

QT 大作业——信科人生模拟器

毛子彦 2300013190

齐致权 2300012989

于海祥 2300012987

一. 程序主体功能介绍

概述：信科人生模拟器是一款旨在帮助玩家体验北京大学信息科学技术学院的学习生活的一款文字类互动游戏。在游戏中，玩家将扮演一名信息科学技术学院的一名大一下新生，在游戏选定的 18 天中体验入学--选课---期中--期末的一学期的学习生活。在特定的时间，玩家扮演的角色可以选择不同的选项来做出自己当前想要做的事，进而影响游戏的走向，以此来度过愉快且充实的学期生活。

1. 游戏主界面的交互功能

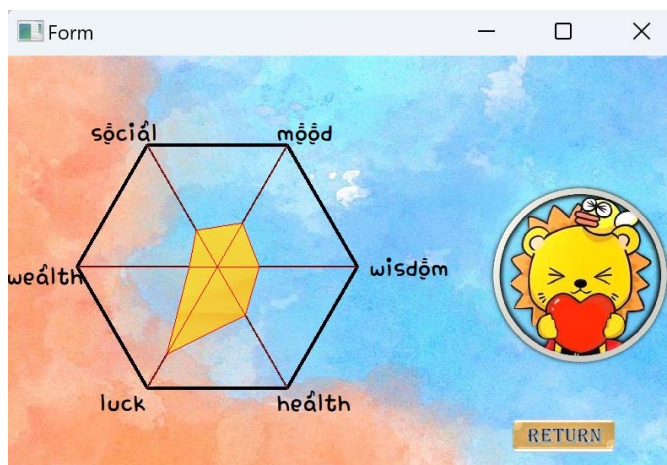
游戏主界面为玩家提供了完成游戏设置，存档，退出，查看课表，查看人物面板等多个交互按钮。此外，玩家主要游戏剧情的推进通过鼠标点击剧情文本框实现。

在游戏剧情遇到分支时，玩家通过点击选项界面的按钮来实现与游戏的交互，根据玩家选择的不同选项，程序会配备与选项场景相配的图片来增强游戏的真实感。



2. 人物面板功能

游戏附带了人物的智慧、健康、财富、运气、心情、社交六项属性来模拟学生正常在学期生活中的变化。人物的数值将根据玩家做出的选择而变化，并在学期结束时根据面板给予评价。玩家可在游戏过程中点击“figure”按钮查看角色的六维雷达图



3. 游戏字体音乐设置

游戏采用独特的字体，并选取“夏休みの過ごし方”作为游戏的背景音乐，玩家可以在“config”界面调节字体的大小、文本框的透明度、以及音乐音量的大小。



4. 存档功能

游戏允许玩家在游戏中的任意时点进入“save”界面存档保留当前的游戏进度到本地，游戏允许至多存取 10 个档，在“load”界面可以读档继续之前未完成的游戏



二 . 程序艺术设计介绍

1 游戏剧情设计

剧情设计方面，为更好模拟信科一学期的学习生活，我们将游戏进程主要分为三段：学期初的准备、上半学期的学习、与下半学期的学习

学期初的准备阶段：

此阶段的剧情主要围绕刚开学时对新学期生活的向往展开，表达主角希望在新学期做出一些改变。在此阶段，玩家将会选择新学期的目标（包括得个高绩点、有个好身体、交很多朋友、开心最重要）作为游戏结束时评价的重要标准。此外，玩家将参与选课，并体验北大刺激的选课投点。为保证玩家的游戏体验，选课概率的计算公式相对偏向于让玩家可以选到自己喜欢的老师（具体公式见游戏机制设计）。

