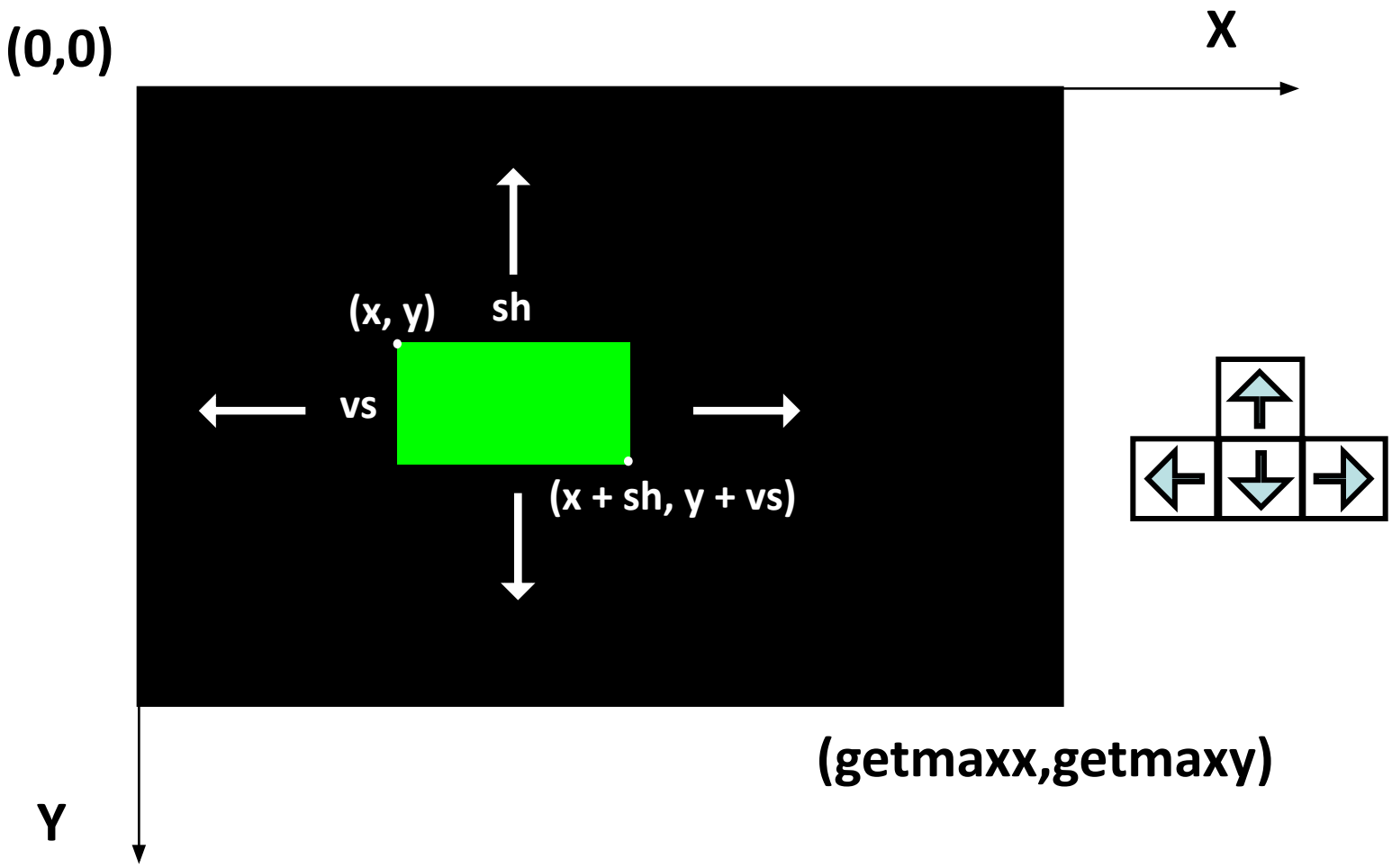
Основы языка Pascal

Управляемое движение объектов

Управляемое движение



Коды клавиш

Символы – это все буквы и значки, которые мы можем увидеть на клавиатуре. Для того чтобы ввести в программу символьные переменные необходимо указать для них символьный тип данных **`Char`**.

