

俄罗斯方块项目总结报告

项目概述

项目目标

实现经典的俄罗斯方块小游戏，可以统计游玩分数与时间并多次重复开始，可在各版本Windows机器上进行游戏。

项目实施细节

本项目采用MVVM框架，将界面和操作逻辑分离，利于小组协作。总体分为app, common, view, viewModel, model层，以下逐一介绍各层作用。

app层

app层主要起连接作用，对view需要的数据（指针）和命令（指针）进行设置，对model viewModel需要的通知进行设置，使它们可以显示数据以及调用对应的命令或通知。

common层

存放各层统一需要的结构声明，实际即CommandBase和NotificationBase两个基类，commandBase即各种命令的父类，通过重载Exec()命令使得拥有对应指针的view层可以在获取用户事件后调用对应函数。NotificationBase为通知类的父类，可以通过para成员参数来标识不同通知，通过setParameter()来设置参数，通过notice()函数来发送通知，使model, viewModel在完成某些特殊事件后让view层打开窗口或绘制某些图片与玩家交互。

```

1  class CommandBase
2  {
3  public:
4      virtual void Exec() = 0;
5  };
6
7  class NotificationBase{
8  public:
9      virtual void notice() = 0;
10     virtual void setParameter(int give) = 0;
11 protected:
12     int para;
13 };

```

view层

存放了具体的通知类，可以用于提醒用户游戏结束。Mainwindow类用于绘制窗口以及事件的接受和执行。

绘制窗口分为两部分，一部分使用qt自带控件实现按钮和数据标签的显示，另一部分使用paintEvent来绘制游戏池、next等具体方块图片。

qt接受键盘输入需要窗口获取焦点，然后我们在keyPressEvent中对接受的键值进行分析，然后调用对应的命令进行处理。

最后是三个QTimer时钟，一个较高频率用于不断刷新界面，防止给玩家操作带来滞后感；一个固定0.1s的时钟长用于游戏的计时；最后一个用于方块的自由下落。

viewmodel层

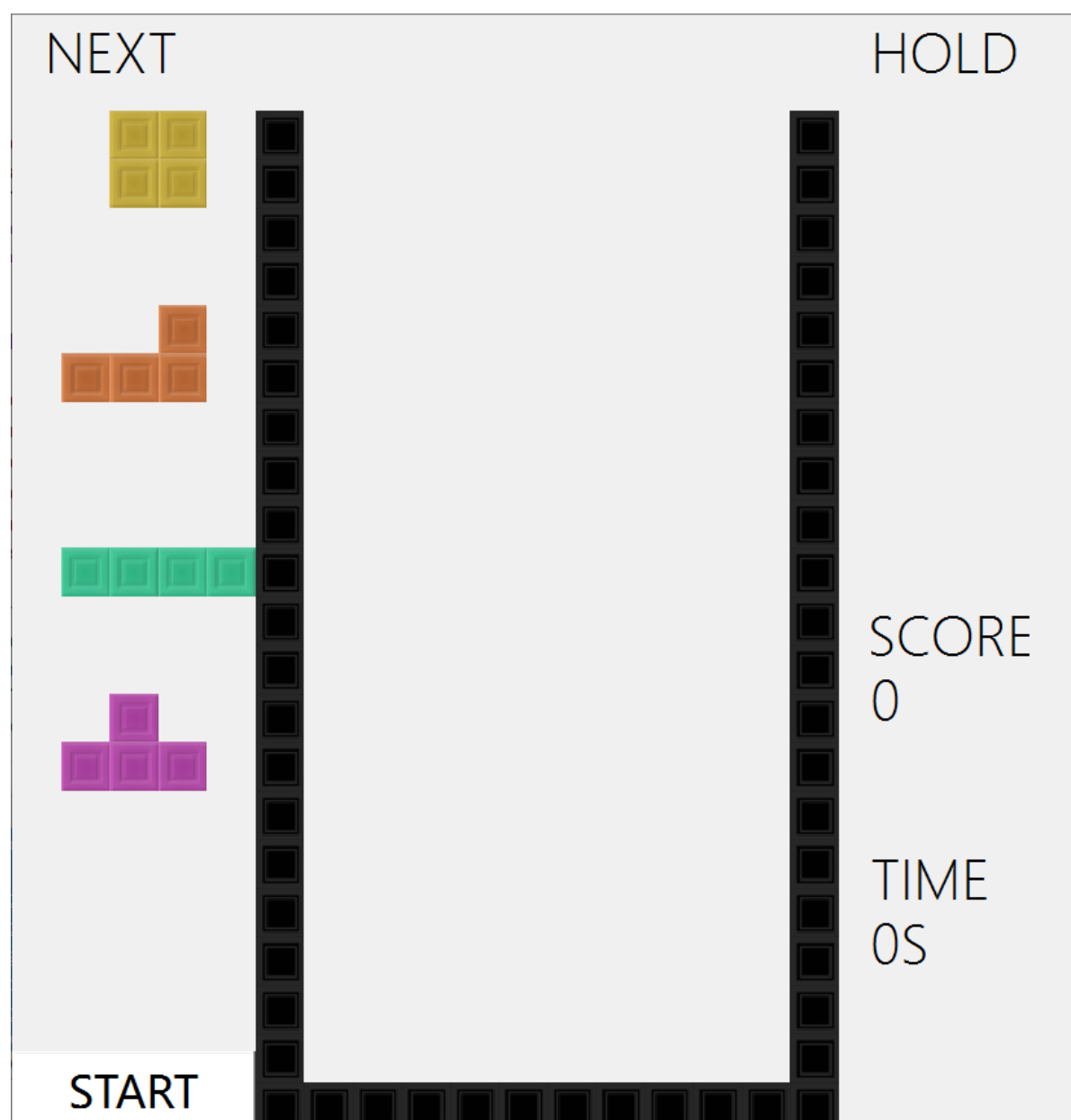
作为view层以及model层的中间层，负责创建命令对象并通过各种get函数对view开放使其能不接触model而执行对应操作。命令对象内部实际即调用model内部函数进行数据操作。另外除命令外，也通过viewmodel层对需要的通知进行设置。

