Lecture Qt015
Graphics II

Instructor: David J. Coe

CPE 353 – Software Design and Engineering

Department of Electrical and Computer Engineering

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Outline

- Graphics View
- Graphics Scene
- Simple Scene/View Example
- Projectile Example
- Collision Example
- Pixmap Animation Example
- Key Points

Graphics View

- **View** is an instance of a QGraphicsView
- Provides ability to scroll or transform rendering of scene

Graphics View

- Framework that provides a 2D canvas that utilizes
  - Items
  - Scene
  - View (one scene may support multiple views)
- Facilitates management of large number of items that the user may wish to interact with via select, drag, or click

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Graphics View

- **Item** (parent class QGraphicsItem)
  - Line — QGraphicsLineItem
  - Rectangle — QGraphicsRectItem
  - Polygon — QGraphicsPolygonItem
  - Ellipse — QGraphicsEllipseItem
  - Pixmap — QGraphicsPixmapItem
  - Text — QGraphicsTextItem or QGraphicsSimpleTextItem
  - Etc.

Graphics Scene

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

- **Scene** is an instance of QGraphicsScene
- Scene has three layers
  - Foreground
  - Items
  - Background
- Scene items rendered in the order they are added
- Provides collision, selection, location, and region detection
- Hierarchical in that items are ultimately treated as children of the scene
- Scene items can be grouped as a QGraphicsItemGroup

Graphics Scene

- selectedItems()
  - Returns a list of items that are currently selected
- collidingItems(…)
  - Returns a list of all items that collide with specified item

Simple Scene/View Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

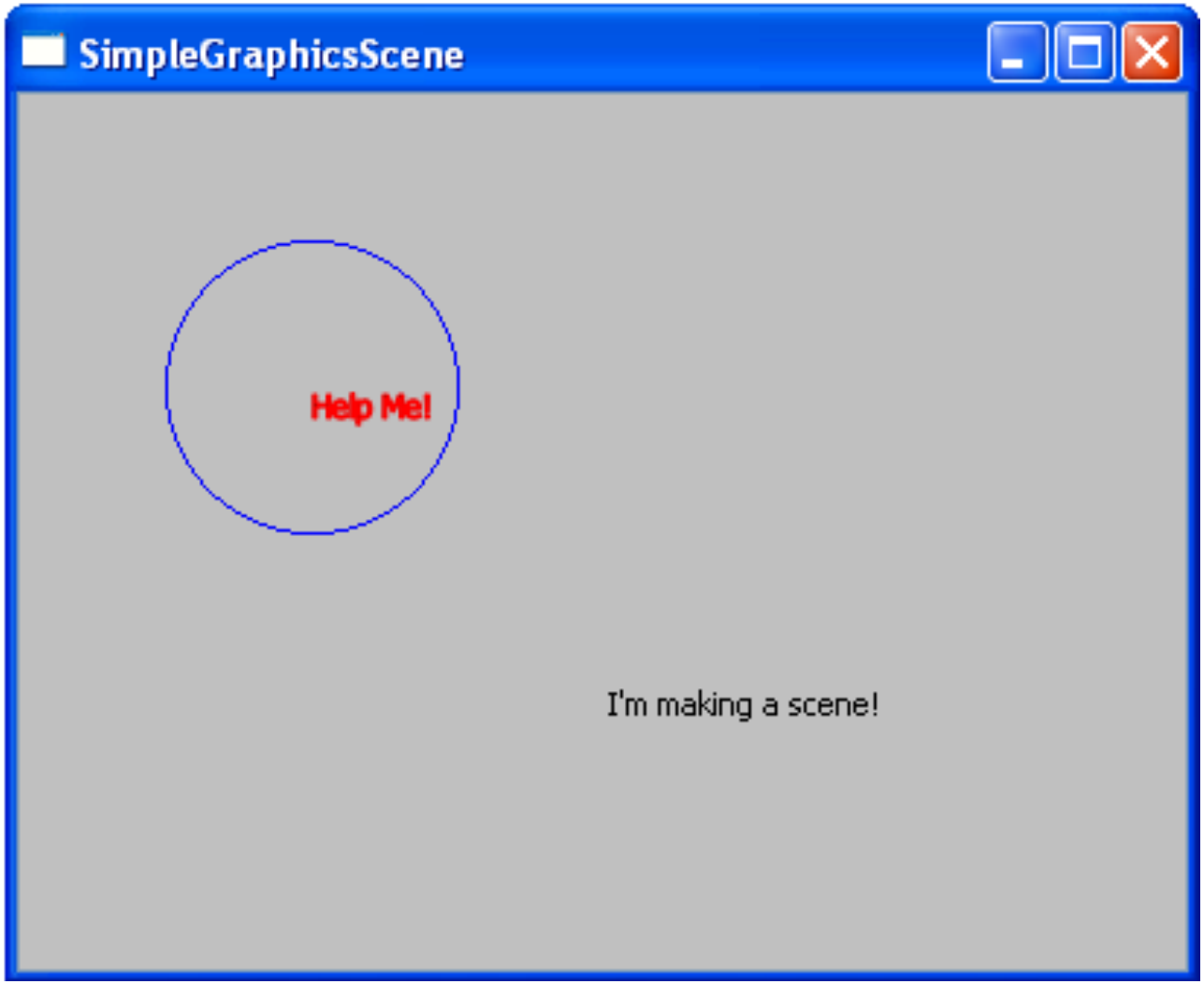
- Goals
  - Create several graphics items
  - Add all items to a scene
  - Display all items in a view

