Angry Trojans

Andrew, Hortencia, Vanessa, Xu

Game Display/Demo

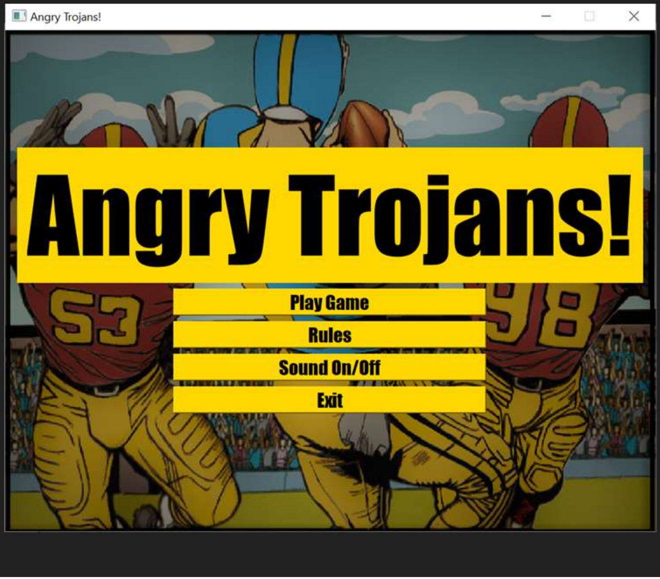
- Install Qt on your computer.
- Install the project zip file.
- Open the .pro file. This will open the project and all files included in the zip.
- Click the green play button, or keyboard shortcut to run the game.
- Click the “Play Game” button on the home screen to begin playing! (or “Rules” to review the rules for the game or “Exit” to exit the application)

System Specifications

- Cross-platform game, compatible with any resolution/both Mac and PC
- Optimized for Windows PC (1920x1080 resolution), with Microsoft Visual C++ Compiler

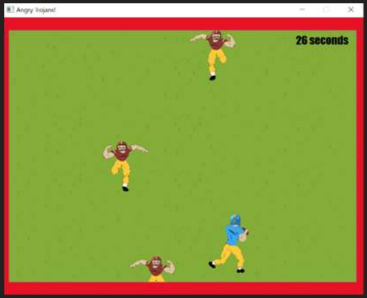
Basic Overview

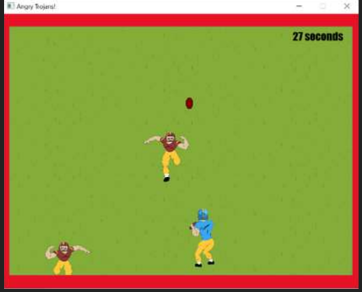
- USC vs. UCLA game
- User plays a UCLA football player defending against enemy USC football players
- Objective: dodge or destroy the USC players
- Total of 10 rounds (30 seconds each)
- Difficulty increases with each level



Rules/Instructions

Use the right and left arrow keys to move and dodge enemies.



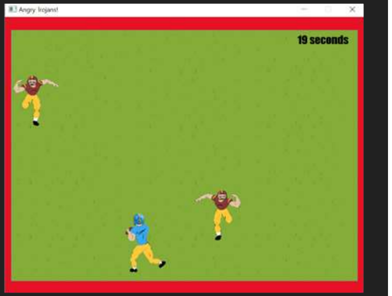


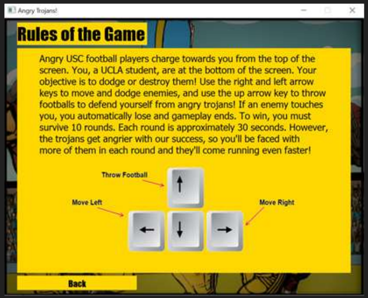
Use the up arrow to throw footballs

to defend yourself from opponents.

If an enemy touches you,

you will lose the round and gameplay will end.