model层

主要的数据存储与处理层，定义了Tetris和matrix结构用于内部处理单个方块。DataModel类存储具体需要显示交互的数据，诸如方块池、操作方块、分数、游戏状态.....实现平移旋转开始游戏等具体操作，viewmodel层仅负责安排调用。

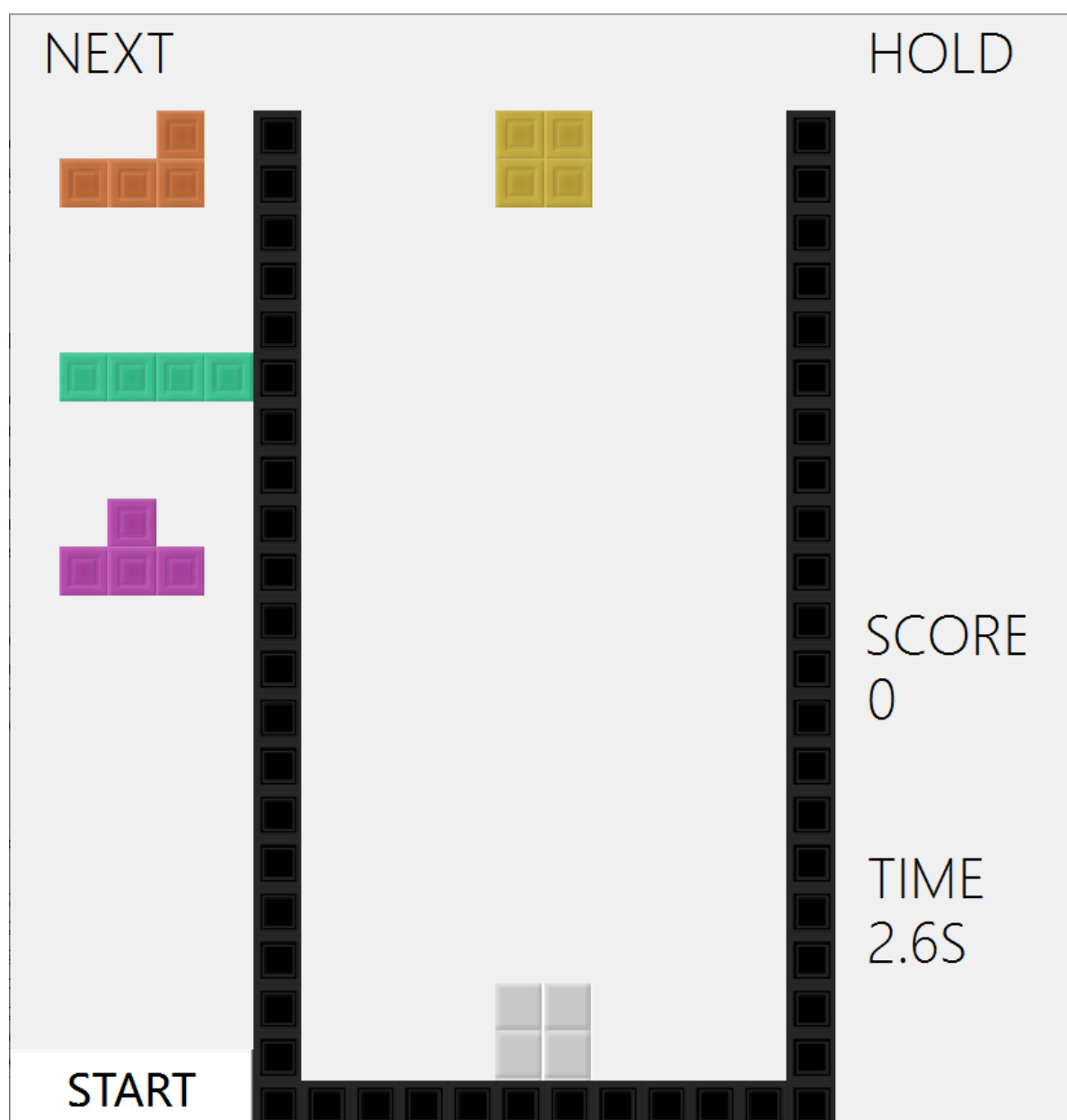
项目成果

游戏开始界面，点击左下角按钮即可开始。



游戏过程

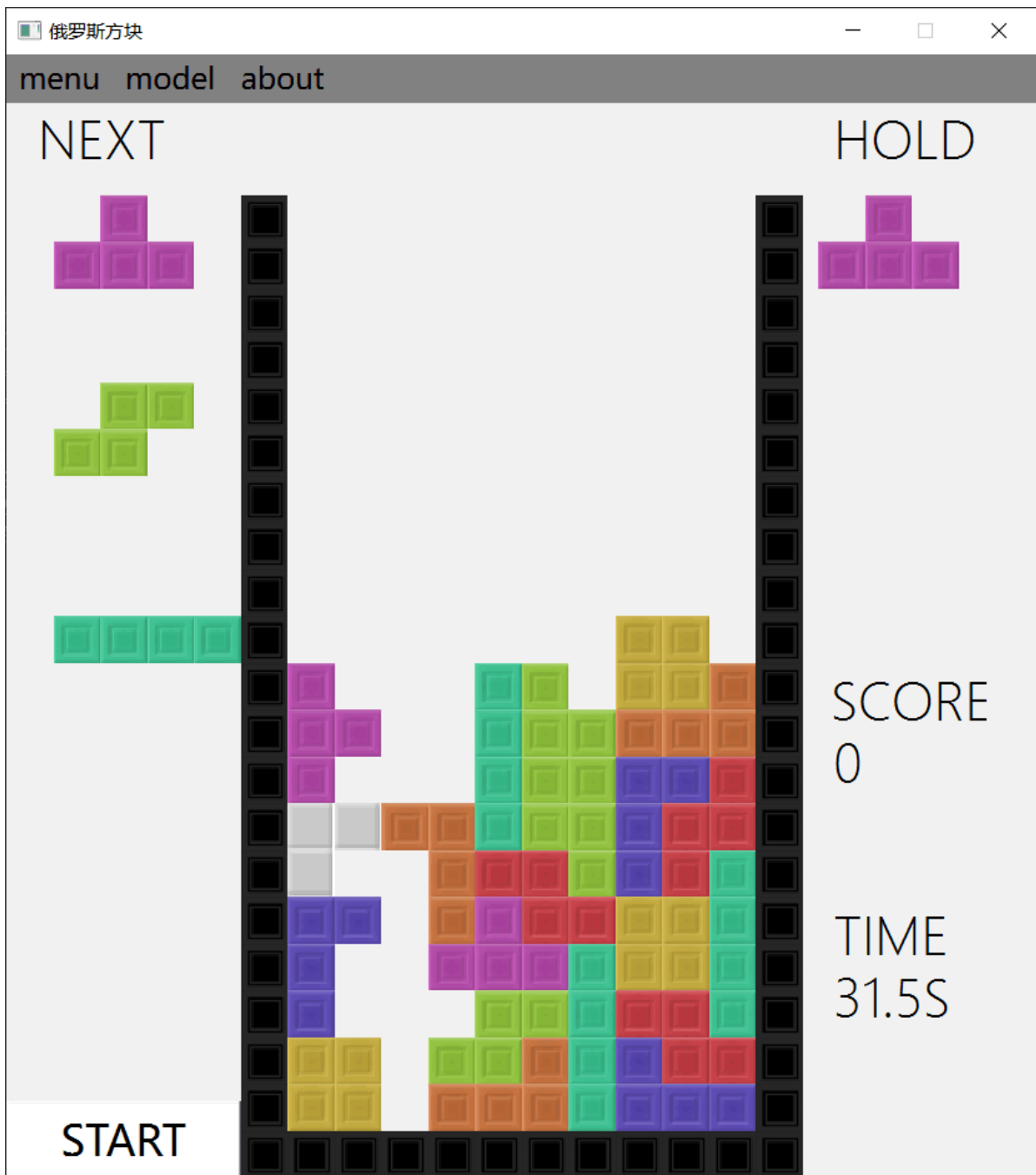
对于当前正在下落的方块，我们有灰色方块作为阴影便于玩家横向定位。



