**План проекта**

**Задача:** Необходимо разработать 2D игру “Змейка”.

**Основные положения, правила игры:**

Игра происходит на **игровом поле** (*Playing field*) прямоугольной формы. Само игровое поле состоит из множества **ячеек** (*Cell*) прямоугольной формы. Размеры одной ячейки задаются в пикселях. Размеры игрового поля задаются количеством ячеек в ширину и длину.

На игровом поле перемещается **змейка** (*Snake*). У змейки есть **головная часть** и **хвостовая** часть. **Направление движения змейки** определяется направлением движением его головной части. У змейки в начале игры есть **исходное положение**, которое он занимает внутри игрового поля и его **изначальный размер**. Его исходное положение, и вообще положение змейки в процессе игры храниться в **массиве координат**, где первый элемент массива — это координата головной части змейки, а последний координата хвостовой части соответственно. В начале игры змейка располагается в заданной точке вертикально относительно нижней стороны игрового поля.

Внутри игрового поля случайным образом **генерируются фрукты**. Изначальное **количество** фруктов внутри игрового поля, а также количество фруктов которое может генерироваться за раз всегда фиксировано и задаётся одним переменным. Змейка передвигаясь по игровому полю **съедает** эти фрукты. Если змейка съест все фрукты в игровом поле, то на поле появятся новые фрукты в количестве равным *fruit\_max\_amount*. Также, когда змейка съедает фрукт то её размер увеличивается на одну ячейку от конца. Причём увеличение змейки на одну ячейку происходит после движения змейки на одну ячейку.

В процессе игры змейка съедая фрукты набирает **очки**, которые записываются в **общий счёт игры**. Изначально общий счёт игры равен нулю. Также во время игры ведётся замер времени. Результаты замера тоже записывается в **общий счёт времени**. Общий счёт времени изначально тоже равен нулю.

Змейка в процессе игры может **умереть**. В случае смерти змейки она **прекращает своё движение**. Смерть змейки происходит **в двух случаях**: головная часть змейки столкнулась с игровым полем, головная часть змейки столкнулась с телом змейки (змейка столкнулась с самим собой). Смерть змейки **фиксируется** в булевой переменной. Она принимает истинное значение если змейка жива. Если она мертва, то переменная принимает ложное значение. Изначально, до того, как пользователь нажмёт клавишу “W” она принимает ложное значение. То есть до того, как началась игра змейка считается мёртвой и соответственно не двигается.

**Используемые переменные, методы:**

|  |  |
| --- | --- |
| **Переменные** | |
| **Переменная** | **Хранимое значение/Назначение** |
| **Игровое поле** | |
| **int** *cell\_width* | Ширина одной ячейки игрового поля. |
| **int** *cell\_length* | Длина одной ячейки игрового поля. |
| **int** *pf\_width* | Количество ячеек игрового поля в ширину. |
| **int** *pf\_length* | Количество ячеек игрового поля в длину. |
| **Змейка** | |
| **int** *snake\_size* | Длина змейки в игровых ячейках. |
| **int** *positions*[*snake\_size*][2] | Координаты ячеек в которых располагается змейка. |
| **bool** *is\_alive* = false | Фиксирует смерть змейки. |
| **Фрукты** | |
| **int** *fruit\_max\_amount* | Количество фруктов, которое изначально располагается внутри игрового поля, а также количество фруктов которое генерируется после съедания змейкой всех фруктов в игровом поле. |
| Счёт | |
| **int** *total\_game\_score* = 0 | Общий счёт игры в очках. Количество оков равно количество фруктов съеденная змейкой за игру. |
| **int** *total\_game\_time* = 0 | Общее время игры. Равно количеству секунд пройдённое от начала игры до смерти змейки. |

**Используемые сокращения:**

pf – *Playing Field* (Игровое поле);