

暑期小学期 Week1 大作业 FlowFree 设计文档

Author: 计 42 殷达 #2014011320

类设计概述:

整个游戏在我的代码中分为两大模块,一个是游戏逻辑,另一个是游戏界面,还有一些是用来辅助游戏运行的以及作为基本的控件出现在游戏界面中的(Qt 的一些控件和我的游戏界面风格不是很搭,所以自己设计了一些控件)。

类功能简介:

游戏逻辑:

GameController: 游戏逻辑核心的类,负责统筹游戏逻辑层面的运行。

GameSetting: 存储游戏相关信息的类,记录当前游戏的完成进度、历史最佳纪录等等。还负责从文件中加载当前关卡的信息。

GameSolver: 用来计算游戏的解决方案,采用深搜,在线提供解决方案。

Pipe: 用来记录各个水管的信息。起始点,经过点,结束点。

PipeManager: 用来管理游戏中的各个水管。

游戏界面:

GamePanel: 游戏界面核心的类,在游戏运行时用来承载并管理游戏中各个部件的布局以及功能连接、实现。

FlowSource: 是游戏源头(Flow),即圆圈的控件,存储源头所在位置以及颜色。

GameBlock: 是游戏的背景格子控件。每一个格子是一个控件,用来记录其上管道的颜色,并决定何时显示。

MouseState: 是游戏的鼠标旁边透明环的控件。在显示透明环的时候,滚动鼠标滑轮可以改变透明环大小。

PipePainter: 用来绘制游戏中各个水管的控件,通过 **PipeManager** 获取水管信息。

游戏辅助:

GameAudioPlayer: 用来播放游戏音效。由于频繁播放音效会让游戏的流畅性降低,所以该控件还觉得是否过滤掉当前的音效。

MyPainter: 用来替代 **QPainter**,设定了一些参数,使得游戏的绘图更加优美。

基础部件:

GameViewPanel: 是我用来写各种游戏面板的基础类,背景略透明,可以拖拉移动,也可以通过右键固定位置。拖拉状态为圆角矩形,固定为普通矩形。

GameButton: 是我自己写的游戏的按钮。具有透明效果,圆角,简洁。

GameSizeChangedButton: 是 **GameButton** 的扩展,在鼠标移至其上的时候大小可以变化。用在主菜单界面中。

GameChooseBox: 是自己设计的下拉选框,由若干 **GameButton** 组成。可以设置下来的项目数量。

GameInfoViewer: 游戏运行过程中可以通过按住 **Ctrl** 键调出,用来显示当前游戏状态和相关信息(**GameSetting** 类的数据)。

GameLevelChooser: 选择游戏界面的控件,用来布局各个 **Label** 以及 **ChooseBox**,同时把选择的关卡信息告诉 **mainwindow** 运行相应关卡。

GameMenuInterface: 主菜单。用来布局各个 **Label** 和 **GameButton**,可以选择 5~7 棋盘大小或者转入选择界面。游戏过程中返回的菜单也是该界面管理。

GameMessageBox: 游戏消息框。游戏胜利的时候跳出消息框,选择重玩关卡或者下

一关。

注：由于类结构稍显复杂，其接口函数有很多都较为简单，便不做解释。

一些类的重要接口函数：

游戏逻辑：

GameController:

bool Valid(const **QPoint**& _coordinate); 判断坐标是否合法。
bool Adjacent(const **QPoint**& p1, const **QPoint**& p2); 判断两个坐标是否相邻。
int Distance(const **QPoint**& p1, const **QPoint**& p2); 计算两个坐标的曼哈顿距离。
bool CheckWin(); 判断是否满足胜利条件。
QPoint Coordinate(const **QPoint**& _pos); 将屏幕坐标转换成游戏坐标。
QPoint Center(const **QPoint**& _coordinate); 将游戏坐标转换成屏幕坐标。
void AutoSearchPath(const **QPoint**& _coordinate); 在游戏坐标跳跃时，补充完成中间空缺的步骤。
void GameStateRecalculate(); 将游戏的运行状态重新计算，更新游戏进度等信息。
void CoordinateSelected(const **QPoint**&);
void CoordinateChanged(const **QPoint**&);
void CoordinateReleased(); 用来在游戏焦点坐标改变时处理游戏，是游戏逻辑层的核心。

GameSetting:

void LoadFile(); 从文件中加载游戏关卡数据。
void SaveFile(); 将玩家的游戏成就保存在文件中。
static GameSetting getNextGame(const **GameSetting**& currentSetting); 获得下一局游戏的游戏信息。
static GameSetting getPreviousGame(const **GameSetting**& currentSetting); 获得上一局游戏的游戏信息。

GameSolver:

void Search(const **QPoint**& ori, int oriColor); 递归深搜。
void Calculate(); 计算路径。
void Analog(); 模拟游戏完成路径。

游戏界面：

GamePanel:

int getBlockHeight(); 获得实时格子高度。
int getBlockWidth(); 获得实时格子宽度。
void UpdatePipeDisplay(); 更新管道信息并绘制。
void UpdateBlockDisplay(); 更新格子信息并绘制。

MouseState:

void StartPaintCircle(const **QColor**& _color, const **QPoint**& _pos); 开始画鼠标透明环。
void EndPaintCircle(); 结束绘制。
virtual void wheelEvent(**QWheelEvent** *); 在鼠标滚轮滑动的时候更新透明环大小。