Discussing the tasks



Let's examine our task

- Product type:
application (for the computer)
- Functionality:
asks a random question from a set; checks the answer



Memo Card

Village

Answer options

☒ wilena ☐ woinstva
☐ woiska ☐ wioska

Answer



Memo Card

Village

Test result
Incorrect!

wioska

Answer

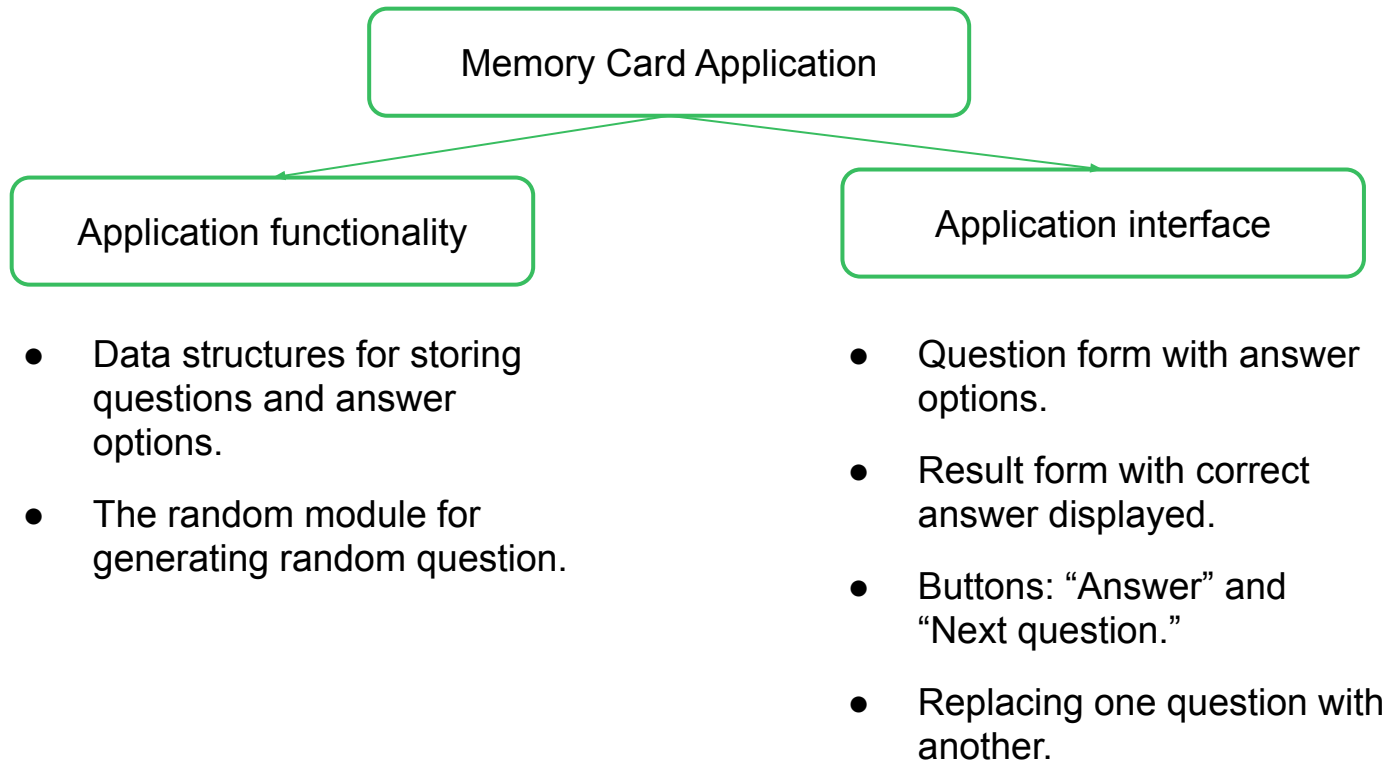
What instruments will we need for this task?



Discussing the tasks



Let's examine our task



Discussing the tasks

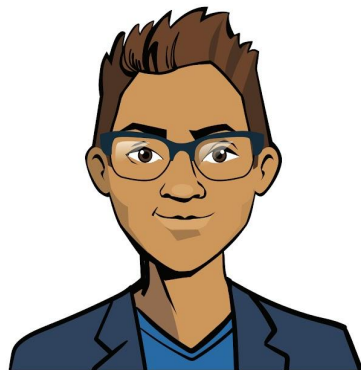


Memory Card Application



With some rare exceptions (for example, grouping radio buttons) **there are no unfamiliar topics.**

But this project will require the **comprehensive use** of various areas of programming.



Discussing the tasks



The goal for the workday —

Make a work plan for the Memory Card application and develop its interface.

Today you will :

- review the layouts and types of widgets, and how to program them using PyQt
- learn new parameters for positioning content in the application window
- create an interface for the Memory Card application



Discussing the tasks



Qualifications



Show off your knowledge of the PyQt library



Qualifications



What is a windowed application ?
What is it made up of?



Qualifications

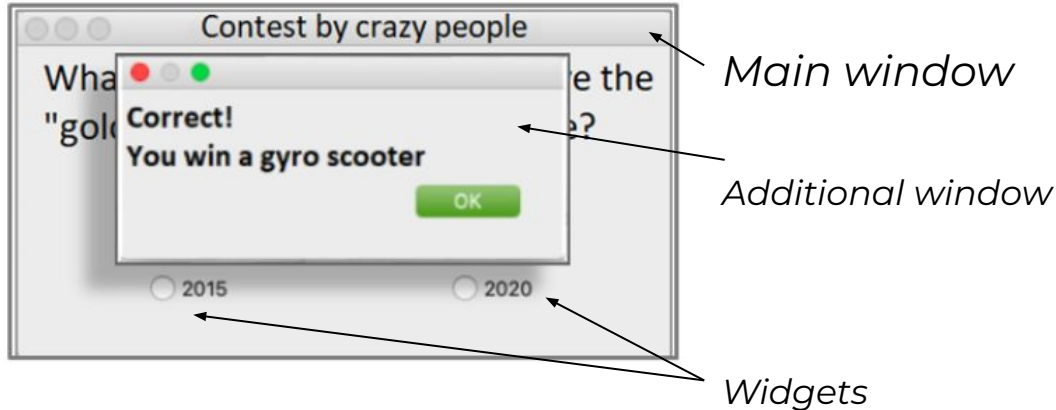


A windowed application

Is a computer program that interacts with a user through a graphical interface.

A standard windowed application is made up of:

- ❑ a main **window**
- ❑ control elements (**widgets**)
- ❑ additional windows (optional)



Qualifications



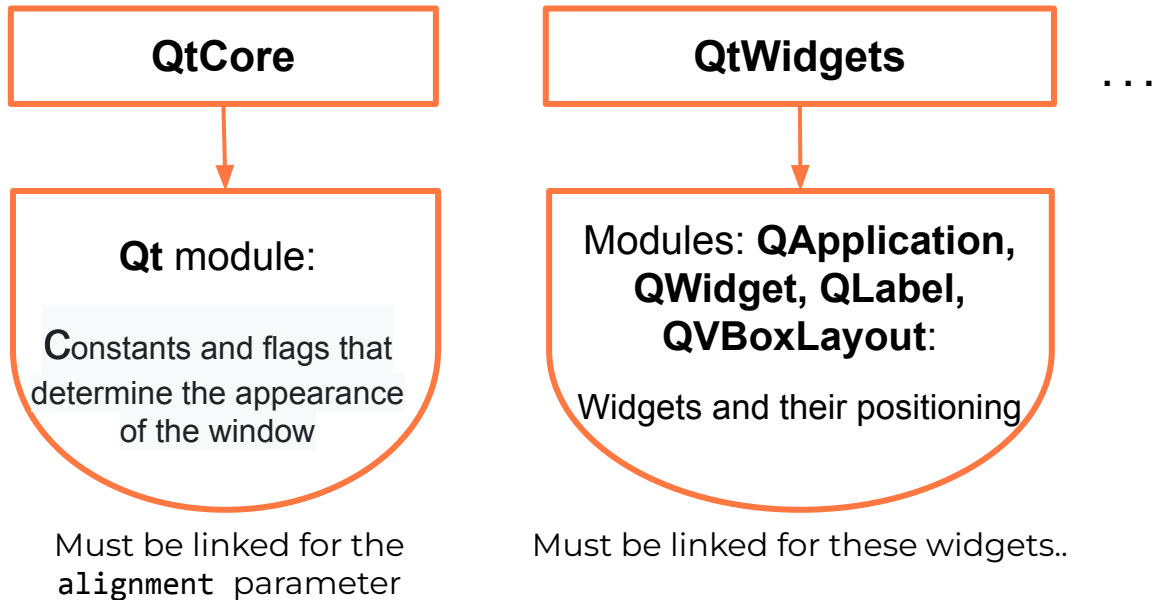
What PyQt **modules do you know? What are they used for?**



Qualifications



Linking PyQt modules



```
from PyQt5.QtCore import Qt
from PyQt5.QtWidgets import (
    QApplication, QWidget, QLabel, QVBoxLayout, QHBoxLayout, ...
)
```



Qualifications



What **widgets** do you know?