Simple Scene/View Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

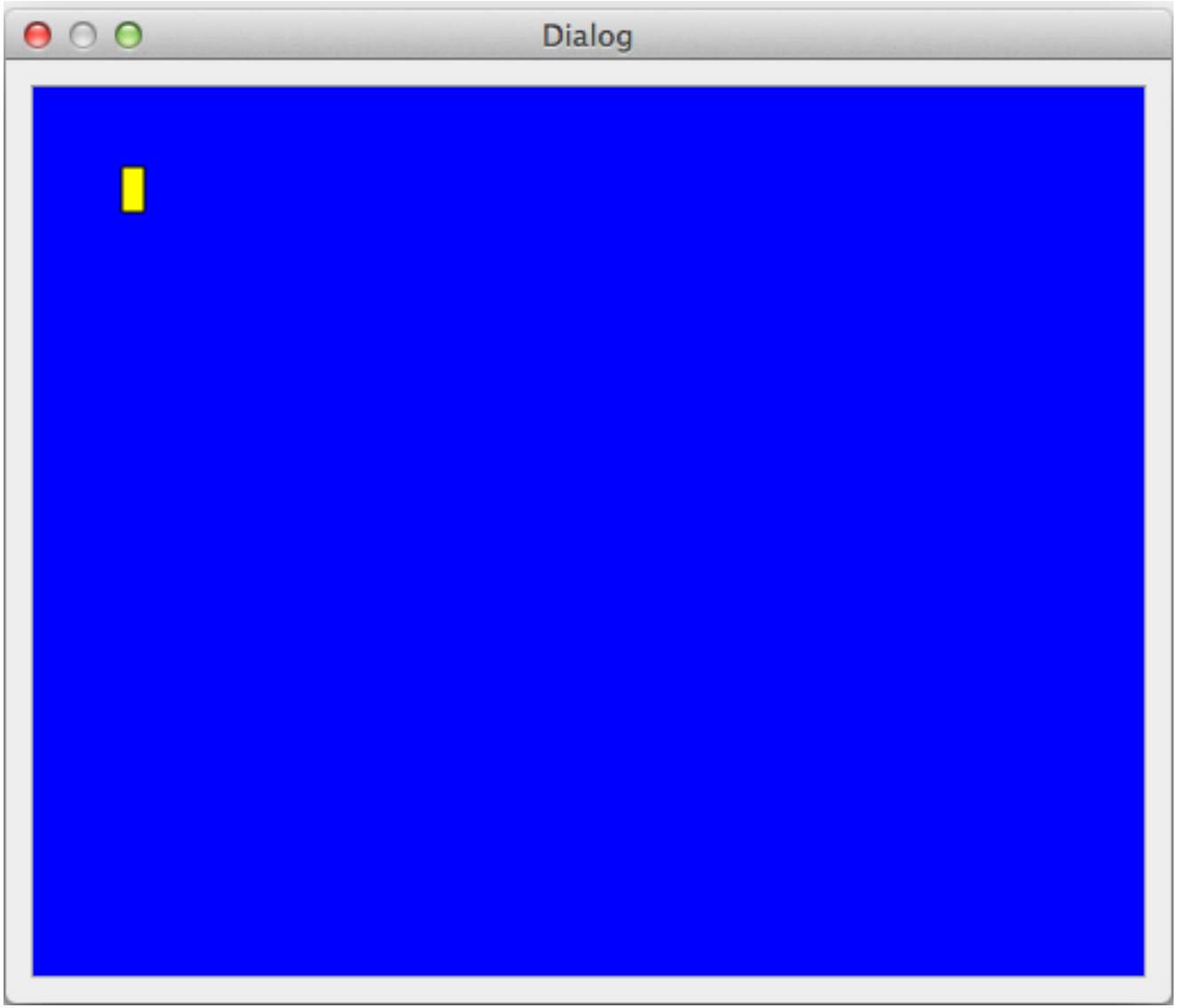
```
#include <QtGui>


int main(int argc, char *argv[])
{
    QApplication myApp(argc, argv);

    QGraphicsScene  myScene(QRect(-100, -100, 400, 300));
    myScene.setBackgroundBrush(QBrush(Qt::lightGray, Qt::SolidPattern));

    QGraphicsSimpleTextItem  simpletext("Help Me!");
    simpletext.setPen(QPen(Qt::red));
    simpletext.setPos(0,0);

    QGraphicsTextItem text("I'm making a scene!");
    text.setDefaultTextColor(Qt::black);
    text.setPos(100, 100);

    QGraphicsEllipseItem  circle(QRect(-50, -50, 100, 100));
    circle.setPen(QPen(Qt::blue));

    myScene.addItem(&simpletext);
    myScene.addItem(&text);
    myScene.addItem(&circle);

    QGraphicsView myView;
    myView.setScene(&myScene);
    myView.show();

    return myApp.exec();
}
```

Simple Scene/View Example



THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Projectile Example



THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Projectile Example

- Key Public Methods inherited from **QGraphicsItem**
  - **QRectF boundingRect() const**;
    // approximate item area within which paint occurs – virtual function
  - **QPainterPath shape() const;**
    // accurate shape of item for collision detection – virtual function
  - **void paint(QPainter *painter,**
    const QStyleOptionGraphicsItem *option,
    QWidget *widget);
    // draws the actual item in local coordinates – virtual function

- Key Protected Methods inherited from **QGraphicsItem**
  - **void advance(int step);**
    // animates item (called twice on step = 0, step = 1) – virtual function

Projectile Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// projectile.h

#ifndef PROJECTILE_H
#define PROJECTILE_H

#include <QGraphicsItem>
#include <QColor>
#include <QPainter>
#include <QRect>

class Projectile : public QGraphicsItem
{
public:
    explicit Projectile();

    QRectF boundingRect() const;
    QPainterPath shape() const;
    void paint(QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget);

protected:
    void advance(int step);

private:
    QColor color;
    qreal dx, dy;
    qreal x, y;
    qreal w, h;

