A. Lý thuyết

Bài 1:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bresenham | |  |  |  |  |  | MidPoint | |  |  |  |
| a | b | Xt | Yt |  |  |  | a | b | Xt | Yt |  |
| 10 | 8 | 7.808688094 | 4.99756038 |  |  |  | 10 | 8 | 7.808688094 | 4.99756038 |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Cung AC** |  |  |  |  |  |  | **Cung AC** |  |  |  |  |
| **x** | **y** |  |  | **P** |  |  | **x** | **y** |  |  | **D** |
| **0** | **8** | **P1** | **10^2\*(1-2\*8)+2\*8^2** | **-1372** |  |  | **0** | **8** | **D1** | **(8^2)-(10^2)\*(8-0.25)** | **-711** |
| **1** | **8** | **P2** | **-1372+2\*(8^2)\*(2\*0+3)** | **-988** |  |  | **1** | **8** | **D2** | **-711+(8^2)\*(3+2\*0)** | **-519** |
| **2** | **8** | **P3** | **-988+2\*(8^2)\*(2\*1+3)** | **-348** |  |  | **2** | **8** | **D3** | **-519+(8^2)\*(3+2\*1)** | **-199** |
| **3** | **8** | **P4** | **-348+2\*(8^2)\*(2\*2+3)** | **548** |  |  | **3** | **8** | **D4** | **-199+(8^2)\*(3+2\*2)** | **249** |
| **4** | **7** | **P5** | **548+2\*(8^2)\*(2\*3+3)+4\*(10^2)\*(1-8)** | **-1100** |  |  | **4** | **7** | **D5** | **249+(8^2)\*(3+2\*3)+(10^2)\*(2-2\*8)** | **-575** |
| **5** | **7** | **P6** | **-1100+2\*(8^2)\*(2\*4+3)** | **308** |  |  | **5** | **7** | **D6** | **-575+(8^2)\*(3+2\*4)** | **129** |
| **6** | **6** | **P7** | **308+2\*(8^2)\*(2\*5+3)+4\*(10^2)\*(1-7)** | **-428** |  |  | **6** | **6** | **D7** | **129+(8^2)\*(3+2\*5)+(10^2)\*(2-2\*7)** | **-239** |
| **7** | **6** | **P8** | **-428+2\*(8^2)\*(2\*6+3)** | **1492** |  |  | **7** | **6** | **D8** | **-239+(8^2)\*(3+2\*6)** | **721** |
| **8** | **5** | **P9** |  |  |  |  | **8** | **5** | **D9** |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Cung CB** |  |  |  |  |  |  | **Cung CB** |  |  |  |  |
| **x** | **y** |  |  | **P** |  |  | **x** | **y** |  |  | **P** |
| **10** | **0** | **P1** | **(8^2)\*(1-2\*10)+2\*(10^2)** | **-1016** |  |  | **10** | **0** | **D1** | **(10^2)-(8^2)\*(10-0.25)** | **-524** |
| **10** | **1** | **P2** | **-1016+2\*(10^2)\*(2\*0+3)** | **-416** |  |  | **10** | **1** | **D2** | **-524+(10^2)\*(3+2\*0)** | **-224** |
| **10** | **2** | **P3** | **-416+2\*(10^2)\*(2\*1+3)** | **584** |  |  | **10** | **2** | **D3** | **-224+(10^2)\*(3+2\*1)** | **276** |
| **9** | **3** | **P4** | **584+2\*(10^2)\*(2\*2+3)+4\*(8^2)\*(1-10)** | **-320** |  |  | **9** | **3** | **D4** | **276+(10^2)\*(3+2\*2)+(8^2)\*(2-2\*10)** | **-176** |
| **9** | **4** | **P5** | **-320+2\*(10^2)\*(2\*3+2)** | **1280** |  |  | **9** | **4** | **D5** | **-176+(10^2)\*(3+2\*3)** | **724** |
| **8** | **5** | **P6** |  |  |  |  | **8** | **5** | **D6** |  |  |