How to create a(n):

- ❑ **radio button,**
- ❑ **button,**
- ❑ **label?**



Qualifications



Popular widgets:

```
from PyQt5.QtCore import Qt
```

```
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton, QLabel, QRadioButton
```

<i>Object</i>	<i>Designation</i>
Application	<code>QApplication</code>
Application window	<code>QWidget</code>
Label	<code>QLabel</code>
Button	<code>QPushButton</code>
Radio button	<code>QRadioButton</code>



Qualifications



**What methods are used to
create **layouts** ?**

To add widgets to layouts?



Qualifications



Positioning widgets in a line

<i>Method</i>	<i>Designation</i>
<code>v_line = QVBoxLayout()</code>	A constructor that creates an object of the type “Vertical line.”
<code>v_line.addWidget(title, alignment = Qt.AlignCenter)</code>	A method that adds a widget to a line and centers it.
<code>my_win.setLayout(v_line)</code>	Add the resulting line and its objects to the application window.

A horizontal line is created the same way.

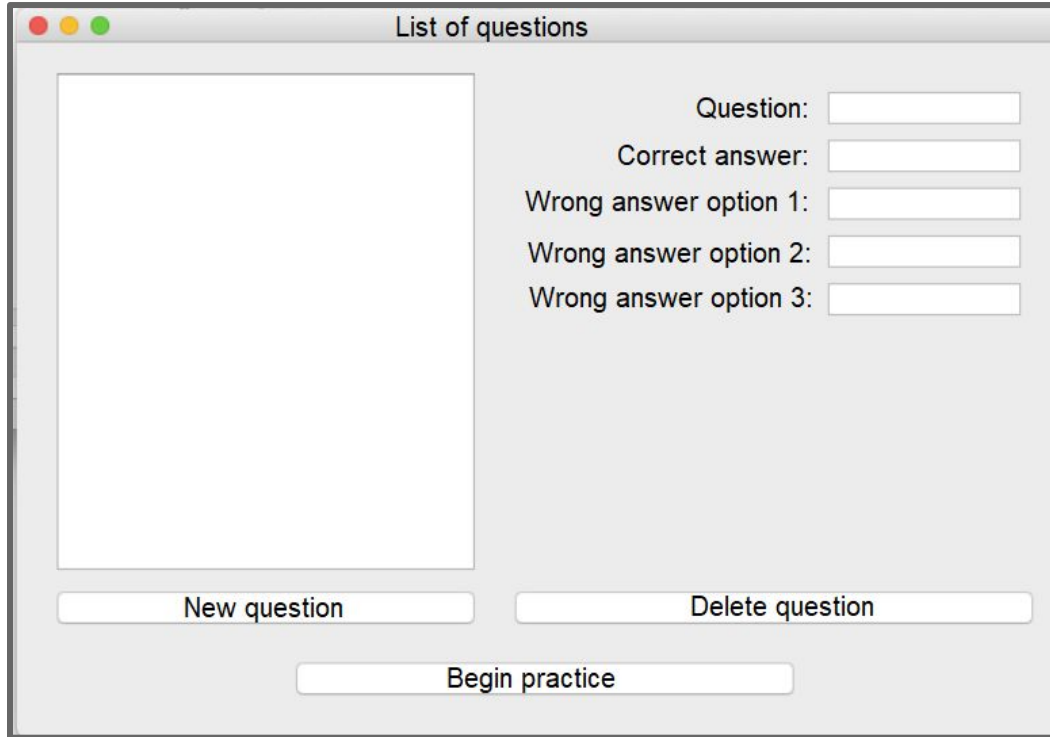
Layouts can be added to other layouts.



Qualifications



Name and show the guiding lines in this window:



List of questions

Question:

Correct answer:

Wrong answer option 1:

Wrong answer option 2:

Wrong answer option 3:

New question

Delete question

Begin practice

Several correct answers are possible!



Review



Possible answer:

Several correct answers are possible!



List of questions

Question:

Correct answer:

Wrong answer option 1:

Wrong answer option 2:

Wrong answer option 3:

Review



Qualification confirmed!

Excellent, you're ready to brainstorming and carry out this job!

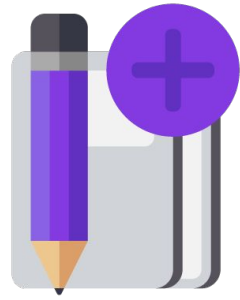


Qualifications



Brainstorming:

The Memory Card Interface

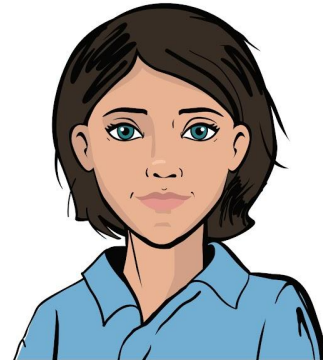


Working on a large project

Memory Card is a big project that can't be finished in one workday.

Before they start working on a large task, real designers split it into sub-tasks and come up with a **work strategy**.

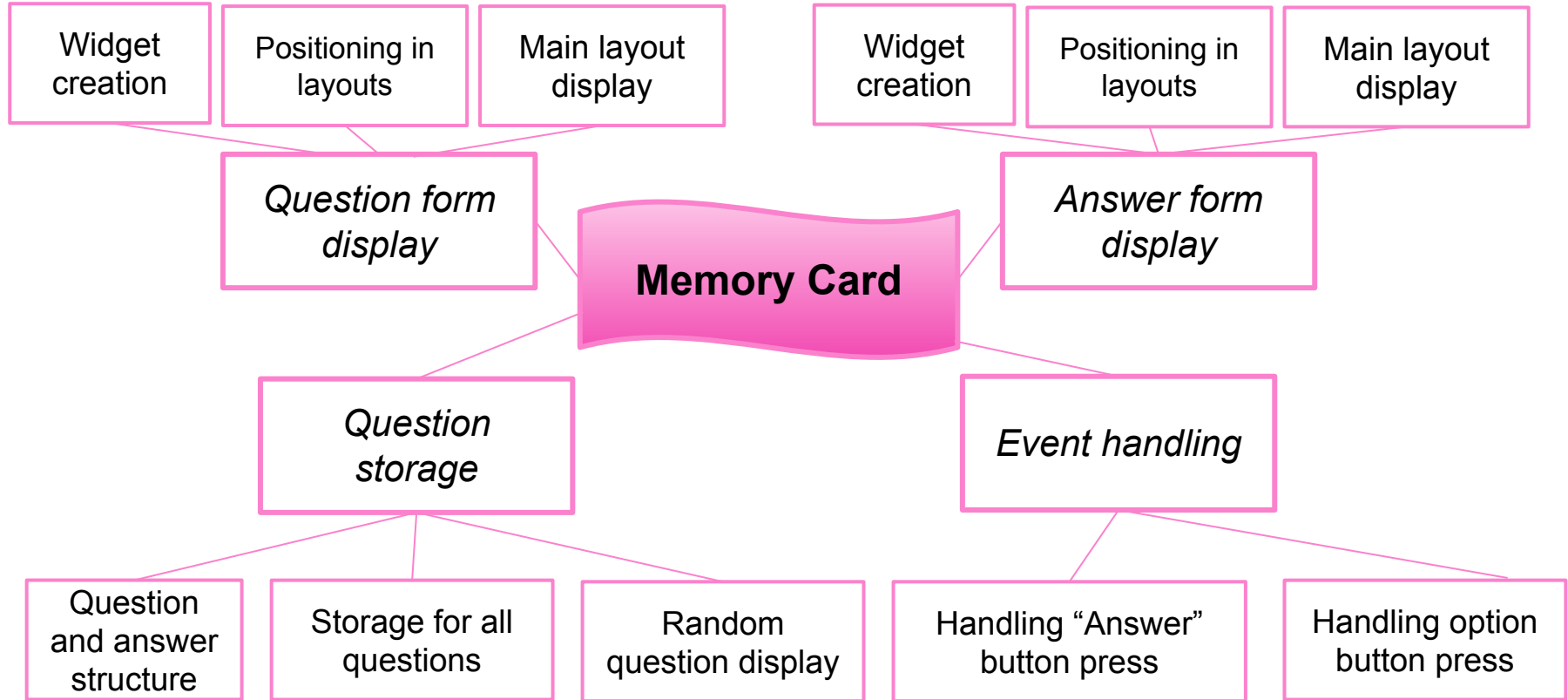
Right now, we're going to describe the expected functionality for Memory Card using a diagram, and plan our tasks for today.



