

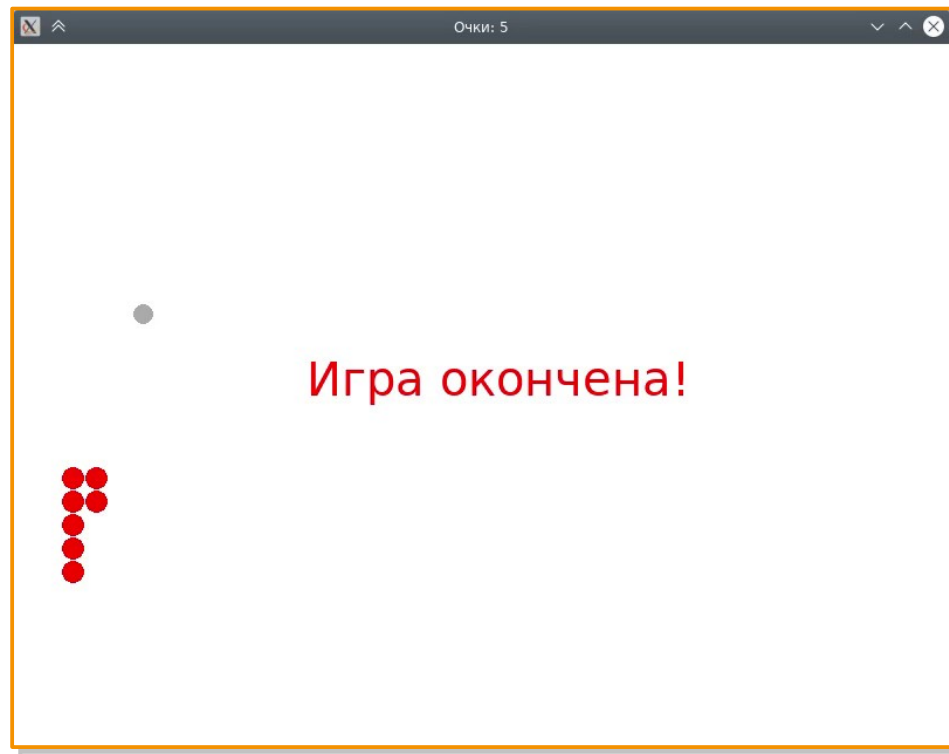
# Машинная графика в школе на свободных библиотеках проекта «qtSimpleGraph»

Проскурнёв Артём Сергеевич

ГБОУ Школа № 830

Москва

# Машинная графика



# Турбо Паскаль

```
File Edit Search Run Compile
[■]===== noname01.pas =====1=[:]
uses graph;
var
d,m:integer;
begin
  d:=detect;
  initgraph(d,m,'egavga.bgi');
  setcolor(15)
  circle(100,100,30);
  readln;
  closegraph;
end.
*===== 12:1 =====
F1 Help F2 Save F3 Open Alt+F9
```

```
File Edit Search Run Compile Debug Tools Options
[.]===== Programs\graphika6.pas =====
Program Graphika6;
Uses Graph;
Procedure Graphinterface;
Var gd, gm, error:Integer;
s:String;
Begin
gd:=detect;
s='';
Initgraph(gd,gm,s);
error:=GraphResult;
if error<>GrOk then
begin
writeln(GraphErrorMsg(Error));
Halt(error)
end
end;
begin
Graphinterface;
MoveTo(100,100);
OutText('PASCAL');
MoveRel(200,200);
OutText('PASCAL');
MoveRel(200,200);
OutText('PASCAL');
ReadLn;
CloseGraph;
End.
```

# Lazarus

The image shows the Lazarus IDE interface with two code editors and a form preview window.

**Left Editor (project1.lpr):**

```
1  unit Unit1;
.  {$mode objfpc}{$H+}
.  interface
.  uses
5  [ Classes, SysUtils, Forms, Controls, Graphics, Dialogs;
.  type
.  { TForm1 }
.  TForm1 = class(TForm)
.      procedure FormPaint(Sender: TObject);
10     private
.     public
.     end;
.  var
.      Form1: TForm1;
15  implementation
.  {$R *.lfm}
.  { TForm1 }
.  procedure TForm1.FormPaint(Sender: TObject);
.  begin
.  20     Canvas.Ellipse(70, 70, 130, 130);
.  end;
.  22  end.
.  23
```

