

**Báo cáo đồ họa máy tính**

Giảng viên hướng dẫn: Ts. Phạm Xuân Hiền



May 2, 2019

NHÓM 21

Đồ họa máy tính

**Đề tài: Vùng quê tôi**

**Người thực hiện:**

Lê Thanh Lương B1609830

Trần Sĩ Đạt B1611134



**Lời nói đầu**

*Đồ họa máy tính là một lĩnh vực của khoa học máy tính nghiên cứu về cơ sở toán học, các thuật toán cũng như các kĩ thuật cho phép tạo, hiển thị và điều khiển hình ảnh trên màn hình máy tính. Đồ họa máy tính liên quan ít nhiều đến một số lĩnh vực như đại số, hình học giải tích và hình học họa hình, quang học……và kĩ thuật máy tính và đặc biệt là chế tạo phần cứng (các loại màn hình, các thiết bị nhập xuất, các vi mạch đồ họa).*

*Nghiên cứu các phép biến đổi hình học trong không gian thực hai chiều là một trong những nội dung quan trọng.*

*Qua thời gian tìm hiểu nhóm quyết định làm đề tài xây dưng phong cảnh hoạt hình và chuyển động (vùng quê tôi).*

*Do quá trình học và tìm hiểu còn nhiều thiếu sót nên chương trình còn nhiều hạn chế, nhóm thực hiện mong nhận được ý kiến đóng góp từ phía cô và các bạn để chương trình được hoàn thiện hơn.*

**CHƯƠNG 1: GiỚI THIỆU**

1. **Đặt vấn đề**

Đề tài vùng quê của tôi cần sử dụng những hàm nào, các phép vẽ hình cơ bản , sử dụng bộ đếm thời gian, bảng giá trị màu phù hợp với từng đối tượng vẽ, bố cục vị trí của từng đối tượng,..

1. **Lịch sử giải quyết vấn đề**

Dựa vào những kiến thức được học trên lớp, và đọc tham khảo các tài liệu thì để tạo ra các hình ảnh 2D, các chuyển động cần sử dụng các phép biến đổi hai chiều cơ bản như tịnh tiến, quay và biến đổi tỉ lệ, hàm random, bộ đếm thời gian (timer) để tạo ra được đối tượng di chuyển và hình ảnh mong muốn.

1. **Mục tiêu của đề tài**

Kiểm tra kiến thức đã học trên lớp, ứng dụng những hàm và thuật toán để áp dụng vào thực tế, góp phần củng cố kiến thức, nắm bắt hiểu sâu rộng hơn các hàm và thuật toán, xây dựng các đối tượng đồ họa 2D cơ bản, các hình ảnh các chuyển động như mong muốn.

1. **Đối tượng và phạm vi nghiên cứu**

**Đối tượng nghiên cứu:** không gian màu, các hệ tọa độ, các đối tượng đồ họa cơ sở, các thuật toán tô màu, các phép biến đổi hai chiều cơ bản như tịnh tiến, quay và biến đổi tỉ lệ.

**Phạm vi nghiên cứu:** đề tài nghiên cứu tập trung vào phạm vi những kiến thức đã học trong lớp, và những kiến thức sinh viên tự tìm hiểu trên mạng hay nguồn tài liệu sách giáo khoa trong thư viện của trường.

1. **Phương pháp nghiên cứu**

Nghiên cứu theo hướng tập trung vào khảo sát các đối tượng đồ họa cơ sở, các hàm, các thuật toán trên giao diện hiển thị ra màn hình.

Nghiên cứu để tạo ra tọa độ chính xác trên từng thiết bị khác nhau, hiển thị một cách phù hợp.

1. **Kết quả đạt được**

Tạo ra được đối tượng như mong muốn, các hàm và thuật toán được áp dụng triệt để, tạo ra được chuyển động của các đối tượng một cách mượt mà.

1. **Bố cục**

Xây dựng các hàm vẽ riêng biệt cho từng đối tượng, mỗi đối tượng được đặt ở một tọa xác định sao cho bố cục của bài được rõ ràng và hợp lí.

