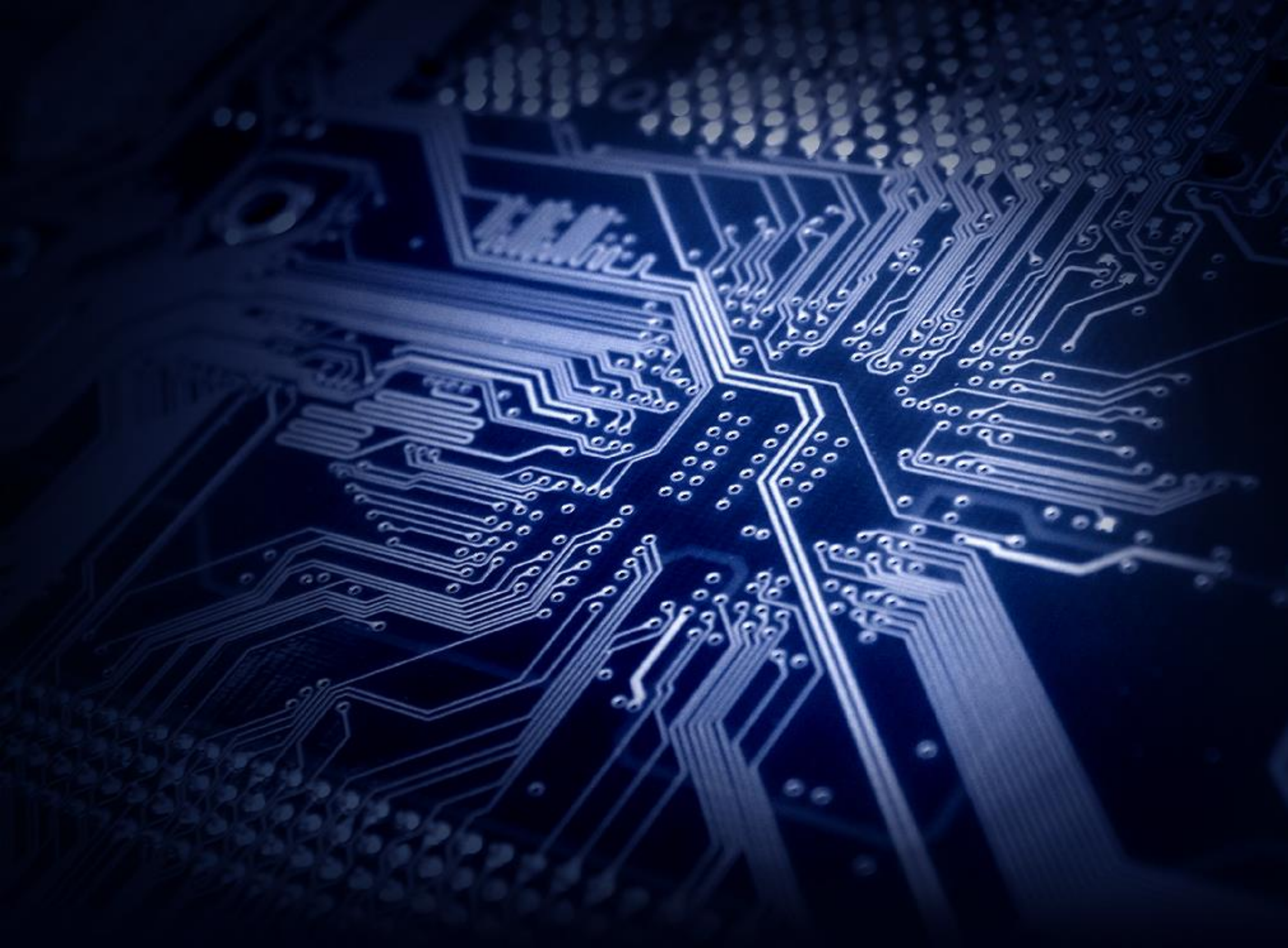


GUI Programming with QT

Final Project (Flappy bird game)

This report is considered as an explanation and clarification of the general concept of the project and all the functionality and classes used and implemented in order to realize this project.

