

Программирование на языках высокого уровня

Лаб. работа 10. Обработка клавиш

Общие сведения

Требуется загрузить из файла двухмерный лабиринт и отобразить его на экране.

Установить позицию начальной точки в лабиринте, можно строку и столбец также загружать из файла. Далее, персонаж в виде прямоугольника определенного цвета должен двигаться в зависимости от нажатия клавиш-стрелок, при этом он не должен проходить сквозь стены лабиринта.

Общие сведения

На оценку «3» - задачи сложности А и В.

На оценку «4» - задачи сложности С

На оценку «5» - задачи сложности D

*Из лабораторной работы №5

**Значения элементов массива задавать
в программе

Задача А

Загрузить лабиринт и получить
возможность проходить по нему.

Задача В

Лабиринт должен быть дополнен точкой финиша, при достижении которой пользователь получает сообщение о выигрыше

Задача С

Программа должна содержать несколько загружаемых из файлов уровней, при прохождении одного должен автоматически загружаться следующий, и персонаж должен устанавливаться на начальную точку следующего уровня

Задача D

Аналогично задаче C, лабиринт должен содержать точки, наступление на которые приводит к проигрышу. Положение точек задается во входном файле. В случае проигрыша пользователь должен получить сообщение о произошедшем и начать игру заново.

Подсказка

1. Массив лабиринта, координаты персонажа следует хранить в глобальных переменных, объявленных до функций, в которых они используются
2. Для обработки нажатия клавиши в функции WndProc, нужно:

Объявить в функции wndproc переменную int key;

Внутри switch (message) добавить – рядом с case WM_COMMAND:

```
case WM_KEYDOWN:
```

```
key = wParam; //Получаем код нажатой клавиши
```

```
... //Собственно здесь делаем обработку нажатия клавиш
```

```
    //в переменной key хранится код клавиши, который при  
    нажатии стрелок может принимать значения VK_LEFT, VK_UP,  
    VK_DOWN, VK_RIGHT.
```

```
break;
```

```
... .
```


Подсказка

Для удобства обработки нажатия клавиш можно использовать следующую конструкцию:

```
switch(key)
{
    case VK_LEFT:
        ...//Делаем то, что нужно делать при нажатии стрелки
        влево
        break;
    case VK_RIGHT:
        ...//Делаем то, что нужно делать при нажатии стрелки
        влево
        break;
}
```

Подсказка

Для обновления окна:

```
InvalidateRect(hWnd, NULL, TRUE)
```