上半学期的学习：

上半学期与下半学期的剧情主要以玩家选择自己想做的事情展开。上半学期事件设计上与体育锻炼和学习的事件量偏多，以便于玩家应对游戏中期到来的期

中考试与体测。为平衡其他事件对角色成长的增益，对于娱乐事件与其他事件适当提高了增益幅度（具体见机制设计）。

另外，上半学期加入了写 Lab1 的特殊剧情，作为学期中相对重要的事件，Lab1 给予了玩家相对丰富的世界线，玩家可以选择苦学多日，自己独立完成 Lab，也可以求助他人帮忙完成，或者一拖再拖，熬夜完成任务。在期中考试和体测时，没有做过多的设计，以模拟考试周学生繁忙的生活，没有更多精力做课外的活动下半学期的学习：

下半学期的事件设计上更偏向于娱乐事件与社交事件，并加入了魔兽大作业与 Lab2 的特殊游戏剧情。Lab2 与 Lab1 类似不再赘述，魔兽大作业的设计上是一个小游戏，玩家通过点击按钮进行 Debug，每次点击会随机减少 bug 数量并列举现实中魔兽的常见问题增强真实感（如旗帜更换问题，剑攻击力为 0 问题等）。魔兽的设计旨在突出学期中大作业的不易，让玩家更能将自己代入游戏中。经历完三个阶段后，会进入总结阶段的剧情中。此阶段会根据考试成绩计算玩家的绩点，并根据人物的各项面板给予三个梯度的评价。根据学期初目标的选择，游戏会评估玩家对目标的完成程度。最后，献上我们组成员对大家学习生活的祝愿。



2.游戏界面设计

(1) 实景拍摄

为还原北京大学的真实环境，我们小组拍摄了未名湖、五四操场、寝室等三四十张实景照片，并随剧情的推进切换到符合场景的照片，尽可能的还原北大的真实情况

(2) 文本框与按钮的美化

对于作业中的全部按钮与文本框，我们在 word 中自制了其图案并截图作为素材引入了游戏中，极大地美化了游戏的整体 UI 界面，让文本框与字体相契合。

(3) 虚拟形象引入

我们小组在对游戏界面的设计中刻意引入了攻城狮的形象，既展现信科的独特文化，也为我们的游戏增加了青春的感觉。让玩家自己代入攻城狮的形象，对游戏产生更浓厚的兴趣。



(所用的部分图片素材)

3.游戏机制设计

(1) 事件池

为在保持游戏随机性的同时控制每种类型的事件出现的比例，我们小组将事件按不同池子分类，并在剧情设计时规定了每天事件所属的池子，这样就可以很好的控制玩家成长的速度与稳定性。池子分类具体如下：娱乐池子，出游池子，社交池子，学习池子，睡觉池子，打工池子，锻炼池子。

```
string sportpool[3]={"去五四操场跑两圈","和同学去邱德拔打一打羽毛球","去一体打乒乓球吧"};
string socialpool[5]={"和同学出去看电影吧","和室友一起出去吃个饭","去电玩城玩一玩吧","去社团参加活动","和同学去酒吧喝点酒"};
string gamepool[5]={"原神,启动!","农","桌(来一把三国杀)","雀魂","去无名湖边走一走"};
string trippool[3]={"去爬香山","北京室内还没转过,去北海公园逛一逛","看看隔壁学校的实力"};
string workpool[3]={"去做学生助理吧","去做一份家教","去当心理学被试"};
```

(2) 面板属性

十四天的游戏进程与较多的不同类型事件的选择使得想要均衡的提高各面板属性变得非常困难。因此，我们小组先将“luck”这一与任何时间没有直接联系的属性做完全随机处理，在评价时取平均值衡量。这样可以减轻一些提高其他面板属性的压力。之后，我们适量调高了每个事件对角色面板的增幅，具体如下：

Luck: random(1,100),每天开始时

Wisdom:上课/自习+random(3,5)

Health:睡觉/锻炼+random(3,7);

Wealth:社交-random(50,100),打工+100/200

Social:社交+random(5,11)

Mood:娱乐+random(3,7);出游+random(5,20)

