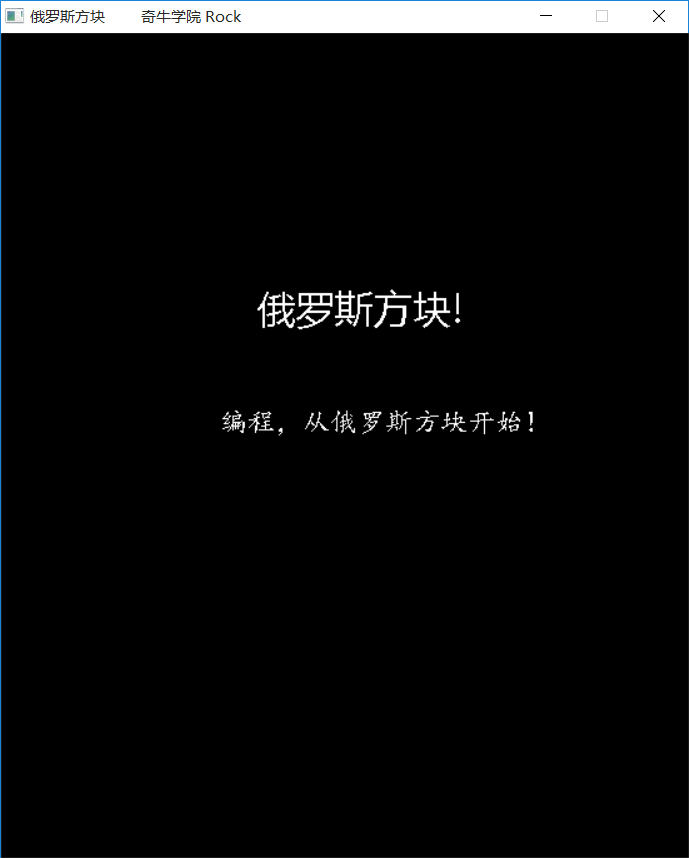
# 项目分析

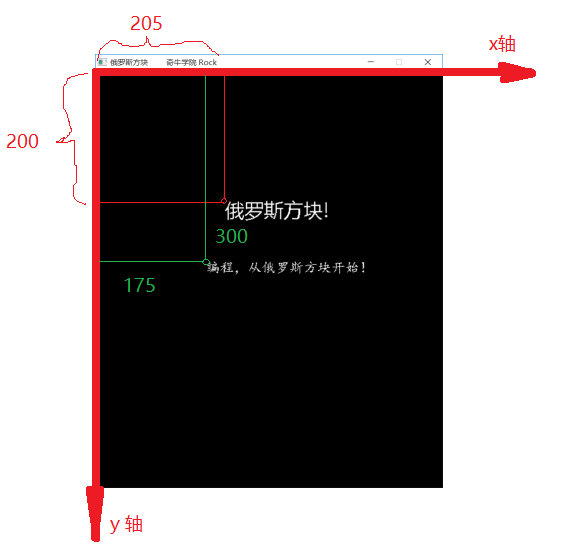
# 项目演示、项目分析

# 启动页面

启动页面



分析：



开发环境搭建

1）安装vc++2010, 或其他vs版本

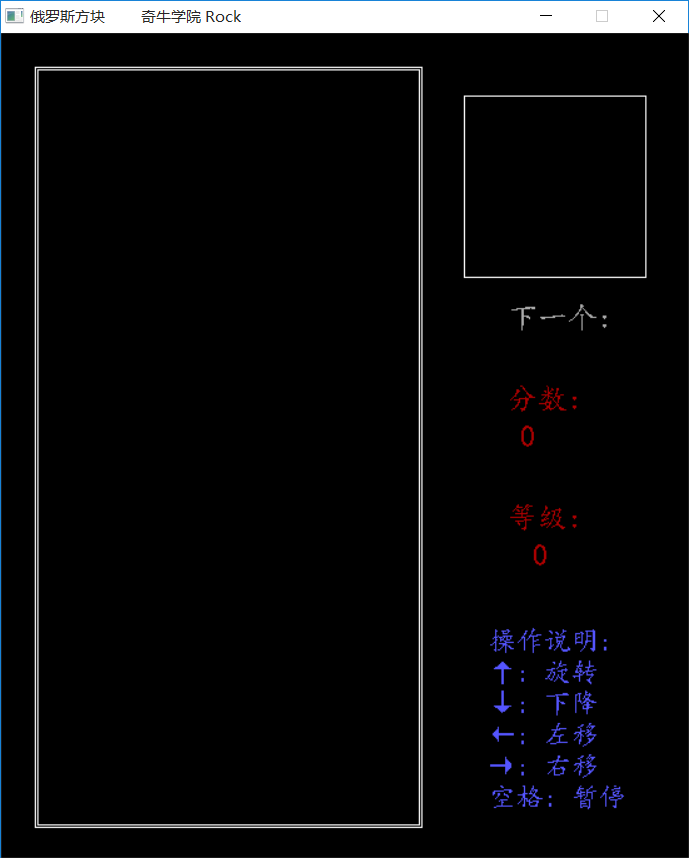
2）安装easyX图形库

代码实现:

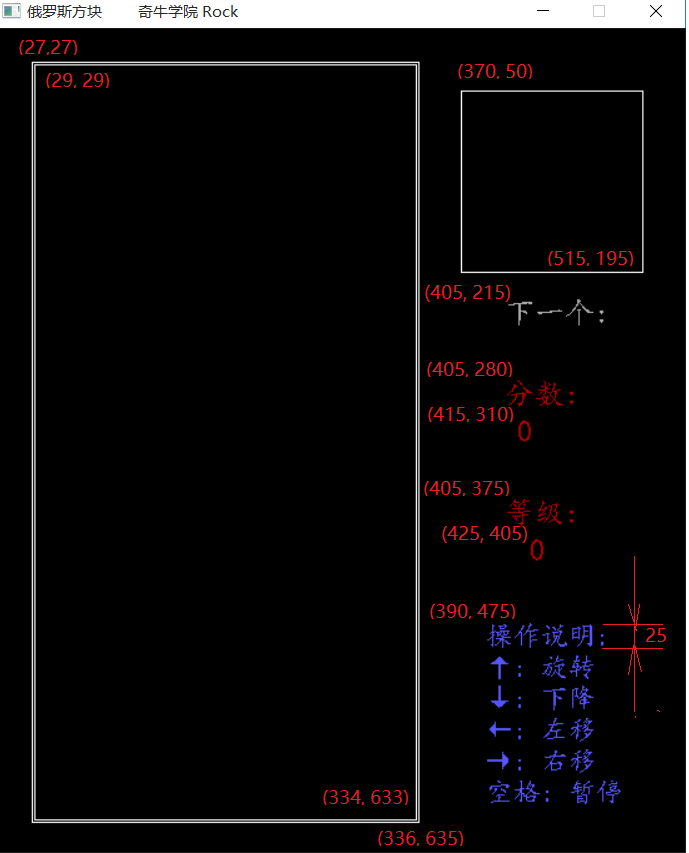
|  |
| --- |
| # include <stdio.h>  # include <graphics.h>  void welcome(void) {  initgraph(550, 660);  // 设置窗口标题  HWND hwnd = GetHWnd();  SetWindowText(hwnd, "俄罗斯方块 奇牛学院 Rock");  //Sleep(2000);  // 游戏标题  setfont(40, 0, "微软雅黑");  setcolor(WHITE);  outtextxy(205, 200, "俄罗斯方块!");  // 游戏副标题  setfont(22, 0, "楷体");  outtextxy(175, 300, "编程，从俄罗斯方块开始！");  Sleep(3000);  }  int main()  {  welcome();    closegraph();  return 0;  } |