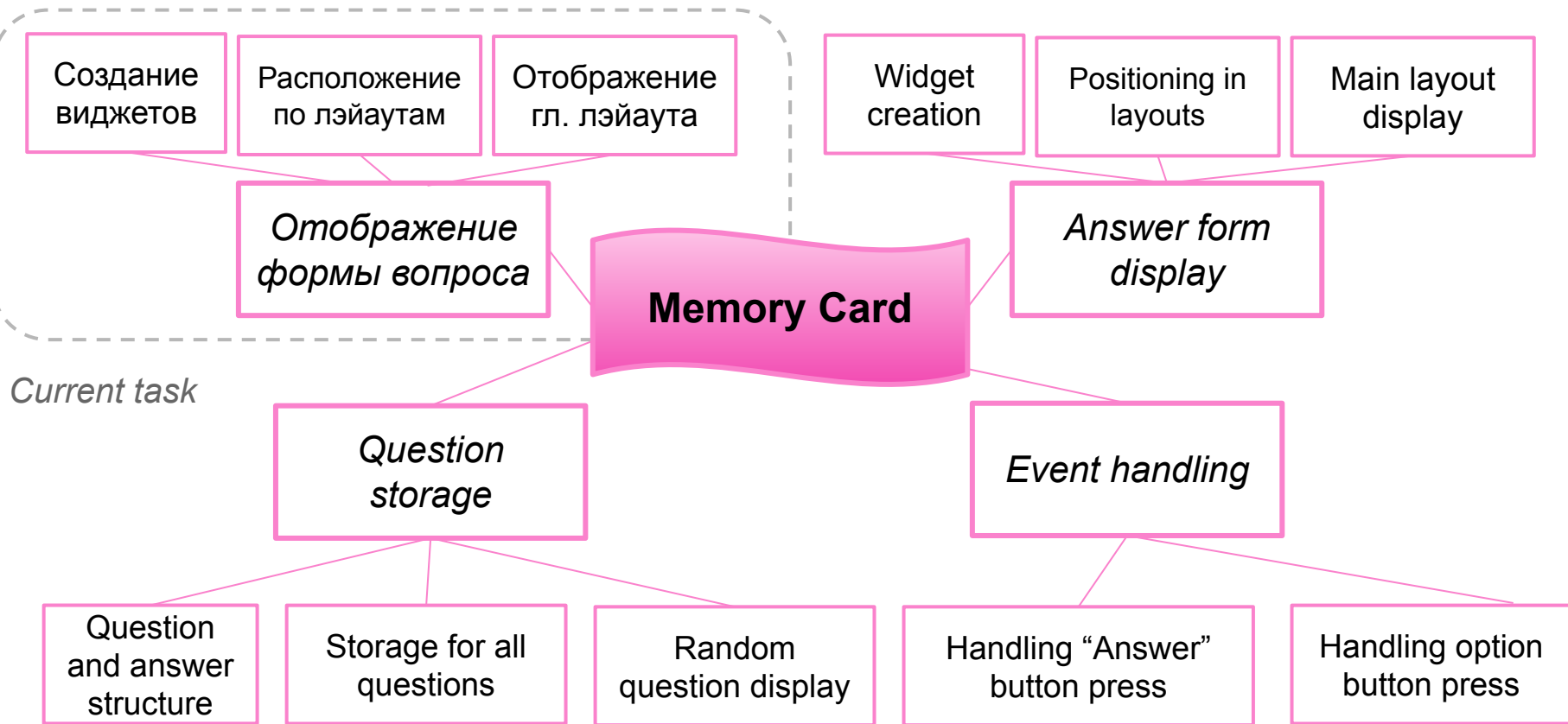
Brainstorming



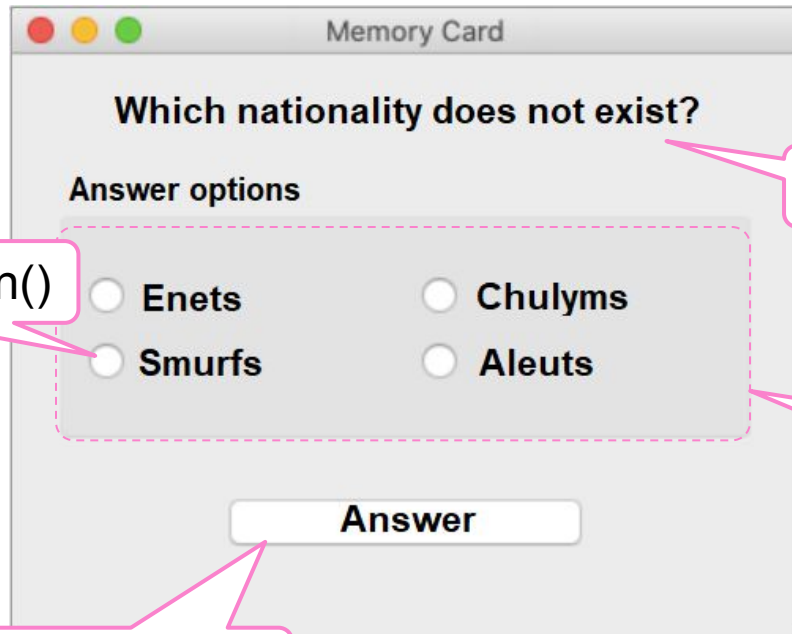
Application property diagram



Application property diagram



What widgets are in this window?



QRadioButton()

QLabel()

?

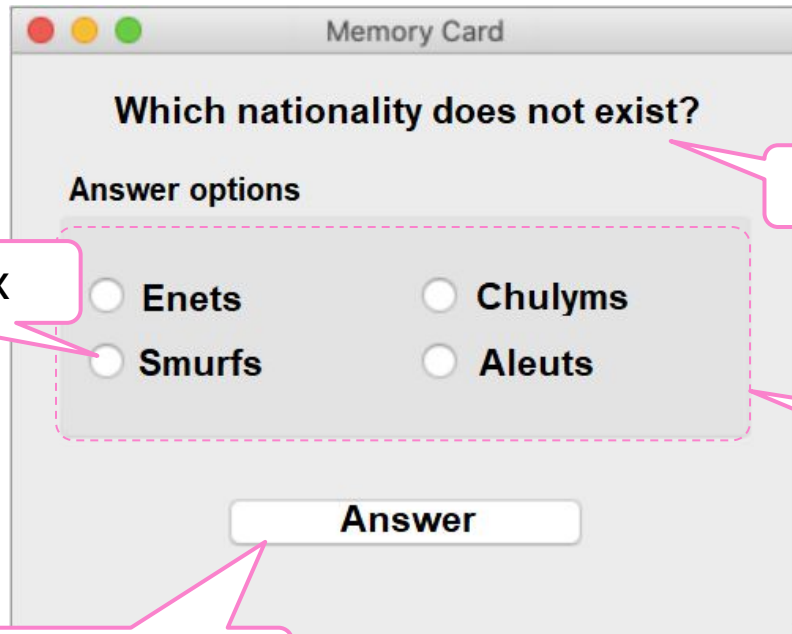
?