(3) 成绩计算

按照游戏进行到中期角色 wisdom 成长至 60 左右，末期成长至 80 左右的估计，通过 wisdom 和 luck 对考试成绩进行计算，具体公式如下：

期 中 : $50 + 40 * \min(1, (\text{wisdom} - 30) / 30) + 20 * (\text{random}(0, 40) + \text{luck} - 70)^2 / 2500$

期 末 : $50 + 40 * \min(1, (\text{wisdom} - 50) / 30) + 20 * (\text{random}(0, 40) + \text{luck} - 70)^2 / 2500$

(4) 选课投点

为增强游戏体验，让玩家可以尽可能选到自己喜欢的老师、在生活中确

实教自己的老师，又由于总点数只有 100 点，无法兼顾所有课程，我们调高了选中课程的概率。具体如下：

$$P = (50 + \min(\text{点数} \times 1.5, 50) + \text{random}(-15, 15)) / 100$$

三 . 程序功能实现

1. 游戏主体交互的实现

(1) 玩家与文本框的交互

首先，通过在文本框上覆盖一个与文本框大小相同的按钮，并将其变为透明的，用于检测玩家鼠标的点击。之后，在作业项目的文件夹里新建 txt 加入常规的剧情内容，以行为单位分开加入编号，定义变量 time2 为记录剧情读入到对应的变量，定义 setup() 为更新文本框内文字的函数。当玩家点击按钮后，调用 gonext() 函数令 time2++，并立即执行 setup()，这样便可实现玩家与文本框的交互。此外，日期功能可直接通过读取 time2 实现

```
void Form::setup(){
    ui->a->setStyleSheet("QPushButton { background-color: transparent; border: none; }");
    ui->a->setFixedSize(1442,222);
    ui->a->setGeometry(ui->txb->geometry());
    ui->a->show();
    connect(ui->a,&QPushButton::clicked,this,&Form::gonext);
    ifstream file("../QT\\savedata\\gamedata.txt");
    int fnum;string fstr;
    file>>fnum;
    getline(file,fstr);
    while(fnum!=time2){
        file>>fnum;
        getline(file,fstr);
    }
    file.close();
}
```

(此代码负责设置透明按钮与读入 time2)

```
0 2月18日，又是美好的一天，今天的运势是——
1 新学期的第一天
2 我成了大一下的鼠鼠了
3 唉，半年好快啊
4 好多事情还没做
5 这学期还是得给自己顶一个目标
6 这学期我一定要
7 新学期的目标是什么呢？
8 嗯
9 好像这学期该选课了
10 看看选课网
11 看看选什么课
12 COURSE1 (转到选课界面)
13 好，就选这些吧
14 接下来干什么呢
```

(此图为剧情文件的部分内容)

(2) 特殊剧情的实现与选项界面的处理

对于游戏中的特殊剧情，我们采取了将对应的编号做分类讨论，特殊处理，在 setup() 函数中检查 time2 的值是否与特殊剧情编号相同，并给予实现（如：程设考试成绩的剧情）。

对于选项界面，首先仿照对特殊事件的处理，用一个函数判断是否该编号剧情有选项。由于 UI 设计上有四个选项，而游戏剧情中，往往有 2,3,4 不同个数的选项。为统一管理，我们另新建一个 txt 文本，加入有选项的剧情对应的编号与选项个数并在其后加入选项所属池子的标签（如：第 28 行剧情对应 3 个选项分别属于编号 1,3,4 的池子，即在文本框中写入 28 3 1 3 4 一行）。到达选项剧情时，隐藏文本框前的透明按钮，显示需要的选项。对于每个选项，定义 chon() 函数（选项按钮点击的槽

函数) 用于做点击选项后的处理。当玩家选择选项后按钮会根据其所属的池子来做相应的人物面板变化与返回文本框上新显示内容。这些在 transans(int 编号)中实现

```
void Form::ch1on(){
    if(ch(time2)){
        ui->txb->setText(QString::fromStdString(transans(tmp1)));
    }
    else if(time2==7){
        target="“得个高点”";
        ui->txb->setText(QString::fromStdString("上学期学得太烂了，这学期得努力了。"));
    }
    ui->a->raise();
    ui->a->show();
    ui->ch1->hide();ui->ch2->hide();ui->ch3->hide();ui->ch4->hide();
    ui->ch1->hide();ui->ch2->hide();ui->ch3->hide();ui->ch4->hide();
}
```