};#endif // PROJECTILE_H
```

Projectile Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// projectile.cpp

#include "projectile.h"
Projectile::Projectile()
{
  color = QColor("yellow");
  dx = 0.00;
  dy = 2.50;
  x = 0.0;
  y = 0.0;
  w = 10.0;
  h = 20.0;
}

void Projectile::paint(QPainter *painter,
                       const QStyleOptionGraphicsItem *option,
                       QWidget *widget)
// paint() paints item in local coordinates
{
  painter->setBrush(color);
  painter->drawRect(-w/2, -h/2, w, h);
}

QRectF Projectile::boundingRect() const
// Determines bounds for repainting
{
  qreal adjust = 1.0;
  return QRectF(-w/2 - adjust, -h/2 - adjust, w + adjust, h + adjust);
}
```

Projectile Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// projectile.cpp -- continued


void Projectile::advance(int step)
// advance() advances item by frame
{
  if (step == 0)
    return;

  x = this->pos().x();
  y = this->pos().y();

  x = x + dx;
  y = y + dy;

  setPos(x, y);         // Set item position in parent coordinates
}

QPainterPath Projectile::shape() const
// Returns shape for collision detection
{
  QPainterPath path;
  path.addRect(-w/2, -h/2, w, h);
  return path;
}
```

Projectile Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// dialog.h

#ifndef DIALOG_H
#define DIALOG_H

#include <QDialog>
#include <QGraphicsScene>
#include <QTimer>
#include "projectile.h"

namespace Ui {class Dialog;}

class Dialog : public Qdialog
{
    Q_OBJECT

public:
    explicit Dialog(QWidget *parent = 0);
    ~Dialog();

private:
    Ui::Dialog *ui;
    QGraphicsScene* scene;
    Projectile* p1;
};

#endif // DIALOG_H
```

Projectile Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// dialog.cpp
#include "dialog.h"
#include "ui_dialog.h"
#include <QtDebug>

Dialog::Dialog(QWidget *parent) : QDialog(parent), ui(new Ui::Dialog)
{
    ui->setupUi(this);

    // Create and configure scene object
    scene = new QGraphicsScene(this);
    scene->setItemIndexMethod(QGraphicsScene::NoIndex);
    scene->setSceneRect(-200, -150, 400, 300);
    scene->setBackgroundBrush(Qt::blue);

    // Configure graphics view object
    ui->graphicsView->setScene(scene);
    ui->graphicsView->setRenderHint(QPainter::Antialiasing);

    // Add projectile to scene
    p1 = new Projectile;
    p1->setPos(-200, -150);
    scene->addItem(p1);