**Right Editor (project1.lpr):**

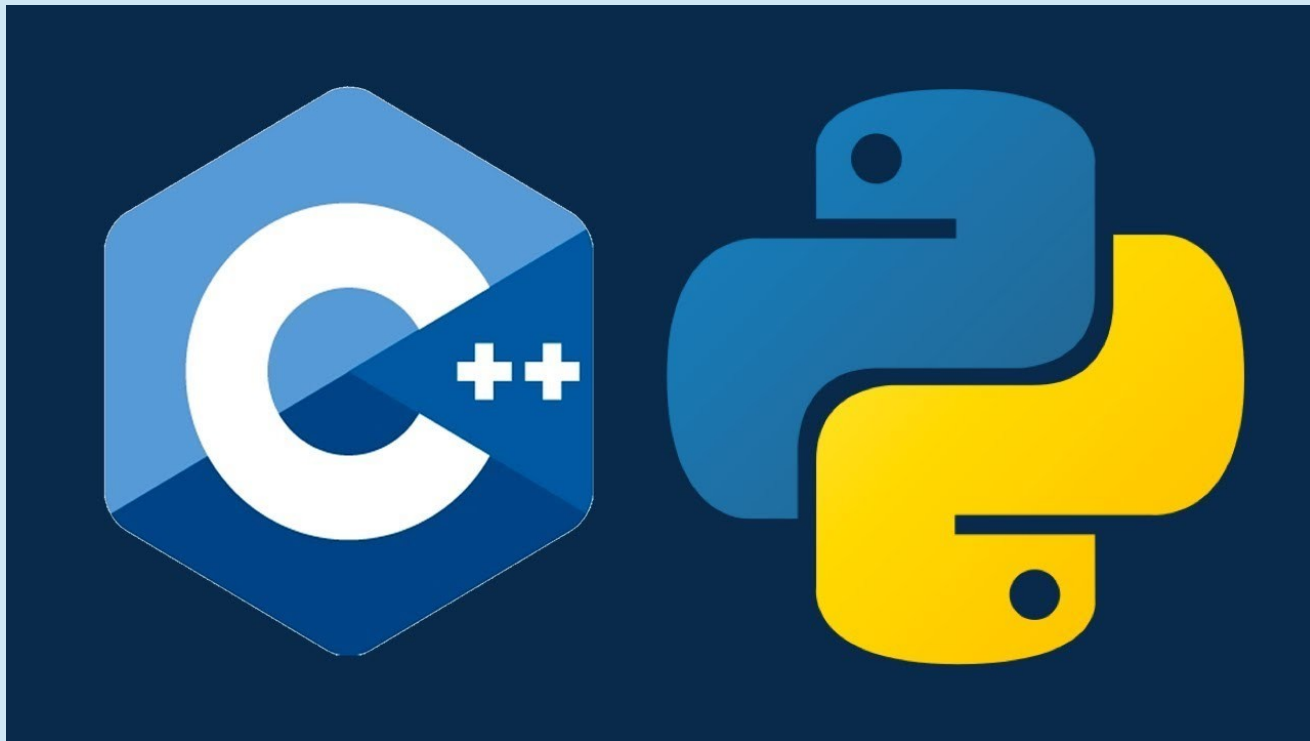
```
1  program Project1;
.
.  {$mode objfpc}{$H+}
.
.  uses
.  { $IFDEF UNIX }
.  cthreads,
.  { $ENDIF }
.  { $IFDEF HASAMIGA }
10  athreads,
.  { $ENDIF }
.  Interfaces, // this includes the LCL widgetset
.  Forms, unit1
.  { you can add units after this };
15
.  {$R *.res}
.
.  begin
.  RequireDerivedFormResource:=True;
20  Application.Scaled:=True;
.  Application.Initialize;
.  Application.CreateForm(TForm1, Form1);
.  Application.Run;
.  end.
25
```

**Form Preview:** A window titled "Form1" with a dark header bar containing a paw print icon and a back arrow. The main area is light gray and contains a white circle with a black outline, centered on the canvas.

# Линейное выполнение и взаимодействие

```
File Edit Search Run Compile D
[■]===== noname01.pas =====1=[:]
uses crt,graph;
var
d,m:integer;
begin
  d:=detect;
  initgraph(d,m,'egavga.bgi');
  putpixel(10,10,15);
  readkey;
  putpixel(20,20,15);
  readkey;
  putpixel(30,30,15);
  readkey;
  closegraph;
end.
*===== 15:1 =====
F1 Help F2 Save F3 Open Alt+F9 C
```

# Мейнстрим на уроках информатики



# tkinter Полякова К. Ю.

 [kpolyakov.spb.ru](http://kpolyakov.spb.ru)

Преподавание, наука и жизнь.