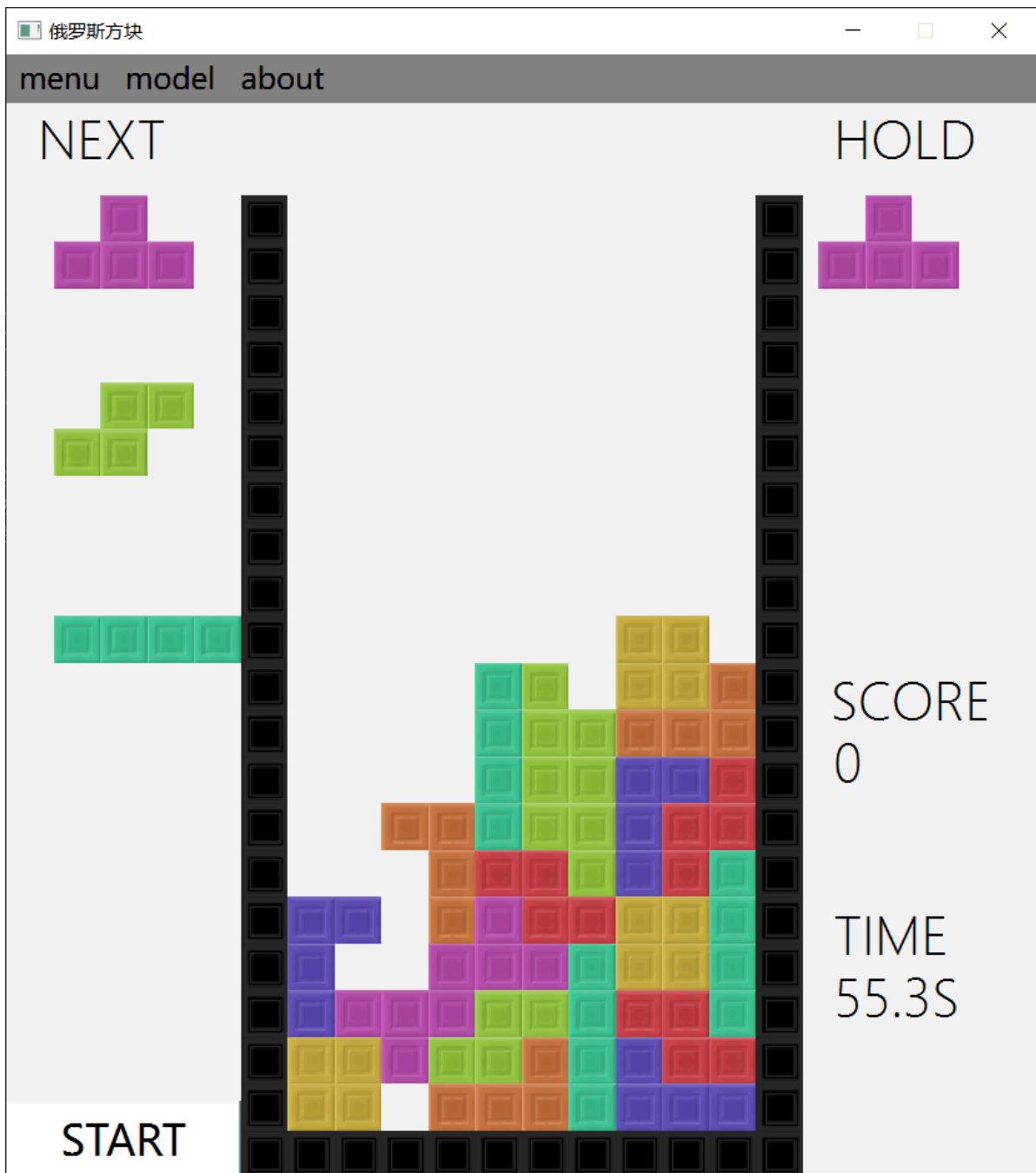
复杂的T-旋转操作

分数、时间右下角显示。

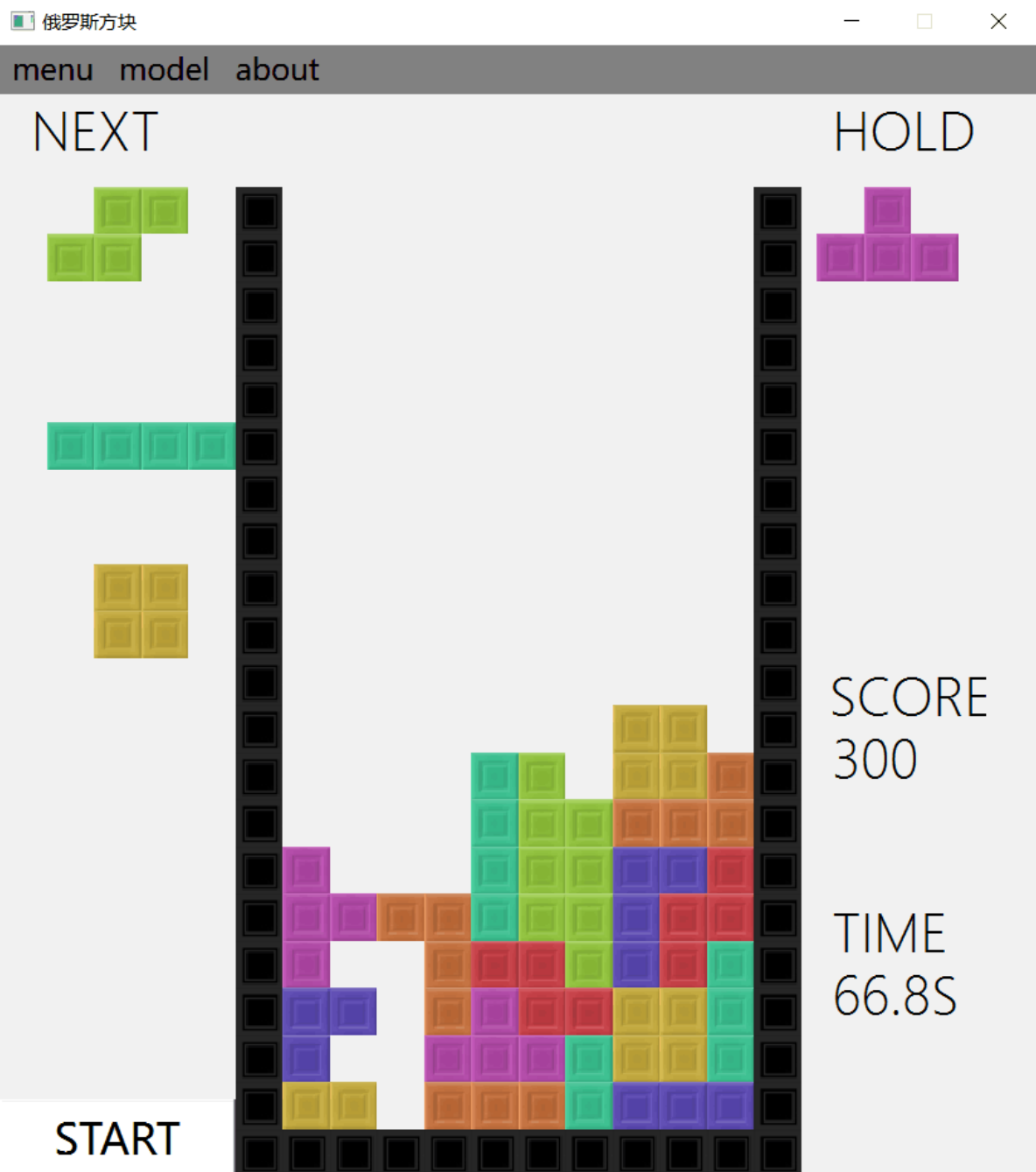
1. 方块即将碰撞

