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

Темы: Двумерные динамические массивы, функции с параметрами, структуры

В этом задании мы также будем изображать движущуюся змейку, которая может съедать «еду», движение змейки будет осуществляться в разных направлениях и им можно будет управлять. Игровое поле будет реализовано на базе двумерного динамического массива. В отличие от предыдущего проекта все данные, относящиеся к отдельным сущностям мы поместим в структуры.

Текущий проект будет разрабатываться на основе предыдущего, но корректировки в части описания данных будут существенными.

Вы уже заметили, что заголовки функций в предыдущем проекте увеличились и передавать большое количество параметров при вызове стало не удобно. Поэтому внесем изменения в организацию данных



**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".

## Данные модуля const

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Идентификатор | тип |  |
| 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 |  |  |

**Задание 2**

Объявите структуры, описывающие основные сущности программы.

## Данные программы

|  |  |  |
| --- | --- | --- |
| Описание | Идентификатор | тип |
| Структура snake:   * Текущий размер змейки * Массив координат змейки по горизонтали (X) * Массив координат змейки по вертикали (X) * Направление движения змейки |  | snake |
| Структура food:   * Координата еды по горизонтали (X) * Координата еды по вертикали (Y) * Признак, что еда установлена |  | food |
| Структура field:   * Массив игрового поля (двумерный динамический) * Длина игрового поля по горизонтали * Длина игрового поля по вертикали |  | field |
| Структура game:   * Таймаут задержки между шагами игры * Признак продолжения игры |  | Game |

## Данные (переменные) функции main

|  |  |  |
| --- | --- | --- |
| Описание | Идентификатор | Тип |
| Экземпляр структуры snake |  | Snake |
| Экземпляр структуры food |  | Food |
| Экземпляр структуры field |  | Field |
| Экземпляр структуры game |  | Game |
| Текущий размер змейки |  |  |
| Текущий признак, что еда установлена |  |  |
| Текущий таймаут задержки между шагами игры |  |  |
| Текущий признак продолжения игры |  |  |
| Текущее направление движения змейки |  |  |

**Задание 3**

Так как изменились данные программы, то придется вносить изменения в функции (в объявления и определения). Однако, сами алгоритмы изменяться не должны.

1. Внесите изменения в функции модулей **field** и **snake.**

## Функции модуля field

|  |  |  |
| --- | --- | --- |
| Описание | Идентификатор | Возможные параметры |
| Инициализация поля. Заполняется массив поля, в том числе границы | ??? init\_ field (???) | Структура Field, [возможно что-то еще] |
| Печать поля | ??? print\_field(???) | Структура Field, [возможно что-то еще] |
| Очистка поля | ???clear\_field(???) | Структура Field, [возможно что-то еще] |
| Создание динамического поля игры | ??? create\_array(???) | Структура Field, [возможно что-то еще] |
| Разрушение динамического поля игры | ??? destroy\_array(???) | Структура Field, [возможно что-то еще] |

### Алгоритм функции init\_field( )

* Создание динамического двумерного массива поля игры (вызов функции **create\_array**)
* граница поля заполняется символом границы .
* Вызов функции **clear\_field()**;

## Функции модуля snake

|  |  |  |
| --- | --- | --- |
| Описание | Идентификатор | Надо передать |
| Инициализация змейки | ??? init\_snake(???) | cтруктура Snake, [возможно что-то еще] |
| Передвижение змейки | ??? move\_snake(???) | cтруктура Snake, [возможно что-то еще] |
| . . . |  |  |

1. Измените объявления и определения функций модуля **game.**

## Функции модуля game

|  |  |  |
| --- | --- | --- |
| Описание | Идентификатор | Возможные параметры |
| Инициализация игры | ??? init\_game(???) | Структура Game, [возможно что-то еще] |
| Установка змейки | ??? set\_snake(???) | Структура Field, cтруктура Snake, [возможно что-то еще] |
| Очистка поля змейки | ??? clear\_snake(???) | Структура Field, cтруктура Snake, [возможно что-то еще] |
| Проверка еды (что змейка съела еду) | ??? check\_eating(???) | Структура Food, cтруктура Snake , [возможно что-то еще] |
| Проверка змейки (что она заползла на границу или укусила себя). | ??? check\_snake(???) | Структура Field, cтруктура Snake, [возможно что-то еще] |
| Установка еды | ???set\_food(???) | Структура Field, cтруктура Food, [возможно что-то еще] |
| Проверка конца игры | ??? check­\_game(???) | Структура Field, cтруктура Snake, [возможно что-то еще] |
| Обработка пользовательского ввода | ??? handle\_cmd(???) | cтруктура Snake, [возможно что-то еще] |

1. В функции **main** создайте экземпляры структур.
2. Откорректируйте вызовы функций
3. Проверьте корректность работы программы.