**CHƯƠNG 2: MÔ TẢ BÀI TOÁN**

1. **Mô tả chi tiết bài toán**

**Vẽ bầu trời và nhiều ngôi sao:**

void graphics::drawSky(QPainter &painter){

int x=0;

int y=0;

int w=width();

int h=height(); painter.setBrush(QBrush("#191970"));

painter.drawRect(x,y,w,h);

}

}

void graphics::drawStars(QPainter &painter){

for(int i=0;i<100;i++){

drawStar(painter,vitriSao.at(i),mausacSao.at(i),kichthuocSao.at(i),0,2);}}

* **Vẽ bầu trời:** tạo ra bầu trời làm nền cho toàn bộ khung cảnh bằng cách vẽ hình chữ nhật với tọa độ (x,y) và chiều dài w, chiều rộng h.
* **Vẽ nhiều ngôi sao:** dựa vào hàm vẽ một ngôi sao, sau đó cho vòng lặp chạy với vị trí, màu sắc, kích thước sẽ tạo ra nhiều sao.

**Vẽ một ngôi sao:**

void graphics::drawStar(QPainter &painter,QPointF ViTri,QString MaMau,double kich,double xoay,double soCanh){

painter.setPen(QColor(MaMau));

painter.setBrush(QColor(MaMau));

painter.drawEllipse(ViTri,kich,kich);

double Do=xoay;

for(int i=0;i<soCanh;i++){

QPointF D1=quay1(ViTri,kich\*2,Do+180);

QPointF D2=quay1(ViTri,kich\*2,Do);

painter.drawLine(D1,D2);

Do=Do+360/(soCanh\*2);

}

}

* **Vẽ một ngôi sao:** sử dụng phép quay1 với một góc xác định, số cánh, vị trí và kích thước.

**Vẽ thân côi xoay gió:**

void graphics::drawWindMill(QPainter &painter){

//than

QPoint than[4]={

QPoint(310,300),

QPoint(270,480),

QPoint(390,480),

QPoint(350,300)

};

painter.setBrush(QBrush("#8B008B"));

painter.drawPolygon(than,4);

//dinh

QPoint noc[3]={

QPoint(300,300),

QPoint(360,300),

QPoint(330,250)

};

painter.setBrush(QBrush("#FF3333"));

painter.drawPolygon(noc,3);

}

* **Vẽ thân cối xoay gió:** dùng hàm polygon vẽ thân cối gồm 4 điểm, đỉnh cối xoay gồm 3 điểm.

**Vẽ bướm bay:**

void graphics::drawButterfly(QPainter &painter){

painter.setPen(Qt::NoPen);

painter.setBrush(QColor(random1(255),random1(255),random1(255)));

int x=random1(width()/2+500);

int y=random1(height()/2+500);

int d=50;

int r=d/3;

QPoint c(x,y);

QPoint p(x,y-r);

for( int i=1;i<=4;i++){

QPoint a6=quay(p,c,i\*62);

painter.drawEllipse(a6,r,r); }}

* **Vẽ bướm bay:** dùng hàm vẽ Ellipse, sau đó cho quay với một góc xác định trong vòng lặp for.

**Vẽ ngôi nhà:**

void graphics::drawHouse(QPainter& painter){

int w=1280;

int h=800;

//than nha

painter.setBrush(QBrush("#FF69B4"));

painter.drawRect(w/120,400,230,100);

//tang1

painter.setBrush(QBrush("#EE00EE"));

painter.drawRect(w/120,300,230,100);

//cuatrentang1

painter.setBrush(QBrush("#FFB90F"));

painter.drawRect(180,320,50,80);

//langcang

painter.setBrush(Qt::VerPattern);

painter.drawRect(w/120,350,230,50);

//nocnha

QPoint noc[4]={

QPoint(90,200),

QPoint(10,300),

QPoint(240,300),

QPoint(160,200)

};

painter.setBrush(QBrush("#EE0000"));

painter.drawPolygon(noc,4);

//cua chinh

painter.setBrush(QBrush("#EE0000"));

painter.drawRect(100,420,50,80)

//cua so

//bentrai

painter.setBrush(QBrush("#00BFFF"));

painter.drawRect(30,410,40,40);

//benphai

painter.setBrush(QBrush("#00BFFF"));

painter.drawRect(180,410,40,40);

} }

* **Vẽ ngôi nhà:** dùng hàm polygon vẽ thân nhà trên/dưới, cửa sổ, mái nhà, cửa chính, lang cang.