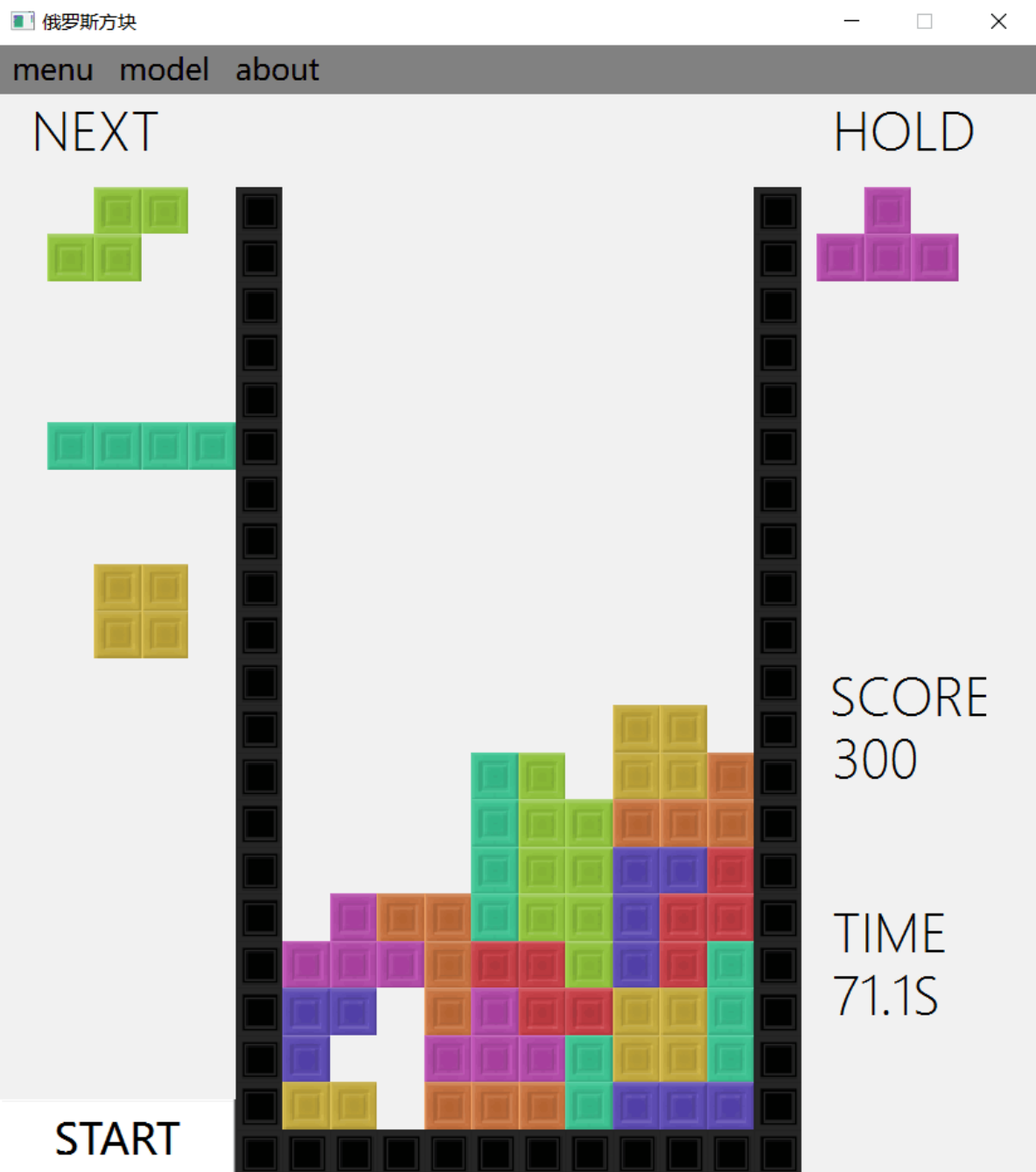


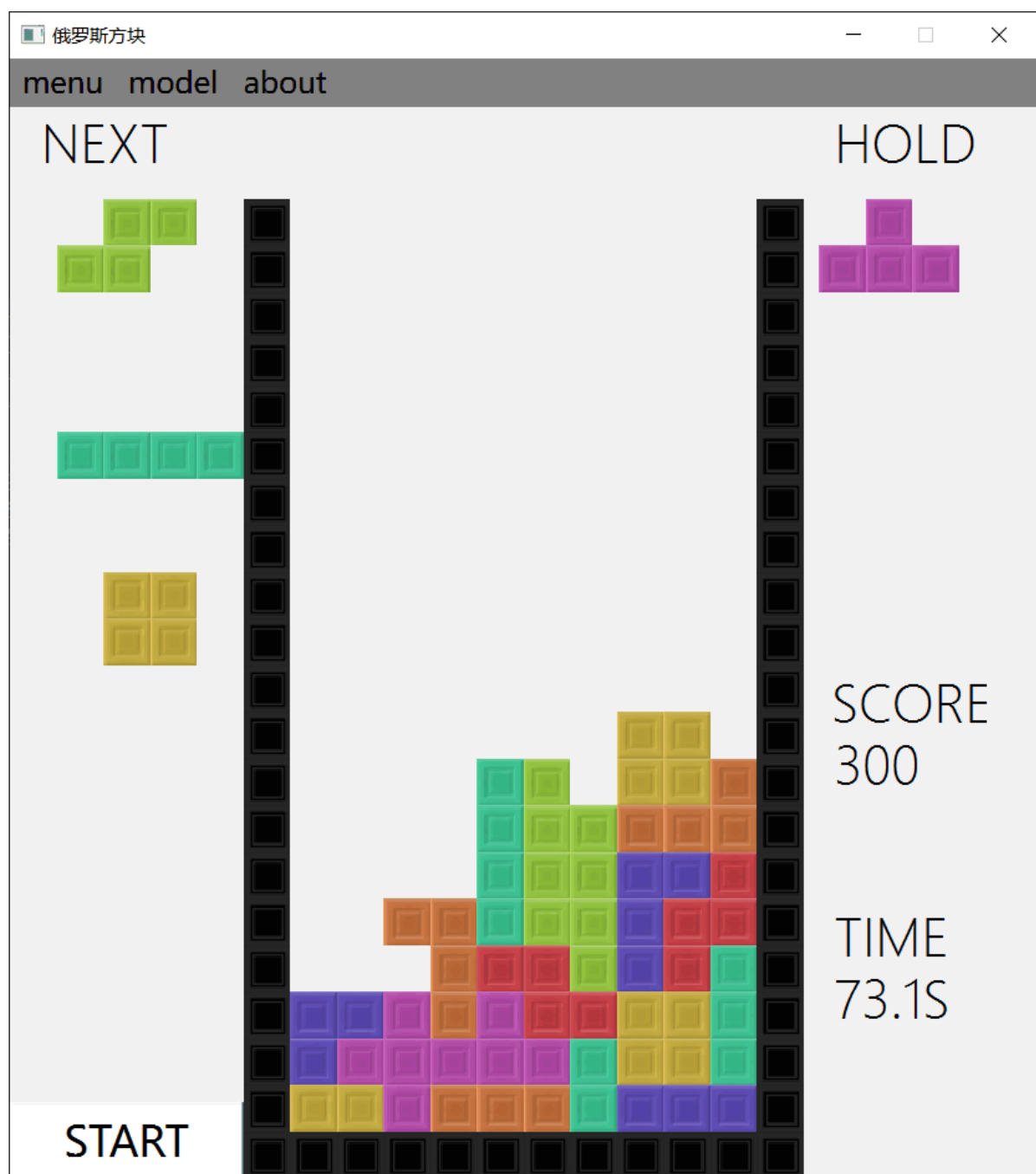
2. 通过旋转将其卡入内部对应形状



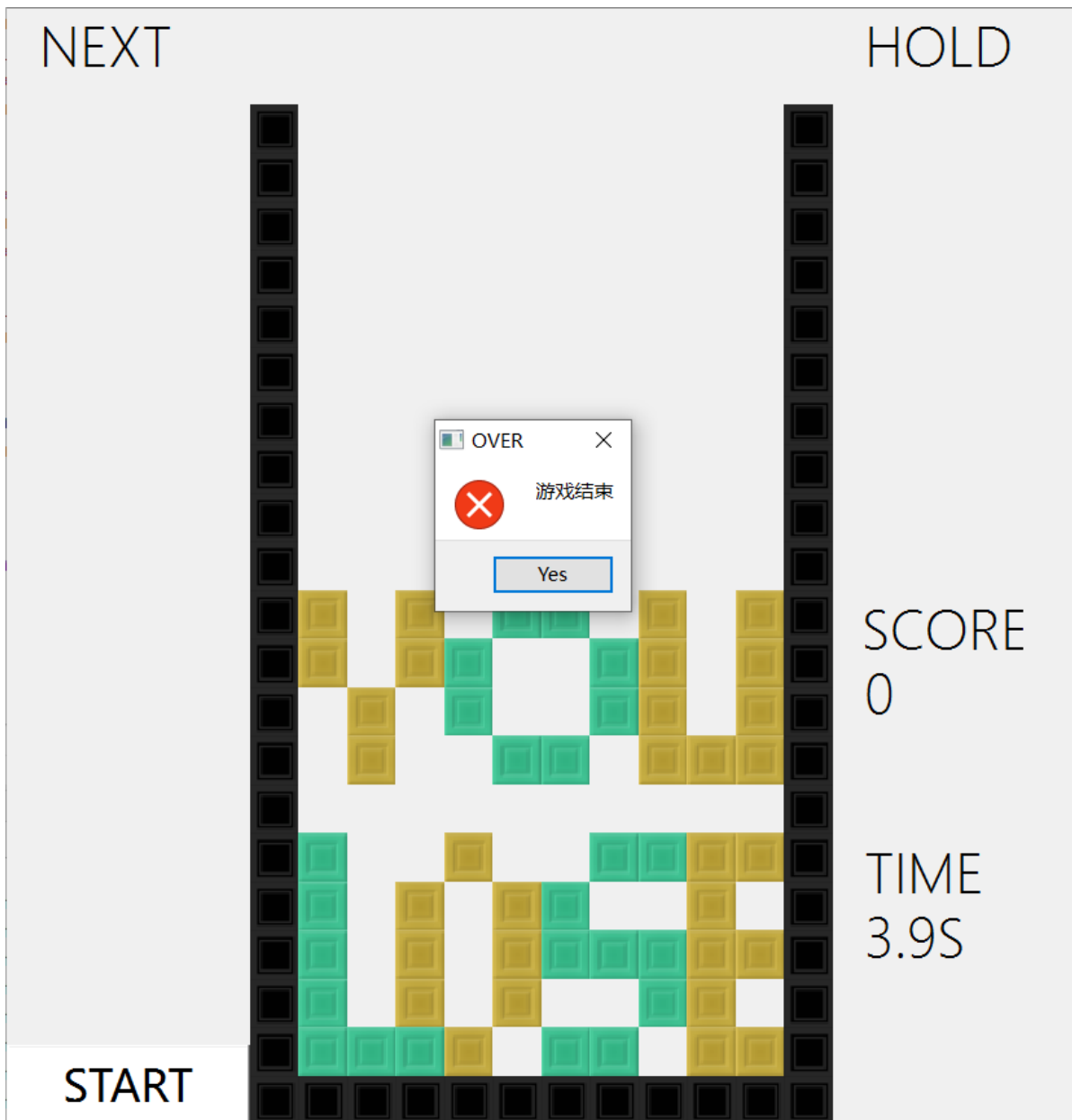
