

2020游戏程序设计报告

课程名称： 游戏程序设计

任课老师： 李仕

姓名： 倪一超、温载旭、钱天宇、徐江航

学号： 19220317、19220320、19220318 、19205329

专业： 数字媒体技术

## 简介：

**1.游戏基本资料**

·游戏名称：飞鸟

·游戏类型：棋牌类游戏

**2.游戏概要**

玩家通过独立作战或与对家配合，通过掷骰子数，决定自己棋子在棋盘上的行进步数，以全部棋子最先到达终点的游戏者为最后胜利者。可以锻炼玩家的策略意识以及灵活变通。

**3.游戏界面分析**

3.1游戏开始界面

从初始界面切换至游戏开始界面（展示棋盘图案、玩家位置）

3.2游戏内容界面

·界面操作说明：

骰子：按左键进行掷骰，得出掷骰点数。

起飞：选择行动的棋子，棋子按点数行走。在掷得6点后，方可将一枚棋子由“基地”起飞至起飞点，并可以再掷骰子一次，确定棋子的前进步数。

棋盘：棋子按照棋盘上的格子进行运动。

玩家：显示此次掷骰是由哪个玩家进行的。

获胜：当有玩家的所有棋子到达终点时，则该玩家获胜。

结束：当所有棋子到达终点时，则游戏结束。

·界面说明：

|  |  |  |
| --- | --- | --- |
| 界面 | 截图 | 说明界面内容解说 |
| 初始界面 | TH_%{96%6Z8FUX14O)7CG)Q | 初始界面 |
| 棋子飞行界面  （棋盘界面） | RIAGI{9Q(FJY]SN@}3{O7VW | 棋盘MAP |
| 掷骰界面 | 8$VOED0_YDKQEQMLZW2P{DT  8$VOED0_YDKQEQMLZW2P{DT |  |
| 某棋子到达终点界 | 6D]UTW)2KQ4Y()6OL{PN[LD |  |
| 玩家选择棋子界面  （棋子制作完成） | 3R%E8$PHV9E%MIP$E]J_7~0(1XNWMUDYZ2NI32GMMANP{Y  (Z{CS6`[}{6]N4QYTS@B~(QCWS@4483]96@NHHD[_RBU$N | 棋子 |

**4.音乐音效**

暂无

**5.游戏世界各元素定义**

（列表对游戏世界包含的各个对象进行说明）

|  |  |  |
| --- | --- | --- |
| 对象 | 作用 | 属性 |
| 图片资源 | 提供显示界面的图片 | 棋子图、功能按键图、骰子图 |
| 棋子 | 玩家操控的对象 | 通过玩家的指令在棋盘上移动 |
| 骰子 | 控制棋子的移动 | 棋子根据骰子的点数以及玩家的选择进行移动 |
|  |  |  |
|  |  |  |
|  |  |  |

