Игра «Змейка». Отображение движения змейки в 2-мерном встроенном массиве

Темы: Двумерные встроенные массивы, функции с параметрами.

В этом задании мы также будем изображать движущуюся змейку, которая может съедать «еду», но в отличие от предыдущих проектов движение змейки будет осуществляться не только по горизонтали, но и по вертикали. И им можно будет управлять. Текущий проект будет разрабатываться на основе предыдущего.



**Задание 1**

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

**Задание 2**

Удобно для моделирования игрового поля использовать встроенный двумерный массив типа **char**, строки которого соответствуют значениям координаты Y, а столбцы - соответствуют значениям координаты X. Массив хранит символы, отображающие: границу, змейку и еду.

**Важно (НОВЫЕ данные)!**

Так как поле игры теперь будет двумерным, то потребуется ввести новые данные:

* Количество строк игрового поля (константа)
* Координата расположения еды по вертикали
* Массив координат змейки в поле игры (по вертикали).

Добавьте новые данные в программу

## Данные модуля const

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Идентификатор | тип |  |
| Длина игрового поля по горизонтали (включает граничные элементы) | columns |  |  |
| Длина игрового поля по вертикали | rows |  |  |
| Символ для хвоста змейки | tail\_symbol |  |  |
| Символ для головы змейки | head\_symbol |  |  |
| Символ для заполнения поля игры | field\_symbol |  |  |
| Символ для границы поля игры | border\_symbol |  |  |
| Максимальный размер массива змейки | L |  | const, не должна быть > длины игрового поля-2 |
| Перечисление «направление движения» | DIRECTION |  |  |
| Константы для обработки клавиатурного ввода (перечисление из “Задание (игра Змейка)\_1 (клавиатурный ввод)” ) | KYES |  |  |
| Символ еды | food\_symbol | const.h |  |

## Данные (переменные) функции main

|  |  |  |
| --- | --- | --- |
| Описание | Идентификатор | Тип |
| Текущий размер змейки |  |  |
| Массив координат змейки по горизонтали (X) |  |  |
| Массив координат змейки по вертикали (X) |  |  |
| Массив игрового поля |  |  |
| Признак, что еда установлена |  |  |
| Координата еды по горизонтали (X) |  |  |
| Координата еды по вертикали (Y) |  |  |
| Таймаут задержки между шагами игры |  |  |
| Признак продолжения игры |  |  |
| Направление движения змейки |  |  |

**Задание 3**

Поскольку теперь игровое поле будет двумерным, то придется изменить объявления и реализации функций, которые используют массив поля.

* Измените объявления и реализации функций, в которых используется поле игры.
* Алгоритмы функций меняться не должны,
* но так как змейка должна двигаться в двумерном поле, то реализации функций изменятся.

Все изменения будем вносить постепенно, чтобы не сломать все, что было сделано.

1. На первом этапе внесем изменения в функции модуля **field.**

## Функции модуля field

|  |  |  |
| --- | --- | --- |
| Описание | Идентификатор | Возможные параметры |
| Инициализация поля. Заполняется массив поля, в том числе границы | ??? init\_ field (???) | Массив поля, [возможно что-то еще] |
| Печать поля | ??? print\_field(???) | Массив поля, [возможно что-то еще] |
| Очистка поля | ???clear\_field(???) | Массив поля, [возможно что-то еще] |

1. Изменим объявления и реализации функций модуля **game.**  Но пока не будем добавлять новых параметров. В качестве Y-координаты будем использовать 0. Змейка будет двигаться только по нулевой строке двумерного массива.
2. Проверьте, что программа работает корректно. (Пока змейка двигается по-прежнему горизонтально)

## Функции модуля game

|  |  |  |
| --- | --- | --- |
| Описание | Идентификатор | Возможные параметры |
| Инициализация игры | ??? init\_game(???) | таймаут, игра, [возможно что-то еще] |
| Установка змейки | ??? set\_snake(???) | Массив поля игры, массив змейки [возможно что-то еще] |
| Очистка поля змейки | ??? clear\_snake(???) | Массив поля игры, массив змейки, [возможно что-то еще] |
| Проверка еды (что змейка съела еду) | ??? check\_eating(???) | Массив змейки, параметры еды, [возможно что-то еще] |
| Проверка змейки (что она заползла на границу). | ??? check\_snake(???) | Массив поля игры, массив змейки, [возможно что-то еще] |
| Установка еды | ???set\_food(???) | Массив поля игры, параметры еды, [возможно что-то еще] |
| Проверка конца игры | ??? check­\_game(???) | Можно проверить, что змейке некуда ползти, а также, что змейка укусила себя |

## Функции модуля snake

|  |  |  |
| --- | --- | --- |
| Описание | Идентификатор | Надо передать |
| Инициализация змейки | ??? init\_snake(???) | Начальную позицию змейки, массив змейки ,[возможно что-то еще] |
| Передвижение змейки | ??? move\_snake(???) | Новую позицию змейки, массив змейки,[возможно что-то еще] |

### Алгоритм функции init\_snake ( )

* Устанавливается начальный размер змейки.
* Заполняются значениями массив змейки по горизонтали и массив змейки по вертикали

Если размер змейки равен 1 (имеется только голова), то заполнение массивов очевидно. А если змейка кроме головы имеет еще хвост, то нужно подумать как заполнить массивы. Особенно интересно, если змейка может начинать движение в зависимости от направления ее движения.

## Управление змейкой

Для управления змейкой используются клавиши клавиатуры. Но в отличие от предыдущих работ, где змейка могла двигаться только по горизонтали и можно было задавать направление движения только в начале игры, в этой версии можно будет менять направление во время работы.

**Важно:**

* При изменении направления движения важно учитывать, что змейка может поворачивать только на 90 градусов. Она не может изменить направление на противоположное, т.е. не может повернуться на 180 градусов.
* Если голова змейки упирается в стенку, то движение змейки идет по кольцу, т.е. она появляется с другого (противоположного) края поля.

## Проверка конца игры

В предыдущих версиях программы игра завершалась, если змейка достигла своего максимального размера. Это условие можно оставить.

При нажатии клавиши ESC также происходит завершение работы программы.

Но добавим еще одно условие. Если змейка «укусила» себя, т.е. координаты головы совтадают с координатами тела, то также происходит завершение работы.

**Задание 4**

1. В модуле **game** дополните функцию, которая будет обрабатывать нажатие клавиш. В качестве управляющих клавиш Вы можете выбрать как клавиши ‘w’ ’a’ ‘s’ ‘d’, так и клавиши стрелочек. При нажатии на ESC программа должна завершаться.
2. Внесите изменения в имеющиеся функции: добавьте, где требуется, массив координат змейки, направление движения змейки.
3. Проверьте, что программа работает корректно:

* змейка умеет передвигаться в горизонтальном и вертикальном направлении (влево, вправо, вверх, вниз)
* змейка умеет корректно преодолевать препятствия (левую, правую, верхнюю, нижнюю границы)
* при поедании еды размер змейки увеличивается на 1
* [Если змейка достигла своего максимального размера (на поле нет места для установки еды), происходит завершение работы программы]
* Если змейка укусила себя, то происходит завершение работы программы.
* при нажатии клавиши ESC происходит завершение работы программы