

“Jump off the bridge” 说明文档

一、程序功能介绍

本项目灵感来自“Jump off the bridge”，是一款街机风格游戏，目标是跳下桥落入水中以获得胜利。

本项目的特点是界面简洁明快，相对同类其它游戏，角色操作更加多样，道具玩法更加丰富。

项目实现的主要功能如下：

1、多样的角色操作，逼真的物理效果：

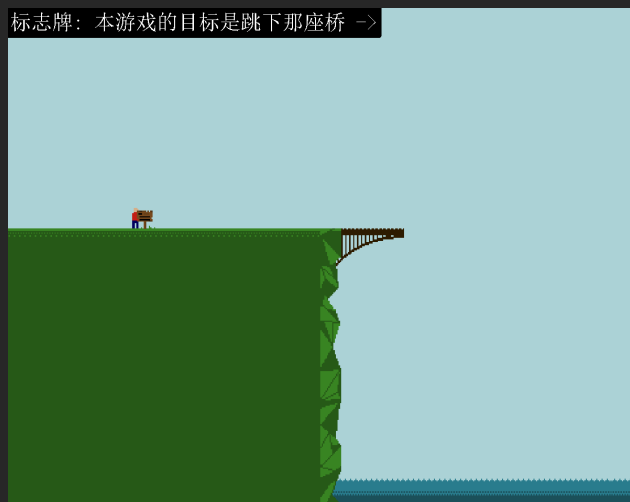
可以用键盘控制角色行走、跳跃、拾取、抛掷物品等动作；在角色操作以外，还实现了丰富的死亡喷血、喷火、射击等特效。

2、丰富的场景体验，较高的完成度：

设计了功能丰富的道具、地形、人物等类，提供新奇的体验；项目完成度较高，共计八个关卡，难度梯度合理，既有充分的新手导引，又有极富挑战性的后期关卡，提供完整的游戏体验。

3、良好的可读性、可扩展性：

具有较合理的类、变量和函数命名，必要的注释，可读性较好；提供了多种物品基类和关卡基类，易于继承实现新的子类，快速设计出新的关卡。



游戏界面示意

功能清单	
<、->方向键	移动、选择关卡
space 键	跳跃
W 键	拾取
E 键	投掷
R 键	重新开始游戏
Esc 键	退出
Enter 键	确认
开始游戏音、背景音、按键音	
木	靠近时显示提示
钥匙	可拾取、投掷，可打开门
门	阻挡玩家，可被钥匙打开，被打开后闪烁消失
各种颜色的公牛	可移动，杀死玩家
老者	靠近时显示提示，呈现说话效果
加长桥面	长度随玩家位置而变化
伪装木牌	靠近时射出子弹，杀死玩家，随后逃跑
老妇	靠近时显示提示，呈现说话效果，阻挡玩家，呈现杖击效果，获得狗后显示提示，允许玩家通过
尖刺	杀死玩家
狗	可拾取、投掷，呈现摇尾巴效果
鬼	可移动，向上弹射玩家
箭	可拾取、投掷
箭匣	生成箭
火球	按正弦函数轨迹移动，杀死玩家
火龙	共三点生命值，受到伤害后红心闪烁变黑，按随机时间间隔吐出火球，受到伤害后有概率进入狂暴状态，连续吐出多发火球

功能清单

二、项目各模块与类设计细节

Gameobject 类

实现游戏中物体的基础功能，包含了物体的绘图、坐标、更新、判断碰撞等等功能。

```
#ifndef GAMEOBJECT_H
#define GAMEOBJECT_H

#include <QObject>
#include <QRectF>
#include <QPixmap>
#include <QPainter>

class GameObject:public QObject{
    Q_OBJECT

public:
    GameObject(qreal x, qreal y, qreal width, qreal height);
    virtual ~GameObject();

    QRectF const rect(){return mRect;}

    virtual void updateRect();
    virtual void draw(QPainter *painter);

    bool collision(GameObject* obj);
    QRectF mRect;

protected:
    QPixmap mGraph;

};

#endif // GAMEOBJECT_H
```

MoveObject 类、StaticObject 类

两个类均是 GameObject 类的子类，StaticObject 类实现静止物体，MoveObject 类多了物体的速度等参数和接口，实现物体移动功能。

```

#ifndef STATICOBJECT_H
#define STATICOBJECT_H

#include "gameobject.h"

class StaticObject:public GameObject{
    Q_OBJECT

public:
    StaticObject(qreal x, qreal y, qreal width, qreal height);
    virtual ~StaticObject();
};

#endif // STATICOBJECT_H

```

```

#ifndef MOVEOBJECT_H
#define MOVEOBJECT_H

#include "gameobject.h"

class MoveObject:public GameObject{
    Q_OBJECT

public:
    MoveObject(qreal x, qreal y, qreal width, qreal height);
    virtual ~MoveObject();

    qreal const xSpeed(){return mxSpeed;}
    qreal const ySpeed(){return mySpeed;}

    void setOnFloor(const bool& _){onFloor=_;}
    bool onFloor(){return onFloor;}

    void setxSpeed(const qreal& xSpeed){mxSpeed=xSpeed;}
    void setySpeed(const qreal& ySpeed){mySpeed=ySpeed;}
    void setPos(const qreal& x,const qreal& y){
        mRect.setRect(x, y, mRect.width(), mRect.height());
    }

    //更新物体位置
    virtual void updateRect();

    void setBeLift(bool val){beLift=val;}

protected:
    qreal mxSpeed;
    qreal mySpeed;
    bool onFloor;
    bool beLift;
    MoveObject* father;

};

#endif // MOVEOBJECT_H