# 初始化游戏环境

效果：



分析：

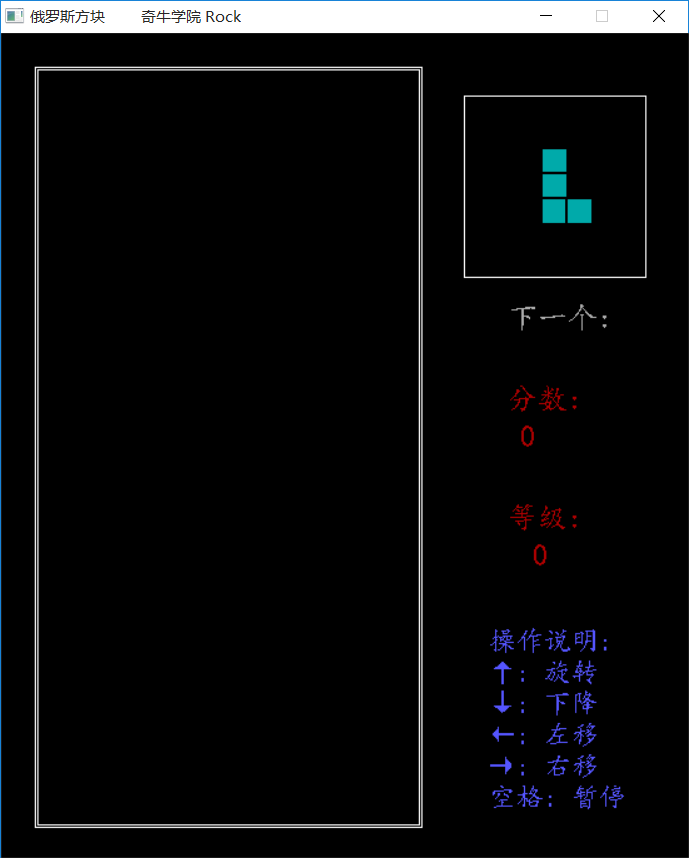


实现：

|  |
| --- |
| int score = 0; // 总分  int rank = 0; //等级  void initGameScene()  {  char str[16];    cleardevice();  setcolor(WHITE);  rectangle(29, 29, 334, 633);  rectangle(27, 27, 336, 635);  rectangle(370, 50, 515, 195);  setfont(24, 0, "楷体");  setcolor(LIGHTGRAY);  outtextxy(405, 215, "下一个:");  setcolor(RED);  outtextxy(405, 280, "分数:");  sprintf(str, "%d", score);  outtextxy(415, 310, str);  outtextxy(405, 375, "等级:");  sprintf(str, "%d", rank);  outtextxy(425, 405, str);  setfont(22, 0, "楷体");  setcolor(LIGHTBLUE);  outtextxy(390, 475, "操作说明:");  outtextxy(390, 500, "↑: 旋转");  outtextxy(390, 525, "↓: 下降");  outtextxy(390, 550, "←: 左移");  outtextxy(390, 575, "→: 右移");  outtextxy(390, 600, "空格: 暂停");  }  int main()  {  welcome();  initGameScene();  system("pause");  closegraph();  return 0;  } |

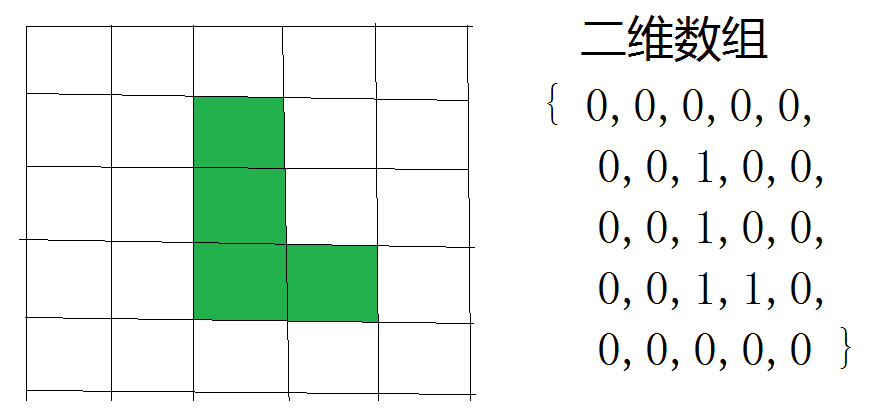
# 新方块

效果：



分析：

L型方块



每个方块有4种形态：

4个方向，所以使用4个二维数组来表示1种方块

|  |
| --- |
| { 0,0,0,0,0,  0,0,1,0,0,  0,0,1,0,0,  0,0,1,1,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,0,0,0,0,  0,1,1,1,0,  0,1,0,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,1,1,0,0,  0,0,1,0,0,  0,0,1,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,0,0,1,0,  0,1,1,1,0,  0,0,0,0,0,  0,0,0,0,0 } |