Поиск  Найти  
карта сайта поиск по тэгам

[главная](#) [школа](#) [вуз](#) [наука](#) [delphi](#) [программы](#) [походы](#) [автор](#) [Новости](#) [Блог](#)

[Программа](#)  
[Учебник 7-9](#)  
[Учебник 10-11\(Б+У\)](#)  
[Учебник 10-11\(У\)](#)  
▼ [Пособие \(Py, C++\)](#)  
    ▶ [Ссылки](#)  
[Конкурсы](#)  
[Презентации](#)  
[PDF \(9 класс\)](#)

Учебное пособие «Пр

## Ссылки

Ресурсы по язы

- [www.python.org](http://www.python.org)
- [wingware.com](http://wingware.com) —
- [sourceforge.net/](http://sourceforge.net/)
- [www.jetbrains.com/pycharm/](http://www.jetbrains.com/pycharm/) — *PyCharm* — среда программирования на Python, версия *Community* — бесплатная;
- [www.pyinstaller.org](http://www.pyinstaller.org) — программа *PyInstaller* для преобразования скриптов на языке Python в исполняемые файлы;
- [pypi.org/project/Pillow](http://pypi.org/project/Pillow) — библиотека *Pillow* для работы с изображениями в Python;

`line(x1, y1, x2, y2)`

нарисовать линию между точками с координатами (x1,y1) и (x2,y2); цвет линии определяется последней командой `penColor`; функция возвращает ссылку на объект-отрезок.

Особенность: **последняя точка линии не закрашивается.**

## ● модуль graph для создания простых графических пр

[Робототехника](#)  
[Arduino](#)  
[MMLogic](#)  
[Исполнители](#)  
[КуМир](#)  
[Язык Си](#)  
[Delphi](#)

- [www.pygame.org](http://www.pygame.org) — библиотека *PyGame* для программирования игр на языке Python;
- [interactivepython.org](http://interactivepython.org) — «Алгоритмы и структуры данных с использованием Python»;
- [younglinux.info/oopython.php](http://younglinux.info/oopython.php) — Python. Введение в объектно-ориентированное программирование;
- [wxpython.org](http://wxpython.org) — библиотека *wxPython* для разработки программ с графическим интерфейсом;
- [pygtk.org](http://pygtk.org) — библиотека *PyGTK* для разработки программ с графическим интерфейсом;
- [www.riverbankcomputing.com/software/pyqt/intro](http://www.riverbankcomputing.com/software/pyqt/intro) — библиотека *PyQt* для разработки программ с графическим интерфейсом;
- [younglinux.info/tkinter.php](http://younglinux.info/tkinter.php) — Tkinter. Программирование графического интерфейса;
- [effbot.org/tkinterbook/tkinter-index.htm](http://effbot.org/tkinterbook/tkinter-index.htm) — учебник по пакету *tkinter*;

```
from graph import *
def keyPressed(event):
    if event.keycode == VK_ESCAPE:
        close()
onKey(keyPressed)
run()
```

# Xcbwin

```
#include "xcbwin.h"
int main() {
```

```
Xcbwin Window;
```

```
Window.Open(400,400); //Opens a Window of size 400x400
```

```
Window.Black(); //Sets the drawing-color to black
```

```
Window.DrawRectangle(20,20,30,30); //Draws a Rectangle
```

```
Window.Wait(); // Waits for a keypress-event
```

```
Window.Screenshot(); //Saves a screenshot of the drawing
```

```
Window.Close(); //Closes the Window
```

```
}
```

```
case XCB_KEY_PRESS:
```

```
{
```

```
    // if any key was pressed, return
```

```
    xcb_key_press_event_t *bp = reinterpret_cast<xcb_key_press_event_t*> (event);
```

```
    uint8_t keycode = bp->detail;
```

```
    cout << bp->detail << endl;
```