Brainstorming



What widgets are in this window?



QLabel

QRadioButton

QGroupBox
(?)

QPushButton



Brainstorming



The **QGroupBox** Group (QtWidgets)

<i>Method</i>	<i>Designation</i>
<code>RadioGroupBox = QGroupBox('Options')</code>	Constructor for creating a group (visibly separate)
<code>rbtn_1 = QRadioButton('Enets')</code> <code>layout_quest = QHBoxLayout()</code> <code>layout_quest.addWidget(rbtn_1)</code>	Create radio buttons, create line layout, add radio button to layout
<code>RadioGroupBox.setLayout(layout_quest)</code>	Make the layout the content of the group



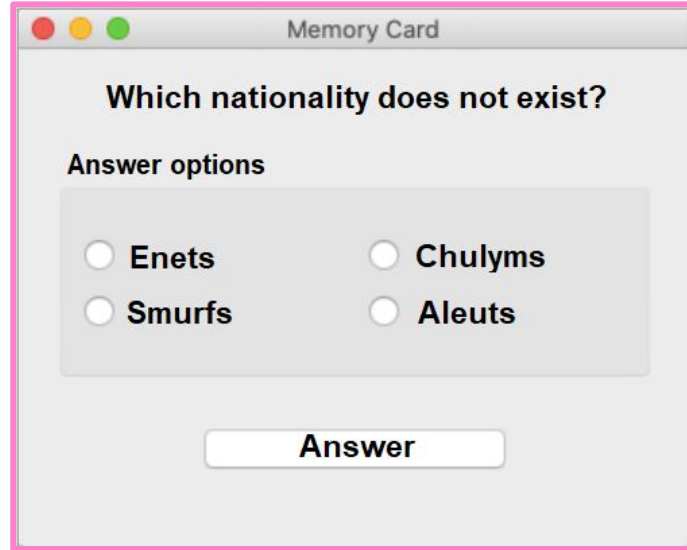
Brainstorming



```
RadioGroupBox = QGroupBox("Answer options")
rbtn_1 = QRadioButton('Enets')
rbtn_2 = QRadioButton('Smurfs')
rbtn_3 = QRadioButton('Chulyms')
rbtn_4 = QRadioButton('Aleuts')
layout_ans1 = QHBoxLayout()
layout_ans2 = QVBoxLayout()
layout_ans3 = QVBoxLayout()

layout_ans2.addWidget(rbtn_1)
layout_ans2.addWidget(rbtn_2)
layout_ans3.addWidget(rbtn_3)
layout_ans3.addWidget(rbtn_4)
layout_ans1.addLayout(layout_ans2)
layout_ans1.addLayout(layout_ans3)

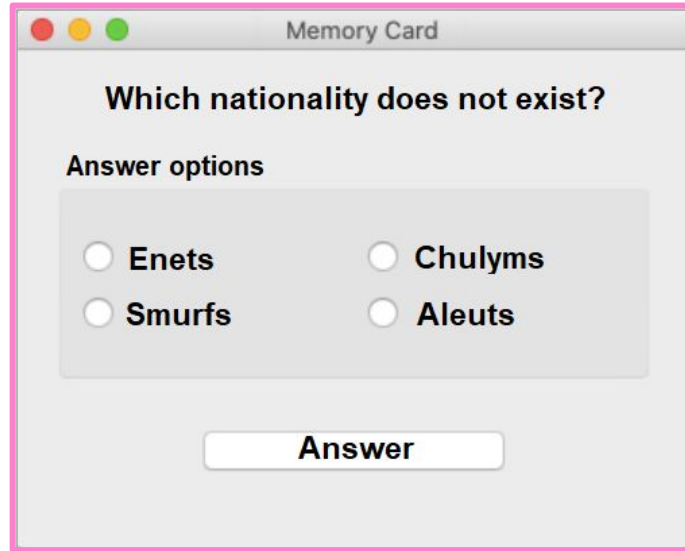
RadioGroupBox.setLayout(layout_ans1)
```



Brainstorming



How is this content positioned in the layout?



Memory Card

Which nationality does not exist?

Answer options

☐ Enets ☐ Chulymys

☐ Smurfs ☐ Aleuts

Answer



Brainstorming



One possibility:

Memory Card

Which nationality does not exist?

Answer options

1 Enets

2 Chulymys

3 Smurfs

4 Aleuts

Answer

A group of widgets with a given layout



Brainstorming



Another possibility:

Memory Card

Which nationality does not exist?

Answer options

<input type="radio"/> Enets	<input type="radio"/> Chulymys
<input type="radio"/> Smurfs	<input type="radio"/> Aleuts

Answer

Layout line 1

Layout group 2

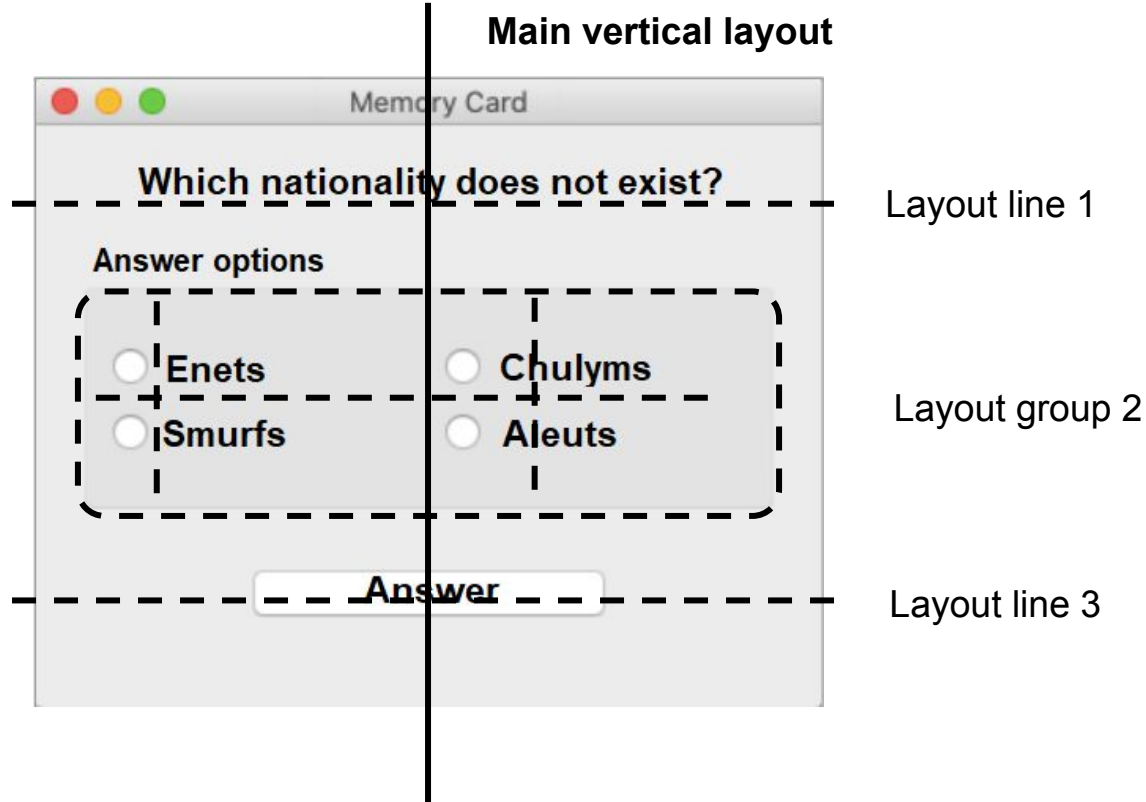
Layout line 3



Brainstorming



Another possibility:

