PKU 怪谈之我要上程设

by Deep Golf

1 程序功能介绍

1.1 概述

本项目是一个剧情向游戏，并在剧情基础上嵌入了两个小游戏：2048 和跑酷，以丰富用户的游戏体验。

在游戏中，玩家需要扮演一名选修了程设的攻城狮，根据校园怪谈规则的指引，闯过重重关卡，实现与程设的双向奔赴。

1.2 机制设置

本游戏共设计有 7 个基础场景，3 张规则纸条，3 个特殊关卡和 1 个结局。

纸条是游戏规则的载体，也是玩家做出选择的依据。玩家需要通过选择正确的互动分支才能触发纸条，知晓规则。

在每个基础场景，玩家都会通过对话，根据选择进入不同的分支。

如果玩家选择正确的分支，将有机会获得更全面的怪谈规则。

如果玩家选择错误的分支，则将进入特殊关卡。

特殊关卡 1、2 是 2048 游戏，玩家需要通关该游戏来进入下一场景

特殊关卡 2 是跑酷游戏，当玩家选择错误分支，将可能触发该关卡。玩家需要完成跑酷游戏以进入特殊场景。

1.3 剧情简介

周三清晨，你悠悠转醒，准备奔赴早十的程设课堂。

然而天色似乎出奇地暗，往常熟睡的室友也不知所踪，你感到一丝不对劲。

忽然，你发现了一张残缺不全的昏黄纸片，上面罗列着从未听说过的校园规则。“学校里不存在任何异常现象，也没有伪装在正常人外表下的神秘怪物”，画蛇添足的言语在你心中埋下了疑惑的种子，而刺猬姐的校园传说、“暂时不能给你明确的答复”的戏谑言语出现在一份书面的学生守则中，更

是让你一头雾水。

然而你太热爱程设课堂了，没有什么能够阻挡你和程设的双向奔赴。那么，根据规则的指引前进吧！解开 2048 的门锁，完成跑酷的挑战，收集足够的信息，在不断靠近程设课堂的同时，你也将解开校园内奇怪氛围的秘密，发现梦境与现实的真相。

1.4 灵感来源

创作这个项目的动机，是想做一个更具有趣味性、更具有北大特色、凝聚更多巧思的游戏，而不仅仅是对坦克大战、贪吃蛇等经典游戏的简化版复刻。

往年优秀项目展示环节中，一个描述校园生活的剧情类游戏吸引了我们的注意力。“讲好故事”正是剧情类游戏的第一要义，也与我们小组对游戏趣味性、北大特色性的追求不谋而合。往年项目取材于校园生活的灵感得到我们的借鉴，而如何弥补其剧情相对单薄、北大特色相对匮乏、游戏内容相对单一的缺点，也成为我们思考的重心。

本项目的构思恰逢对强基计划的讨论如火如荼之时，“暂时不能给你明确的答复”成为独属于北大同学的新梗。于是，一个自然的体现北大特色的想法便油然而生：将“暂时不能给你明确的答复”和刺猬姐的校园传说等北大圈内文化融入到剧情中去，让北大的同学们在看到该游戏时能够默契地会心一笑。

在这些北大亚文化因素载体的选择上，我们选择了近几年比较流行的校园怪谈。这些圈内文化本身就具有一种戏谑性，定位也是茶余饭后聊天谈资，与“怪谈”以严肃形式表达戏剧内容的特点较为匹配。而在具体圈内文化的选择上，我们一方面与时俱进，选择了“原神怎么你了”“疯人院学生”等日常调侃中常见的元素，力求增加趣味性；另一方面加入小白房、博实超市等元素，试图起到向同学们进行北大校园变迁史科普的作用。至于一些三观不是那么正的亚文化，则被提前排除在外。同时，我们也在树洞平台上进行了搜索，借鉴了 3087374 等树洞，以求元素本身更具北大的代表性，更能引发同学们的共鸣。

当然，如果只有场景的切换和剧情的推进，这个项目就不像一款游戏，

而更像一篇带插图的文章。为了增加游戏的可玩性与趣味性，我们设置了玩家在对话时的选项，以增加游戏的交互性。同时，我们还制作了 2048 和跑酷两款小游戏，分别嵌入到了两个特殊关卡中，以丰富玩家的游戏体验、增加游戏的多元性。

