南开大学 计算机科学与技术

姓名：陈秋彤

学号：2311815

班级：计算机科学与技术卓越班

2024年5月7日

高级语言程序设计

实验报告

目录

[高级语言程序设计大作业实验报告 2](#_Toc166003479)

[**一.** **作业题目** 2](#_Toc166003480)

[**二.** **开发软件** 2](#_Toc166003481)

[**三.** **课题要求** 2](#_Toc166003482)

[**四.** **主要流程** 2](#_Toc166003483)

[**1． 整体流程** 2](#_Toc166003484)

[**2． 对象功能设计** 3](#_Toc166003485)

[**（1） Monkey类** 3](#_Toc166003486)

[**（2） Map类** 4](#_Toc166003487)

[**（3） Phase & Stage类** 4](#_Toc166003488)

[**（4） Gold & Barrier类** 4](#_Toc166003489)

[**（5） 综合运用** 4](#_Toc166003490)

[**3． 测试运行** 5](#_Toc166003491)

[**五.** **测试效果** 5](#_Toc166003492)

[遇到的困难及解决方法 5](#_Toc166003493)

[**六.** **收获** 6](#_Toc166003494)

[**1． 面向类设计程序** 6](#_Toc166003495)

[**2． 绘制贝塞尔曲线** 7](#_Toc166003496)

[**3． 记录鼠标的点击时间** 7](#_Toc166003497)

高级语言程序设计大作业实验报告

1. **作业题目**

基于“跳一跳”及“Monkey Rope”小游戏，创新开发的跑酷类小游戏。

1. **开发软件**

Visual Studio 2022，Qt Creator 4.11.1（Community）

1. **课题要求**
2. 对象设计。
3. 成员函数及功能实现。
4. 对象之间的互动反应。
5. 测试运行。
6. **主要流程**
   1. **整体流程**

首先分析游戏流程，基本流程如下：

在操作环节，玩家通过控制长按鼠标的时间，控制游戏角色依次弹跳，角色弹跳后，检测是否落在下一平台上，如果成功降落，则继续游戏，若失败，则单轮游戏结束，玩家可选择清屏重新开始游戏。

其中，为提升可玩性，玩家可依次操控两个游戏角色P1、P2，当P1、P2相距过远时，会触发“拉伸”事件，通过“牵拉”函数调整P1、P2的位置，增大游戏的操纵难度；同时，在平台上会随机出现道具Gold及Barrier，角色如果接触到Gold，可进行加分，若接触到“Barrier”，则结束游戏进程。

通过以上分析，可将游戏中的不同区块功能抽象为以下几类：Monkey（玩家所控制的角色）、‘Phase & Stage（游戏中角色站立的平台）、Barrier & Gold（游戏中的道具，提升可玩性）、Map（游戏地图），在实现过程中，需要通过对上述几类对象单独设计各自所需的功能，从而实现游戏内容。

* 1. **对象功能设计**
     1. **Monkey类**

包含游戏角色Player1和Player2（以下用P1、P2简称）。

对每个角色，包含以下基本属性：

* 位置坐标、跳跃速度
* 边框（用于判断，角色起跳后是否降落在平台上）、生命状态（若降落失败，生命状态调整为false，若通过牵拉“复活”，重新回到true）
* 起跳状态、着陆状态（辅助判断角色是否降落在平台上，或被牵拉后是否掉出平台）

还包含以下函数：

* 基础功能：位置更新、位置设定
* 基础动作：跳跃、降落、复活
* 牵拉函数：当P1及P2相距过远时，调整P1、P2的位置；若P1或P2掉下平台，而仍存在存活的游戏角色时，在存活的角色起跳时，牵拉死亡的角色，从而有几率“复活”已死亡的角色。为确保牵拉的视觉效果，绘制贝塞尔曲线
* 滚动函数，主要考虑：如果角色跳跃到画面最右侧边界，需要滚动屏幕，将游戏角色调整至画面左侧，确保角色不会跳出画面
  + 1. **Map类**

