**2022-23春季学期程序设计实习**

**Qt大作业《星穹铁船》实验报告**

组长：黎奕欣

组员：董博瑄，张洋

**一、项目简介**

在一个月黑风高的晚上，北大信科的三位同学在一起商讨Qt大作业的立项事宜。作为一名资深科幻迷和游戏迷，组长黎奕欣对游戏《星际拓荒》和刚上线不久的《崩坏：星穹铁道》颇感兴趣。凭借这些游戏提供的灵感，模拟太空飞行小游戏《星穹铁船》诞生了。

**二、功能介绍**

本游戏是一款模拟太空飞行的2D小游戏。玩家在游戏中需要驾驶飞船，在具有真实引力环境的太空中躲避星体、前往一个又一个得分点，并在有限的时间和飞船生命值下得到尽可能高的分数。

**1 主菜单**

程序运行时进入。主界面中可以选择开始游戏、阅读教程以及查看制作团队。 

图 1-主菜单

**2 游戏教程与制作人员**

主菜单的“游戏教程”与“制作人员”按钮分别打开两个窗口，展示相关信息。

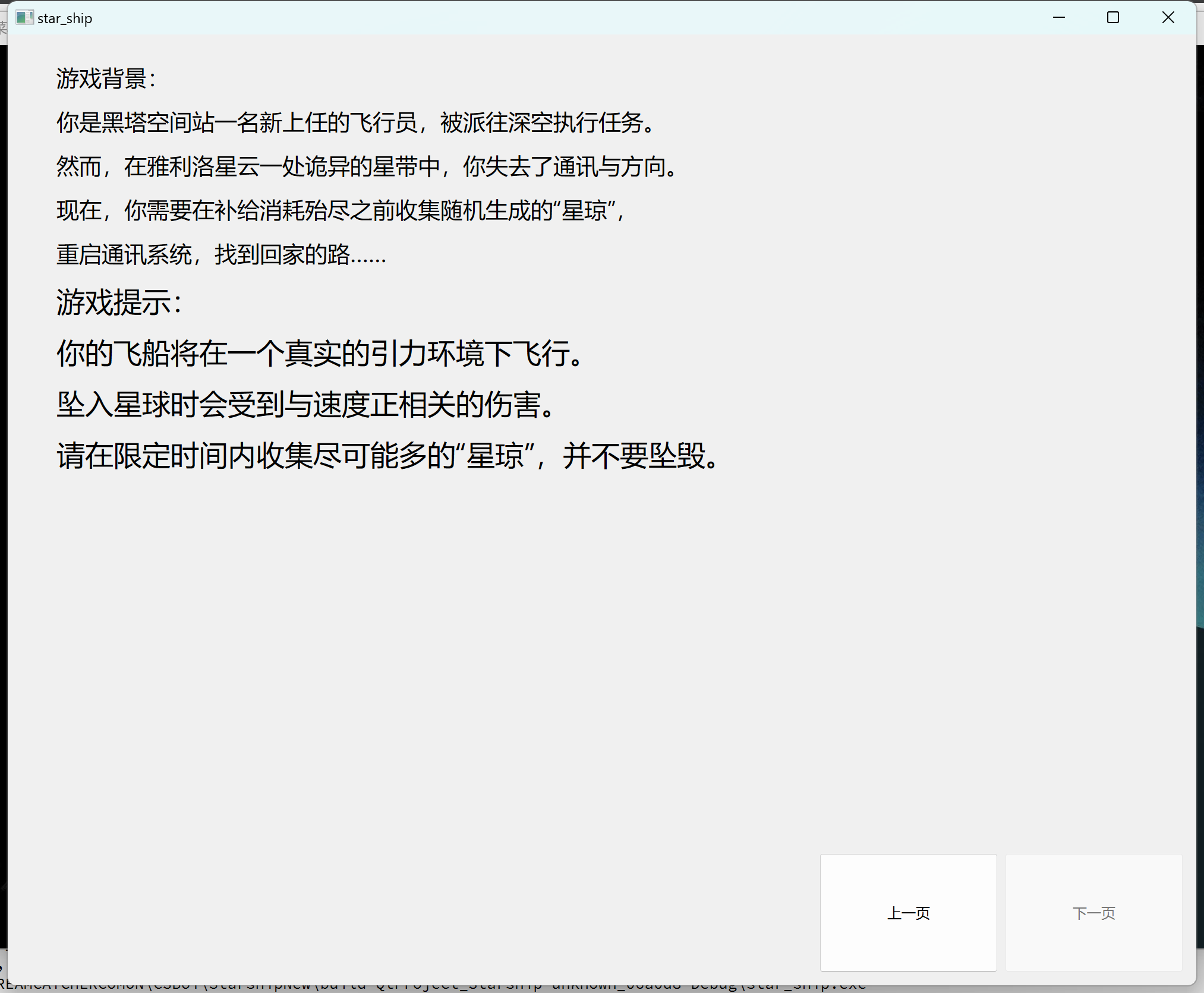


图 2-游戏背景与提示介绍

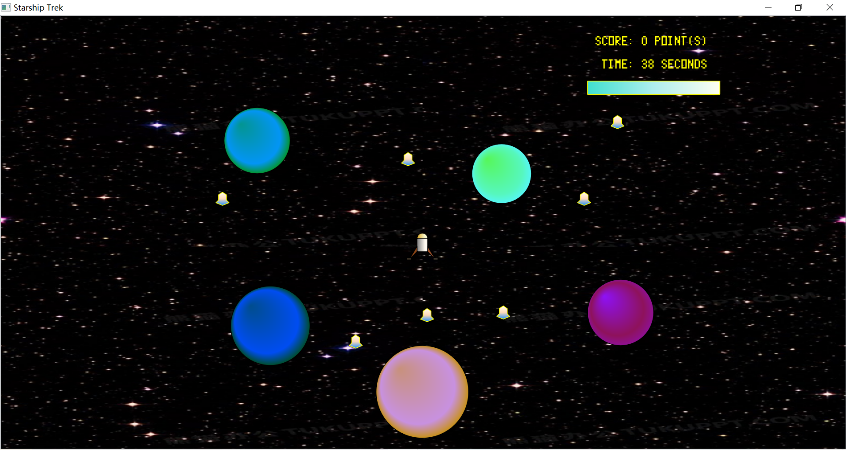


图 3-游戏画面

点击“开始游戏”即可开始。游戏初始界面如图所示。地图中间为玩家操作的飞船（外形参考《星际拓荒》中的太空船）。地图中会出现5个设定好的星球和若干随机生成的积分（外形参照《崩坏：星穹铁道》中的货币“星琼”）。右上显示了玩家的当前得分、消耗时间和剩余生命值。