    // Create and configure timer object
    QTimer* timer = new QTimer;
    connect(timer, SIGNAL(timeout()), scene, SLOT(advance()));
    timer->setInterval(1000/33);
    timer->start();
}
Dialog::~Dialog() { delete ui;}
```

Disable indexing to speed up animation

Advancing scene results in all items in the scene advancing

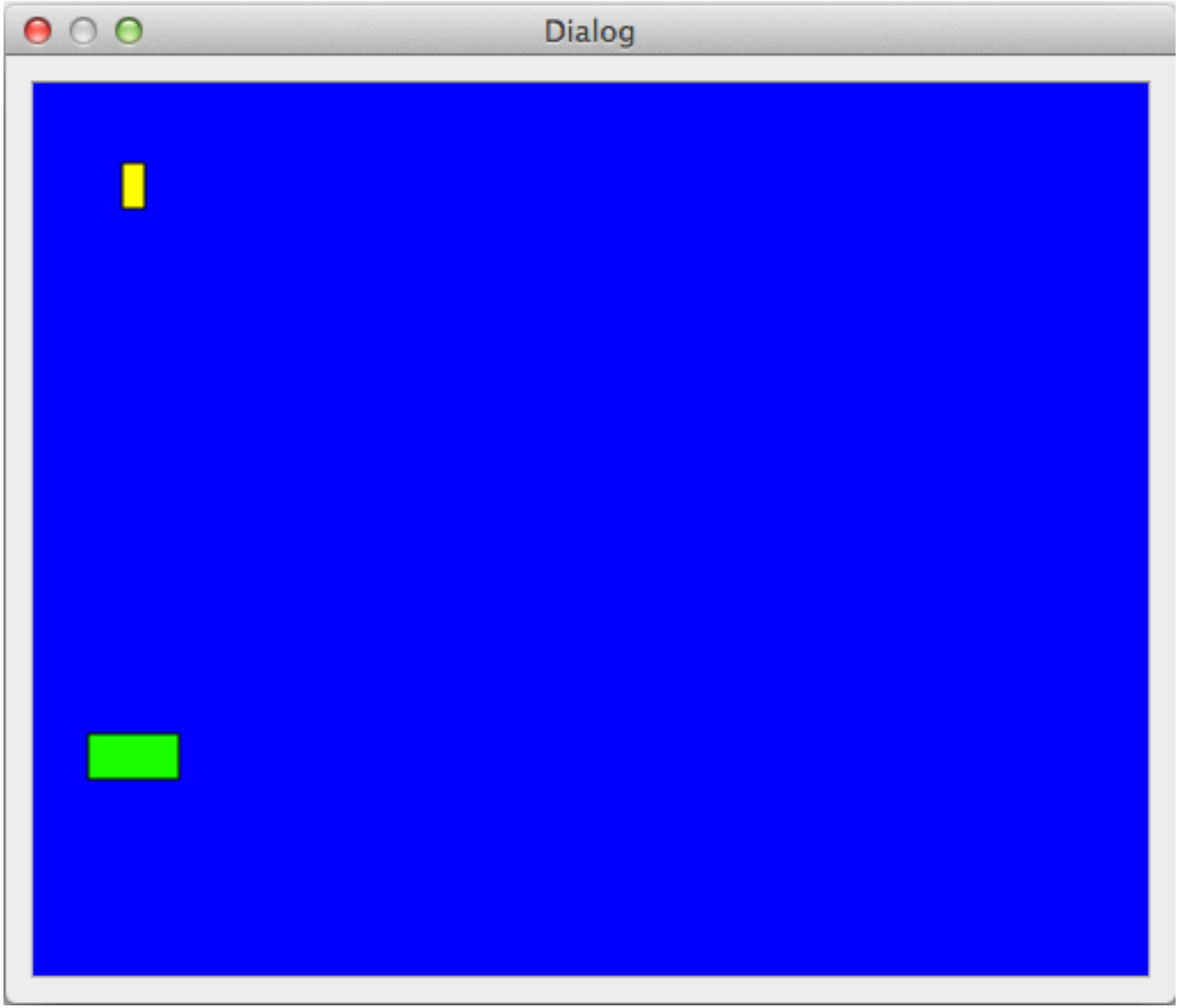
Collision Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

- Goals
  - Create a Projectile class that inherits from **QGraphicsItem**
  - Create a custom Dialog which uses a **QGraphicsView** object to display a **QGraphicsScene** object
  - Add a **Projectile** object to the scene and animate its travel
  - Add a **Target** object to the scene in the path of the **Projectile** to stage a collision
  - Detect collision and highlight region of overlap

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Collision Example



Collision Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// target.h
#ifndef TARGET_H
#define TARGET_H
#include <QGraphicsItem>
#include <QColor>
#include <QPainter>
#include <QRectF>

class Target : public QGraphicsItem
{
public:
    explicit Target();

    QRectF boundingRect() const;

    QPainterPath shape() const;

    void paint(QPainter *painter,
               const QStyleOptionGraphicsItem *option,
               QWidget *widget);

protected:
    void advance(int step);

private:
    qreal dx, dy;
    qreal x, y;
    qreal w, h;
};
#endif // TARGET_H
```