实现角色起跳后，自动向后滚动，形成角色跳跃前进的效果

* + 1. **Phase & Stage类**

需要满足，当角色起跳后，随地图的滚动向后滚动，并不断刷新出现新的平台，出现机制使用随机数函数，提升随机性和可玩性

* + 1. **Gold & Barrier类**

配合Phase的出现而随机出现，依然采用随机数函数辅助刷新

* + 1. **综合运用**

Monkey起跳后，更新地图及平台，Monkey降落后停止滚动地图

Monkey与Phase接触后，Monkey停止运动

Monkey与道具类对象接触后，触发相应效果，若接触Barrier，则游戏结束

完善游戏开始界面和清屏重启的功能函数，完善绘制函数，完善将Monkey的起跳速度与鼠标按动时间建立联系

* 1. **测试运行**

针对上述功能，依次进行编写调试，优化代码，实现功能

1. **测试效果**

### 遇到的困难及解决方法

1.牵连函数的实现：

在实现过程中，一开始设计了Monkey类的两个对象P1、P2，此时，想计算并调动拉伸函数代码过于复杂，此处将P1、P2合并在一个对象monkeys中，使距离的计算及判断更为简洁，同时，因为在牵拉效果中，P1、P2的速度是相关联的，这样设计有助于设置P1、P2的牵连速度。

2.Monkey降落的实现：

在Monkey降落时，因平台的高度较高，只判断其边框与平台的边框是否接触，可能会出现，Monkey未达到降落要求，但“闪现”降落的问题，经过进一步对Monkey和Phase的坐标判断，确保降落效果。在降落后，如果不重新设置Monkey的位置，会使Monkey的边框与平台持续相交，产生运行问题，此时要重新更新Monkey的位置，避免问题。

3.Barrier 和 Gold的位置设置：

根据设计，道具类物品应只会出现在平台附近，如果只用随机数函数随机生成道具类的位置坐标，会无法控制这一点，同时，可能会导致Barrier和Gold重合的问题。因此，将这两类对象同时设置在Phase类当中，有利于与Phase建立坐标上的牵连关系，确保一个平台上最多出现一个道具，避免拥挤，也可控制二者不会同时出现在相同位置。

4.跳跃顺序的实现：

需使程序自动切换，依次控制P1、P2的运动。如果只用一个key来控制，会在有角色死亡时出现无法条约的“空键”问题，因此，需要进一步判断，并完善对key值改变的方式和次数。

1. **收获**
   1. **面向类设计程序**

学会了将拥有相同功能的对象抽象成一个类，在设计过程中只需考虑一个类所需的功能，最后再考虑类与类之间的互动关系。

举例：对于Monkey类函数，实现以下Monkey所需的功能：

void jumpSet(int l);

void jumpFollow(int l);

void updatePosition();

void updatePositionN();

void setPosition(int x, int y);

void setPositionN(int x, int y);

void down();

void downN();

void slideBack();

void sildeBackN();

void disElastic();

之后，只需要整体考虑Monkey类与Phase类的互动，不需要再设计两类各自的细节：

void land();

void landN();

* 1. **绘制贝塞尔曲线**

在优化牵连函数的视觉效果时，使用可自由伸缩的光滑曲线，将曲线经过的几个节点与Monkey的坐标建立联系，此时效果是最好的，此处，学习并运用了贝塞尔曲线的绘制：

QList<QPointF> points = QList<QPointF>() <<QPointF(m\_monkey.m\_X+m\_monkey.m\_Monkey.width()\*0.1, m\_monkey.m\_Y-m\_monkey.m\_Monkey.height()\*0.8)<< QPointF(m\_monkey.m\_X\*0.5+m\_monkey.n\_X\*0.5, (m\_monkey.m\_Y+m\_monkey.n\_Y)\*0.5+20) << QPointF(m\_monkey.n\_X+m\_monkey.n\_Monkey.width()\*0.1, m\_monkey.n\_Y-m\_monkey.n\_Monkey.height()\*0.8);

QPainterPath path(points[0]);

for (int i = 0; i < points.size() - 1; ++i) {

QPointF sp = points[i];

QPointF ep = points[i+1];

QPointF c1 = QPointF((sp.x() + ep.x()) / 2, sp.y());

QPointF c2 = QPointF((sp.x() + ep.x()) / 2, ep.y());

path.cubicTo(c1, c2, ep);

}

painter.setRenderHint(QPainter::Antialiasing, true);

painter.setPen(QPen(Qt::green, 5));

painter.translate(40, 130);

painter.drawPath(path);

* 1. **记录鼠标的点击时间**

使用了以下Qt自带函数，将鼠标点击的时间记录下来，作为一个变量，供之后设置Monkey的速度时使用：

time\_len = QTime::currentTime();

time\_lens = time\_len.msecsTo(QTime::currentTime());