**Vẽ con đường:**

void graphics::drawRoad(QPainter& painter){

//duongdi

int xd=0;

int yd=510;

int dd=width();

int rd=height()/4;

painter.setBrush(QBrush("#A9A9A9"));

painter.drawRect(xd,yd,dd,rd);

//vach ke duong

int y=560;//vi tri xuat hien vach ke duong tren truc tung

int d=200;//chieu dai vach ke duong

int r=25;//chieu rong vach ke duong

QString vachkeduong="D://vachkeduong.png";

painter.drawPixmap(position1,y,d,r,QPixmap(vachkeduong));

painter.drawPixmap(position2,y,d,r,QPixmap(vachkeduong));

painter.drawPixmap(position3,y,d,r,QPixmap(vachkeduong));

painter.drawPixmap(position4,y,d,r,QPixmap(vachkeduong));

}

**=**

* **Vẽ con đường:** dùng hàm drawRect vẽ con đường, dùng ảnh làm vạch kẻ đường

**Vẽ núi:**

//VE NUI

void graphics::drawMountain(QPainter& painter){

int w=1280;

int h=800;

int x=400;

int y=400;

int r=100;

//nui

QPoint nui[7]={

QPoint(x,y),

QPoint(w+(r/2),y),

QPoint(w-r,r),

QPoint(w-(2\*r),y-r),

QPoint(w-x,0),

QPoint(w-(1.5\*x),y-r),

QPoint(w-(x+3\*r),y/2)

};

painter.setBrush(QBrush("#000000"));

painter.drawPolygon(nui,7);

}

**Vẽ cây xanh:**

void graphics::drawTree(QPainter &painter){

//cay 1

painter.setPen(Qt::NoPen);

int x=h+200+400;

int y=w+200;

int d=100;

int r=d/3;

QPoint c(x,y);

QPoint p(x,y-r);

painter.setBrush(QBrush("#A0522D"));

painter.drawRect(h+180+400,350,r,r\*4);

for( int i=1;i<=5;i++){

QPoint d5=quay(p,c,i\*62);

painter.setBrush(QBrush("#008B00"));

painter.drawEllipse(d5,r\*2,r\*2);

//cay 2

painter.setPen(Qt::NoPen);

int x=h+200+400+200;

int y=w+200;

int d=100;

int r=d/3;

QPoint c(x,y);

QPoint p(x,y-r);

painter.setBrush(QBrush("#A0522D"));

painter.drawRect(h+180+400+200,350,r,r\*4);

for( int i=1;i<=5;i++){

QPoint d5=quay(p,c,i\*72);

painter.setBrush(QBrush("#008B00"));

painter.drawEllipse(d5,r\*2,r\*2);

//cay3

painter.setPen(Qt::NoPen);

int x=h+200+400+200+200;

int y=w+200;

int d=100;

int r=d/3;

QPoint c(x,y);

QPoint p(x,y-r);

painter.setBrush(QBrush("#A0522D"));

painter.drawRect(h+180+400+200+200,350,r,r\*4);

for( int i=1;i<=5;i++){

QPoint d5=quay(p,c,i\*72);

painter.setBrush(QBrush("#008B00"));

painter.drawEllipse(d5,r\*2,r\*2);

//cay4

painter.setPen(Qt::NoPen);

int x=h+200+400+200+200+200;

int y=w+200;

int d=100;

int r=d/3;

QPoint c(x,y);

QPoint p(x,y-r);

painter.setBrush(QBrush("#A0522D"));

painter.drawRect(h+180+400+200+200+200,350,r,r\*4);

for( int i=1;i<=5;i++){

QPoint d5=quay(p,c,i\*52);

painter.setBrush(QBrush("#008B00"));

painter.drawEllipse(d5,r\*2,r\*2);

}

}

}

}

}

* **Vẽ cây xanh:** dùng hàm drawRect để vẽ thân cây, dùng hàm drawEllipse vẽ tán cây sau đó cho quay trong vòng lặp xác định tạo ra tán cây lớn.