Rules of the Game

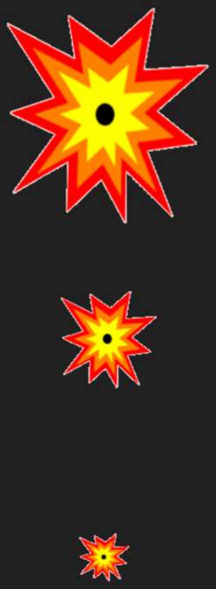
Angry USC football players charge towards you from the top of the screen. You, a UCLA student, are at the bottom of the screen. Your objective is to dodge or destroy them! Use the right and left arrow keys to move and dodge enemies, and use the up arrow key to throw footballs to defend yourself from angry trojans! If an enemy touches you, you automatically lose and gameplay ends. To win, you must survive 10 rounds. Each round is approximately 30 seconds. However, the trojans get angrier with our success, so you'll be faced with more of them in each round and they'll come running even faster!

To win, you must survive 10 rounds.

Each round is approximately 30 seconds long. Speed and amount of enemies will increase with each level!

Game Design

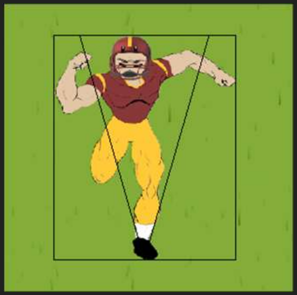
UCLA Bruin

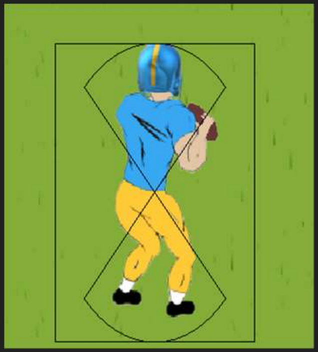


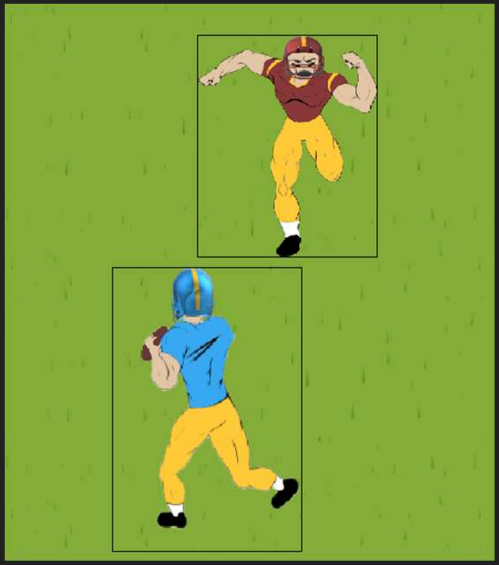