Written by: El HARRARI ANAS
Date: 07/02/2022

École d'Ingénierie Digitale et d'Intelligence Artificielle

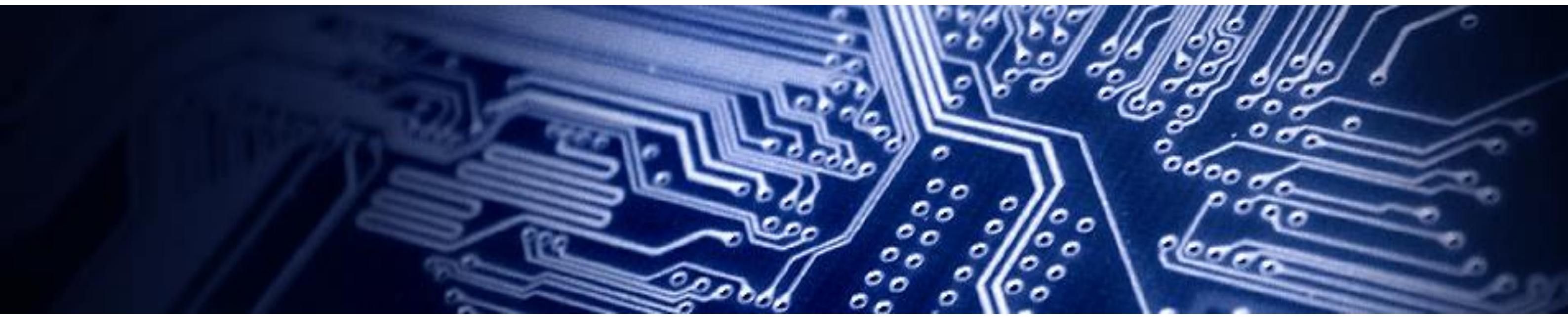
The Game development with QT

Qt has traditionally had a very strong bond to OpenGL, and this bond has served Qt very well, as OpenGL is portable and preferment over a wide range of platforms. It has an impressive set of features and a lot of good thinking has gone into its API to ensure that it can leverage the full potential of modern graphics hardware while staying flexible and preferment. GUI code is Qt's forte. It has great tools for working with GUIs that for some small'ish 2D games will actually be usable as level editors. The Game Development shows a great opportunity to dive into the world of QT GUI Frameworks. This interesting concept helps to practice a large amount of c++ concepts and the learning of the most powerful QT frameworks and classes.