3. 另一个T-旋转,另外可以看到上一步消行已经有了对应分数





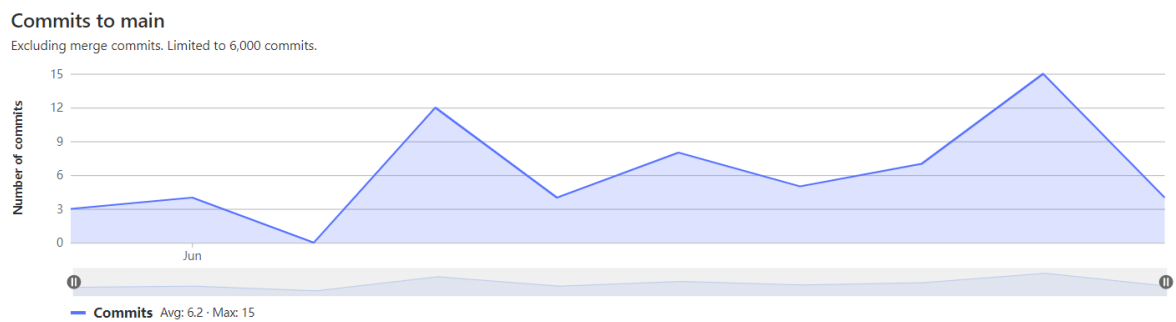


游戏结束，弹窗并通过方块池图片提醒



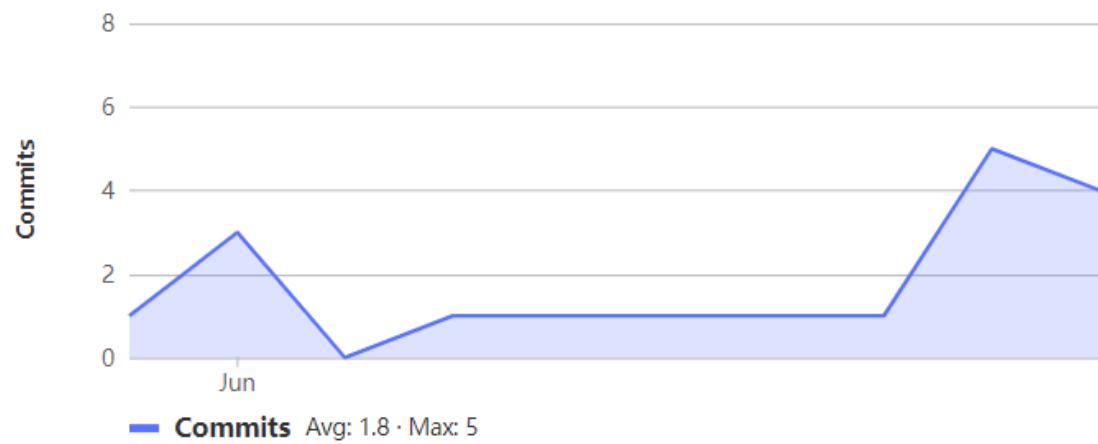
gitlab使用情况

