

## 6. Мемори

### Постановка задачи

В этом задании сделаем игру на тренировку памяти, известную под названиями «Мемори» или «Угадай пару». Игра должна проходить по следующим правилам.

- Играет один игрок.
- Перед началом игры перед игроком находятся 16 перевёрнутых карточек с цифрами от 0 до 7, по две карточки с каждой цифрой.
- Игрок делает ход: выбирает щелчком мыши и переворачивает две карточки. Если на этих карточках написаны одинаковые цифры, то карточки и дальше останутся открытыми. Если на карточках написаны разные цифры, то после того, как игрок выберет следующую карточку, выбранная ранее пара карточек закроется.
- Задача игрока: открыть все карточки за минимальное число ходов.
- При запуске программы игра должна начаться без дополнительного нажатия на кнопку «Перезапустить игру».
- После того, как игра закончится, игрок должен нажать на кнопку «Перезапустить игру» для её начала.

Для упрощения работы рекомендуется использовать шаблон `memory_template.py`.

### Пример игры

#### Начало игры

Все карты скрыты.



Рис. 6.1. Экран в начале игры

#### Первый ход

Игрок выбирает одну карту, щелкает по ней мышью, она открывается.

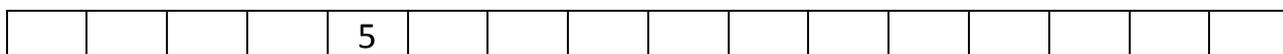


Рис. 6.2. Экран игры в начале первого хода

Игрок выбирает вторую карту, щелкает по ней, она тоже открывается.

				5					3						
--	--	--	--	---	--	--	--	--	---	--	--	--	--	--	--

Рис. 6.3. Экран игры в конце первого хода

Карты разные, в начале следующего хода они будут закрыты.

### *Второй ход*

Игрок выбирает ещё одну карту. Выбранная им карта открывается, непарные карты, открытые на предыдущем ходу закрываются.

		4													
--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--

Рис. 6.4. Экран игры в начале второго хода

Игрок выбирает следующую карту.

		4				4									
--	--	---	--	--	--	---	--	--	--	--	--	--	--	--	--

Рис. 6.5. Экран игры в конце второго хода

Карты совпали, поэтому дальше они останутся открытыми.

### *Третий ход*

Игрок выбирает карту, она открывается, предыдущие открытые карты остаются открытыми.

		4				4						7			
--	--	---	--	--	--	---	--	--	--	--	--	---	--	--	--

Рис. 6.6. Экран игры в начале третьего хода

Игрок выбирает вторую карту, она открывается.

		4				4	6					7			
--	--	---	--	--	--	---	---	--	--	--	--	---	--	--	--

Рис. 6.7. Экран игры в конце третьего хода

Карты не совпали, они будут закрыты.

### *Четвёртый ход*

Игрок выбирает карту, выбранная карта открывается, карты с третьего хода закрываются.

		4				4								3	
--	--	---	--	--	--	---	--	--	--	--	--	--	--	---	--

Рис. 6.8. Экран игры в начале четвёртого хода

Игрок выбирает ещё одну карту.

		4				4			3					3	
--	--	---	--	--	--	---	--	--	---	--	--	--	--	---	--

Рис. 6.9. Экран игры в конце четвёртого хода

Найдена вторая пара карт. Эти карты также останутся открытыми до конца игры.

### Конец игры

Игра кончается, когда игрок откроет все карты.

1	2	4	2	5	6	4	6	1	3	8	8	7	9	3	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Рис. 6.10. Экран в конце игры

### Наблюдения за ходом игры

Обратите внимание на следующие моменты:

- Один ход игры включает **два** щелчка мышью по **закрытым** картам. Подсчёт ходов должен вестись именно таким образом.
- Щелчки по открытым картам не должны приводить ни к каким изменениям в состоянии игры.
- Карты открываются только по выбору пользователя: одна карта по щелчку мышью.
- Закраться могут только те две карты, которые были открыты на предыдущем ходе. Карты, открывавшиеся больше одного хода назад, не закрываются.

### Рекомендованный порядок работы над заданием

1. В игре «Мемори» используется набор из 16 карточек с цифрами в интервале  $[0,8)$ , каждая цифра встречается дважды. Предполагается, что для создания полного списка карточек вы произведёте конкатенацию двух списков с цифрами от 0 до 8. Такой список можно сделать руками или с помощью функции `range()`.
2. Напишите обработчик рисования, который перебирает игровые карточки, используя цикл `for`, и рисует карточки с цифрами на холсте, используя `draw_text()`.
3. Перемешайте карточки, используя `random.shuffle()`. Не забудьте отладить код, рисующий холст, перед тем, как перемешивать карточки, это упростит отладку.

4. Модифицируйте обработчик рисования так, чтобы рисовался пустой зелёный прямоугольник либо значение карточки. Для реализации такого поведения программы предлагается создать второй список с названием `exposed`, каждая запись которого имеет значение `True`, если соответствующая ей карта перевернута и её значение видно, и `False`, если карта лежит рубашкой вверх и её значение недоступно. Для проверки кода следует инициализировать список `exposed` какой-либо последовательностью значений.

5. Далее добавьте функциональность, позволяющую определить, на какую из карт нажали мышкой.

Сначала реализуйте функцию `canvas2card(pos)`, которая по координатам щелчка вернёт номер карты. Чтобы упростить определение, какая из карт выбрана, в шаблоне использован такой размер холста, чтобы карты его полностью равномерно заполняли. Сделайте для этой функции несколько `doctest`'ов и проверьте её работу.