Collision Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// target.cpp
#include "target.h"
#include <QList>
#include <QGraphicsScene>

Target::Target()
{
    dx = 0.00;
    dy = 0.00;
    x = 0.0;
    y = 0.0;
    w = 40.0;
    h = 20.0;
}

void Target::paint(QPainter *painter,
                   const QStyleOptionGraphicsItem *option,
                   QWidget *widget)
// paint() paints item in local coordinates
{
    static QColor color = Qt::green;
    if ( !scene()->collidingItems(this).isEmpty() )
    {
        color = Qt::red;
    }
    painter->setBrush(  color  );
    painter->drawRect(-w/2, -h/2, w, h);
}
```

Is list of items colliding with this target item not empty?

Collision Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// target.cpp -- continued

QRectF Target::boundingRect() const
// Determines bounds for repainting
{
    qreal adjust = 0.5;
    return QRectF(-w/2 - adjust, -h/2 - adjust, w + adjust, h + adjust);
}

void Target::advance(int step)
// Advances item by frame
{
    if (step == 0)
        return;

    x = this->pos().x();
    y = this->pos().y();
    setPos(x, y);       // Set item position in parent coordinates
}

QPainterPath Target::shape() const
// Returns shape for collision detection
{
    QPainterPath path;
    path.addRect(-w/2, -h/2, w, h);
    return path;
}
```

Collision Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// dialog.h

#ifndef DIALOG_H
#define DIALOG_H

#include <QDialog>
#include <QGraphicsScene>
#include <QTimer>
#include "projectile.h”
#include “target.h”

namespace Ui {class Dialog;}

class Dialog : public Qdialog
{
    Q_OBJECT

public:
    explicit Dialog(QWidget *parent = 0);
    ~Dialog();

private:
    Ui::Dialog *ui;
    QGraphicsScene* scene;
    Projectile* p1;
    Target* t1;
};

#endif // DIALOG_H
```

Collision Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// dialog.cpp
#include "dialog.h"
#include "ui_dialog.h"
#include <QtDebug>

Dialog::Dialog(QWidget *parent) :  QDialog(parent),  ui(new Ui::Dialog)
{
    ui->setupUi(this);

    // Create and configure scene object
    scene = new QGraphicsScene(this);
    scene->setItemIndexMethod(QGraphicsScene::NoIndex);
    scene->setSceneRect(-200, -150, 400, 300);
    scene->setBackgroundBrush(Qt::blue);

    // Configure graphics view object
    ui->graphicsView->setScene(scene);
    ui->graphicsView->setRenderHint(QPainter::Antialiasing);

    // Add projectile to scene
    p1 = new Projectile;
    p1->setPos(-200, -150);
    scene->addItem(p1);

    // Add target to scene
    t1 = new Target;
    t1->setPos(-200, 100);
    scene->addItem(t1);

    // Create and configure timer object
    QTimer* timer = new QTimer;
    connect(timer, SIGNAL(timeout()), scene, SLOT(advance()));
    timer->setInterval(1000/33);
    timer->start();
}
Dialog::~Dialog(){ delete ui; }
```