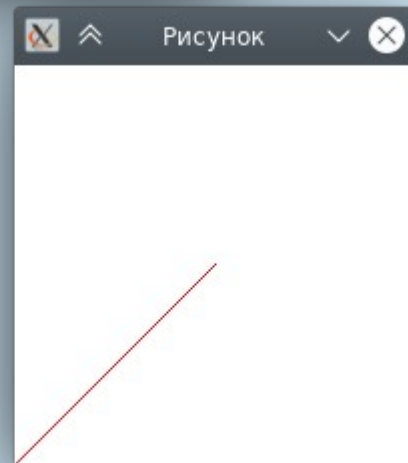
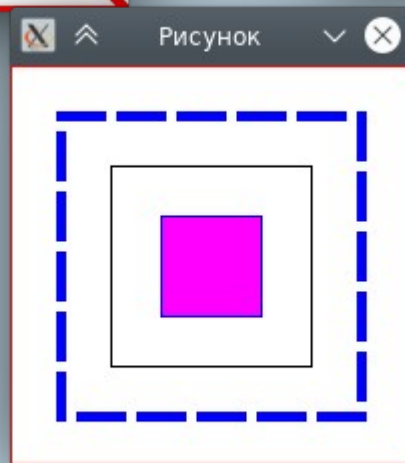
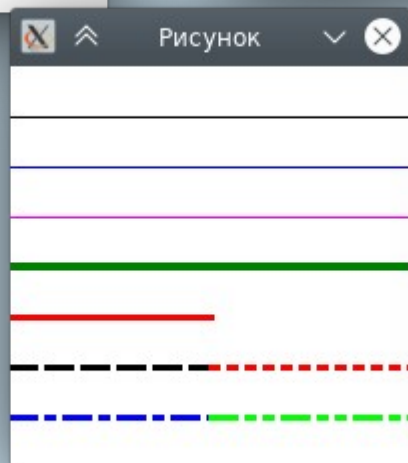
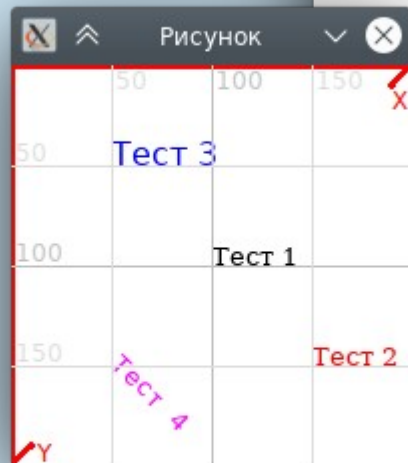
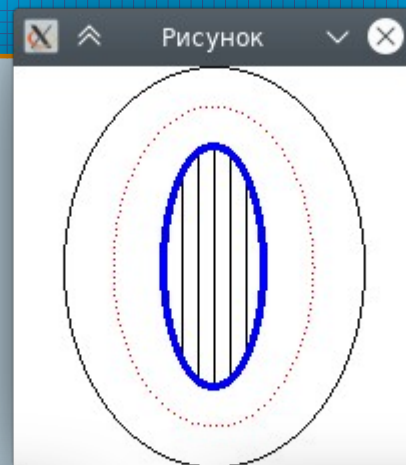
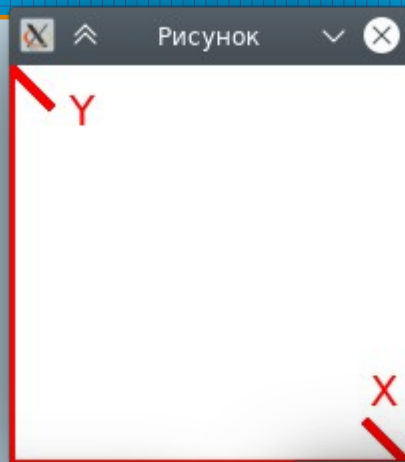
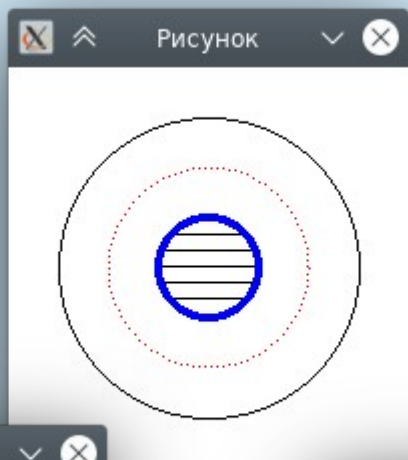
```
    free(event);
```

```
    return keycode;
```

```
}
```