最后则是主题的选取。网上常见的怪谈主题一般指向克苏鲁恐怖神话或对现实的反讽。作为一款以北大为背景的怪谈游戏，以阴暗的克系神话为背景显然不妥，可如果失去这层神秘而紧张的气氛，游戏的悬念性与趣味性则将大打折扣。所以，我们最后将游戏背景设定为，怪谈中的一切都是玩家因过于思念程设课堂，日有所思夜有所梦而创造出的梦境。这样既能体现信科学子一心向学的风气，展示程设作为信科知名好课的吸引力，又能够为怪谈规则的生效、校园环境的改变做合理的解释。

2 项目各模块与类设计细节

2.1 2048 模块

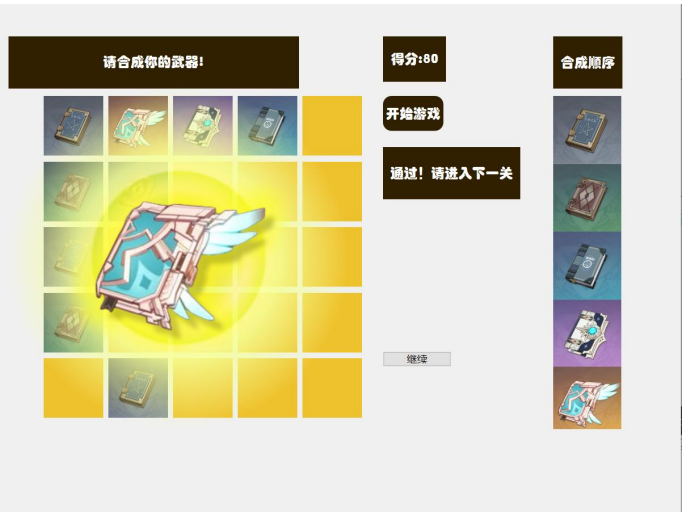
在 2048 模块中,玩家需要通过一个类似于 2048 的游戏合成属于自己的武器。我们为游戏设置了两种武器：剑和魔法书。简便起见，一共只有五个层次，对应 2048 原版游戏中的 2、4、6、8、16。

游戏按照传统的 2048 算法实现，使用 label 来表示图片，玩家每次按动一个按钮，都会触发对应的上下左右移动。图片合并的逻辑是先按照方向将所有图片移动到连续的位置，然后从对应方向开始将连续且相等的图片合成。为了简化代码，我们只写了一个方向的移动（up()）、合并代码，其他方向使用翻转、旋转所有图片，调用 up()，然后再翻转为原位实现。同时，在每次移动过后，都需要判断游戏是否成功、失败，为了界面的美观，我们为成功的合成制作了一个动画（请参考提交的视频）。