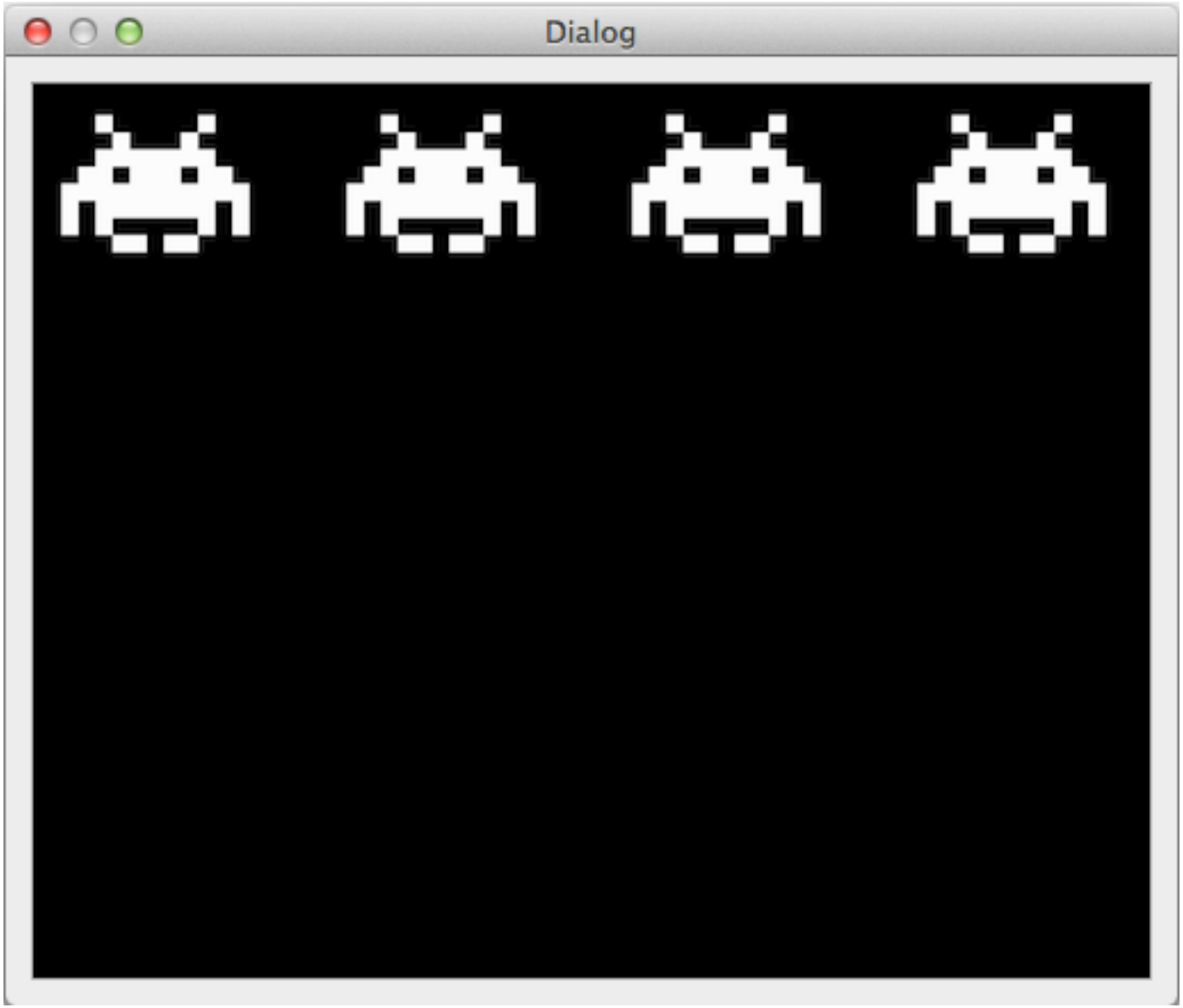
Pixmap Animation Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

- Goals
  - Create a **Alien** class that inherits from **QGraphicsItem**
  - Create a custom Dialog which uses a **QGraphicsView** object to display a **QGraphicsScene** object
  - Add an array of **Alien** objects to the scene and animate their travel
  - Define a series of **QPixmap** images to be displayed on each scene advance