**6.游戏逻辑实现（举例如下）**

**6.1 初始化游戏世界**

（1）加载图片和音频资源

加载棋盘图片、棋子图片、玩家图片、骰子图片。

（2）初始化地图

棋子全部位于初始位置。

（3）初始化鼠标状态

将鼠标状态置为未按下。

**6.2 游戏循环**

**（1）输入**

鼠标左键按下：设置鼠标状态为左键按下。

鼠标左键松开：设置鼠标状态为未按下。

鼠标右键按下：设置鼠标状态为右键按下。

鼠标右键松开：设置鼠标状态为未按下。

**（2）逻辑处理**

1.骰子逻辑

//初始化棋子位置

//初始化棋子信息

Player\_logic()：

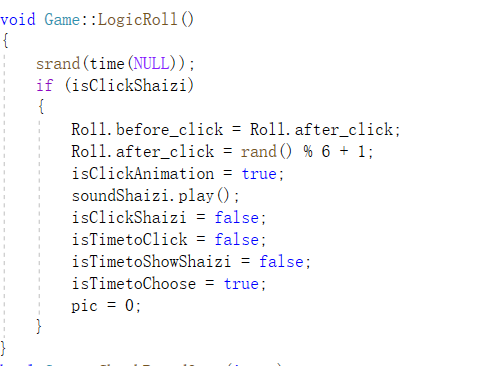
骰子抛到6，则棋子可以飞行。并且玩家可以再次投骰。

骰子未抛到6，则棋子无法飞行，下一位玩家。

AutoPlayer\_logic：

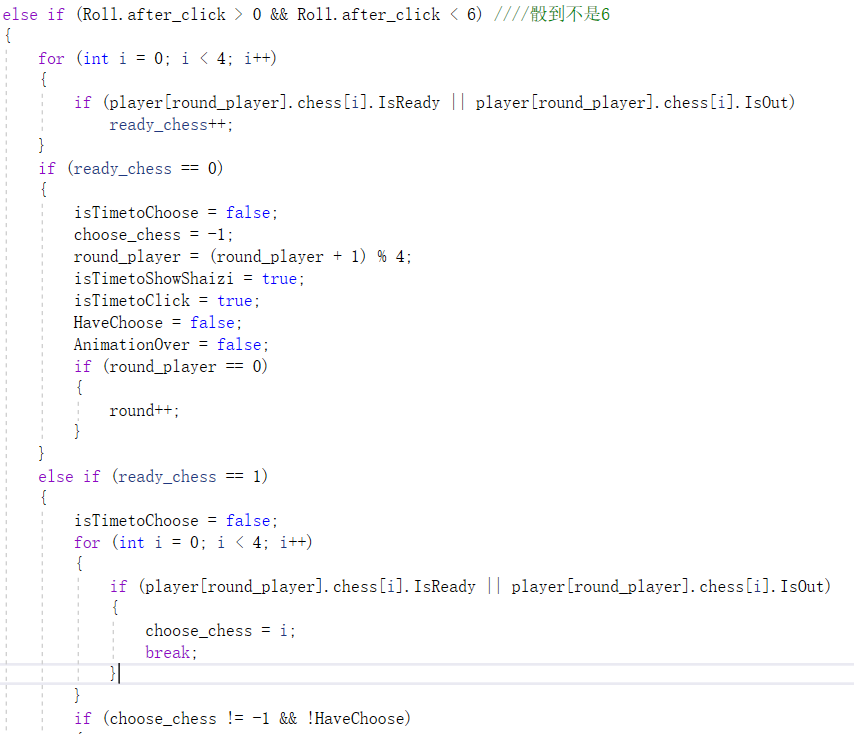
骰子抛到6，则棋子可以飞行。并且玩家可以再次投骰。

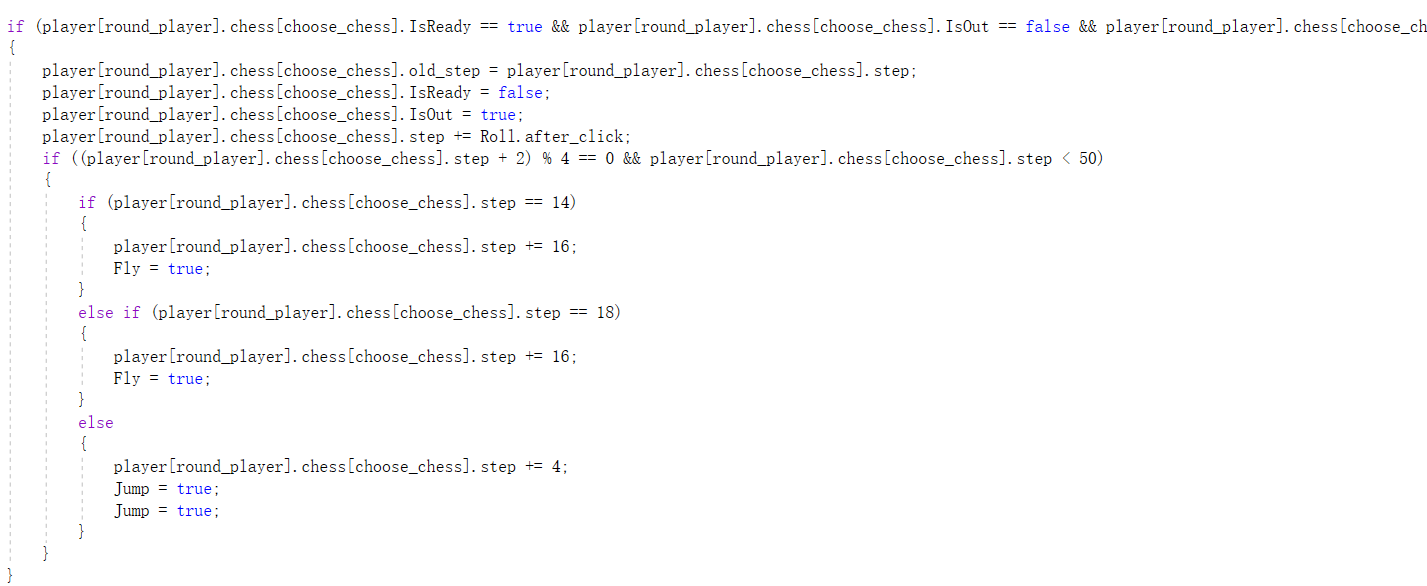
骰子未抛到6，则棋子无法飞行，下一位玩家。

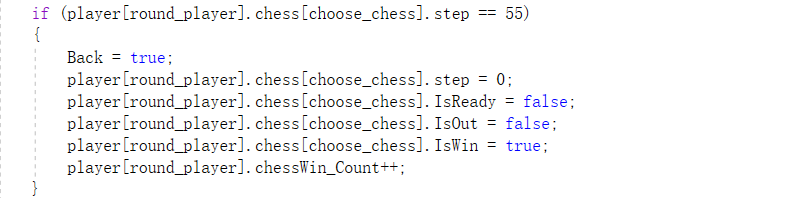


1. 小鸟的行走逻辑。

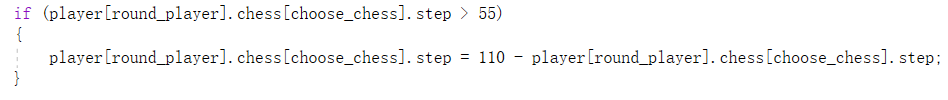