# Qt Creator



# qtSimpleGraph

```
#include "qtsgraph.h"
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QTSGraph w(300, 300);
    w.show();
    return a.exec();
}
void QTSGraph::PaintBox()
{
    PutPixel(10,10,0);
    ReadKey();
    PutPixel(20,20,0);
    ReadKey();
    PutPixel(30,30,0);
    ReadKey();
}
```

```
File Edit Search Run Compi
[■]===== noname01.pas =====1=
uses crt,graph;
var
d,m:integer;
begin
    d:=detect;
    initgraph(d,m,'egavga.bgi');
    putpixel(10,10,15);
    readkey;
    putpixel(20,20,15);
    readkey;
    putpixel(30,30,15);
    readkey;
    closegraph;
end.
*===== 15:1 =====
```

# Основные решения

- Последовательное мгновенное выполнение инструкций без «заморозки» окна во время, например, паузы используется способ «прокрутить» ждущие события, пока окно рисования «стоит на паузе»:

```
QCoreApplication::processEvents(QEventLoop::AllEvents, 50);
```

- Сама же отрисовка производится на объект QPixmap Canvas, который просто «отпечатывается» потом на холст окна:

```
QPainter p(this);  
p.drawPixmap(0, 0, Canvas);
```

# Проблемы

- Если просто закрыть окно приложения во время выполнения цикла с условием выхода, например, **while(KeyPressed())**, то процесс не завершается

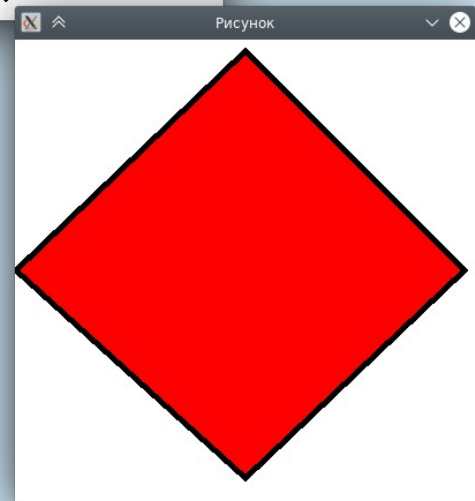
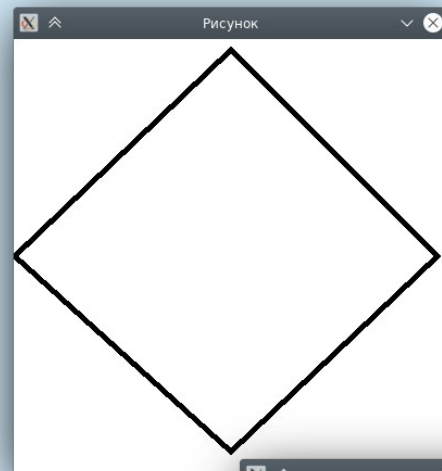
```
void QTSGraph::closeEvent(QCloseEvent *event)
{
    delete ResetTimer;
    delete StartTimer;
    exit(0);
}
```

# Работа с кодом модуля

- Реализация своего метода прямой, окружности
- Реализация заливки
- Реализация своих объектов для игр
- Изучение основ ООП

# Пример изучение рекурсии на заливке

```
.....  
void FloodFill(int x, int y, QRgb c = 0x00000000);  
.....  
• void QTSGraph::FloodFill(int x, int y, QRgb c)  
  {  
    QRgb z = GetPixel(x, y);  
    PutPixel(x, y, c);  
    if(GetPixel(x + 1, y) == z) FloodFill(x + 1, y, c);  
    if(GetPixel(x - 1, y) == z) FloodFill(x - 1, y, c);  
    if(GetPixel(x, y + 1) == z) FloodFill(x, y + 1, c);  
    if(GetPixel(x, y - 1) == z) FloodFill(x, y - 1, c);  
  }
```



# Игры

```
x[i] && yrabbit == y[i])
= true;

A);
,0);

ght && napr != 3) napr = 1;
wn && napr != 0) napr = 2;
ft && napr != 1) napr = 3;
&& napr != 2) napr = 0;

hite);
,r);
--)

+= 20;
+= 20;
-= 20;
-= 20;

ed);
);
x[0] <= 0 || y[0]>=600 || y[0]<=0)break;
i++)

] && y[0] == y[i])
```

[github.com/temaps/qtSimpleGraph](https://github.com/temaps/qtSimpleGraph)



[github.com/temaps/qtSimpleGraphPy](https://github.com/temaps/qtSimpleGraphPy)