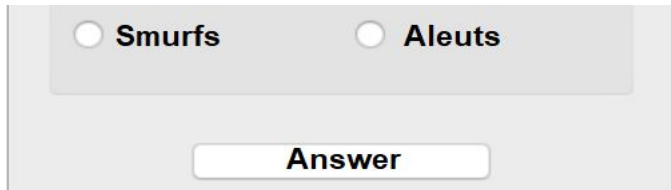


Brainstorming

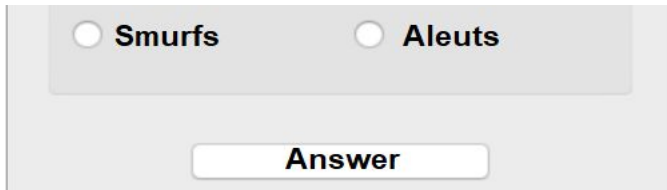


Widget placement (QtCore)

<i>Command</i>	<i>Designation</i>
<code>alignment=Qt.AlignHCenter</code> <code>alignment=Qt.AlignVCenter</code>	Align to center (horizontally) Align to center (vertically)
<code>stretch=2</code>	Stretch a widget (for example, a button)



`stretch=1`



`stretch=3`



Brainstorming



Widget placement

<i>Command</i>	<i>Designation</i>
<code>alignment=Qt.AlignHCenter</code>	Align to center (horizontally)
<code>alignment=Qt.AlignVCenter</code>	Align to center (vertically)
<code>stretch=2</code>	Stretch a widget (for example, a button)
<code>layout_card.setSpacing(5)</code>	Set spacing between content in the layout (for example, between horizontal lines)



Brainstorming



Example:

<i>Command</i>	<i>Designation</i>
<code>alignment=Qt.AlignHCenter</code>	Align to center
<code>stretch=2</code>	Stretch a widget (button)
<code>layout_card.setSpacing(5)</code>	Set spacing between lines

```
layout_line1.addWidget(lb_Question, alignment=Qt.AlignHCenter)
```

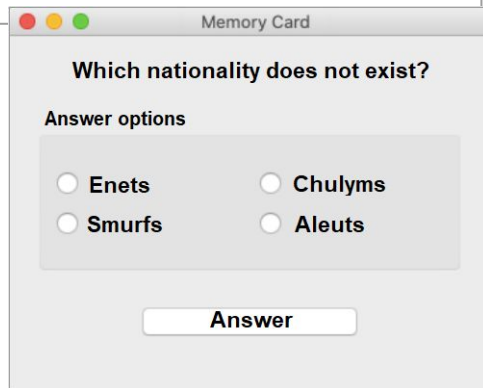
```
#...
```

```
layout_line3.addWidget(btn_OK, stretch=3)
```

```
#...
```

```
layout_card.setSpacing(5)
```

*Question in the center, large “Answer” button,
there is spacing between question, answers and button.*



Memory Card

Which nationality does not exist?

Answer options

☐ Enets ☐ Chulymys

☐ Smurfs ☐ Aleuts

Answer



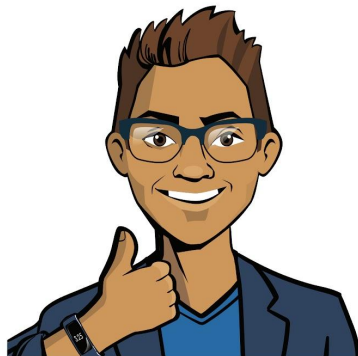
Brainstorming



Your goal :

Begin creating an interface for the Memory Card application in the development environment.

1. Create an application window; set sizes; heading.
2. Create a question display form:
 - ☐ link the necessary modules
 - ☐ create the necessary widgets
 - ☐ position these widgets in layouts
 - ☐ set spacing and center content where needed



Brainstorming

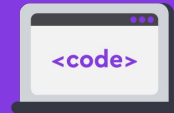
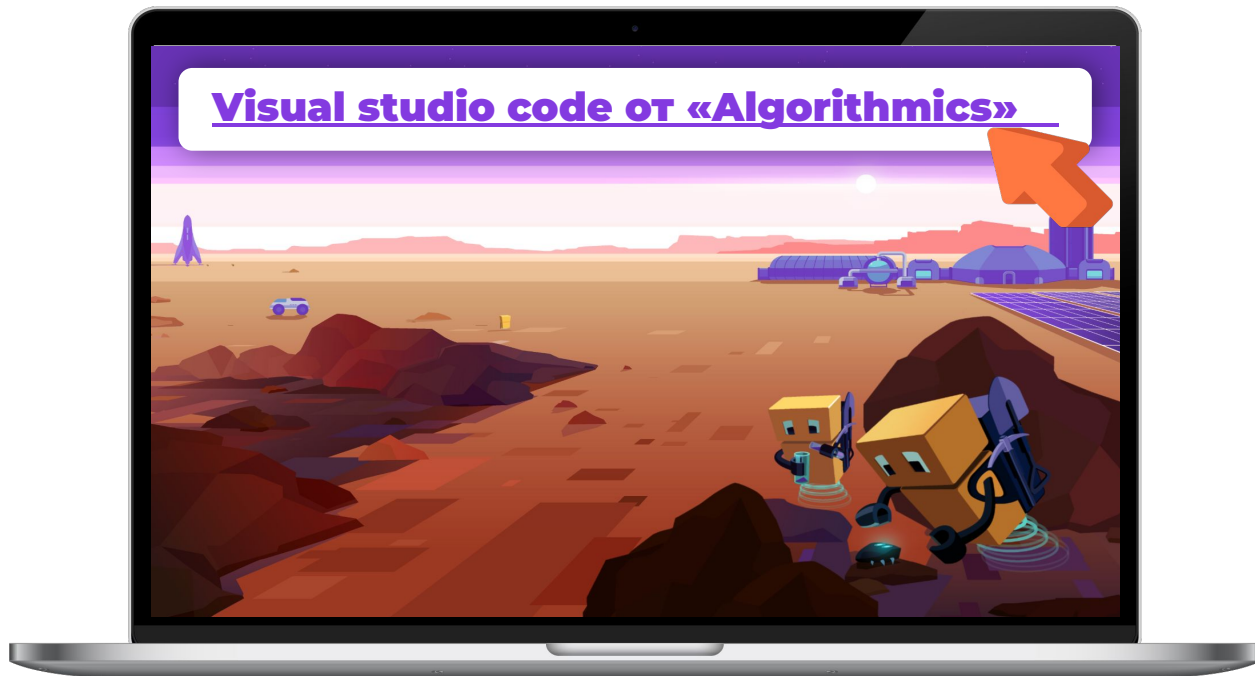


Visual Studio Code: Memory Card Application



Do the task in VS Code

➡ «VSC. PyQt. Memory Card»