В обработчике кликов мыши вызовите эту функцию, и распечатайте результат. Зарегистрируйте обработчик событий кликов мыши. Проверьте, что в консоли печатаются правильные номера карт.

6. В обработчике событий кликов мышки уберите печать и добавьте вызов функции `card_clicked(card)`, передав ей номер карты из `canvas2card(pos)`. В этой функции должна быть реализована основная логика игры.

Сначала сделайте так, чтобы переворачивалась карточка, которую щелкнули мышкой. Если игрок кликнул `i`-ю карточку, следует изменить значение `exposed[i]` с `False` на `True`. Если карта до клика уже была открыта – игнорируйте этот клик мыши.

Напишите `doctest`'ы для проверки этой функции. Обратите внимание, что её работу надо проверять по изменению глобальных переменных, например так:

```
>>> exposed = [False] * 8
>>> card_clicked(1)
>>> exposed
[False, True, False, False, False, False, False,
 False]
```

```
>>> card_clicked(3)
>>> exposed
[False, True, False, True, False, False, False,
False]
>>> card_clicked(1)
>>> exposed
[False, True, False, True, False, False, False,
False]
```

На данном этапе основа для игры написана. Проверьте, что в интерфейсе открывание карт тоже работает корректно.

7. Далее следует добавить логику в функцию `card_clicked(card)` для выбора двух карточек и определения, совпадают они или нет.

Совет: Введите глобальную переменную, которая будет отслеживать состояние игры со следующими значениями:

- a) состояние 0 соответствует началу игры,
- b) 1 – открытию первой карты во время хода,
- c) 2 – открытию второй карты во время хода (ход завершен).

Игра начинается в состоянии 0. После первого щелчка мыши игра переходит в состояние 1. Далее после каждого щелчка по закрытой карте она переходит между состояниями 1 и 2 по цепочке  $2 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow \dots$ . Такое поведение можно реализовать следующим кодом (не забудьте добавить проверку на попадание в закрытую карту):

```
def card_clicked(card):
    global state
    ...
    if state == 0:
        state = 1
    elif state == 1:
        state = 2
    else:
        state = 1
    ...
```

Протестируйте корректность изменения состояния игры с помощью дополнения `doctest`'ов.

8. В состоянии 2 вам следует определить, не были ли парными две предыдущие карты. Если они были непарными, вам следует их опять перевернуть вверх рубашкой. Для хранения индексов обеих карт, выбранных в предыдущий ход, следует использовать глобальные переменные.

Измените doctest'ы так, чтобы они соответствовали правильной работе игры. Вам потребуется изменить старый тест и добавить новых, чтобы проверить все варианты работы функции.

9. Добавьте счётчик, отслеживающий число ходов. Увеличивайте значение счётчика в функции `card_clicked(card)`. Проверьте правильность изменения счётчика, дополнив doctest'ы соответствующими проверками.

В обработчике щелчков мыши используйте функцию `set_text()` для вывода счётчика в метку (англ. label) на боковой панели.

Для создания текстовой метки необходимо после создания окна (frame) выполнить

```
label = frame.add_label("Ход = 0")
```

После этого для обновления текста метки используйте `label.set_text("Ход = " + str(turn))`

10. В завершении реализуйте функцию `new_game()` так, чтобы кнопка «Перезапустить игру» перемешивала карты, обнуляла счётчик ходов и перезапускала игру. В начале игры все карты должны быть закрыты.

### Дополнительные задания

1. Замените `draw_text()` на `draw_image()` для каждой карты, используя 8 различных изображений с цифрами.
2. Организуйте свой код так, чтобы можно было изменять число карт в игре, меняя только значение `NUM_PAIRS` в начале шаблона.

На первый взгляд проект может показаться сложным, но окончательная реализация, проведённая согласно пунктам, займёт около ста строк кода с комментариями и пробелами. Если у вас возникают трудности, сосредоточьтесь на разработке вашего проекта до 6-го шага, т.к. на этом пункте проще увидеть игру целиком. Далее разработка пойдёт проще.

### Оценка задания

В начале игры правильно производится перемешивание карт – 10 %.

- Игра рисует карточки с цифрами на холсте – 10 %.
- В начале игры щелчок по закрытой карте приводит к её открытию – 10 %.
- Игнорируются щелчки на открытых картах – 10 %.
- Если открыта одна непарная карта (состояние=1), то щелчок на закрытой карте приводит к её открытию – 10 %.
- Если открыты две **непарные** карты (состояние=2), то щелчок на закрытой карте приводит к её открытию и закрытию непарных карт – 10 %.
- Если открыты две **парные** карты (состояние=2), то щелчок на закрытой карте приводит к её открытию, парные карты остаются открытыми – 10 %.
- Парные карты, открытые на одном ходе, остаются открытыми до перезапуска игры – 10 %.
- Игра ведёт корректный подсчёт ходов (номер хода может увеличиваться или после первого, или после второго выбора игрока внутри хода) – 5 %.
- Номер хода отображается в текстовой метке на панели управления – 10 %.
- Имеется кнопка перезапуска игры, при нажатии на которую счётчик числа ходов сбрасывается, карты перемешиваются и закрываются – 5 %.

В случае отсутствия тестов, за пункты, для которых они предусмотрены, будет поставлено 75% от указанных выше баллов.

### **Дополнительные баллы**

- Для вывода карт используются изображения вместо текста – 10 %.
- Число карт может быть изменено переменной NUM\_PAIRS – 10 %.

### **Срок сдачи задания**

Группа, занимающаяся по вторникам – 25.10.2016; по пятницам – 28.10.2016.