配合剧情，我们一共设计了两种版本的 2048 游戏，分别为武器和魔法书两张，增强游戏的可玩性，体现了代码的复用性。

【初始界面】

【游戏 01 结束界面】

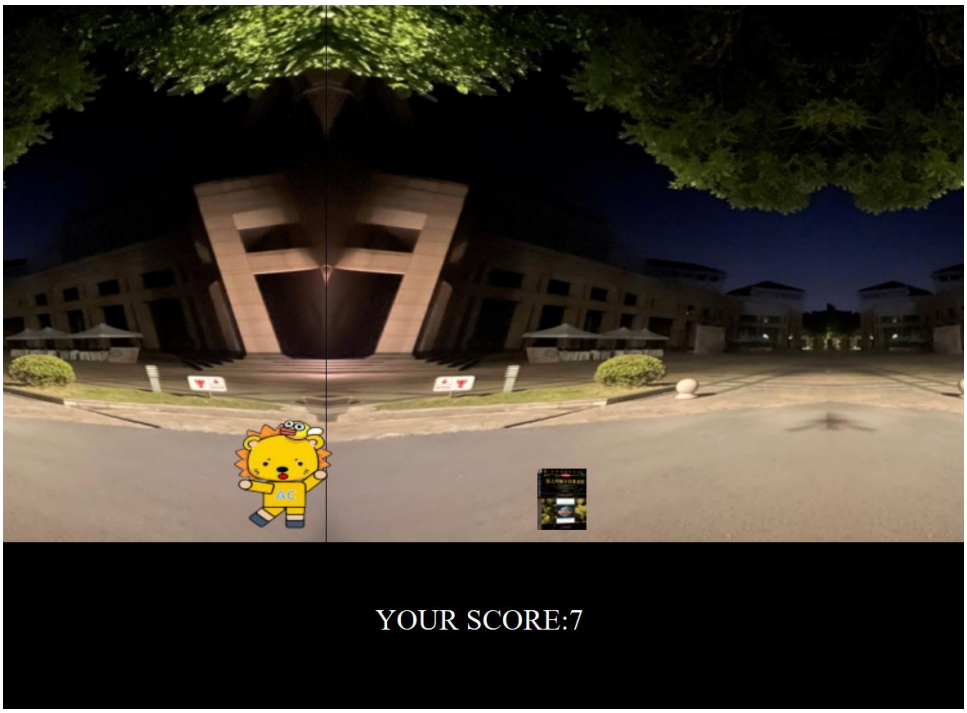


【游戏 02 结束界面】

2.2 跑酷模块

在跑酷模块中，玩家通过上键（up()）来控制工程狮的跳跃，通过背景的相对运动来实现跑步的效果。在背景的设置方面，选择了循环的百讲，表现了无穷无尽的逃亡。

同时对于工程师形象，增加跑步的动画效果增强游戏的可玩性，通过设置定时器进行实现。在障碍物方面，使用《计算机系统导论》作为障碍物，表达信科同学在学习 ICS 路上的种种艰辛，也表达即便艰难也要迎难而上的信心和对于专业的热爱。对于是否和障碍物进行碰撞，设置 QLabel 的几何属性来实现。在图片的正下方，设置有分数，当分数达到 30 分时，该关卡顺利通过。

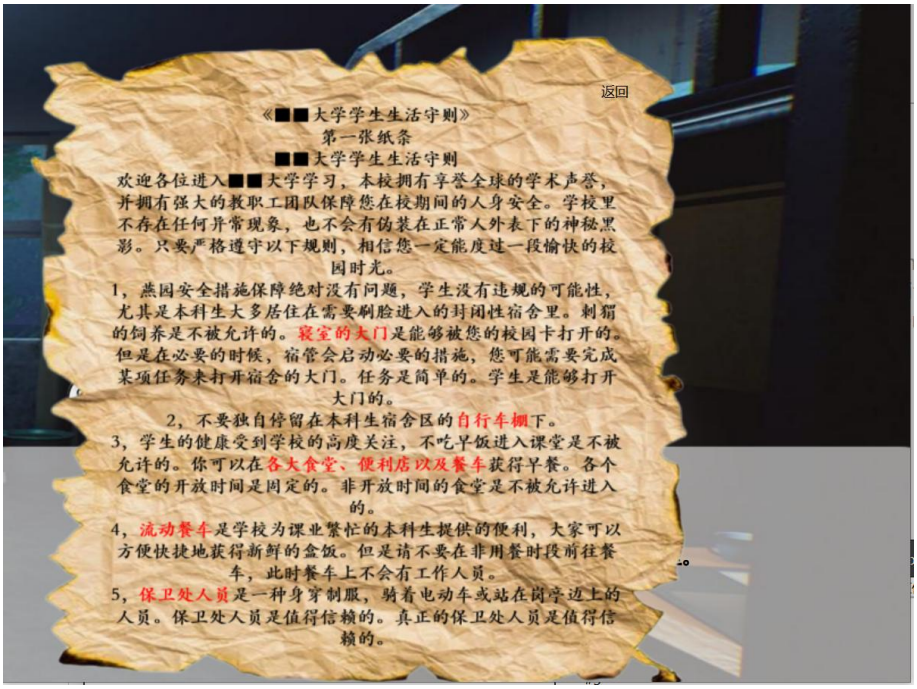


【以百周年纪念讲堂为背景的跑酷游戏】

2.3 纸条模块

为了配合校园怪谈的背景，我们在场景中加入了线索纸条，玩家可以随机点开纸条查看其中的怪谈。在具体实现上，对于每一个场景类 scene1 都有一个代表 paper 的 button 指针，对于有纸条的场景，我们初始化 paper 指针。当玩家点击页面中的任意一点，程序使用 keyPressEvent 捕获并判断是否在 paper 的范围内，如果是，则显示纸条具体内容并配备一个 close