**Vẽ ảnh như mây, máy bay, xe, người đi bộ có di chuyển:**

void graphics::drawPicture(QPainter& painter){

int dx=700; //chieu dai xe

int rx=500;//chieu rong xe

int d=200;//chieu dai cua may va may bay

int r=100;// chieu rong cua may va may bay

int w=1280/6;//vi tri xe xuat hien tren truc hoanh

int h=210;//vi tri xe xuat hien tren truc tung

int h1=40;// vi tri xuat hien may tren truc tung

int h2=20;//vi tri xuat hien cua may tren truc tung

QString filename="D://xe.PNG";//tai anh may bay

QString filename2="D://bay3.PNG";//tai anh may bay

QString filename3="D://may1.PNG";//tai anh may 1

QString filename4="D://may2.PNG";//tai anh may 2

painter.drawPixmap(position-200,h,dx,rx,QPixmap(filename));//ve xe

painter.drawPixmap(position,h1,d,r,QPixmap(filename2));//ve may bay

painter.drawPixmap(position01,h2,d,r,QPixmap(filename3));//ve may 1

painter.drawPixmap(position1,h2,d,r,QPixmap(filename4));//ve may 2

}

* **Vẽ ảnh như mây, máy bay, xe, người đi bộ có di chuyển:** dùng ảnh chèn vào kết hợp với timer và hàm drawPixmap tạo ra chuyển động phù hợp.

**Vẽ bánh xe :**

void graphics::drawWhell(QPainter &painter, int x,int y, int angle){

int r=80;

int n=31;

pcenter.setX(x);

pcenter.setY(y);

pcenter1.setX(x+210);

pcenter1.setY(y+300);

pstart.setX(pcenter.x());

pstart.setY(pcenter.y()-n);

pstart1.setX(pcenter.x());

pstart1.setY(pcenter.y()+n);

pstart2.setX(pcenter.x()+n);

pstart2.setY(pcenter.y());

pstart3.setX(pcenter.x()-n);

pstart3.setY(pcenter.y());

pnew=quay(pstart,pcenter, angle);

pnew1=quay(pstart1,pcenter, angle);

pnew2=quay(pstart2,pcenter, angle);

pnew3=quay(pstart3,pcenter, angle);

painter.setPen(QPen(Qt::black,5,Qt::SolidLine));

painter.setRenderHint(QPainter::Antialiasing,5);

painter.drawEllipse(pcenter,2\*r/5,2\*r/5);

painter.setBrush((Qt::white));

painter.setPen(QPen(Qt::black,5,Qt::SolidLine));

painter.drawLine(pcenter,pnew);

painter.drawLine(pcenter,pnew1);

painter.drawLine(pcenter,pnew2);

painter.drawLine(pcenter,pnew3);

}

* **Vẽ bánh xe:** dùng hàm drawEllipse để vẽ bánh xe, dùng drawLine để vẽ cây căm xe kết hợp với phép quay làm căm xe quay.

**Vẽ người đi bộ:**

void graphics::drawWalkMan(QPainter &painter){

QPixmap spritesheet("D://nguoi1.png");

double wHinh=spritesheet.width()/6;

double hHinh=spritesheet.height();

int r=5;

QRect displayWindow(wHinh\*hinh,0,wHinh,hHinh);

QRect pos(position\*r,height()/2+200,wHinh/2,hHinh/2);

painter.drawPixmap(pos,spritesheet,displayWindow);

}

* **Vẽ người đi bộ:** sử dụng ảnh gồm từng cử động của người đi bộ, kết hợp timer, hàm drawPixmap tạo ra chuyển động hoàn chỉnh.

**Vẽ mặt trăng:**

void graphics::drawMoon(QPainter &painter){

int r1=100;

QString moon="D://moon.png";

painter.drawPixmap(11\*r1,0,r1,r1,QPixmap(moon));

}

* **Vẽ mặt trăng:** sử dụng hàm drawPixmap vẽ ảnh mặt trăng.

**Vẽ hàng rào và bụi cỏ:**

void graphics::drawHaw(QPainter &painter){

int r1=100;

QString hangrao="D://hangrao.png";

QString coxanh="D://coxanh.png";

painter.drawPixmap(6\*r1-30,4\*r1-20,r1,r1,QPixmap(hangrao));

painter.drawPixmap(9\*r1-130,4\*r1-40,r1,r1,QPixmap(hangrao));

painter.drawPixmap(10\*r1-30,4\*r1-20,r1,r1,QPixmap(hangrao));

painter.drawPixmap(12\*r1-30,4\*r1-40,r1,r1,QPixmap(hangrao));

painter.drawPixmap(8\*r1+50,4\*r1-120,r1\*2,r1\*2,QPixmap(coxanh));

painter.drawPixmap(8\*r1,4\*r1-120,r1\*2,r1\*2,QPixmap(coxanh));

painter.drawPixmap(6\*r1+20,4\*r1-120,r1\*2,r1\*2,QPixmap(coxanh));

painter.drawPixmap(10\*r1+30,4\*r1-120,r1\*2,r1\*2,QPixmap(coxanh));

painter.drawPixmap(12\*r1,4\*r1-120,r1\*2,r1\*2,QPixmap(coxanh));

