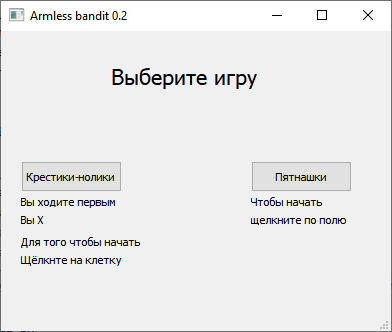
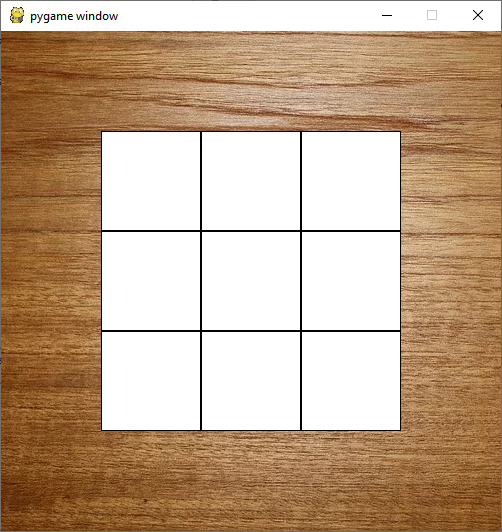
ArmlessBandit (безрукий бандит) 0.2

Выполнил: Алексеев Антон

Основная идея: создание приложения с маленькими короткими играми, которые позволят отвлечься во время долгой рутинной работы. Игры, созданные в приложении: крестики-нолики и пятнашки.

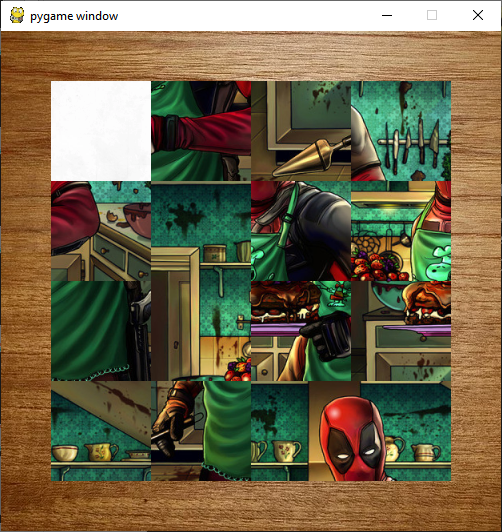


Реализация. Была использована библиотека pygame и pyqt. В главном окне выбирается и запускается игра.



В крестиках – ноликах большая часть кода приходится на функцию хода компьютера. В каждой ячейке записано значение (либо 0, т.е она пустая, либо 1, занятая игроком, или 2 занята компьютером). Так же записано кто сейчас должен делать ход, если игрок, то компьютер ждет его хода, если компьютер, то он совершает ход. В ходе компьютера прописано, что если есть возможность победить он побеждает, иначе, если противник может победить он ему мешает, иначе, если центральная ячейка пустая, то он её заполняет, иначе он выбирает с помощью функции random.choice ячейку, которую он будет заполнять. Подобная «неидеальность» позволяет игроку выигрывать компьютер, который иначе будет либо побеждать, либо сводить всё к ничьей.

В пятнашках большая часть кода находится в функции нажатия на кнопку (ячейку поля). Изначально есть список из 16 элементов заполненный числами 0, 1, 2, 3 … 15, который с помощью метода render() преобразуется в названия картинок в папке data и потом выводит эти картинки в нужном порядке. С помощью библиотеки random список «перемешивается». После этого проверяется можно ли собрать такие пятнашки по формуле, найденной мной в интернете. Если их собрать нельзя, то функция рекурсивно повторяется до тех пор, пока не будет собираемый вариант. Так как количество вариантов собираемых и не собираемых вариантов пятнашек равно (тоже в интернете нашёл), то вероятность того что первый раз будет удачная последовательность 50%, второй 75% и т.д. Поэтому подобных повторений будет 1-3 максимум и тормозить программа не будет. После успешного заполнения списка значения списка выводятся в соответствующие ячейкам списка ячейки поля (кнопки). При нажатии на ячейку поля, проверяется идёт ли игра, если да, то проверяется соседняя ли к ней пустая ячейка и, если да, то меняются значения в списке. Так до победы.



Также хочется отметить что в каждом окне я предусмотрел последствия их закрытия, т.е: при закрытии главного – прекращается вся программа, а при закрытии игр главное окно остается.

Используемые библиотеки: sys, pygame, random, os, uic, QtGui, QApplication, QMainWindow

В разработке также использовались QtDesigner, PyCharmCommunity