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

Темы: Двумерные динамические массивы, функции с параметрами.

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. Но в отличие от предыдущего проекта двумерное игровое поле будет реализовано на базе двумерного динамического массива.

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**

Для моделирования игрового поля будем использовать динамический двумерный массив типа **char**, строки которого соответствуют значениям координаты Y, а столбцы - соответствуют значениям координаты X. Массив хранит символы, отображающие: границу, змейку и еду. Использование динамического массива позволит задавать произвольные размеры поля в начале игры (а не использовать константы).

Так как в процессе игры размер поля меняться не будет, то имеет смысл создавать двумерный динамический массив по схеме «одномерный массив + вспомогательный массив указателей».

## 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 |  |  |

## Данные (переменные) модуля main

|  |  |  |
| --- | --- | --- |
| Описание | Идентификатор | Тип |
| Текущий размер змейки |  |  |
| Массив координат змейки по горизонтали (X) |  |  |
| Массив координат змейки по вертикали (X) |  |  |
| Массив игрового поля (динамический) |  |  |
| 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 |  |  |
| Длина игрового поля по горизонтали |  |  |
| Длина игрового поля по вертикали |  |  |

**Задание 3**

Так как теперь в качестве игрового поля будет использоваться двумерный динамический массив, 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. Скорее всего не потребуется менять и реализации функций.

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

1. Удалите константы rows и columns из файла const.h.
2. Прежде всего, в модуле main создайте динамический двумерный массив. Размеры массива задайте при вводе
3. Если максимальную длину змейки задавали константой, которая зависела от константных rows и columns, то придется сейчас задать абсолютное значение.
4. Внесите изменения в функции модуля **field.** Так как теперь будет использоваться динамический массив для игрового поля, то потребуется сообщить функциям его размеры.
5. Можно разработать отдельную функцию, которая будет создавать динамический двумерный массив, а также функцию, которая будет этот массив разрушать после окончания работы.

## 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] |
| Создание динамического поля игры | ??? create\_field(???) | Массив поля, [возможно что-то еще] |
| Разрушение динамического поля игры | ??? destroy\_field(???) |  |

1. Измените объявления и определения функций модуля **game.** Так как теперь будет использоваться динамический массив для игрового поля, то потребуется сообщить функциям его размеры.

## 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 или укусила себя). | ??? 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 |
| Обработка пользовательского ввода | ??? handle\_cmd(???) | Направление движения, признак окончания игры [возможно что-то еще] |

1. В функции модуля snake вносить изменения не придется, т.к. изменения затрагивают только игровое поле.
2. Не забудьте в конце работы программы очистить динамическую память.
3. Проверьте, что программа работает корректно:

* При запуске программы можно задавать размеры игрового поля с консоли.
* 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