Effect

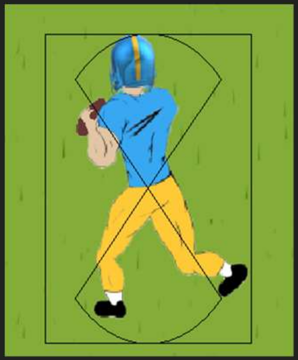
USC Trojan

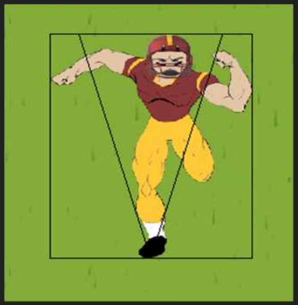
Bounding Shapes & Collision Detection



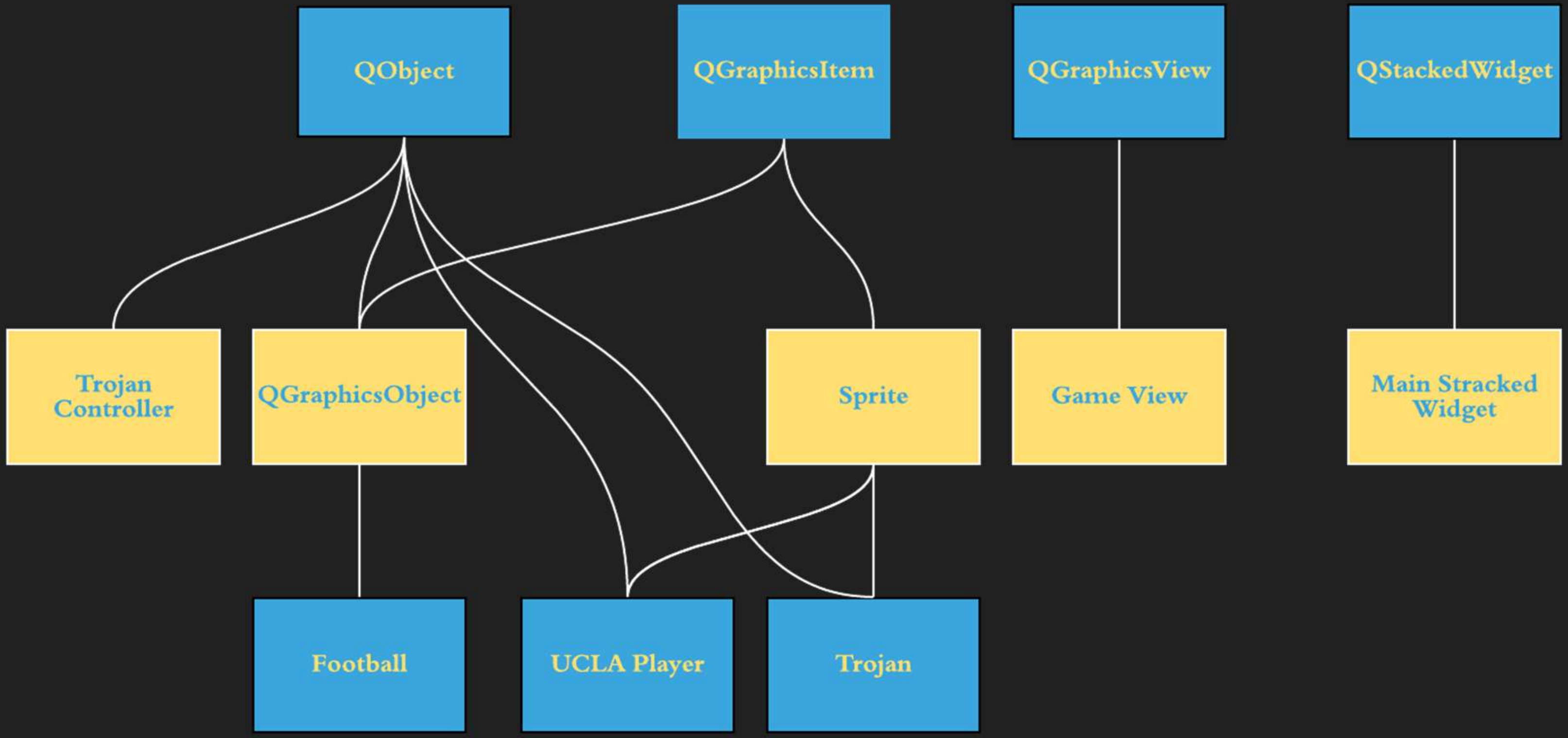








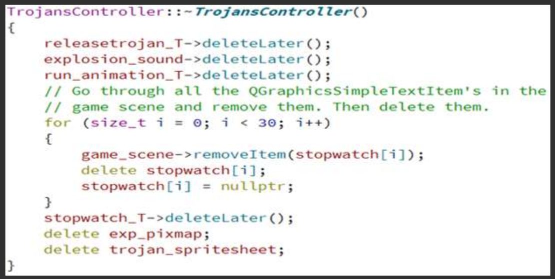
Class Hierarchy



Memory Management

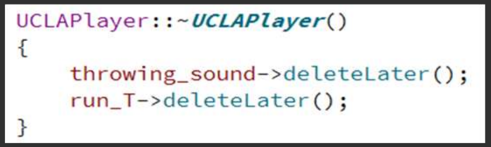
Common Destructors:

TrojanController Destructor:



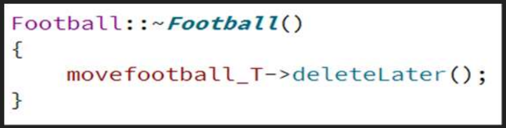
```
TrojansController::~TrojansController()
{
    releasetrojan_T->deleteLater();
    explosion_sound->deleteLater();
    run_animation_T->deleteLater();
    // Go through all the QGraphicsSimpleTextItem's in the
    // game scene and remove them. Then delete them.
    for (size_t i = 0; i < 30; i++)
    {
        game_scene->removeItem(stopwatch[i]);
        delete stopwatch[i];
        stopwatch[i] = nullptr;
    }
    stopwatch_T->deleteLater();
    delete exp_pixmap;
    delete trojan_spritesheet;
}
```