button,同理，玩家点击页面时也会判断是否在 close button 范围之内。



【纸条展开的场景】

2.4 场景切换

在使用堆叠窗口进行实现的时候，我们发现会出现场景之间的覆盖，导致工程师的动画无法显示，在经过多次尝试以后，我们使用 connect 以及槽函数进行实现

3 项目分工

贾越如：跑酷模块、场景切换

梁小雨：纸条模块、2048 模块

叶杨轶：场景切换、场景设计

4 项目总结与反思

在项目中，我们小组使用 qt 编写了一个剧情向关卡游戏，游戏主要包含场景切换、2048 游戏、跑酷游戏三个模块。通过这个项目，我们小组成员不仅学到了很多关于 Qt 和游戏开发的知识，还锻炼了团队合作和解决问题的能力。在完成该过程中，我们遇到了一些问题，最终通过网络资料的搜索和组员之间的讨论解决，这极大锻炼了我们的工程能力和搜索信息的能力。

在项目完成过程中，我们也意识到了自己的不足。比如，我们的分工并不非常合理，导致两个人共同编写一个模块的情况发生，这种情况下给开发和代码质量保证都带来了很大的困难。同时，由于对 qt 项目 debug 时长估计错误，我们的时间安排出现了一些问题，最后一个星期的开发任务量极大。

以下为项目初期、中期、后期所做的主要工作，遇到的困难及反思：

4.1 项目初期

在项目初期，我们把主要的精力投入到了怪谈规则的设计与北大元素的收集，通过①树洞、BBS 等校内论坛的搜索；②对身边同学的问卷访谈；③借鉴网上流行的校园怪谈规则，进行了规则的书写和背景框架的设定，力求实现富有趣味性和北大特色这一目标。

这一阶段存在的主要问题是，由于缺乏 qt 开发经验，低估了编写场景、实现跳转所需的时间，导致刚开始的大纲过于宏大，后面经历多次删改，浪费了时间。（大纲初稿也发布在 github 上了）

4.2 项目中期

在项目中期，我们完成了 2048 小游戏模块与基础的页面设计，并书写了场景切换的 ui 文档。作为一个剧情向游戏，不同选择会触发很多剧情分支，此时界面的正确调用与复用就显得尤为重要。ui 文档将跳转逻辑可视化，减少了界面调用语句书写时的错误。

这一阶段的主要问题出现在复用模块的过程中。比如，我们在游戏中设置了两个相近的 2048 关卡，最开始我们在初始化函数中调用了 grapKeyboard(),所以当执行到第一个 2048 游戏时键盘实际上是被第二个 2048 游戏劫持的。于是导致在第一个 2049 游戏中点击按键没有反映。于是我们将 grabKeyboard()函数写入开始游戏（init()）函数，事实证明这个方法是有效的。

4.3 项目后期

在项目后期，我们完善了跑酷小游戏模块，加入了纸片的实现了部分跳转逻辑的纠正，并进行了 debug 工作。需要承认的是，在这个阶段，我们对 debug 的时间估计出现了一些失误。Qt 虽然是一个 C++框架，但在 C++语法的基础上，关于 ui 文件的构建、多个槽与信号的连接、连接的更新、页面的刷新与时间控

制、页面跳转、先后逻辑等还有自己的规则。由于场景的多次复用和复杂的跳转逻辑，我们在过程中遇到了比较多的 bug，虽然最后通过网络资料的查询一一得以解决，但花费了大量的时间。下面是一些具有代表性的问题和解决方案：

（1）在测试 2048 游戏模块的过程中，出现了键盘上下键无法正常操纵的问题。最后通过资料查询得知，在模块中加入“this-> grabKeyboard();”语句，解决了该问题。