Проверить, была ли нажата клавиша, можно вызовом функции **`KeyPressed`**.

`ReadKey` считывает коды клавиш.

При нажатии некоторых специальных клавиш (стрелки и функциональные клавиши) **`ReadKey`** возвращает нулевой символ (#0), а повторный вызов **`ReadKey`** возвратит сканкод нажатой клавиши.

Функция **`Ord`**, преобразовывает букву в ее числовой код. Коды всех букв и символов можно посмотреть в кодовой таблице ASCII.

←	#0, #75	Esc	#27
↑	#0, #72	Space	#32
→	#0, #77	Enter	#13
↓	#0, #80	и др.	

Как узнать код клавиши

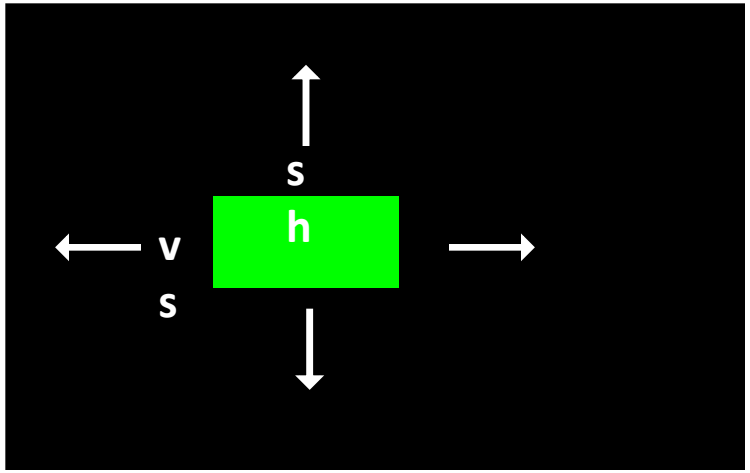
```
uses wingraph, wincrt;
const
  esc=#27;
var gd, gm: integer;
  ch: char; s: string;
begin
Initgraph(gd,gm,'');
repeat
  if keypressed then
  begin
    cleardevice;
    ch:=readkey;
    str(ord(ch),s);
    outtextxy(100,100,s);
  end;
until ch=esc;
closegraph;
end.
```

Процедура `Str` переводит целое число в строку, чтобы потом вывести её на экран с помощью `OutTextxy`

Анимация

Принцип анимации:

1. Нажимаем управляющую клавишу
2. стираем объект
3. изменяем координаты (x,y)
4. рисуем объект в точке (x,y)
5. задержка на несколько миллисекунд
6. переходим к шагу 1



Процедура (рисование и стирание)

```
procedure racket(x, y, sh, vs:integer; col:longint);
begin
  setfillstyle(1, col);
  bar(x, y, x + sh, y + vs);
end;
procedure Upr(var x, y: integer; sh, vs, h:integer;
col: longint);
begin
  racket(x, y, sh, vs, black);
  ch:=readkey;
  if ch=#0 then
  begin
    ch:=readkey;
    case ch of
      right: x := x + h;
    end;
  end;
  racket(x, y, sh, vs, col);
  Delay(50);
end;
```

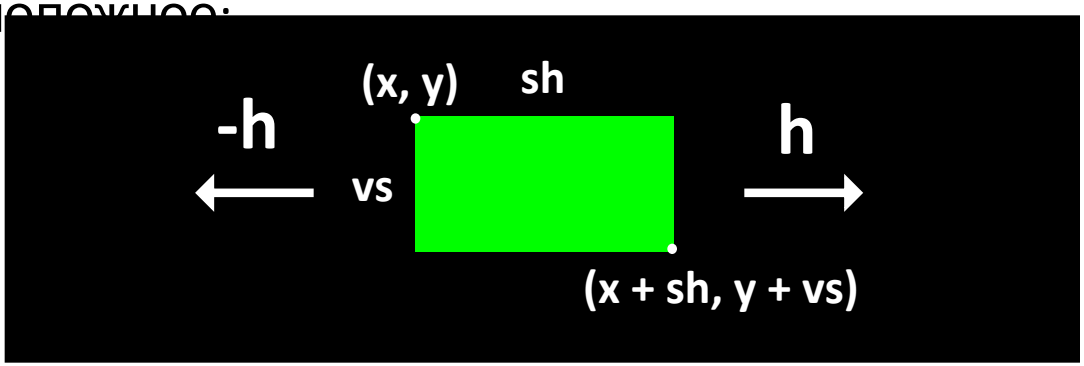
Стираем ракетку

Сдвиг по оси X на шаг h

Рисуем в новом месте цветом col

Обработка границ экрана

Приращение h при достижении границы меняет своё значение на противоположное:




```
procedure upr…
...
  case ch of
    left: if (x > 0) then x := x - h;
  end;
...
```