玩家在自身提供的动力与星球提供的引力构成的场中运动；到达积分后会将其吃掉，并使得积分增加；坠入星球时，会受到与速度呈正相关的伤害。时间停止或生命值降为0后，弹出结算界面，显示最终得分。

1. **项目模块与类设计细节**

项目包含下图所示的这些类。其中”clock\_prompt”与生命值和时间的显示、记录有关，”coin” “planet” “starship”分别处理积分点、星球与飞船的构建和运行，这些类都继承于QGraphicsItem；”game\_scene”与”myScene”是游戏运行的场景类，继承于QGraphicsScene；剩余的类与游戏教程、结算页面、主选单等的展示有关，继承于QWidget。

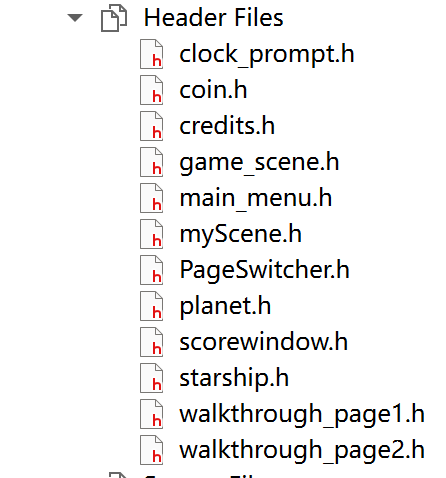


图 4-头文件列表

游戏的运行主要由两个计时器发出的信号推动，计时器一以1/33秒为单位，发出信号给槽函数advance，推动飞船改变运动状态、生命值，以及myScene对象更新生命值与分数、检测游戏是否结束；计时器二以一秒为单位，发出信号给槽函数Stop，更新时间并检测是否超时。

游戏的美工大多用QPainter类对象手绘；添加了合适的背景图片，为星球添加了渐变色、飞船添加了与运动方向匹配的尾迹，画面效果更细腻。积分点的位置随机生成，并确保分布大致合理，使得游戏有随机性、可玩性。