实现：

|  |
| --- |
| #define BLOCK\_COUNT 5  #define BLOCK\_WIDTH 5  #define BLOCK\_HEIGHT 5  #define UNIT\_SIZE 20 //小砖块的宽度和高度  int color[BLOCK\_COUNT] = {  GREEN,CYAN,MAGENTA,BROWN,YELLOW  };  int NextIndex = -1;  int block[BLOCK\_COUNT \* 4][BLOCK\_HEIGHT][BLOCK\_WIDTH] = {  // | 形方块  { 0,0,0,0,0,  0,0,1,0,0,  0,0,1,0,0,  0,0,1,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,0,0,0,0,  0,1,1,1,0,  0,0,0,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,0,1,0,0,  0,0,1,0,0,  0,0,1,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,0,0,0,0,  0,1,1,1,0,  0,0,0,0,0,  0,0,0,0,0 },  // L 形方块  { 0,0,0,0,0,  0,0,1,0,0,  0,0,1,0,0,  0,0,1,1,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,0,0,0,0,  0,1,1,1,0,  0,1,0,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,1,1,0,0,  0,0,1,0,0,  0,0,1,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,0,0,1,0,  0,1,1,1,0,  0,0,0,0,0,  0,0,0,0,0 },  // 田 形方块  { 0,0,0,0,0,  0,1,1,0,0,  0,1,1,0,0,  0,0,0,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,1,1,0,0,  0,1,1,0,0,  0,0,0,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,1,1,0,0,  0,1,1,0,0,  0,0,0,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,1,1,0,0,  0,1,1,0,0,  0,0,0,0,0,  0,0,0,0,0 },  // T 形方块  { 0,0,0,0,0,  0,1,1,1,0,  0,0,1,0,0,  0,0,0,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,0,0,1,0,  0,0,1,1,0,  0,0,0,1,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,0,1,0,0,  0,1,1,1,0,  0,0,0,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,1,0,0,0,  0,1,1,0,0,  0,1,0,0,0,  0,0,0,0,0 },  // Z 形方块  { 0,0,0,0,0,  0,1,1,0,0,  0,0,1,1,0,  0,0,0,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,0,1,0,0,  0,1,1,0,0,  0,1,0,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,1,1,0,0,  0,0,1,1,0,  0,0,0,0,0,  0,0,0,0,0 },  { 0,0,0,0,0,  0,0,1,0,0,  0,1,1,0,0,  0,1,0,0,0,  0,0,0,0,0 },  };  void clearBlock(int x, int y) {  setcolor(BLACK);  setfont(23, 0, "楷体");  for (int i = 0; i<BLOCK\_HEIGHT; i++)  for (int j = 0; j<BLOCK\_WIDTH; j++)  outtextxy(x + UNIT\_SIZE\*j, y + UNIT\_SIZE\*i, "■");  }  void drawBlock(int x, int y, int next) {  setfont(23, 0, "楷体");  setcolor(color[NextIndex]);  for (int i = 0; i<5; i++)  for (int j = 0; j<5; j++)  if (block[4 \* NextIndex][i][j] == 1)  outtextxy(x + 20 \* j, y + 20 \* i, "■");  }  void nextblock()  {  int x = 391, y = 71;  //在右侧的提示区清除原来的方块  clearBlock(x, y);    // 在右侧的提示区绘制新方块  // 1. 产生新的方块  srand(time(NULL));  NextIndex = rand() % BLOCK\_COUNT;  // 2. 绘制  drawBlock(x, y, NextIndex);  }  int main()  {  welcome();  initGameScene();  nextblock();  system("pause");  closegraph();  return 0;  } |

# 降落方块

## 5.1）使用“访问数组”来确定是否有方块

|  |
| --- |
| int visit[30][15], Color[30][15]; // visit[i][j] == 1 表示该位置有方块  int main()  {  welcome();  initGameScene();  nextblock();  // 清空访问数组  Sleep(500);  memset(visit, 0, sizeof(visit));  system("pause");  closegraph();  return 0;  } |

## 5.2）设计游戏循环

|  |
| --- |
| int BlockIndex = -1; //当前方块的序号  void newblock() { //新方块下降  }  int main()  {  welcome();  initGameScene();  nextblock();  // 清空访问数组  Sleep(500);  memset(visit, 0, sizeof(visit));  // 最开始时， 第一个方块，就是下一个方块  BlockIndex = NextIndex;  while (1)  {  newblock();  }  system("pause");  closegraph();  return 0;  } |

## 5.3) 方块下降

详见视频。

# 消行检测

详见视频。

# 更新积分

# 进一步优化

1. 游戏音效
2. 消除行的音效
3. 多行同时消除的附加音效
4. 下降时的音效
5. 网络对战功能

双人对战

多人对战

1. 3D化