

# 恐龙末日小游戏 作业报告

刘老师班 2 队 组号：89 组员：宋一昀

## 一、程序功能

### 1、概述

这是一款操控角色躲避障碍，以存活一定时间的动作类游戏。玩家可以自主选择角色与关卡，或者挑战无尽模式。（下图为第二关界面）



### 2、具体功能

（1）主界面（第一关为例）



- 游戏说明：点击后会出现游戏说明小窗口。
- 两只恐龙的按钮：点击后可以更换玩家角色。点击后按钮贴图会发生奇妙的变化（变为手画的、游戏过程中恐龙被击中时的闪烁特效贴图），主页的恐龙会相应更新。
- 开始游戏：开始游戏。
- 关卡选择：唤起选关页面。

（2）游戏说明窗口：有文字和图片介绍。可以滚动鼠标滚轮上下查看文本

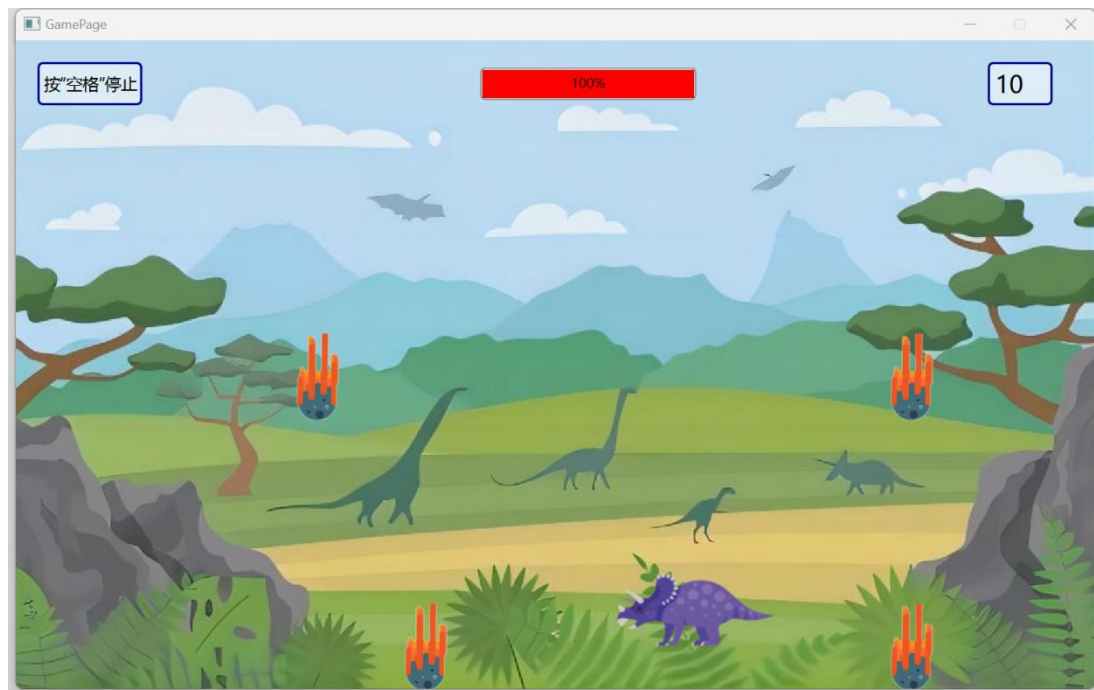


（3）关卡选择界面

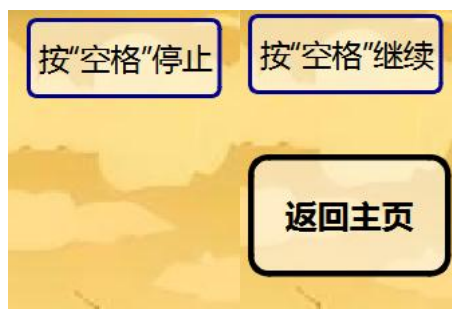


- 可以选择关卡。点击关卡按钮后回到主页面，同时根据关卡改变背景贴图。共有四个关卡可以选择。

#### (4) 开始游戏界面（无尽模式为例）



• 左上角提示：按空格暂停/开始。按下空格后文本框内容会改变、会出现返回主页选项。再次按下空格游戏开始、返回主页按钮消失。



- 右上角时间：一二三关是 90 秒倒计时，无尽模式是从 0 正计时
- 陨石：天空随机掉落。掉落的间隔、速度、数量等逻辑根据关卡和时间都会变化
- 恐龙：玩家操控移动
- 血条：被击中后扣血，血条会有相应的变化
- 动画：恐龙被击中后，血槽减少。恐龙会有闪烁提示。击中的陨石会有爆炸特效并且消失。  
恐龙吃到血包，血槽回复，血包处会有红心动画。