}

1. **Vấn đề có liên quan đến bài toán**

**Bài toán sử dụng các hàm quay như sau:**

QPoint graphics::quay(QPoint p, QPoint c, int deta){

QPoint pnew;

double goc=deta\*3.14/180;

pnew.setX(c.x()+(p.x()-c.x())\*cos(goc)-(p.y()-c.y())\*sin(goc));

pnew.setY(c.y()+(p.x()-c.x())\*sin(goc)+(p.y()-c.y())\*cos(goc));

return pnew;

}

QPointF graphics::quay1(QPointF BD,double kc,double huong){

return QPointF(BD.x()+kc\*cos(M\_PI\*huong/180),BD.y()+kc\*sin(M\_PI\*huong/180));

}

QPointF graphics::quaynew(QPointF p1, QPointF c1, int deta1){

QPointF pnewgoc;

double goc=deta1\*3.14/180;

pnewgoc.setX(c1.x()+(p1.x()-c1.x())\*cos(goc)-(p1.y()-c1.y())\*sin(goc));

pnewgoc.setY(c1.y()+(p1.x()-c1.x())\*sin(goc)+(p1.y()-c1.y())\*cos(goc));

return pnewgoc;

}

**Các hàm random:**

**Các hàm Random:**

int graphics::random(int n1,int n2){

return rand()%n2+n1;

}

int graphics::random1(int n3){

return rand() % n3;

}

1. **Mô tả giải pháp cho bài toán**

Đối với việc thu nhỏ khung nhìn giao diện hình ảnh bị xáo trộn, không đúng vị trí, chúng ta cần cho tọa độ chuẩn và chính xác.

Sắp xếp thứ tự vẽ các đối tượng hợp lí không xếp chồng lên nhau, đối tượng nào nằm trước thực hiện vẽ trước, đối tượng nằm sau thực hiện vẽ sau.

Điều chỉnh thời gian timer sau cho các đối tượng có chuyển động sẽ chuyển động một cách mượt mà và hợp lí nhất.

Màu sắc cho từng đối tượng phải phù hợp với góc nhìn thưc tế, có sự hài hòa, tương phản và sinh động.

Đối với các đối tượng chuyển động ta có thể điều chỉnh khoảng cách dịch chuyển để đối tượng chuyển động nhanh hoặc chậm hơn.

Các đối tượng cần vẽ có độ phức tạp cao thì có thể chèn ảnh đã được cắt và xử lí những phần thừa của ảnh để tạo độ chân thực hơn.

Đối với chuyển động của người đi độ thì cần xem kĩ kích thước và khoảng cách giữa từng ảnh để có thể chia khung nhìn một cách chính xác tạo nên chuyển động mượt mà và chân thật nhất.

**CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT**

1. **Thiết kế hệ thống**

* Để thực hiện đề tài ta chúng tôi sử dụng phần mềm đồ họa: Qt creator
* Hệ thống được cài đặt sẽ bao gồm các file:
* tên\_project.pro: file chính của dự án
* mainwindow.ui: định nghĩa giao diện giao diện đồ họa (theo cấu trúc XML)
* mainwindow.h: định nghĩa lớp MainWindow
* main.cpp: chứa hàm main()
* mainwindow.cpp: cài đặt các hàm có trong lớp MainWindow
* Tạo một lớp mới kế thừa lớp Qwidget:
* Graphics.h: khai báo các hàm, biến, các sự kiện.
* Graphics.cpp: lời gọi các hàm, các phép biến đổi đồ họa 2 chiều cơ bản, thực hiện vẽ .

1. **Thiết kế và cài đặt giải thuật**
2. **Ý tưởng**

**Tạo ra các file: graphics.h, graphics.cpp,**

**Graphics.h**

#ifndef GRAPHICS\_H

#define GRAPHICS\_H

#include <QWidget>

class graphics : public QWidget

{

Q\_OBJECT

public:

explicit graphics(QWidget \*parent = 0);

//Cac su kien ve

void *paintEvent*(QPaintEvent \*);