我们使用了浙江大学Git代码托管服务，各自负责各自的部分并积极与他人联系。



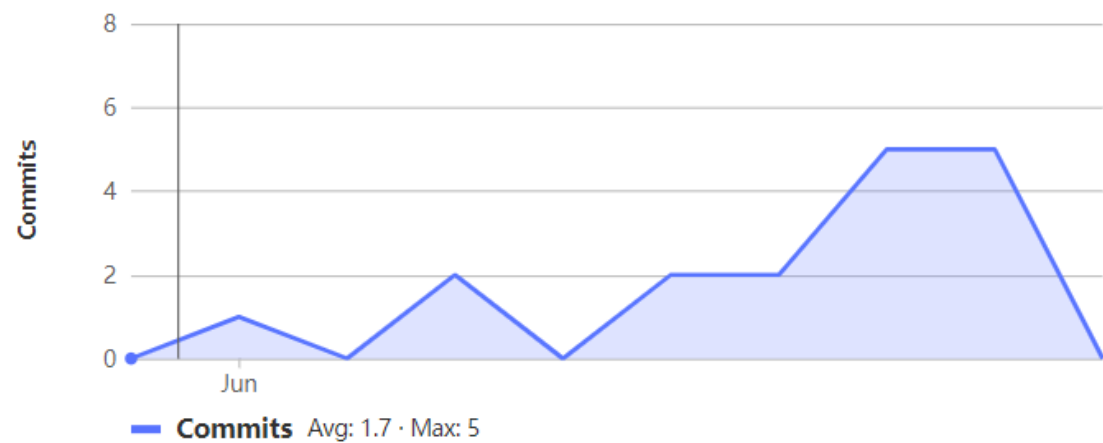
Gitnothingya

18 commits (chinayanhan@outlook.com)