Полная программа

```
Uses wingraph, wincrt;
var x, y, sh, vs, h, gd, gm: integer; ch: char;
  procedure racket(x,y,sh,vs: integer; col: longint);
  begin
  ...
  end;
  procedure upr(var x, y: integer;
  sh, vs, h: integer; col: longint);
  begin
  ...
  end;
Begin
  Initgraph(gd, gm, '');
  x := 30;
  y := getmaxy div 2;
  sh := 70; vs := 20;
  h := 5;
  racket(x, y, sh, vs, yellow);
  repeat
    if keypressed then
    upr(x, y, sh, vs, h, yellow);
  until ch=esc;
  Closegraph;
end.
```

процедуры

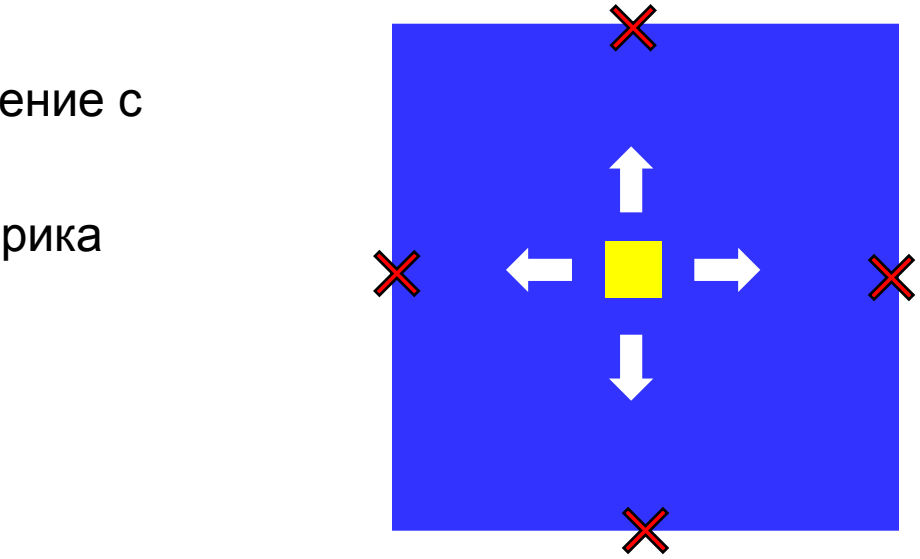
начальные условия

первая отрисовка ракетки

выход при нажатии клавиши Esc

Задание

1. Нарисовать ракетку (прямоугольник), управляемую стрелками, не выходящий заграницы экрана.
2. Нарисовать ракетку, управляемую стрелками, который не может выйти за границы прямоугольной области.
3. Нарисовать ракетку, которая движется непрерывно по экрану и отталкивается от границ экрана, при нажатии стрелок она меняет направление движения.
4. Нарисовать 2 управляемых ракетки, одна из них управляется стрелками, другая клавишами w, a, s, d.
5. Соединить неуправляемое движение с управляемым. Шарик + ракетка.
6. Добавить условия отбивания шарика от ракетки.



Объединение неуправляемого и управляемого движения

```
repeat
  neupr(x2, y2, hx, hy, r, yellow);
  if keypressed then
    upr(x1, y1, sh, vs, h, green)
until ch=esc;
```

Процедура (отрисовки круга)

По заданным координатам, радиусу и цвету отрисовываем круг в нужном месте:

```
procedure Krug(x, y, r: integer; col: longint);
begin
  setcolor(col);
  setfillstyle(1, col);
  fillellipse(x, y, r, r);
end;
```