(在 chon 中调用 transans 函数来获取新文本，并显示透明按钮，隐藏选项)

```
-1 1是睡觉2是社交3是娱乐4是体育5是出游6是上课7是自习8是打工11\12\13是lab
7 2 8 8
15 4 10 2 3 4
17 2 1 6
18 2 6 2
19 3 7 3 1
21 3 1 2 4
22 2 6 2
23 2 3 7
25 3 6 1 3
26 2 2 9
27 3 1 4 3
```

(此图为记录选项信息的截图)

(3) 不同界面跳转的实现

每个界面都创建一个类。对于每个界面都定义游戏主体内容所需的变量，在跳转时创建新的窗口界面的对象，并用旧窗口的变量将其初始化，然后关闭旧窗口，这样就实现了不同窗口的跳转。

2. 存档功能的实现

(1) 存档

在项目文件夹中新建 10 个 txt 文本。在跳转到存档界面时，复制游戏当前状态的全部变量信息。在存档界面的 confirm 按钮的点击后，调用槽函数将文件写入对应存档 txt 文件中

(2) 读档

打开存档读档界面时，读取所有存档文件获取存档名并显示。记录读档界面被点击的最后一个按钮对应存档的编号，在玩家点击加载后读取其中的变量，创建新的游戏窗口，调用 setup 函数即可。

3. 按照剧情切换图片的实现

在 setup()与 transans()函数中将图片与对应的池子编号或 time2 编号对应即可知道该放哪张图片。用 QPixmap 库加载事先放在项目文件夹中的图片，并在 QWeight 主界面绘制即可。

```
string Form::transans(int x){
    QPixmap chuang("../QT\\savedata\\picture\\床.jpg");
    QPixmap dianwancheng("../QT\\savedata\\picture\\电玩城.jpg");
    QPixmap dianying("../QT\\savedata\\picture\\电影.png");
    QPixmap jiaolou("../QT\\savedata\\picture\\教楼.jpg");
    QPixmap jiaoshi("../QT\\savedata\\picture\\教室.jpg");
    QPixmap jiaoshi2("../QT\\savedata\\picture\\教室2.jpg");
    QPixmap jiaoshi3("../QT\\savedata\\picture\\jiaoshi3.jpg");
    QPixmap lu("../QT\\savedata\\picture\\路.jpg");
    QPixmap hu("../QT\\savedata\\picture\\湖.jpg");
    QPixmap fang("../QT\\savedata\\picture\\房.jpg");
    QPixmap men("../QT\\savedata\\picture\\门.jpg");
```

(用 QPixmap 导入图片)

4. 音乐与文本设置的实现

用 QMediaPlayer 库创建播放音乐的对象，并定义全局变量用于记录音乐音量。用 QFont 库设置文本字体，并定义全局变量用于记录字体大小。在游戏开始时，初始化字体与音乐，并当玩家在 config 界面调节文字大小与音量时，改变全局变量的值，并更新当前正在显示的文本，以此实现音乐与文本的设置。

```
player=new QMediaPlayer();
QMediaPlaylist *playlist=new QMediaPlaylist();
QUrl source(QUrl::fromLocalFile("../QT\\savedata\\bgm.wav"));
playlist->addMedia(source);
playlist->setPlaybackMode(QMediaPlaylist::Loop);
player->setPlaylist(playlist);
player->play();
```

(播放音乐)