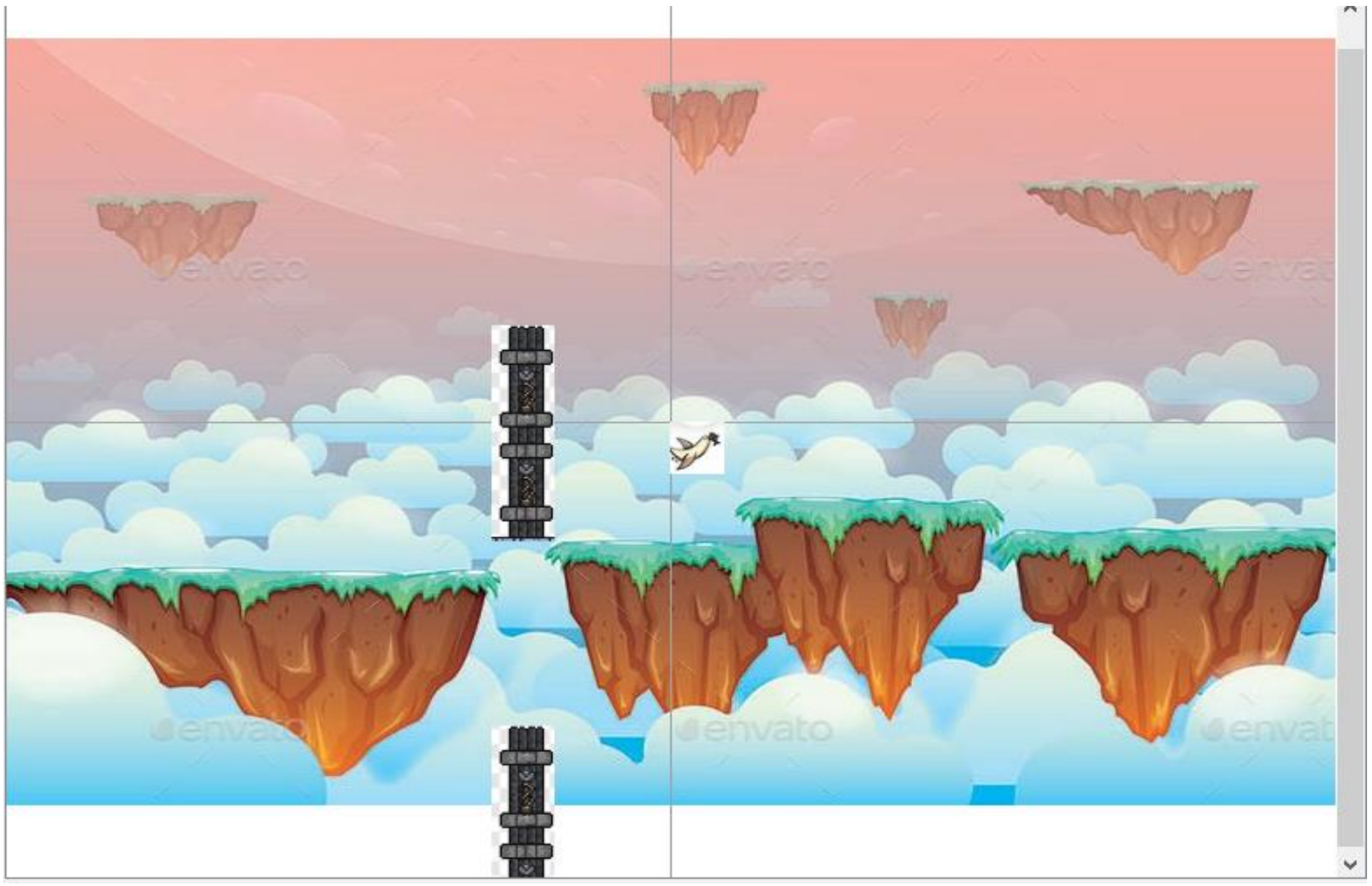
In this project, we had made a full functional flappy bird game using the classes and the frameworks of QT.

Flappy Bird Game

The Flappy Bird game is a popular mobile game developed for the first time in Vietnam. The game is a side scoller where the player controls a bird, attempt to fly between columns of green pipes without hitting them. The game is considered as a fully functionality game that necessity a good level of programming. The flappy Bird Game is a game where the programmer will need many frameworks and a good manipulation of all the c++ concepts. The game presents an interesting project that goes through all what we have been through in our course of GUI Programming with QT.It helps to practice all the c++ concepts and the powerful classes of QT. In order to realize this project, we had used many Qt Classes such as (QGraphicsItems, QPropertyAnimation, QWidget...).These classes provide large functionalities that make the programming of such a complex project easier



Preview :