//Khai bao ham de thuc hien ve

void drawPicture(QPainter& painter); //ve anh

void drawWhell(QPainter& painter, int x, int y, int angle); //banh xe sau

void drawHouse(QPainter& painter); //ve ngoi nha

void drawMountain(QPainter& painter); //ve nui

void drawRoad(QPainter& painter);

void drawTree(QPainter& painter); //ve cay

void drawCar(QPainter& painter,int x, int y, double position); //ve xe hoi

void drawWindMill(QPainter& painter); //ve than coi xoay gio

void drawSky(QPainter& painter); //ve bau troi

void drawStar(QPainter &painter); //ve mot ngoi sao

void drawStars(QPainter &painter); //ve nhieu ngoi sao

void drawButterfly(QPainter& painter);//ve buom bay

void drawRotary(QPainter& painter); //ve canh coi xoay gio

void drawWalkMan(QPainter& painter); //ve nguoi di bo

void drawMoon(QPainter& painter); //ve mat trang

//cac phep quay

QPoint quay(QPoint p, QPoint c,int deta); //banh xe, buom bay

QPointF quay1(QPointF BD,double kc,double huong);//ngan sao

QPointF quaynew(QPointF p1, QPointF c1, int deta1); //coi xoay gio

//khai bao ve coi xoay gio

int angle1;

int h1=height();

int w1=width();

void Rotary(QPainter& painter);

QPointF pstart10,pcenter10,pnew00,pnew11,pnew12,pnew13,pnewgoc,pnew14,pnew15;

//khai bao ve ngan sao

int random(int n1,int n2);

void drawStar(QPainter &painter,QPointF ViTri,QString MaMau,double kich,double xoay,double soCanh);

QVector<QPointF> vitriSao;

QVector<QString> mausacSao;

QVector<double> kichthuocSao;

QPointF HT;

double ViTri;

//ve buom nay

int random1(int n3);

//ve nguoi di bo

double hinh;

QPixmap spritesheet;

//khai bao su kien timeEvent

void *timerEvent*(QTimerEvent \*);

double position, position01, position1, position2, position3,position4;

double timerId;

double h=height();

double w=width();

QPoint pstart,pstart1,pstart2,pstart3,pcenter,pcenter1,pnew,pnew1,pnew2,pnew3;

int angle;

signals:

public slots:

};

#endif // GRAPHICS\_H

**Graphics.cpp**

#include "graphics.h"

#include<math.h>

#include<QPainter>

#include<QString>

#include<QPen>

graphics::graphics(QWidget \*parent) :

QWidget(parent)

{

position=0;

position01=300;

position1=0;

position2=350;

position3=700;

position4=1050;

timerId=startTimer(5);

angle=0;

angle1=30;

QVector<QString> NhieuMau;

NhieuMau<<"#FFFFFF"<<"#0099FF"<<"#33CC33"<<"#FF3300"<<"#6633CC"<<"#FF33FF"<<"#FFFF00";

ViTri=0;

HT=QPointF(200,200);

double w=width()+1280;

double h=height();

for(int i=0;i<100;i++){

vitriSao<<QPointF(random(0,w),random(0,h));

mausacSao<<NhieuMau.at(random(0,NhieuMau.size()));

kichthuocSao<<random(1,3);

}

}

void graphics::*paintEvent*(QPaintEvent \*){

QPainter painter(this);

painter.setRenderHint(QPainter::Antialiasing);

drawSky(painter);

drawHouse(painter);

drawRoad(painter);

drawWindMill(painter);

drawMoon(painter);

drawMountain(painter);

drawTree(painter);

drawPicture(painter);

drawStars(painter);

drawStars(painter);

drawButterfly(painter);

drawRotary(painter);

drawCar(painter,0, height()/5,position);

drawWalkMan(painter);

}

void graphics::*timerEvent*(QTimerEvent \*){

position1+=-1;

angle1+=30;

int r=200;//kich thuoc vat di chuyen 1

int r2=200;//kich thuoc vat di chuyen 2

int r3=200;//kich thuoc vat di chuyen 3`

position+=1;

position01+=1;

position1+=-1;

position2+=-1;

position3+=-1;

position4+=-1;

angle+=1;

if(position==width()+r){

position+=1;

position=-r;

}

if(position01==width()+r2){

position01+=1;

position01=-r2;}

if(position2==-r3){

position2+=-1;

position2=width();

}

if(position3==-r3){

position3+=-1;

position3=width();

}

if(position4==-r3){

position4+=-1;

position4=width();

}

//VONG LAP NGAN SAO

for(int i=0;i<100;i++){

int at=random(1,2);

vitriSao[i].rx()+=((at==1)? 1:-1);

vitriSao[i].ry()+=((at==1)? 1:-1);

kichthuocSao[i]=random(1,3);