5. 美化文本框与按钮的实现

在文本框上粘贴实现准备好的图片，并在其上方创建与其大小相同的透明 Label。将文本内容在 Label 上绘制，二者叠加在玩家视角中就是美化后的文本框与图案。

```
QString fontFilePath = "../QT\\savedata\\anna.ttf";
int fontId = QFontDatabase::addApplicationFont(fontFilePath);
QStringList fontFamilies = QFontDatabase::applicationFontFamilies(fontId);
QString familyName = fontFamilies.first();
QFont font;
QIcon icon("../QT\\savedata\\picture\\start.png");
ui->start->setIcon(icon);
ui->start->setIconSize(QSize(300, 90));
QIcon icon1("../QT\\savedata\\picture\\load.png");
ui->load->setIcon(icon1);
ui->load->setIconSize(QSize(300, 90));
QIcon icon2("../QT\\savedata\\picture\\config.png");
ui->config->setIcon(icon2);
ui->config->setIconSize(QSize(300, 90));
QIcon icon3("../QT\\savedata\\picture\\exit.png");
ui->exit->setIcon(icon3);
ui->exit->setIconSize(QSize(300, 90));
font.setFamily(familyName);
font.setPointSize(24);
QFont font1;
font1.setFamily(familyName);
font1.setPointSize(20);
ui->config->setFont(font);ui->start->setFont(font);ui->load->setFont(font);ui->exit->setFont(font);ui->label->setFont(font);
```

(设置字体与美化按钮)

四 . 小组成员分工

具体分工如下：

| | |
|------------------------|---------|
| 游戏主体剧情推进与交互功能和存读档功能的实现 | 毛子彦 |
| 游戏剧情与界面设计与美化 | 于海祥 |
| 游戏内机制设计 | 于海祥、毛子彦 |
| 选课界面实现 | 齐致权 |

| | |
|--------|-----|
| 作业报告撰写 | 毛子彦 |
| 人物面板界面 | 齐致权 |
| 实景拍摄 | 于海祥 |

五. 项目总结与反思

1. 大量图片资源占用内存过大——仍需解决的问题

本项目使用了大量的图片资源，使得项目所占用的内存较大。针对该问题，我们小组提出了以下两种可能的解决方案：

- (1) 使用图像压缩技术：将所有图片进行拼接然后压缩，这样可以大大减小文件所占用的内存空间。但由于本游戏对界面的美观程度有较高的要求，而压缩后的图片可能存在分辨率过低等问题，造成对游戏整体体验的破坏，因此该方案仍需改进。
- (2) 建立服务器：将所有的图片与文本信息放在云端保存，并在运行时读取，这样既节约了内存，也便于后期对游戏进行更新与统一管理，但由于搭建服务器与数据库的难度过大，因此在小组完成作业时没有采用该方案，可以在之后完善该部分的内容。

2. 界面跳转时的冗余处理——应当吸取的教训

在开始着手处理界面跳转时，首先没有考虑应用全局变量来存储游戏的状态信息，致使在游戏主体功能实现后，跳转界面需要进行十分繁琐的初始化步骤来传递变量。也对之后添加图片背景与美化 UI 的任务增加了阻碍。让游戏改动一点就需要修改所有对应变量的代码，不方便后续对游戏的优化。

其次，每种类型的界面在游戏中只需要一种即可，完全可以通过 `hide()` 和 `show()` 实现跳转功能。但由于未采取全局变量的设置，在项目完成的初期采用了更冗余效率更低的创建新界面的方式，占用了更多的运行内存。

在之后进行类似项目的编写时，应在着手完成之前考虑出最优的解决方案后再予以实施，以防止在后续的工作时想到更优解，但为时已晚。

3. 未加入成就系统——游戏设计的遗憾

在游戏进程结束时的结算针对玩家的各项数值进行了评估，并给予了相应的称号。在项目完成后，我们小组发现完全可以加入一个成就系统，用于记录玩家已解锁那些称号，还有哪些尚未解锁。这样可以更好的激发玩家游玩该游戏的热情，并尝试更极端、更冒险的游玩方式，为游戏增加了更多可能。我们小组考虑会在之后的时间中完成该功能的添加