Application creation

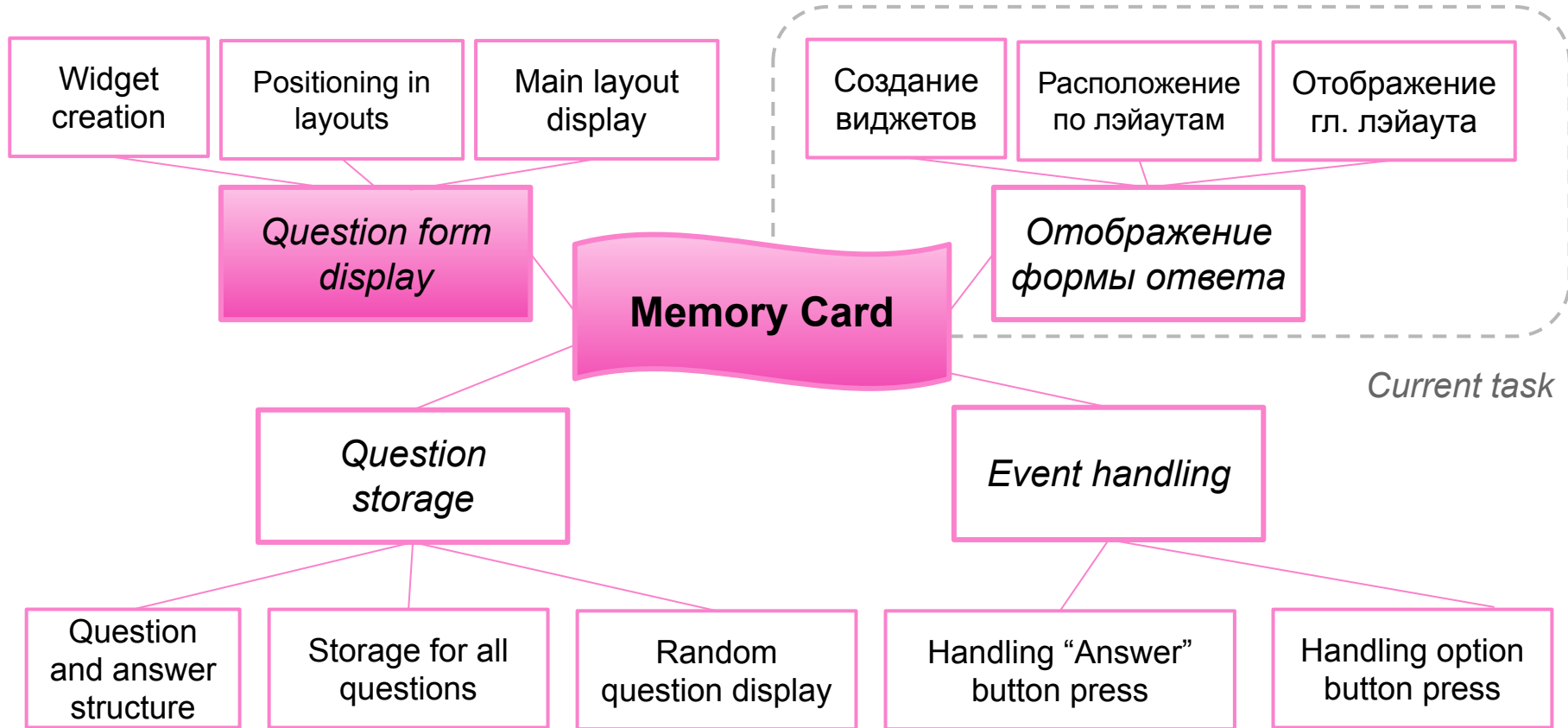


Brainstorming:

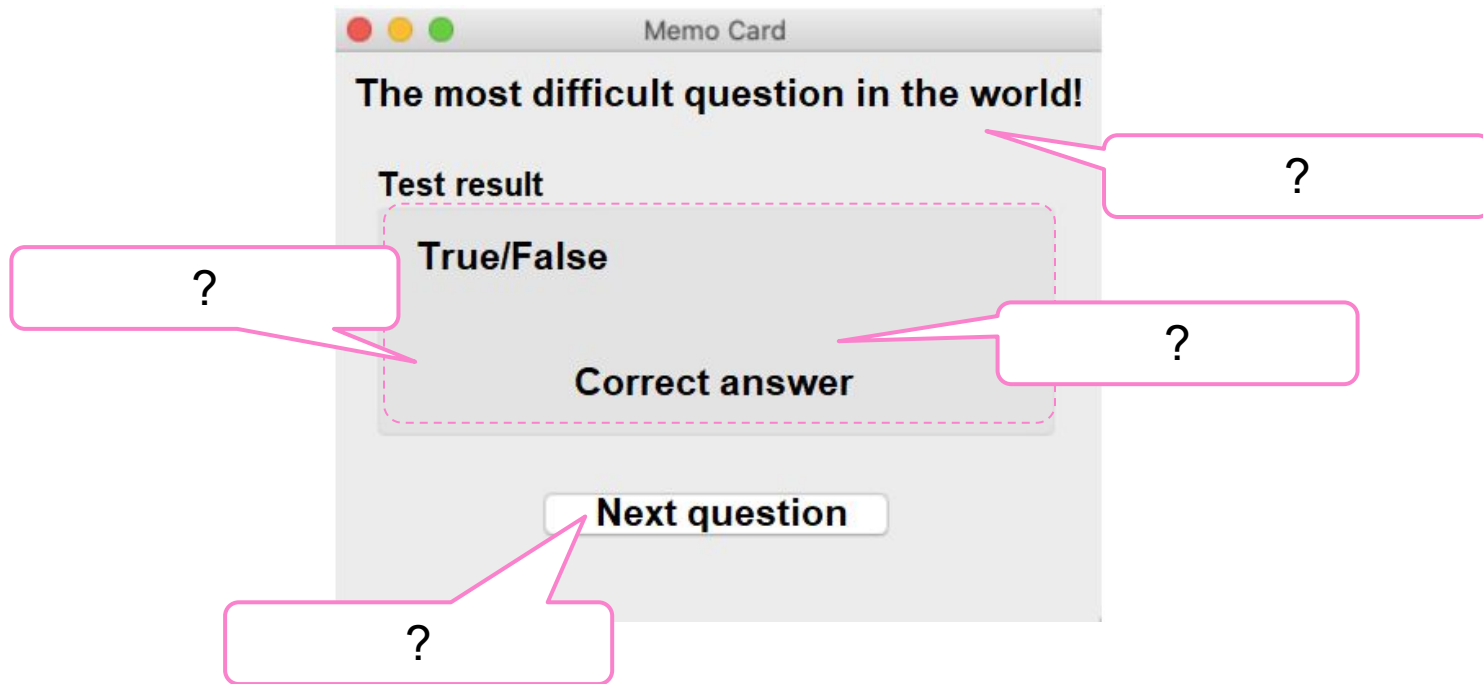
The Memory Card Interface



Application property diagram



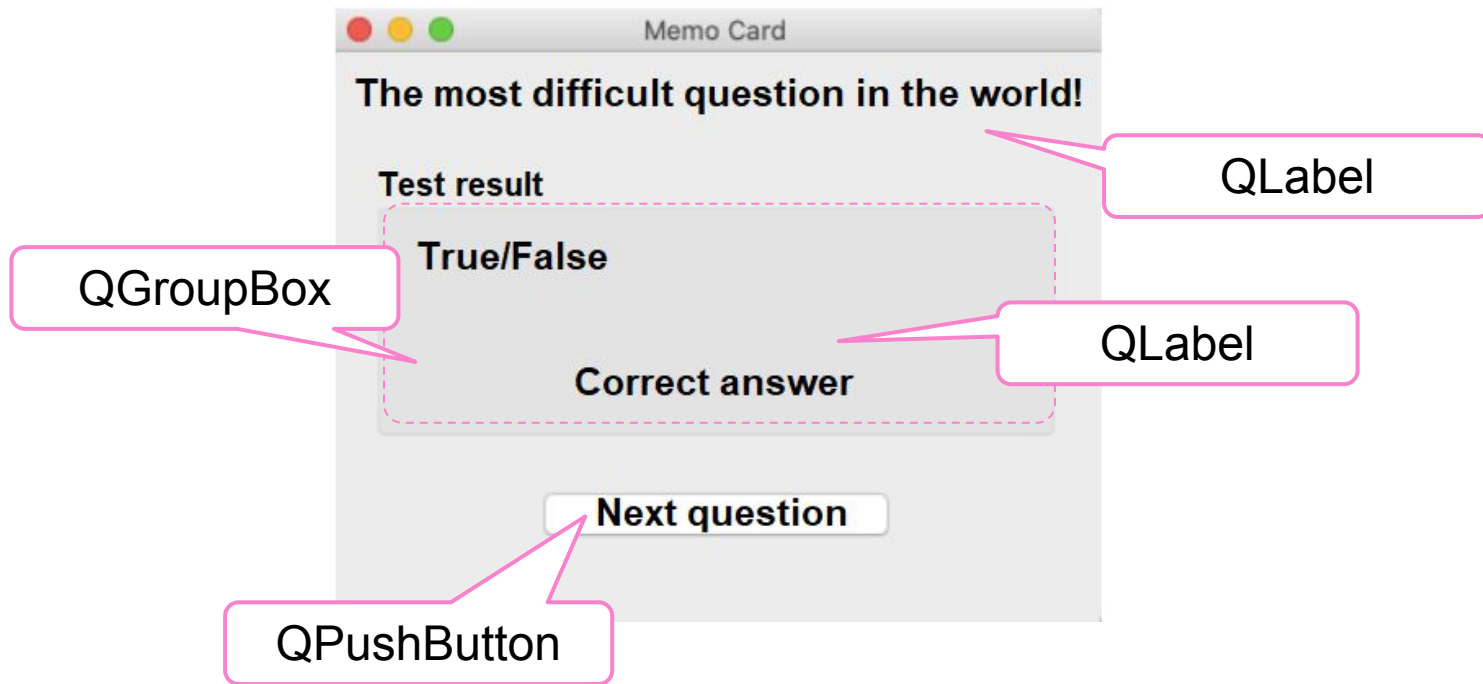
What widgets are in this window?