Bài 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MidPoint | |  |  |  |
| a | b | Xt | Yt |  |
| 10 | 8 | 7.808688094 | 4.99756038 |  |
|  |  |  |  |  |
| **Cung AC** |  |  |  |  |
| **x** | **y** |  |  | **D** |
| **0** | **8** | **D1** | **(8^2)-(10^2)\*(8-0.25)** | **-711** |
| **1** | **8** | **D2** | **-711+(8^2)\*(3+2\*0)** | **-519** |
| **2** | **8** | **D3** | **-519+(8^2)\*(3+2\*1)** | **-199** |
| **3** | **8** | **D4** | **-199+(8^2)\*(3+2\*2)** | **249** |
| **4** | **7** | **D5** | **249+(8^2)\*(3+2\*3)+(10^2)\*(2-2\*8)** | **-575** |
| **5** | **7** | **D6** | **-575+(8^2)\*(3+2\*4)** | **129** |
| **6** | **6** | **D7** | **129+(8^2)\*(3+2\*5)+(10^2)\*(2-2\*7)** | **-239** |
| **7** | **6** | **D8** | **-239+(8^2)\*(3+2\*6)** | **721** |
| **8** | **5** | **D9** |  |  |
|  |  |  |  |  |
| **Cung CB** |  |  |  |  |
| **x** | **y** |  |  | **P** |
| **10** | **0** | **D1** | **(10^2)-(8^2)\*(10-0.25)** | **-524** |
| **10** | **1** | **D2** | **-524+(10^2)\*(3+2\*0)** | **-224** |
| **10** | **2** | **D3** | **-224+(10^2)\*(3+2\*1)** | **276** |
| **9** | **3** | **D4** | **276+(10^2)\*(3+2\*2)+(8^2)\*(2-2\*10)** | **-176** |
| **9** | **4** | **D5** | **-176+(10^2)\*(3+2\*3)** | **724** |
| **8** | **5** | **D6** |  |  |

**Bài 3**: Hãy so sánh 2 giải thuật và 2 chương trình Bresenham và Midpoint để vẽ đường ellipse.

1. Thuật toán Bresenham:

**Ưu điểm:**

Giải thuật Bresenham chủ yếu được thiết kế để vẽ các đoạn đường thẳng, nhưng có thể được áp dụng để vẽ ellipse.

Độ phức tạp thời gian của thuật toán Bresenham thấp hơn so với Midpoint trong một số trường hợp.

**Nhược điểm:**

Cần phải tính toán các giá trị đạt phương sai, điều này có thể tốn kém từ quan điểm tính toán.

2. Thuật toán Midpoint:

**Ưu điểm:**

Giải thuật Midpoint được thiết kế trực tiếp cho việc vẽ ellipse, giảm độ phức tạp tính toán so với Bresenham.

Có thể hiệu quả hơn đối với vẽ ellipse so với thuật toán Bresenham.

**Nhược điểm:**

Dù có độ phức tạp thời gian thấp hơn so với Bresenham trong một số trường hợp, nhưng độ phức tạp thời gian của Midpoint vẫn tăng theo kích thước của ellipse.

B. Thực hành

#**include** <GL/glut.h>  
#**include** <iostream>  
#**include** <stdlib.h>  
  
**struct** **Point** {  
 **int** x;  
 **int** y;  
};  
  
**struct** **Polygon** {  
 **int** Vertex;  
 Point\* Vertexs;  
};  
  
Point\* **Input**(**int**& Vertex) {  
  
 **do** {  
 system("cls");  
 std::cout << "Nhap so dinh: ";  
 std::cin >> Vertex;  
 } **while** (Vertex <= 3);  
  
 Point\* Vertexs = **new** Point[Vertex];  
  
 **for** (**int** i = 0; i < Vertex; i++) {  
 std::cout << "x: ";  
 std::cin >> Vertexs[i].x;  
  
 std::cout << "y: ";  
 std::cin >> Vertexs[i].y;  
  
 std::cout << std::endl;  
 }  
  
 **if** (Vertexs != nullptr) **return** Vertexs;  
}  
  
**void** **DrawPolygon**() {  
 **int** Vertex;  
 Point\* Vertexs = Input(Vertex);  
  
 glClear(GL\_COLOR\_BUFFER\_BIT);  
 glColor3f(1.0f, 0, 0);  
 glBegin(GL\_POLYGON);  
  
 **for** (**int** i = 0; i < Vertex; i++) {  
 glVertex2i(Vertexs[i].x, Vertexs[i].y);  
 }  
  
 glEnd();  
}  
  
**void** **display**() {  
 DrawPolygon();  
 glFlush();  
}  
  
**int** **main**(**int** agrc, **char**\* agrv[]) {  
 glutInit(&agrc, agrv);  
 glutInitWindowPosition(0, 0);  
 glutInitWindowSize(500, 500);  
 glutCreateWindow("Bai 1");  
 gluOrtho2D(-250, 250, -300, 300);  
 glutDisplayFunc(display);  
 glutMainLoop();  
}