}

//VE NGUOI DI BO

hinh+=1;

if(hinh==6) hinh=0;

repaint();

}

1. **Lưu đồ các thuật toán**

**THUẬT TOÁN**

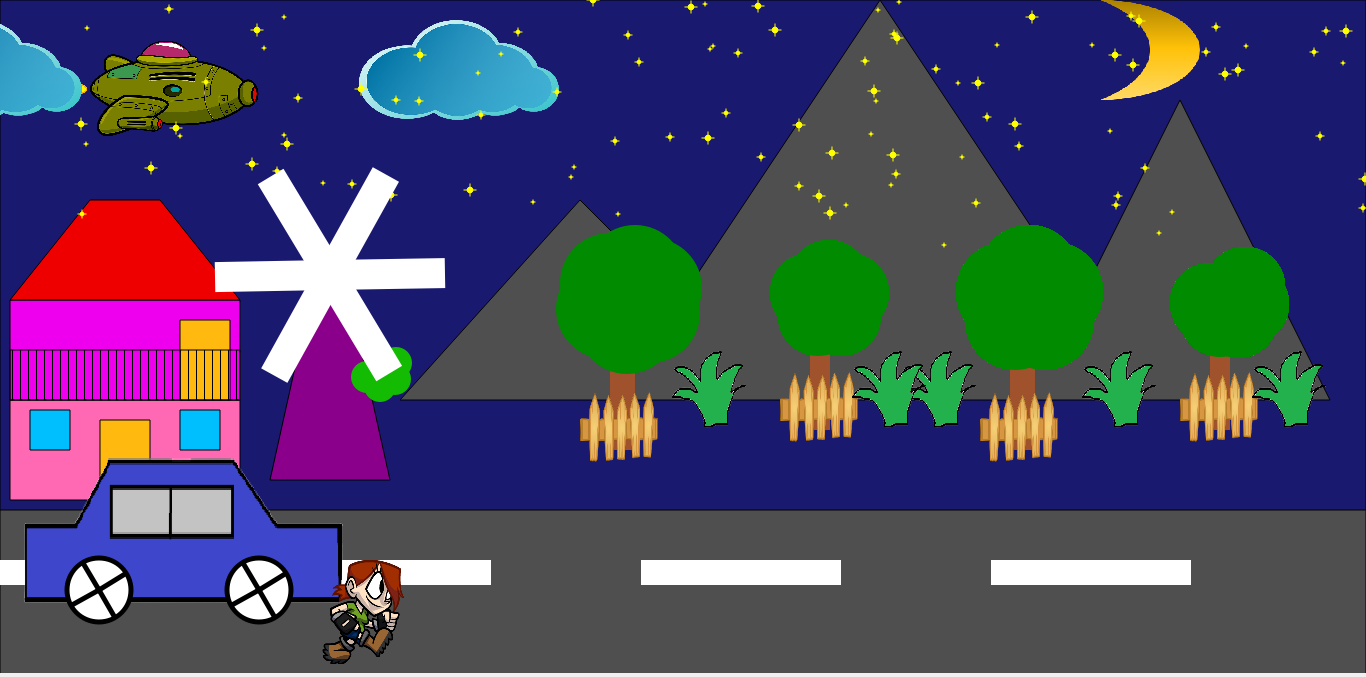
**PHÉP BIẾN ĐỔI**

HÀM

**drawLine, drawEllipse, drawPixmap, drawPolygon, random, random1**

**Các phép quay: quay, quay1, quaynew**

1. **Giao diện**

****

**CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC**

1. **Nội dung chương trình code**

//Cac su kien ve

void *paintEvent*(QPaintEvent \*);