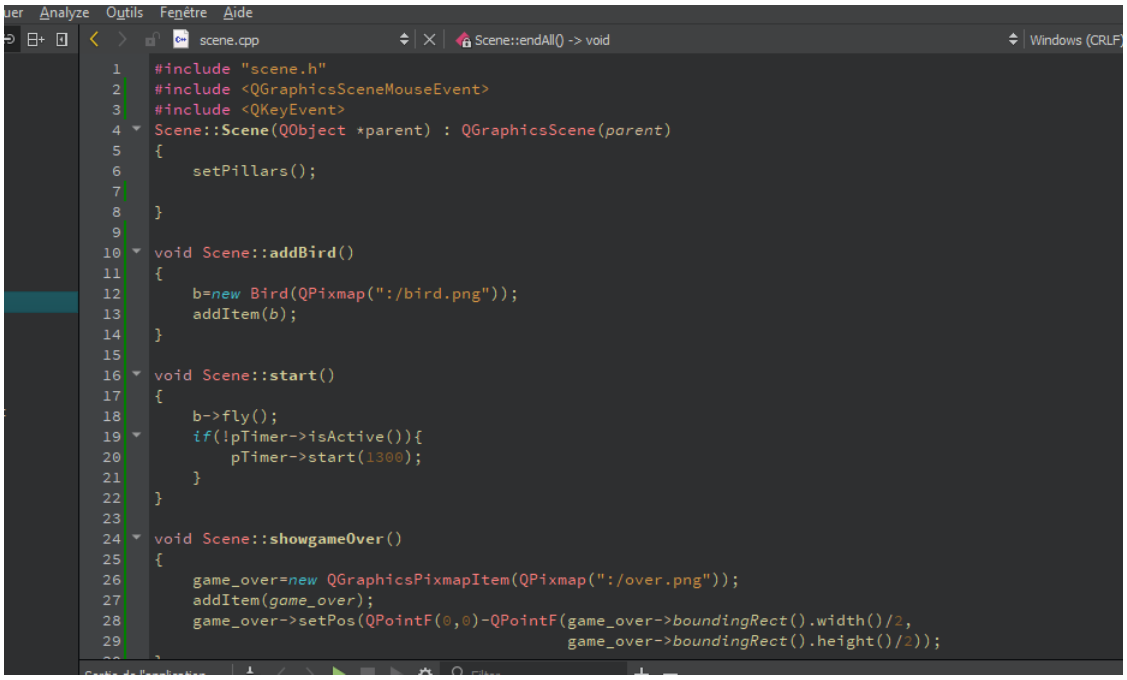
This project, as it has been said before, is a flappy bird game. A game where the player can control a bird using the key board (The Space Key) or the mouse (The left button). The game ends when the bird hits one of the pillars. The concept of the game may look simple but the implementation of these using the concepts of programming and the theories of Math and physics presents a big challenge. In this report, we will be going throw all the principal points of this project in order to clarify our code and the choice of the existence methods.

Across the code:

Our work in this project was essentially divided into for main classes: each class is responsible of a large part of our result (The Scene Class, Bird, Pillar and the widget one)

- **The scene Class:**

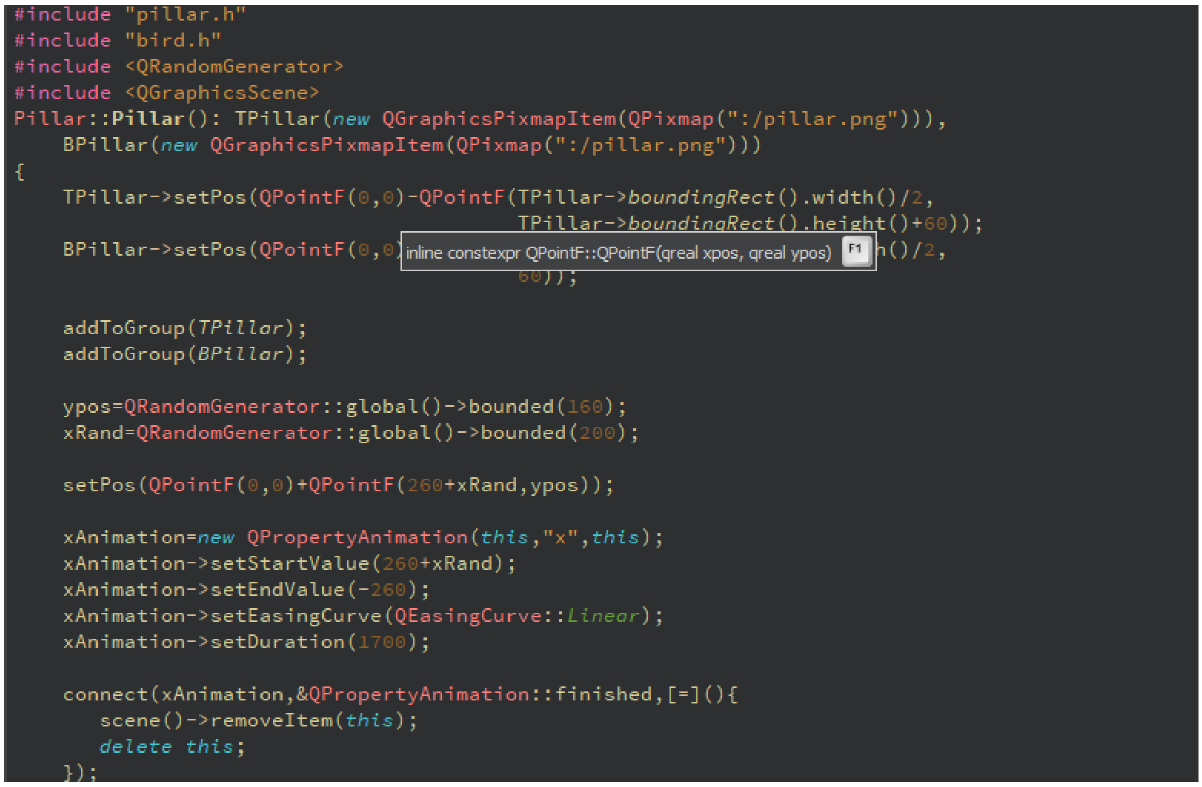
The scene class is the main and the most important class in our project. This class aims to define the scene of the game and responsible of all the movement and the interactions between the other classes and componments.It also the classes where we define the start and the ending of the game.



```cpp
#include "scene.h"
#include <QGraphicsSceneMouseEvent>
#include <QKeyEvent>
Scene::Scene(QObject *parent) : QGraphicsScene(parent)
{
    setPillars();

}

void Scene::addBird()
{
    b=new Bird(QPixmap(":/bird.png"));
    addItem(b);
}

void Scene::start()
{
    b->fly();
    if(!pTimer->isActive()){
        pTimer->start(1300);
    }
}

void Scene::showgameOver()
{
    game_over=new QGraphicsPixmapItem(QPixmap(":/over.png"));
    addItem(game_over);
    game_over->setPos(QPointF(0,0)-QPointF(game_over->boundingRect().width()/2,
                                           game_over->boundingRect().height()/2));
```

