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

Темы: Двумерные динамические массивы, функции с параметрами, структуры, внедренные 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 и , кроме того, структура game будет содержать в качестве полей структуры snake, field, food.

Текущий проект будет разрабатываться на основе предыдущего, корректировки в части описания данных затронут только структуру game.



**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 * Экземпляр структуры snake * Экземпляр структуры food * Экземпляр структуры field |  | game |

## Данные (переменные) модуля main

|  |  |  |
| --- | --- | --- |
| Описание | Идентификатор | Тип |
| Экземпляр структуры game |  | Game |
| Текущий размер змейки |  |  |
| Таймаут задержки между шагами игры |  |  |
| Направление движения змейки |  |  |
| Длина игрового поля по горизонтали |  |  |
| Длина игрового поля по вертикали |  |  |
|  |  |  |

**Задание 3**

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

Так как теперь вся информация об объектах игры сосредоточена в структуре game, то логично практически весь код (вызов функций) перенести в модуль game и поместить в функции **next\_step()** и **exec\_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). | ??? 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] |
| Выполнение одного шага игры | ??? next\_step(???) | Структура Game , [возможно что-то еще] |
| Выполнение процесса игры | ??? exec\_game(???) | Структура Game , [возможно что-то еще] |

### Алгоритм функции init\_game

* инициализация игрового поля (вызов функции **init\_field ()**)
* инициализация змейки (вызов функции **init\_snake ()**)
* инициализация еды (вызов функции **init\_food ()**)
* “Установка” змейки в поле игры (вызов функции **set\_snake ()**)
* Вывод поля игры на экран (вызов функции **print\_field ()**)
* Устанавливаются начальные значения параметров игры.

### Алгоритм функции next\_step

* Установка еды на поле (вызов функции **set\_food ()**)
* Очистка старых позиций змейки в поле (вызов функции **clear\_snake ()**)
* Проверка не съела ли еду змейка (вызов функции **check\_eating ()**)
* Передвижение змейки (вызов функции **move\_snake ()**)
* Проверка не уперлась ли змейка в граничный элемент (вызов функции **check\_snake ()**)
* Установка новых позиций змейки в поле игры (вызов функции **set\_snake ()**)
* Вывод поля игры на экран (вызов функции **print\_field ()**)
* Проверка завершения игры (вызов функции **check\_game ()**)

### Алгоритм функции exec\_game

В цикле, пока идет игра выполняется алгоритм одного шага

* Вызов функции **next\_step()**
* Если работа идет в автоматическом режиме, то

Временная задержка (**Sleep()**)

Если нажата какая-либо клавиша, то

Обработка пользовательского ввода (вызов функции **handle\_cmd**())

* Если пошаговый режим работы, то

Обработка пользовательского ввода (вызов функции **handle\_cmd**())

### Алгоритм функции main (один из возможных)

1. Подготовка к игре
   1. инициализация игры (вызов функции **init\_game ()**)
   2. Вывод приглашения к игре
2. Запуск процесса игры (вызов функции **exec\_game ()**)
3. Завершение игры. Освобождение динамической памяти.

**Задание 4**

1. Измените объявления и определения функций модуля **game.**
2. Выполните ввод исходных данных в функции main()
3. Отредактируйте алгоритм функции **main()**
4. Проверьте корректность работы программы.