BAI 2:

#**include** <GL/glut.h>  
  
int x1 = 250, x2 = 300;  
  
void idle() {  
 glutPostRedisplay();  
}  
  
void display() {  
 glClear(GL\_COLOR\_BUFFER\_BIT);  
 glColor3f(0, 1.0f, 1.0f);  
  
 glBegin(GL\_TRIANGLES);  
 glVertex2i(0, x2);  
 glVertex2i(35, x1);  
 glVertex2i(-35, x1);  
 glEnd();  
  
 x2--;  
 x1--;  
  
 glFlush();  
}  
  
int main(int agrc, char\* agrv[]) {  
 glutInit(&agrc, agrv);  
 glutInitWindowPosition(0, 0);  
 glutInitWindowSize(500, 500);  
 glutCreateWindow("Bai 2");  
 gluOrtho2D(-250, 250, -300, 300);  
 glutDisplayFunc(display);  
 glutIdleFunc(idle);  
 glutMainLoop();  
}

BAI 3:

#include <GL/glut.h>  
#include <stdlib.h>  
#include <cmath>  
  
#define PI 3.14159265  
  
int SpecialKey;  
  
void OnKeyDown(int key,int x, int y) {  
 SpecialKey = key;  
  
 **if** (SpecialKey == GLUT\_KEY\_END) {  
 exit(0);  
 } **else** {  
 glutPostRedisplay();  
 }  
}  
  
  
void display() {  
 glClear(GL\_COLOR\_BUFFER\_BIT);  
 glColor3f(1.0f, 0, 0);  
  
 glBegin(GL\_TRIANGLES);  
  
 **if** (SpecialKey == GLUT\_KEY\_DOWN) {  
 glVertex2i(100, 0);  
 glVertex2i(0, 200);  
 glVertex2i(200, 200);  
 }  
 **else** **if** (SpecialKey == GLUT\_KEY\_UP) {  
 glVertex2i(0, 0);  
 glVertex2i(200, 0);  
 glVertex2i(100, 200);  
 }  
 **else** **if** (SpecialKey == GLUT\_KEY\_RIGHT) {  
 glVertex2i(200, 100);  
 glVertex2i(0, 0);  
 glVertex2i(0, 200);  
  
 }  
 **else** **if** (SpecialKey == GLUT\_KEY\_LEFT) {  
 glVertex2i(0, 100);  
 glVertex2i(200, 0);  
 glVertex2i(200, 200);  
 }  
 **else** {  
 glVertex2i(0, 0);  
 glVertex2i(200, 0);  
 glVertex2i(100, 200);  
 }  
  
 glEnd();  
 glFlush();  
}  
  
int main(int agrc, char\* agrv[]) {  
 glutInit(&agrc, agrv);  
 glutInitWindowPosition(0, 0);  
 glutInitWindowSize(500, 500);  
 glutCreateWindow("Bai 3");  
 gluOrtho2D(-250, 250, -300, 300);  
 glutDisplayFunc(display);  
 glutSpecialFunc(OnKeyDown);  
 glutMainLoop();  
}

BAI 4:

#**include** <GL/glut.h>  
  
int x1 = 250, x2 = 300;  
  
void Timer(int value) {  
 glutPostRedisplay();  
 glutTimerFunc(30, Timer, 0);  
}  
  
void display() {  
 glClear(GL\_COLOR\_BUFFER\_BIT);  
 glColor3f(0, 1.0f, 1.0f);  
  
 glBegin(GL\_TRIANGLES);  
 glVertex2i(0, x2);  
 glVertex2i(35, x1);  
 glVertex2i(-35, x1);  
 glEnd();  
  
 x2--;  
 x1--;  
  
 glFlush();  
}  
  
int main(int agrc, char\* agrv[]) {  
 glutInit(&agrc, agrv);  
 glutInitWindowPosition(0, 0);  
 glutInitWindowSize(500, 500);  
 glutCreateWindow("Bai 2");  
 gluOrtho2D(-250, 250, -300, 300);  
 glutDisplayFunc(display);  
 glutTimerFunc(0, Timer, 0);  
 glutMainLoop();  
}