This class inherits from QGraphicsScene, which is a powerful class in QT that provides a lot of fonctionnalities.In this is project, we had used a lot of QT classes but importantly The QGraphicsItem, The QGraphicsScene, QObject and the QPropertyAnimation.

This project contains many interesting parts but generally, this work is divided in two main parts: the first one consist of the establishment of all the graphical and the designer componments.However, the other part remains an implementation of Mathematical and physical laws that were presented in this project in an interesting way.



```
#include "pillar.h"
#include "bird.h"
#include <QRandomGenerator>
#include <QGraphicsScene>
Pillar::Pillar(): TPillar(new QGraphicsPixmapItem(QPixmap(":/pillar.png"))),
    BPillar(new QGraphicsPixmapItem(QPixmap(":/pillar.png")))
{
    TPillar->setPos(QPointF(0,0)-QPointF(TPillar->boundingRect().width()/2,
                                         TPillar->boundingRect().height()+60));
    BPillar->setPos(QPointF(0,0 inline constexpr QPointF::QPointF(qreal xpos, qreal ypos) F1 h()/2,
                                         60));

    addToGroup(TPillar);
    addToGroup(BPillar);

    ypos=QRandomGenerator::global()->bounded(160);
    xRand=QRandomGenerator::global()->bounded(200);

    setPos(QPointF(0,0)+QPointF(260+xRand,ypos));

    xAnimation=new QPropertyAnimation(this,"x",this);
    xAnimation->setStartValue(260+xRand);
    xAnimation->setEndValue(-260);
    xAnimation->setEasingCurve(QEasingCurve::Linear);
    xAnimation->setDuration(1700);

    connect(xAnimation,&QPropertyAnimation::finished,[=](){
       scene()->removeItem(this);
       delete this;
    });
```

Conclusion:

The Game development with QT present an interesting concept and a great opportunity to practice the majority of the programming concepts. During this project , we were throw a lot of concepts that we had seen during our course of "GUI PROGRAMMING WITH QT".We also had the opportunity to practice all the oriented programming concepts in order to realize one of the popular games in the world "the flappy bird game"