Игра «Змейка». Отображение движения змейки в одном направлении + еда.

Темы: Инструкции цикла (while, for), перечисления, директивы препроцессора, одномерные массивы

В этом задании мы также будем изображать движущуюся змейку, но в дополнение к змейке на поле появляется еда.

1. Еда для змейки возникает на поле в случайном месте (но только **пустом** месте).
2. Если змейке удается «съесть» еду (координаты головы змейки совпадают с координатами еды), то длина змейки увеличивается на 1 элемент.

**Замечание**

Текущая версия программы все еще будет упрощенной.

* Игровое поле будет состоять только из одной строки.
* Обработка нажатия клавиш производиться не будет
* Весь рабочий код пока будет располагаться в функции main.
* Завершение игры происходит тогда, когда на поле негде будет поставить еду.

В текущей работе, как и в работе «Отображение змейки\_2» будем применять одномерные массивы.

Текущий проект будет разрабатываться на основе предыдущего.

**Задание1**

1. Добавьте в проект файлы, которые были созданы в предыдущем задании:

* snake.cpp и snake.h, которые будут содержать информацию, относящуюся к понятию "Змейка"
* game.cpp и game.h, которые будут содержать информацию, относящуюся к понятию "Игра".
* Заголовочный файл const.h, который будет содержать константные данные
* Файл main.cpp, который будет содержать все управляющие инструкции
* field.cpp и field.h, которые будут содержать информацию, относящуюся к понятию "Поле игры".

1. Добавьте в проект новые файлы:

food.cpp и food.h, которые будут содержать информацию, относящуюся к понятию "Еда".

1. В файл const.h добавьте константу для отображения символа еды

## Глобальные данные

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Описание | переменная | | Модуль | | Примечания |
| Текущий размер змейки | snake\_size | | snake | | >0 |
| Таймаут задержки между шагами игры | timeout | | game | | >100 (ms) |
| Длина игрового поля (включает граничные элементы) | columns | | const.h | | const, >0 |
| Символ для головы змейки | head\_symbol | | const.h | | const |
| Символ для хвоста змейки | tail\_symbol | | const.h | | const |
| Символ для заполнения поля игры | field\_symbol | | const.h | | Const |
| Символ для границы поля игры | border\_symbol | | const.h | | Const |
| Максимальный размер массива змейки | L | | const.h | | Const, не должна быть > длины игрового поля-3.(2 граничных элемента +1 элемент еды) |
| Массив координат по горизонтали (X) хвоста змейки | snake\_x[L] | | snake | | >=0 |
| Массив игрового поля | field [columns] | | field | | >=0, пока одномерный |
| Символ еды | food\_symbol | const.h | |  | |
| Признак, что еда установлена | food\_flag | food | |  | |
| Координата еды по горизонтали (X) | food\_x | food | |  | |
| Признак продолжения игры | game\_on | game | |  | |

Желтым отмечены новые данные (по сравнению с предыдущим проектом)

**Задание 2**

Определите и проинициализируйте в соответствующих модулях нужные переменные и константы:

* Символ еды
* Признак, что еда установлена

**Генерация еды**

* Позиция еды определяется с помощью генерации случайного значения. Причем это значение должно попадать в пределы поля и в то же время эта позиция поля не должна быть занята (ни граничными символами, ни символами змейки). Позиция еды не меняется до тех пор, пока змейка не съест ее.
* В том случае, если змейка “съела” еду, требуется сгенерировать новую позицию еды.
* Выполнять генерацию имеет смысл до тех пор, пока есть место на поле, куда можно поставить еду. Если такого места нет, то нужно заканчивать игру.

**Задание 3**

* Сгенерируйте еду и установите ее на поле.
* В том случае, если змейке удается «съесть» еду (координаты головы змейки совпадают с координатами еды), то длину змейки надо увеличить на 1 элемент. При этом надо будет пересчитать элементы «хвоста» и передвинуть змейку.
* При условии, что еда съедена, необходимо повторить процесс генерации и установки новой порции еды.

 