（2）在测试跑酷效果时，发现跑酷人物落在障碍物上时，会出现图片覆盖的问题。最后通过精细确定图片的相对位置得到解决。（**需要补充**）

（3）由于场景切换的逻辑较为复杂，showScene 函数因为参数细节的问题出现了好几次报错，经历了好几次修改，确定为以下版本：

```
void showScene(int sceneIndex)
{
    if(sceneIndex == 444){
        scene4_game->start_game();
    }
    scene1->setVisible(sceneIndex == 1);
    scene1_2->setVisible(sceneIndex == 12);
    scene2->setVisible(sceneIndex == 2);
    scene3->setVisible(sceneIndex == 3);
    scene3_2->setVisible(sceneIndex == 32);
    scene3_game->setVisible(sceneIndex == 333);
    scene4->setVisible(sceneIndex == 4);

    scene4_game->setVisible(sceneIndex == 444);

    scene5->setVisible(sceneIndex == 5);
    scene5_2->setVisible(sceneIndex == 52);
    scene5_3->setVisible(sceneIndex == 53);
    scene5_4->setVisible(sceneIndex == 54);
    scene5_5->setVisible(sceneIndex == 55);

    scene6->setVisible(sceneIndex == 6);
    scene6_2->setVisible(sceneIndex == 62);
    scene6_3->setVisible(sceneIndex == 63);
    scene7->setVisible(sceneIndex == 7);
    scene8->setVisible(sceneIndex == 8);

    scene9->setVisible(sceneIndex == 8);

}
```

```
void MainWindow::setupScenes()
{
    /*可以根据scene实例化出不同的对象出来，然后把所有的场景加入到堆叠窗口中去*/
    showScene(1);
    connect(scene1->dialogWidget, &DialogWidget::dialogFinished, this,  [this]() { switchToScene(12); });
    connect(scene1_2->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(2); });
    connect(scene2->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(3); });
    connect(scene3->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(32); });
    connect(scene3_2->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(333); });
    connect(scene3_game, &game2048::gameFinished, this, [this]() { switchToScene(4); });
    connect(scene4->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(444); });

    connect(scene5->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(52); });
    connect(scene5_2->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(53); });
    connect(scene5_3->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(54); });
    connect(scene5_4->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(55); });
    connect(scene5_5->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(8); });


    connect(scene6->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(62); });
    connect(scene6_2->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(63); });
    connect(scene6_3->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(7); });

    connect(scene4_game, &running::gameFinished, this, [this]() { switchToScene(8); });
    connect(scene7->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(8); });
    connect(scene8->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(9); });
    connect(scene9->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(12); });
```

并加上了“connect(scene1->dialogWidget, &DialogWidget::dialogFinished, this, [this]() { switchToScene(12); });”语句以实现场景之间的跳转。

4.4 回顾与展望

回顾这次项目的实现过程，虽然收获良多，却也确实有一些遗憾之处：

（1）没能完全实现第一版大纲的丰富内容

我们在前期花费了很多精力收集资料、书写规则，却由于时间的限制，在最终的版本中进行了大幅删改，导致大量努力付诸东流。

我们希望能够在未来的版本中加入更多的场景与分支，添加更多如 2048 一样的小游戏关卡，引入 npc 人物和对话机制，增加剧情厚度，提高游戏的交互性与趣味性。

（2）可以引入更丰富的动画效果

为了丰富视觉效果、提升游戏的流畅性，我们引入了一定的动画效果。比如在跑酷过程中，加入攻城狮嘴巴张合的动画；在 2048 的合成过程中，增强，合

我们希望在将来的游戏版本中，引入更多的动画效果，提升游戏在视觉上的吸引力。

（3）分工重叠等问题可以避免

由于开发经验的不足，我们在开发过程中出现了一些配合问题，浪费了一些

时间：如，在分工时，出现了两名同学编写同一模块的问题，既导致了重复劳动，又由于代码风格的不同给维护和调试带来了一些困难；又如，在进行代码工作时，没有预先将调用和复用的关系整理出来，导致编写跳转语句时思路较为混乱，debug 浪费了较长的时间。

在后续的开发过程中，我们希望可以实现更合理的分工，在开始编写代码前将需求和不同模块间的关系明确列出，减少无效劳动，提高开发效率。

4.5 总结

当然，虽然有诸多遗憾，但我们实现了项目开始时的预设目标：设计一款融趣味性和北大特色于一体的剧情向游戏。在代码编写的过程中，我们熟悉了 Qt 这一开发框架，实操了程设课堂中所学的 C++技巧；在小组合作的过程中，我们获得了合作开发项目的经验，认识到了合理分工的重要性。总体而言，还是满意多于遗憾的。也希望助教学长在体验我们的 Pku 怪谈时，能露出专属于 Pkuer 的会心一笑。