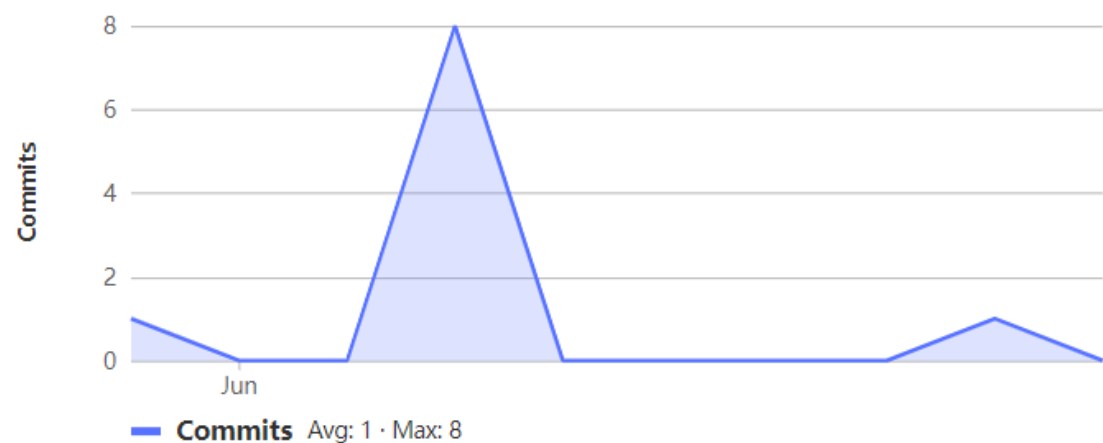
yxn

17 commits (1348624696@qq.com)



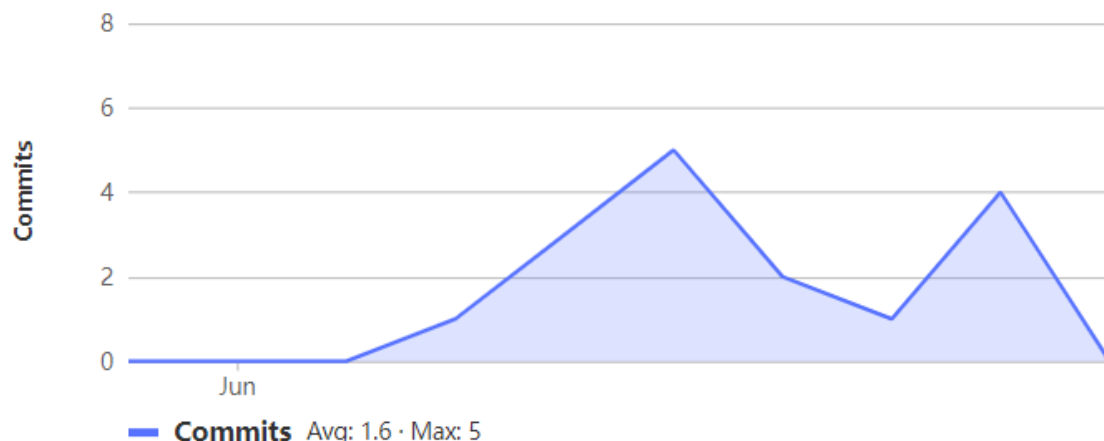
殷信楠

10 commits (3200105938@zju.edu.cn)



李真泉

16 commits (3200105207@zju.edu.cn)



总结与个人感悟

该项目中我们成功复刻了经典的俄罗斯方块游戏，经过测试，游戏可流畅进行。

颜晗：

本次开发我主要负责各层框架的构建，管理项目进度并协调各层的任务。算是一次比较难忘的经历吧，第一次这么正式地使用工程框架以及代码托管工具来协助开发，相比较以往基本一个项目虽然分配给小组但是真实由一个人开发的情况有很大不同。也是第一次没有经验，在组队协作过程中有许多的不足，有时也会想还不如独自开发，但是本次项目其实比较简单，遇到真正的大型项目，是不可能一个人干活的，尽早适应协作开发有巨大的好处。这次经历算是给自己以后积累了宝贵的经验。最后感谢两位队友在开发过程中积极配合，及时完成任务，同样感谢袁老师细心指导，让我对 `MVVM` 框架有了较深的认识。

殷信楠：

本次开发我负责了项目中较为重要的数据和操作。通过上千行代码的编辑，我对 `C++` 语言更加熟练了，在不断的 `deBug` 中锻炼了代码能力，也了解了不少新知识，这对于往后继续深入学习 `C++` 有相当大的帮助。在与组员的不断交流中，对 `MVVM` 框架的了解也更加深入了，也体会到了使用框架开发的好处。

最后，感谢两位队友在短学期的陪伴以及付出。感谢袁老师数日在机房的教导和陪伴。

李真泉：

在本次的 `Project` 中我负责程序的 `view` 层。通过这次的项目，学习了 `mvvm` 框架，对于软件项目的分工有了具体的感受，更加理解工程中解耦合的重要。我们使用 `Qt` 图形界面开发框架，并且用 `Git` 来进行版本控制和项目管理。我负责的部分的难点在于对后端接口的联系，对 `ui` 界面的设计。`Qt` 提供了许多的控

件，构建时需要熟练使用这些控件，过程中 `ui` 参数的调整比较繁琐。通过 `Qt` 中对象树的使用，我对于 `C++` 对象的构造有了更深的理解。此次项目的经验对我之后开发较大型项目非常有帮助。

项目过程中，队友都很好地交流协作，非常感谢队友的付出。

最后非常感谢袁老师的教授和指导，我在此次课程中受益匪浅。