```

Player 类

MoveObject 类的子类，实现了通过按键控制玩家小人行动，如左右方向键控制小人左右移动，空格键实现小人跳跃，W 键和 E 键分别实现小人拾取物品和丢弃物品，同时也实现了玩家小人死亡等多种不同的状态。

```

#ifndef PLAYER_H
#define PLAYER_H

#include <QGraphicsPixmapItem>
#include <QKeyEvent>
#include <QVector>

#include "moveobject.h"
#include "blood.h"

class Player:public MoveObject{
    Q_OBJECT

private:
    MoveObject *liftObj;
    //0为静止, 1为移动
    QPixmap right[2];
    QPixmap left[2];
    QPixmap liftRight[2];
    QPixmap liftLeft[2];
    QPixmap death;
    int indexRight;
    int indexLeft;
    int indexLiftRight;
    int indexLiftLeft;
    bool indexDirection;
    int deathTimer;
    void setIndex();

public:
    Player(qreal x, qreal y, qreal width, qreal height);
    ~Player();

    void updateRect();
    void draw(QPainter *painter);

    void updateState();

    void jump();
    void pickUp(MoveObject* obj);
    void throwing();

    void moveRight();
    void moveLeft();
    void moveNo();

    void setDeath();

    QVector<Blood*> bloodList;
    enum State{
        Static,
        Jump,
        MoveLeft,
        MoveRight,
        Lift,
        LiftMoveLeft,
        LiftMoveRight,
        Death
    }playerState;
};

```

Game（不同 level 的关卡）类

每一个关卡类实现了每一个难度和场景不同的关卡，实现游戏的主要界面及功能，共有 8 个关卡。

```

#ifndef LEVEL1_H
#define LEVEL1_H

#include <QGraphicsView>
#include <QTimer>
#include <QPainter>
#include <QWidget>
#include <QMediaPlayer>

#include "player.h"
#include "floor.h"
#include "signboard.h"

class Level1:public QWidget{
    Q_OBJECT

private:
    Level1(QWidget *parent=NULL,QWidget *next=NULL);
    virtual ~Level1();

    static Level1* uniqueInstance;
    Player *player;
    QPixmap buffer;
    QPixmap ground;
    QPixmap water;
    QPixmap banner;
    QPixmap ret[2];
    QTimer engine;
    Floor *floor;
    int indexWater;
    SignBoard* board;
    bool leftKeyPressed;
    bool rightKeyPressed;
    QWidget* main;
    bool pause;
    bool indexPause;
    QWidget* next;
    QMediaPlayer cursor;
    QMediaPlayer enter;

public:
    static Level1* instance(QWidget *parent,QWidget *next);
    void drawBanner(QPainter *);
    void drawToBuffer();
    void reset();

protected:
    void showEvent(QShowEvent *event);
    virtual void keyPressEvent(QKeyEvent *e);
    virtual void keyReleaseEvent(QKeyEvent *e);
    virtual void paintEvent(QPaintEvent *);

public slots:
    void advance();
};

```

以 level1 为例

各种物品类：Floor, ActiveBoard 等

这些类根据使用需要而继承自 MoveObject 类或 StaticObject 类，实现不同的功能和动态效果（详见功能清单）。

Parameter.cpp

包含了在项目实现中需要用到的参数。不同文件调用相同参数时均使用相同参数名，便于修改和维护。

三、成员分工

朱锦瑜实现了游戏的基本逻辑、基本道具、基本地形、基本人物、主界面、关卡选择、关卡切换、背景音乐。

他是 Floor.h, floor.cpp, game.h, game.cpp, gameobject.h, gameobject.cpp, mainwindow.h, mainwindow.cpp, moveobject.h, moveobject.cpp, player.h, player.cpp, staticObject.h, staticObject.cpp 的主要实现者。

张立恒实现了游戏的道具、地形、人物、部分关卡、特效、说明文档、视频制作。

他是 activeBoard.h, activeBoard.cpp, arrow.h, arrow.cpp, arrowshelf.h, arrowshelf.cpp, blood.h, blood.cpp, bomb.h, bomb.cpp, bullet.h, bullet.cpp, dragon.h, dragon.cpp, floor2.h, floor2.cpp, key.h, key.cpp, level1.h, level1.cpp, level3.h, level3.cpp, level4.h, level4.cpp, level5.h, level5.cpp, level8.h, level8.cpp, signboard.h, signboard.cpp, wall.h, wall.cpp 的主要实现者。

张新海实现了游戏的道具、地形、人物、部分关卡、图片素材、资源和参数文件配置。

他是 bulls.h, bulls.cpp, dog.h, dog.cpp, floor1.h, floor1.cpp, ghost.h, ghost.cpp, level2.h, level2.cpp, level6.h, level6.cpp, level7.h, level7.cpp, oldLady.h, oldLady.cpp, trap.h, trap.cpp, parameter.cpp 的主要实现者。

四、总结与反思

经过本次项目设计，我们更加意识到合理的团队分工和团队合作的重要性。在协作开发中，交流是重要的一部分，正因为开发过程中的不断沟通，我们的项目开发过程相对顺利和高效。

这款游戏从设计到开发都经过了细致的学习、思考和探索，培养了我们的克服困难的意识品质，也希望玩家能在游戏工程中获得愉悦的体验。

针对本次项目开发过程中出现的波折，我们认识到在提交自己负责的文件供团队成员调用之前，需要确保函数的正确合理。