Brainstorming



What widgets are in this window?



Brainstorming

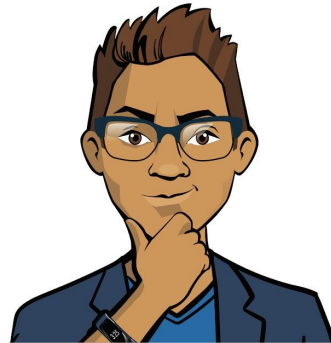


Switching from form to form

Optimally, both the question form and the answer form should be placed in the same window.

In that case, the switch from the question form to the answer form will happen when the “Answer” button is pressed.

We'll do event handling next time. For now, let's **hide the question form** and **put the correct answer form** above it.



Brainstorming



Show/hide group

You already know the commands for covering and displaying widgets. You can hide and show groups of widgets the same way.

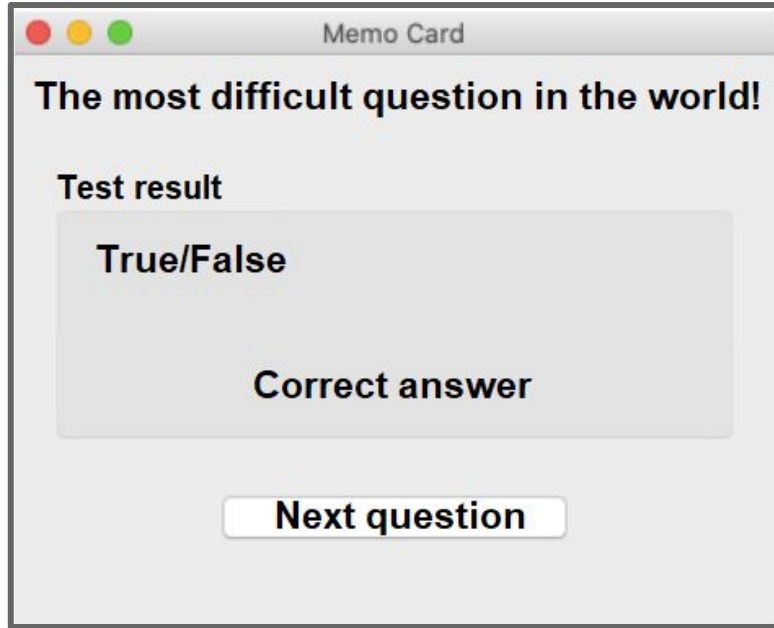
<i>Command</i>	<i>Designation</i>
<code>RadioGroupBox.hide()</code>	Cover answer option panel
<code>AnsGroupBox.show()</code>	Show correct answer panel (shown by default)



Brainstorming



How do we position the content in the layouts?



The mockup shows a window titled "Memo Card" with a standard macOS-style title bar (red, yellow, green buttons). The content is organized as follows:

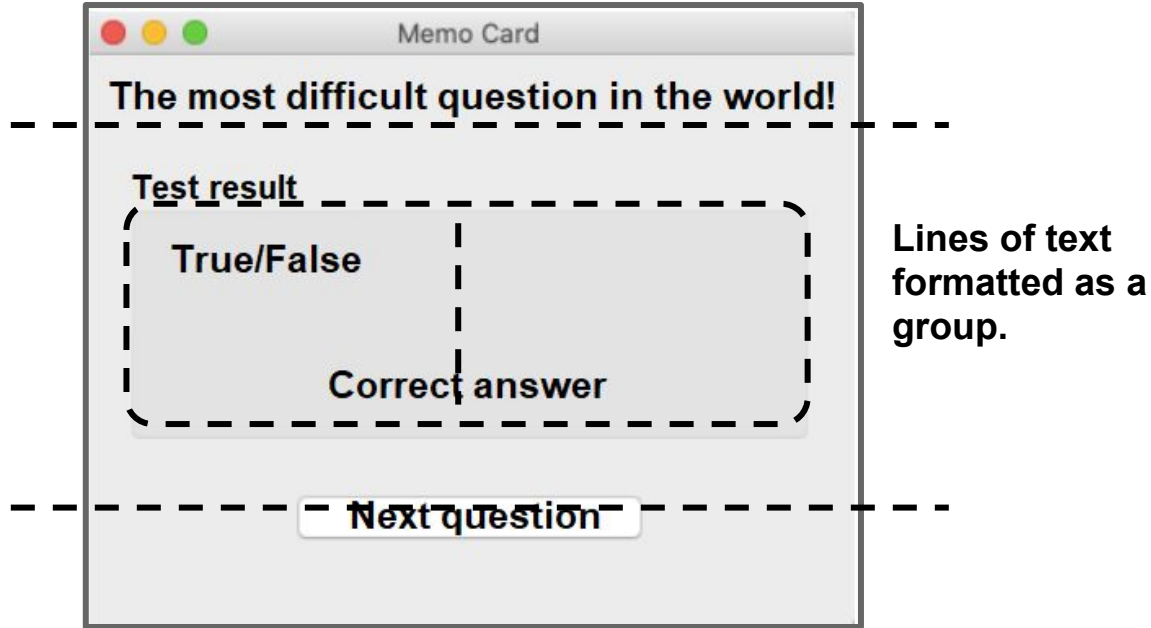
- Title:** "The most difficult question in the world!"
- Section Header:** "Test result"
- Content Area:** A light gray rectangular box containing:
 - Text:** "True/False"
 - Text:** "Correct answer"
- Action:** A button labeled "Next question" at the bottom.



Brainstorming



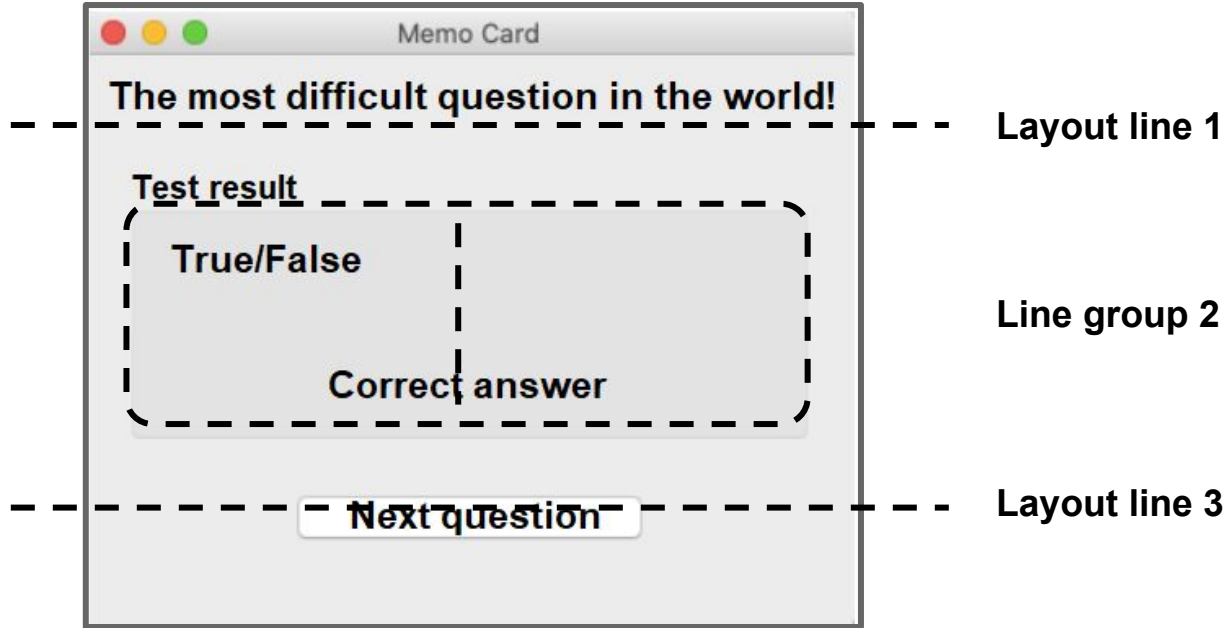
One possibility:



