**Пояснительная записка**

# Идея проекта:

Создать работающую игру

# Задачи, выполнение которых необходимо для реализации проекта

* Создать визуальную составляющую
* Написать логические алгоритмы программы
* Соединить два этих пункта

# Что будет реализовываться

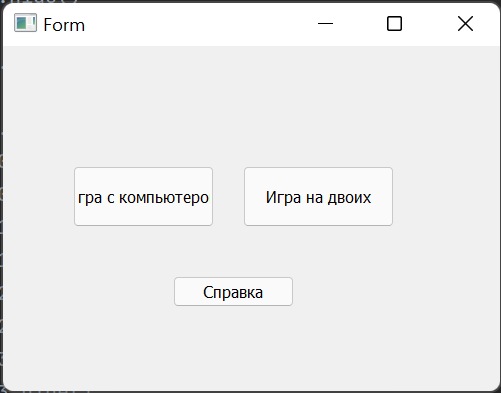
* Игра против компьютера
* Игра на одном устройстве для двоих человек
* Справочная информация

# Использовавшиеся технологии

* PyQt5 – для создания окон программы (визуальная составляющая)
* Sqlite3 – для хранения данных об игроках
* Random – для расставления кораблей компьютера

# Использовавшиеся алгоритмы

При запуске программы показывается окно, созданное с помощью библиотеки PyQt5, где пользователь может увидеть 3 кнопки: 1) игра с компьютером 2) игра на одном устройстве 3) Справка

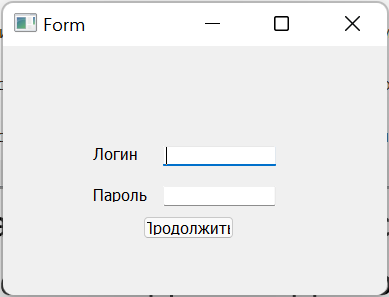


При нажатии на кнопку «справка» на экране появляется окно (экземпляр класса Spravka), где описаны правила игры (реализовано через текстовый файл и последующее его считывание и добавление текста на экран)

Изображение выглядит как текст

Автоматически созданное описание

При нажатии на кнопку «игра с компьютером» появляется окно для ввода логина и пароля пользователя



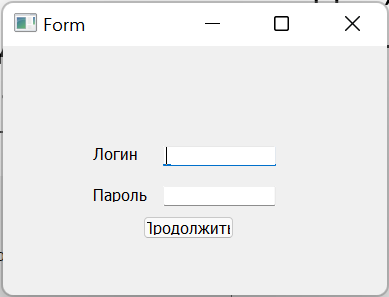
Затем происходит проверка на наличие пользователя в базе данных (работа файла check\_user1.py, реализовано с помощью библиотеки sqlite3)

Если пользователь не найден, то появляется окно предлагающее создать нового пользователя

Изображение выглядит как текст

Автоматически созданное описание

После нажатия на кнопку «создать» на экран появляется окно для ввода логина и пароля, после чего пользователь добавляется в базу данных с введенными логином и паролем



Теперь, когда пользователь есть в базе данных и его пароль верный, появляется окно для расстановки кораблей

Изображение выглядит как стол

Автоматически созданное описание

Здесь пользователь расставляет свои корабли. Для начала в правом меню он выбирает количество палуб корабля и потом, в меню под полем, он выбирает расположение: горизонтальное или вертикальное, потом нажимает на клетку внутри поля и тогда проверяется количество возможных расставленных кораблей данного типа, если это значение больше нуля, то происходит проверка можно ли туда поставить корабль с помощью функции check из файла structure. Функция check, если всё хорошо ставит корабль по заданным координатам, направлению и количеству палуб. Если количество кораблей данного типа превышено, то появляется сообщение о том, что нельзя ставить столько кораблей. Если функция check возвращает значение False, то выводится сообщение о том, что нельзя ставить корабль в заданные координаты. Если есть корабли, которые можно поставить, а пользователь нажал кнопку далее, выводится сообщение о том, что расставлены не все корабли. Таким образом, пользователь вынужден расставить все корабли в соответствии с правилами игры. Затем пользователь нажимает кнопку далее и на экране появляется окно сражения.

Изображение выглядит как стол

Автоматически созданное описание

В то же время вызывается функция makerasstanovka. Таким образом расставляются корабли у компьютера.

Начинается сражение, первым ходит игрок, он выбирает клетку, в которую хочет выстрелить и если у поля компьютера в этой клетке находится значение 4, то пользователь попал и это значение становится 2, одновременно с этим выполняется функция dead, которая проверяет убит ли корабль. Иначе это значение становится 1 (что означает промах). Ход переходит к компьютеру. вызывается функция comphod. Каждый раз когда либо игрок, либо компьютер убивают корабль, происходит проверка на количество оставшихся живых кораблей. Если в какой-то момент либо у компьютера, либо у игрока не остаётся живых кораблей вызывается окно победы или проигрыша игрока.

Аналогично работает режим на одном устройстве. В момент передачи устройства другому игроку (для расстановки кораблей) появляется окно (объект класса For\_2\_players)

Изображение выглядит как текст

Автоматически созданное описание

Также если живых кораблей не осталось выводится победа первого игрока или победа второго игрока (экземпляры классов win\_1 или win\_2 соответственно)

## Как работает функция check

Эта функция получает на вход координаты x и y (значения от 0 до 9), а также расположение корабля (1 – если вертикально, 2 – если горизонтально), количество палуб и поле, где это нужно выполнить (в данном случае поле игрока).

Изначально поле игрока заполнена нулями, если все клетки, принадлежащие кораблю это 0, то клетки, в которых должен стоять корабль, заполняется 4 и вызывается функция fillmas, которая окружает корабль 3

## Как работает функция fillmas

Данная функция получает на вход координаты x, y, направление, количество палуб и поле, где это нужно выполнить. Затем приходит по каждой клетке принадлежащий кораблем и если клетки вокруг нее являются 0, то эти клетки становятся 3.

## Как работает функция dead

Эта функция принимает параметры x, y и поле в котором это нужно проверить И смотрит в 4 стороны и проверяет есть ли клетка с номером 4 до которой можно добраться по клеткам со значением 2 если такая клетка есть то корабль не убит иначе убит и тогда все эти клетки со значением 2 заменяется на клетки со значением 5

## Как работает функция makerasstanovka

Данная функция с помощью метода randint получает координаты x, y и направление, вызывает функцию check, только вместо поля игрока передаёт поле компьютера

## Как работает функция fillneardead

Эта функция является полной копией функции fillmas, но действует во время сражения для убитых кораблей

## Как работает функция comphod

С помощью метода randint получает координаты x, y и затем выполняем проверку на то: стрелял ли компьютер уже в это место, Значение клетки поля игрока равно 5 или 2 или 1