//Khai bao ham de thuc hien ve

void drawPicture(QPainter& painter); //ve anh

void drawWhell(QPainter& painter, int x, int y, int angle); //banh xe sau

void drawHouse(QPainter& painter); //ve ngoi nha

void drawMountain(QPainter& painter); //ve nui

void drawRoad(QPainter& painter);

void drawTree(QPainter& painter); //ve cay

void drawCar(QPainter& painter,int x, int y, double position); //ve xe hoi

void drawWindMill(QPainter& painter); //ve than coi xoay gio

void drawSky(QPainter& painter); //ve bau troi

void drawStar(QPainter &painter); //ve mot ngoi sao

void drawStars(QPainter &painter); //ve nhieu ngoi sao

void drawButterfly(QPainter& painter);//ve buom bay

void drawRotary(QPainter& painter); //ve canh coi xoay gio

void drawWalkMan(QPainter& painter); //ve nguoi di bo

void drawMoon(QPainter& painter); //ve mat trang

//cac phep quay

QPoint quay(QPoint p, QPoint c,int deta); //banh xe, buom bay

QPointF quay1(QPointF BD,double kc,double huong);//ngan sao

QPointF quaynew(QPointF p1, QPointF c1, int deta1); //coi xoay gio

//khai bao ve coi xoay gio

int angle1;

int h1=height();

int w1=width();

void Rotary(QPainter& painter);

QPointF pstart10,pcenter10,pnew00,pnew11,pnew12,pnew13,pnewgoc,pnew14,pnew15;

//khai bao ve ngan sao

int random(int n1,int n2);

void drawStar(QPainter &painter,QPointF ViTri,QString MaMau,double kich,double xoay,double soCanh);

QVector<QPointF> vitriSao;

QVector<QString> mausacSao;

QVector<double> kichthuocSao;

QPointF HT;

double ViTri;

//ve buom nay

int random1(int n3);

//ve nguoi di bo

double hinh;

QPixmap spritesheet;

//khai bao su kien timeEvent

void *timerEvent*(QTimerEvent \*);

double position, position01, position1, position2, position3,position4;

double timerId;

double h=height();

double w=width();

QPoint pstart,pstart1,pstart2,pstart3,pcenter,pcenter1,pnew,pnew1,pnew2,pnew3;

int angle;

void graphics::*paintEvent*(QPaintEvent \*){

QPainter painter(this);

painter.setRenderHint(QPainter::Antialiasing);

drawSky(painter);

drawHouse(painter);

drawRoad(painter);

drawWindMill(painter);

drawMoon(painter);

drawMountain(painter);

drawTree(painter);

drawPicture(painter);

drawStars(painter);

drawStars(painter);

drawButterfly(painter);

drawRotary(painter);

drawCar(painter,0, height()/5,position);

drawWalkMan(painter);

}

1. **Tài liệu tham khảo**

**Sách giáo trình Đồ Họa Máy Tính**

**Trang web:** https://vi.wikipedia.org/wiki/Đồ\_họa\_máy\_tính

[www.laptrinhtanbinh.com/course/1/Lap-trinh-voi.../3.1.-Gioi-thieu-ve-thu-vien-Qt](http://www.laptrinhtanbinh.com/course/1/Lap-trinh-voi.../3.1.-Gioi-thieu-ve-thu-vien-Qt )

Phụ Lục

[**Lời nói đầu** 1](#_Toc7695091)

[**CHƯƠNG 1: GiỚI THIỆU** 1](#_Toc7695092)

[**1.** **Đặt vấn đề** 1](#_Toc7695093)

[**2.** **Lịch sử giải quyết vấn đề** 1](#_Toc7695094)

[**3.** **Mục tiêu của đề tài** 1](#_Toc7695095)

[**4.** **Đối tượng và phạm vi nghiên cứu** 1](#_Toc7695096)

[**5.** **Phương pháp nghiên cứu** 1](#_Toc7695097)

[**6.** **Kết quả đạt được** 1](#_Toc7695098)

[**7.** **Bố cục** 1](#_Toc7695099)

[**CHƯƠNG 2: MÔ TẢ BÀI TOÁN** 1](#_Toc7695100)

[**1.** **Mô tả chi tiết bài toán** 1](#_Toc7695101)

[**2.** **Vấn đề có liên quan đến bài toán** 1](#_Toc7695102)

[**3.** **Mô tả giải pháp cho bài toán** 1](#_Toc7695103)

[**CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT** 1](#_Toc7695104)

[**1.** **Thiết kế hệ thống** 1](#_Toc7695105)

[**2.** **Thiết kế và cài đặt giải thuật** 1](#_Toc7695106)

[**a)** **Ý tưởng** 1](#_Toc7695107)

[**b)** **Lưu đồ các thuật toán** 1](#_Toc7695108)

[**3.** **Giao diện** 1](#_Toc7695109)

[**CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC** 1](#_Toc7695110)

[**1.** **Nội dung chương trình code** 1](#_Toc7695111)

[**2.** **Tài liệu tham khảo** 1](#_Toc7695112)