Brainstorming



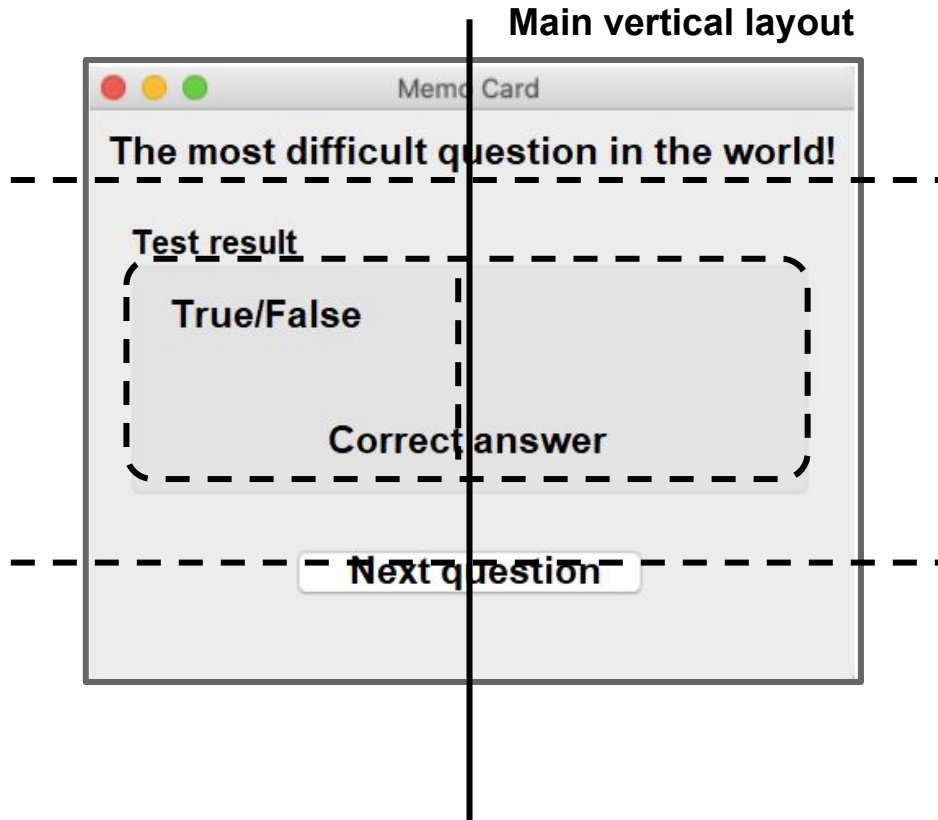
Another possibility:



Brainstorming



Another possibility:



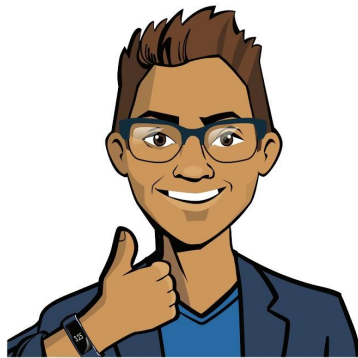
Brainstorming



Your task:

Continue creating the interface for the Memory Card application in the development environment.

1. Hide the question and answer option form.
2. Create the correct answer form:
 - ☐ link the necessary modules
 - ☐ create the necessary widgets
 - ☐ place them in the layouts
 - ☐ set spacing and center content where needed



Brainstorming

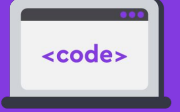
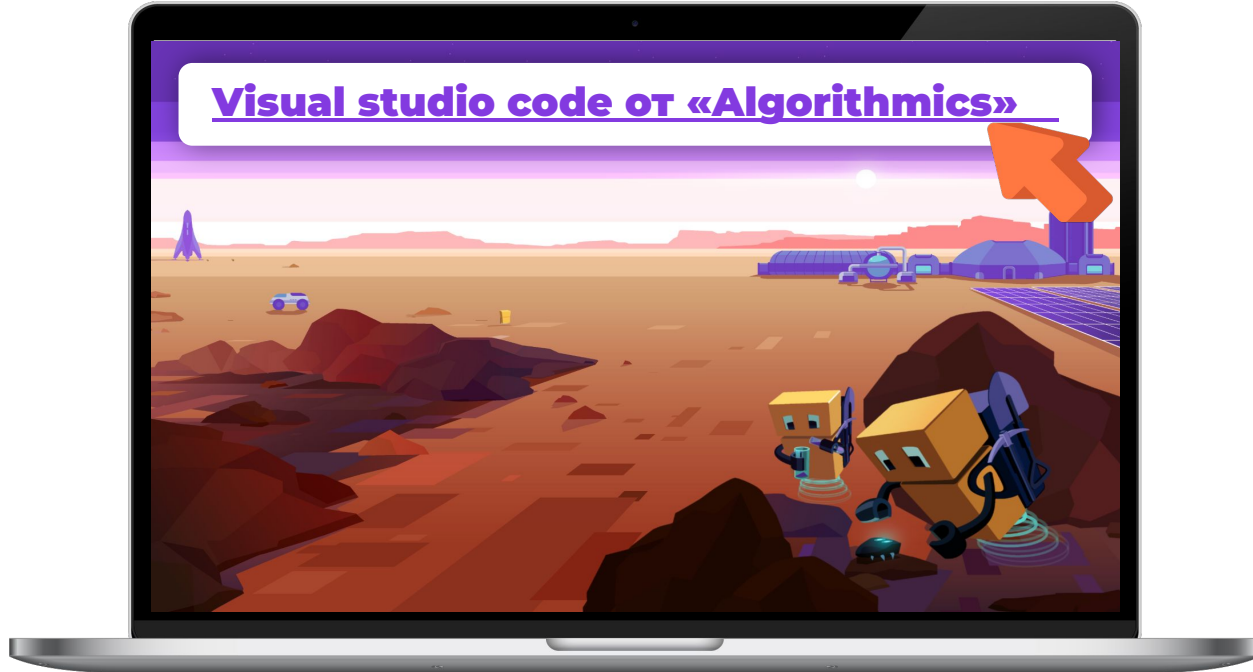


Visual Studio Code: Memory Card Application



Do the task in VS Code

➡ «VSC. PyQt. Memory Card»



Application creation

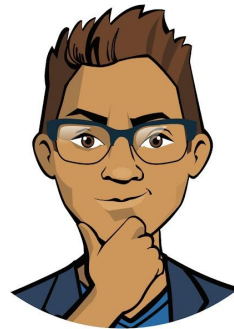


Wrapping up the workday



To wrap up the workday, complete a technical interview:

1. List all the possible objects that can be added to the layout.
2. How do you center a widget? How do you align it to the left? What module needs to be linked for that?
3. How do you stretch a widget? What module needs to be linked for that?



*Cole,
Senior developer*



*Emily,
Project manager*



Wrapping up the
workday

Excellent work!

Today, you:

- 1) began working on a large project and broke it into tasks
- 2) completed two blocks of tasks for programming the interface
- 3) learned new parameters for aligning and stretching widgets



Wrapping up the
workday

Evaluating the effectiveness of today's work

Answer these questions with your colleagues:

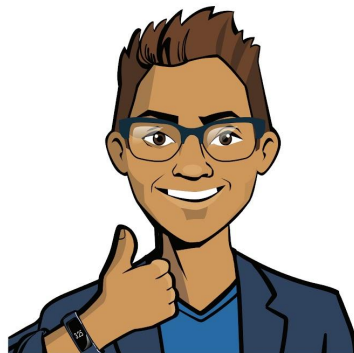
1. What went especially well today?
2. What didn't turn out the way you wanted?
3. What should you do next time to ensure a success?



Wrapping up the
workday

Additional tasks

- ❑ Examine the code you've written one more time.
- ❑ Finish writing the code, if necessary.
- ❑ **Add additional comments to the code to explain** which part of the code does what.



Wrapping up the
workday