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Pixmap Animation Example



Pixmap Animation Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// alien.h
#ifndef ALIEN_H
#define ALIEN_H
#include <QGraphicsItem>
#include <QColor>
#include <QPainter>
#include <QRect>
class Alien : public QGraphicsItem
{
public:
    explicit Alien();
    QRectF boundingRect() const;
    QPainterPath shape() const;
    void paint(QPainter *painter, const QStyleOptionGraphicsItem *option,
               QWidget *widget);
protected:
    void advance(int step);
private:
    QPixmap pmaps[8];
    qreal dx, dy;
    qreal x, y;
    qreal w, h;
    int index;
};
#endif // ALIEN_H
```

Pixmap Animation Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// alien.cpp

#include "alien.h"

Alien::Alien()
{
    dx = 0.00;
    dy = 2.50;
    x = 0.0;
    y = 0.0;
    w = 100.0;
    h = 72.0;
    index = 0;
    pmaps[0] = QPixmap(":/images/invader1.png");
    pmaps[1] = QPixmap(":/images/invader1.png");
    pmaps[2] = QPixmap(":/images/invader1.png");
    pmaps[3] = QPixmap(":/images/invader1.png");
    pmaps[4] = QPixmap(":/images/invader2.png");
    pmaps[5] = QPixmap(":/images/invader2.png");
    pmaps[6] = QPixmap(":/images/invader2.png");
    pmaps[7] = QPixmap(":/images/invader2.png");
}
```

Pixmap Animation Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// alien.cpp -- continued

void Alien::paint(QPainter *painter,
                  const QStyleOptionGraphicsItem *option,
                  QWidget *widget)
// paint() paints item in local coordinates
{
    index = (index + 1) % 8;
    painter->drawPixmap(-w/2, -h/2, pmaps[index]);
}

QRectF Alien::boundingRect() const
// Determines bounds for repainting
{
    qreal adjust = 1.0;
    return QRectF(-w/2 - adjust, -h/2 - adjust, w + adjust, h + adjust);
}

void Alien::advance(int step)
// Advances item by frame
{
    if (step == 0)
        return;
    x = this->pos().rx();
    y = this->pos().ry();
    x = x + dx;
    y = y + dy;
    setPos(x, y);
}
```

Pixmap Animation Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// alien.cpp -- continued


QPainterPath Alien::shape() const
// Returns shape for collision detection
{
    QPainterPath path;
    path.addRect(-w/2, -h/2, w, h);
    return path;
}
```

Pixmap Animation Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// dialog.h
#ifndef DIALOG_H
#define DIALOG_H
#include <QDialog>
#include <QGraphicsScene>
#include <QTimer>
#include "alien.h"

namespace Ui {class Dialog;}

class Dialog : public Qdialog
{
    Q_OBJECT
public:
    explicit Dialog(QWidget *parent = 0);
    ~Dialog();
private:
    Ui::Dialog *ui;
    QGraphicsScene* scene;
    Alien* aliens[4];
};

#endif // DIALOG_H
```

Pixmap Animation Example

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

```
// dialog.cpp
#include "dialog.h"
#include "ui_dialog.h"
#include <QtDebug>
Dialog::Dialog(QWidget *parent) : QDialog(parent), ui(new Ui::Dialog)
{
    ui->setupUi(this);

    // Create and configure scene object
    scene = new QGraphicsScene(this);
    scene->setItemIndexMethod(QGraphicsScene::NoIndex);
    scene->setSceneRect(-200, -150, 400, 300);
    scene->setBackgroundBrush(Qt::black);

    // Configure graphics view object
    ui->graphicsView->setScene(scene);
    ui->graphicsView->setRenderHint(QPainter::Antialiasing);

    // Add projectiles to scene
    for(int k = 0; k < 4; k++)
    {
        aliens[k] = new Alien;
        aliens[k]->setPos(-190 + k * 125, -150);
        scene->addItem(aliens[k]);
    }

    // Create and configure timer object
    QTimer* timer = new QTimer;
    connect(timer, SIGNAL(timeout()), scene, SLOT(advance()));
    timer->setInterval(1000/33);
    timer->start();
}
Dialog::~Dialog(){ delete ui;}
```

Key Points

- Graphics scene objects provide a convenient means of managing multiple graphical items
- The advance() and paint() methods of graphics items may be used in conjunction with a timer to animate display of the item
- Be mindful of coordinate systems