UCLAPlayer Destructor:



```
UCLAPlayer::~UCLAPlayer()
{
    throwing_sound->deleteLater();
    run_T->deleteLater();
}
```

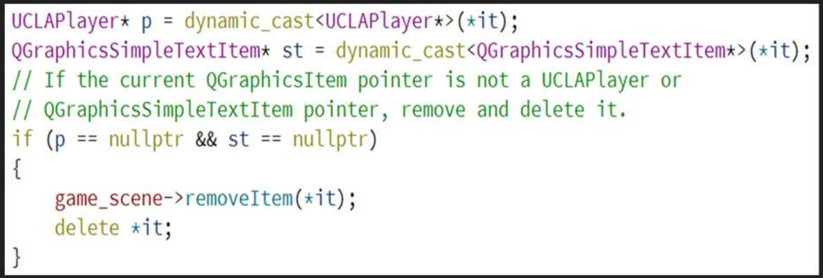
Football Destructor:



```
Football::~Football()
{
    movefootball_T->deleteLater();
}
```

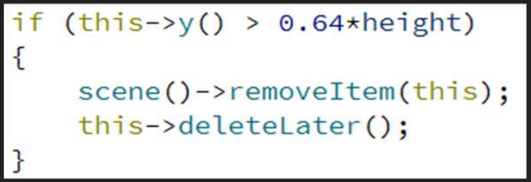
Other Deallocation Methods:

Deletion of items after round ends:



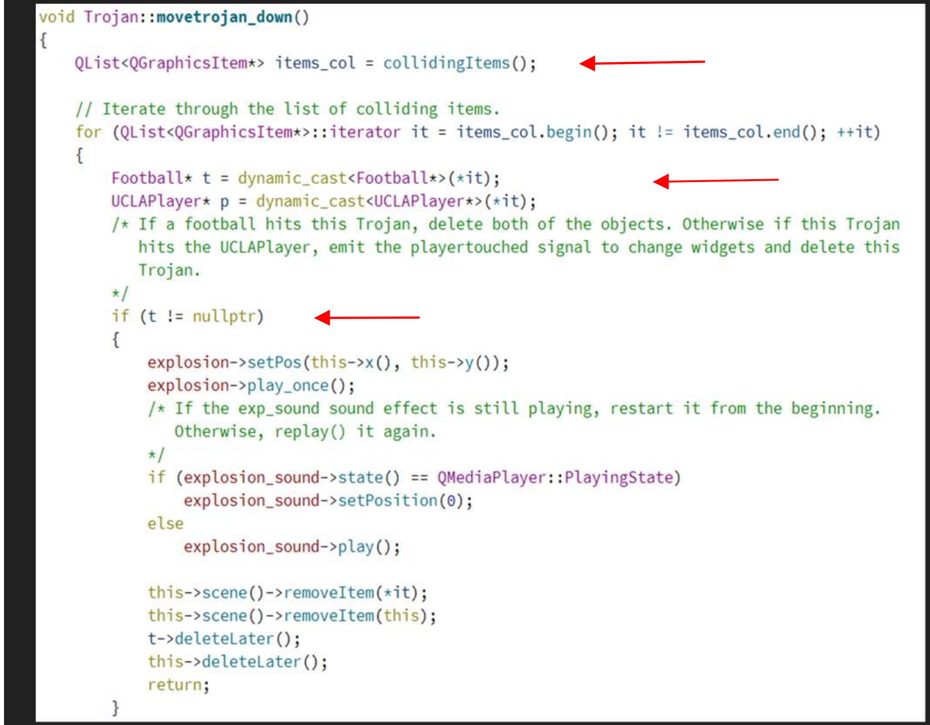
```
UCLAPlayer* p = dynamic_cast<UCLAPlayer*>(*it);
QGraphicsSimpleTextItem* st = dynamic_cast<QGraphicsSimpleTextItem*>(*it);
// If the current QGraphicsItem pointer is not a UCLAPlayer or
// QGraphicsSimpleTextItem pointer, remove and delete it.
if (p == nullptr && st == nullptr)
{
    game_scene->removeItem(*it);
    delete *it;
}
```