当掷到六时，玩家可以选择小鸟进入准备状态或者行走。当没有掷到六时，需要根据准备区是否有小鸟来实现走或不走。没有掷到六时的部分代码逻辑的实现如下：



小鸟的行走步数由step来记录。当小鸟的行走步数小于50步时，还要判断小鸟是否需要跳跃或飞行。当小鸟行走后所在的格子为同色，则可能发生跳跃或飞行。按照地图的规定，小鸟出门的第二格就是同色的，然后每隔四格就是同色的格子。当小鸟恰好走到第14步或第18步时，小鸟会实现跳跃+飞行，共跨越16步。具体代码逻辑的实现如下：

当step的步数刚好等于55步时，小鸟回到鸟巢，产出一颗鸟蛋。具体代码逻辑的实现如下：

当step步数大于55步时，小鸟触底后要往回退。具体代码逻辑的实现如下：

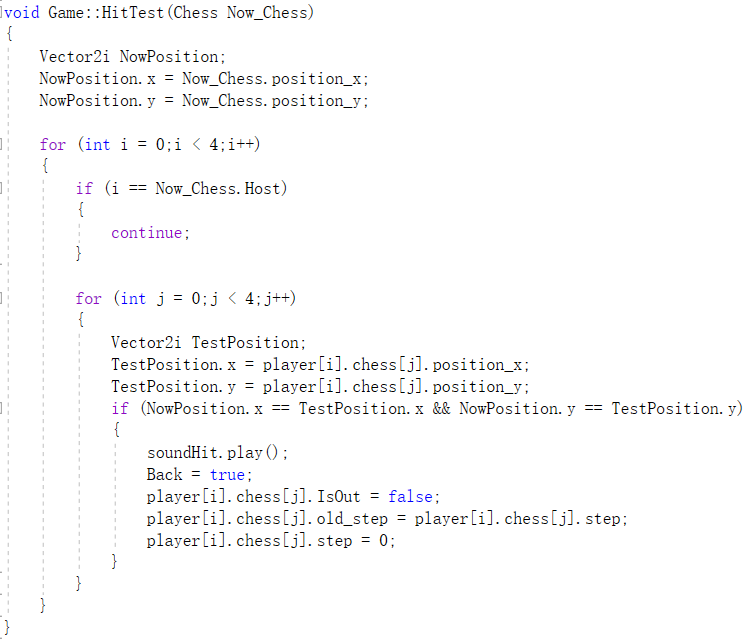


游戏逻辑的代码很长，但核心逻辑就是上述的不同step步数相对应的逻辑代码。

3.碰撞检测

小鸟行走的过程中，可能会和其他不同色的小鸟走到同一格，这时就会将那只小鸟撞回。所以，小鸟的每次行走都要通过碰撞检测HitTest()函数检测。

HitTest()函数的思路是遍历所有非同色的小鸟，若其位置与当前小鸟的位置相同，则撞回。具体代码逻辑的实现如下：



**（3）绘制**

**6.3 游戏结束**

当有一玩家的四个棋子全部到达终点，则游戏结束。

## 游戏SWOT分析

|  |  |  |  |
| --- | --- | --- | --- |
| S优势 | 界面美观 | O机会 |  |
| W劣势 |  | T威胁 |  |