(5) 结算页面（第三关为例）



- 如果血量到 0，恐龙会变成黑色（烤焦了），同时玩家不再能操控恐龙
- 显示红色的你输了或绿色的你赢了字样
- 按钮为开始游戏和关卡选择

## 二、各模块与类设计

1、Ds 类：恐龙。

包含了恐龙的{

碰撞箱信息（由 8 个点组成的多边形）；

贴图四张。分别为一张正常、两张掉血贴图、一张纯黑色贴图；

尺寸信息；

运动速度和扣血量；

}

2、Tm 类：存储了 7 个用于各种功能的时钟。例如陨石速度、恐龙移动、受伤动画播放等。

3、Lv 类：抽象类，派生出了四个子类 Lv1-Lv4。用虚函数实现了陨石召唤逻辑的差异。

包含召唤陨石、计算陨石间隔、召唤血包等功能

包括：{

```
virtual void meteorite_summon_react_lv() = 0
```

```
virtual void fall_react_lv() = 0
```

```
virtual int summon_interval() = 0
```

}

这些功能将会在选择关卡后实现内容的重载。

3、Cl 类：窗口 QDialog 类。包含选择关卡的按钮和对应关卡的图片。通过改变 Lv 的指针类型来实现关卡的变化。

4、Dialog:窗口 QDialog 类。只包含一个文本窗口和一个退出按钮。

5、GamePage:主界面。包含多个函数，以及其他子类的成员对象。具体包含的函数有：

{

private slots:（这部分是控件的槽函数 slots）

void longPress_left();	左移恐龙的计时器开启
void longPress_right();	右移恐龙的计时器开启
void on_startGamebt_clicked();	游戏开始
void on_pushButton_clicked();	选择红色恐龙
void on_pushButton_2_clicked();	选择紫色恐龙
void on_startGamebt_2_clicked();	唤出游戏说明页面
void on_startGamebt_3_clicked();	唤出关卡选择页面
void on_pushButton_3_clicked();	退出关卡/重新开始

protected:（这部分是游戏内逻辑和 signal）

void updateTimerStatus_r(bool startTimer);	更新左移恐龙的时钟
void updateTimerStatus_l(bool startTimer);	更新右移恐龙的时钟
void keyReleaseEvent(QKeyEvent *event);	松开键盘的信号集合
void paintEvent(QPaintEvent *event);	绘图
void keyPressEvent(QKeyEvent *event);	按下键盘的信号集合
void maintimer_react();	游戏开始时每秒的变化
void meteorite_summon_react();	召唤陨石
void stopcounting();	停止所有计时器
void fall_react();	陨石掉落和碰撞检测
void newstart();	重新开始的函数。

由于不仅按钮会重新开始，游戏失败和主动退出都会重新开始，所以在槽函数外这里还有一个重新开始函数

bool checkcollide(int,int);	检测陨石和恐龙碰撞
bool checkcollide2(int,int);	检测血包和恐龙碰撞
void switchdino_react();	恐龙掉血闪烁动画判断和播放
void gameover();	游戏结束，根据输赢显示不同内容
void restart();	重新开始一个关卡时的绘图
void updatelvlnum(int lvlnum);	重新选关时，level 指针的改变和相应的绘图

}

6、所有的按钮和控件都涉及了风格统一的样式表。这部分在 ui 界面中。下面是一些例子



7、几个设计细节的展示：

（1）恐龙的碰撞箱设计，用原本的 QT 碰撞箱类实现精度不够。所以设计了多边形的方案。恐龙的形态特征使得它们分别在不同关卡中更占优势。红色适合第一关、紫色适合第二关。

（2）陨石爆炸动画、恐龙受伤掉血特效调用多个互相关联的 QTimer，对恐龙类贴图的调用也使用多态，比较复杂

（3）关卡的设计。每一关内部，随着时间流逝，陨石的数量和速度和频率都在增加。

第一关：间隔一段时间，中速掉下一行陨石。时钟倒计时。

第二关：高频率地随机位置慢速掉下一颗陨石。时钟倒计时。

第三关：陨石出现但是短暂悬停，随后急速落下。时钟倒计时。

第四关：每隔一分钟，在第一、二关的模式内交替，但是难度上升地更慢。时钟正计时。

（4）美工设计。尽量形成一种游戏界面的卡通风格。寻找、ai 生成、手绘后，裁剪、适配游戏、计算碰撞箱花了很大精力，最后处理出 17 张游戏贴图。

### 三、分工

宋一昞完成所有。另外两个组员都中期退课了。

### 四、改进

一个人精力有限。其实游戏还预留了可扩展性。

掉落物的多态函数还可以继续写，除了陨石和血包，可以有更多道具，增加可玩性。还能有可操控的道具，例如护盾、缩小恐龙、加速恐龙等，接口都很方便。关卡平衡性测试上，花了不少时间，但是还是不够尽如人意。

游戏应该有金钱、最佳纪录、升级的功能。升级的扩展性已经预留了。但是没有时间继续写。

美工上，一些页面的美工做的不够。比如关卡选择界面。

关卡应该按顺序解锁、有分数计算的制度，但是还没能实现。