Deletion of Trojan going out of scene:



```
if (this->y() > 0.64*height)
{
    scene()->removeItem(this);
    this->deleteLater();
}
```

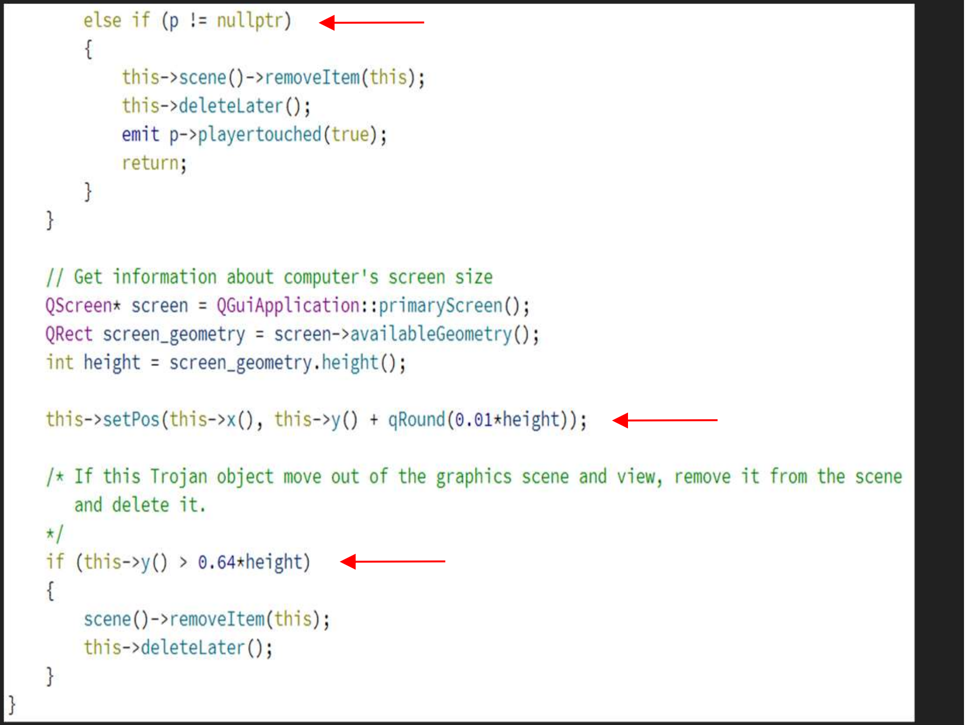
Notable function



```
void Trojan::movetrojan_down()
{
    QList<QGraphicsItem*> items_col = collidingItems();

    // Iterate through the list of colliding items.
    for (QList<QGraphicsItem*>::iterator it = items_col.begin(); it != items_col.end(); ++it)
    {
        Football* t = dynamic_cast<Football*>(*it);
        UCLAPlayer* p = dynamic_cast<UCLAPlayer*>(*it);
        /* If a football hits this Trojan, delete both of the objects. Otherwise if this Trojan
           hits the UCLAPlayer, emit the playertouched signal to change widgets and delete this
           Trojan.
        */
        if (t != nullptr)
        {
            explosion->setPos(this->x(), this->y());
            explosion->play_once();
            /* If the exp_sound sound effect is still playing, restart it from the beginning.
               Otherwise, replay() it again.
            */
            if (explosion_sound->state() == QMediaPlayer::PlayingState)
                explosion_sound->setPosition(0);
            else
                explosion_sound->play();

            this->scene()->removeItem(*it);
            this->scene()->removeItem(this);
            t->deleteLater();
            this->deleteLater();
            return;
        }
```



```
        else if (p != nullptr)
        {
            this->scene()->removeItem(this);
            this->deleteLater();
            emit p->playertouched(true);
            return;
        }
    }

    // Get information about computer's screen size
    QScreen* screen = QGuiApplication::primaryScreen();
    QRect screen_geometry = screen->availableGeometry();
    int height = screen_geometry.height();

    this->setPos(this->x(), this->y() + qRound(0.01*height));

    /* If this Trojan object move out of the graphics scene and view, remove it from the scene
       and delete it.
    */
    if (this->y() > 0.64*height)
    {
        scene()->removeItem(this);
        this->deleteLater();
    }
}
```

thank you!