**关于每个类：**

1. **clock\_prompt**

这个类公有继承了QGraphicsItem和QObject两个类。它的主要功能为记录并显示时间、生命和分数。其中主要的成员变量和成员函数如下（均为public）：

static qreal TIME\_LIMIT;//游戏时间限制

QTimer \*timer;//计时器，实现计时功能

qreal seconds=0;//已消耗的时间

qreal life=MAX\_LIFE;//飞船最大生命值

qreal score=0;//目前得分

void update\_Time();//更新当前时间

inline QRectF boundingRect() const override;

inline QPainterPath shape() const override;

inline void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option,

QWidget \*widget) override;

Inline void advance(int step) override;

我们覆写了Qt内置的boundingRect，shape，paint和advance函数，从而实现绘制和实时更新时间、生命、得分的功能。

1. **Coin**

这个类公有继承了QGraphicsItem。它实现了“星琼”这个可拾取物品。其主要函数和成员变量如下（均为public）：

qreal px, py;//Coin的位置

bool eaten = 0;//用于标记Coin是否已被采集

qreal radius = 10;//Coin的尺寸，boundingRect为20\*20的正方形

inline QRectF boundingRect() const override;

inline QPainterPath shape() const override;

inline void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option,

QWidget \*widget) override;

inline int type() const override;

我们覆写了Qt内置的boundingRect，shape，paint函数，实现了Coin的碰撞箱和绘制。我们还覆写了Qt内置的type函数，将Coin类的type返回值设为UserType+3，从而使Coin可以被区分于其他item。

1. **Credits**

这个类公有继承了QDialog，用于显示制作人员。制作人员界面使用credits.ui绘制。

1. **game\_scene**

这个类公有继承了QGraphicsView，用于实现背景图片、飞船等对象的呈现。其主要函数和成员变量如下：

public:

game\_scene(QGraphicsScene\* scene);//构造函数

public slots:

void Stop();//用于检测游戏是否结束的槽函数

protected:

void showEvent(QShowEvent\* event) override;//展示窗口

void resizeEvent(QResizeEvent\* event) override;//调整窗口大小

1. **main\_menu**

公有继承了QDialog，实现了主选单，并实现了从主菜单到游戏场景的跳转。主选单界面使用main\_menu.ui绘制。

1. **myScene**

公有继承了QGraphicsScene，是游戏界面的主类。它实现了时间、生命与分数的更新、coins的随机绘制、行星与飞船的绘制与运作。其主要成员函数和成员变量如下：

//星球的生成

Planet star = Planet(300, 100, 50, 1284379128);

Planet star\_2 = Planet(-230, 120, 60, 19854);

Planet star\_3 = Planet(120, -110, 45, 1314322266);

Planet star\_4 = Planet(0, 220, 70, 13144444);

Planet star\_5 = Planet(-250, -160, 50, 234897);

Coin coins[7];//7个硬币，在构造函数中随机位置生成

Starship ship;//飞船，在构造函数中添加至myScene

myScene();//构造函数

//实现飞船键盘控制

void keyPressEvent(QKeyEvent\* event);

void keyReleaseEvent(QKeyEvent\* event);

//实现星琼随机生成

bool validRandom(qreal x, qreal y);

//计时器

clock\_prompt\* newClock = new clock\_prompt();

qreal stopTime = 40;

qreal Time = 0;

public slots:

void Stop();//计时器槽函数

我们覆写了Qt内置的keyPressEvent和keyReleaseEvent，实现了WASD四键控制飞船。

1. **pageSwitcher**

公有继承了QWidget，用于实现教程中的翻页。其主要成员函数和成员变量如下：

public:

PageSwitcher(QWidget \*parent = nullptr);//构造函数

void addPage(QWidget \*page);//加页

//槽函数

private slots:

void switchToPrevPage();//上一页

void switchToNextPage();//下一页

private:

void updateButtonStatus();//更新按钮状态

QStackedWidget \*stackWidget;//页面切换控件

QPushButton \*prevButton;//上一页按钮

QPushButton \*nextButton;//下一页按钮

1. **Planet**

公有继承了QGraphicsItem。它实现了游戏中的物品“星球”。其主要成员函数和成员变量如下：

Planet(qreal x, qreal y, qreal r, int color);//Planet的构造函数

inline QRectF boundingRect() const override;

inline QPainterPath shape() const override;

inline void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option,

QWidget \*widget) override;

inline int type() const override;

qreal radius;//星球半径

qreal mass;//星球质量

qreal px, py;//星球位置

int color;//星球主体颜色

我们覆写了Qt内置的boundingRect，shape，paint函数，实现了Planet的碰撞箱和绘制。我们还覆写了Qt内置的type函数，将Coin类的type返回值设为UserType+1，从而使Planet可以被区分于其他item。在此基础上，我们覆写了QGraphicsScene中的collidingItems函数，使飞船只与星球产生物理碰撞交互。

1. **Scorewindow**

公有继承自QWidget，用于展示结算界面。

1. **Starship**

公有继承自QGraphicsItem，实现了飞船的碰撞体积、绘制，并实现了游戏中的全部物理现象，是游戏后端的核心类。其主要成员函数和成员变量如下所示：

Starship();

Starship(QGraphicsItem \*item);

inline QRectF boundingRect() const override;

inline QPainterPath shape() const override;

inline void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option,

QWidget \*widget) override;

inline void advance(int step) override;

qreal angle = 0;//角度

qreal angular\_v = 0;//角速度

qreal angular\_a = 0;//角加速度

qreal velocity = 0;//竖向速度

qreal velocity\_x = 0;//横向速度

qreal acceleration = 0;//喷气加速度

qreal impulse = 0;//冲量

qreal impulse\_x = 0;//横向冲量

qreal angular\_I = 0;//角冲量

qreal gravity\_y = 0;//重力竖向分量

qreal gravity\_x = 0;//重力横向分量

qreal field\_x = 0, field\_y = 0, angular\_f = 0;//限制飞船不出边界的冲力

bool detector = 0;//判断飞船是否进入引力区

qreal life = MAX\_LIFE;//飞船最大生命值

bool operate=1;//判断游戏是否结束

bool ended=0;//判断是否显示了结算界面

我们覆写了Qt内置的boundingRect，shape，paint函数，实现了Starship的碰撞箱和绘制。我们在advance中构建了飞船运行的物理规律，包括飞船的驱动力、星球引力、碰撞冲量等，更新飞船的运动装态。同时，advance更新飞船的生命和得分。

1. **Walkthrough\_page1 & Walkthrough\_page2**

均公有继承自QWidget，用于展示游戏教程。这两个界面均使用ui文件绘制。

1. **main**

项目的主函数，产生并呈现一个main\_menu实例，实现了程序的运行。

**四、分工**

|  |  |  |
| --- | --- | --- |
| 类（函数）名称 | 贡献者 | 大致功能 |
| clock\_prompt | 董博瑄、张洋 | 记录并显示时间、生命与分数 |
| Coin | 董博瑄 | 实现积分点的绘制与功能 |
| Credits | 黎奕欣 | 展示制作人员 |
| game\_scene | 黎奕欣 | 实现背景图片、飞船等对象的呈现 |
| main\_menu | 黎奕欣 | 实现主选单 |
| myScene | 董博瑄、张洋 | 游戏界面的主类，实现时间、生命与分数的更新、coins的随机绘制、行星与飞船的运作。 |
| pageSwitcher | 黎奕欣 | 展示教程 |
| Planet | 董博瑄 | 实现星球的绘制与功能 |
| Scorewindow | 黎奕欣 | 展示结算界面 |
| Starship | 董博瑄、张洋 | 实现飞船的绘制与功能 |
| Walkthrough\_page1 & walkthrough\_page2 | 黎奕欣 | 展示教程 |
| main函数 | 黎奕欣、张洋 | 实现程序的运行 |

报告撰写：董博瑄、张洋

讲解视频：黎奕欣

**五、项目总结与反思**

此次项目鼓励我们从零开始认识Qt、并完成了一款可以运行且有一定游戏性的游戏，是较为成功的。我们正确、高效地继承QGraphicsItem与QGraphicsScene类，简洁地实现了需要的效果，是值得肯定的。

然而，此次项目仍有许多不足：

1. 前期学习时间投入较少。在缺乏对Qt的熟练掌握时便开始构思项目，导致有大量的时间花在了方向错误的尝试上，且到最后都有未能实现的功能；

2. 项目时间安排不完全合理。由于对所需时间的错误估计，项目的许多功能不得不搁浅，例如滚动大地图、赛道模式、故事任务等等；

3. 项目分工不完全合理。考虑到组员都是初学者、能力有限，有些功能需要多个人一起钻研、实现，因此没有做到分工明确、并行工作，开发效率有待提高。

**六、致谢**

感谢张勤健老师和助教们为本项目提供的指导与帮助。特别鸣谢《星际拓荒》和《崩坏：星穹铁道》为本项目提供的创作灵感。