1 简介

游戏名：《合成大西瓜》

组名：五月瓜

组员：

陈衍德 2018013382 搭建框架；团队手册与报告撰写；物理引擎实现；鼠标操作实现；代码评审与集成测试

段祎然 2018010358 原版游戏素材搜集；物理引擎测试与优化；水果的碰撞检测；分数计算与分数动画显示

杨健藩 2020011533 背景图片设计与绘制；按钮设计、绘制与功能实现；水果绘制

杨祖怡 2020030013 背景音乐播放；水果合并的逻辑；水果合并时的果汁动画

陈益良 2020030025 水果的运动（更新位置、速度、加速度）；合成出大西瓜闪金光的动画

2 设计思想

本游戏应用自顶向下可分为**应用层、逻辑层**和**物理层**。

应用层负责背景、水果、按钮、分数等视觉元素的显示，并接收用户的点击按钮、释放水果、进入/退出设置界面的鼠标操作；

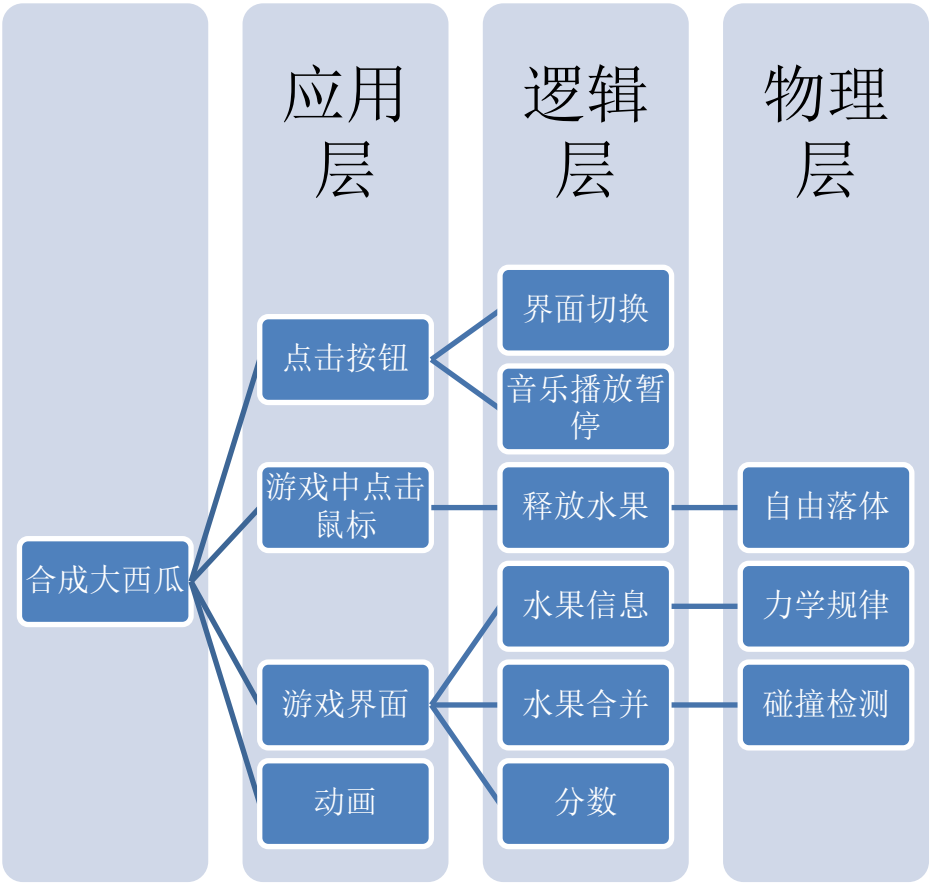
逻辑层负责游戏状态（分数、现有水果等）的维护，实现水果合并等功能，并根据用户的鼠标操作执行音乐播放暂停、界面切换等逻辑；

物理层保证所有水果遵循物理规律，让水果做符合力学规律的运动。

层与层之间的独立性将原问题分解，使实现更容易。

3 系统框架

模块分解：



流程图（游戏主逻辑）：

4 游戏功能

4.1 界面切换

在开始界面点击 PLAY 按钮可以进入游戏界面；

在开始界面点击左上角的齿轮可以进入设置界面；

在游戏界面点击右键可以进入设置界面；

在设置界面点击右键可以回到上一个界面。

4.2 设置

在设置界面点击音乐开关可以播放/暂停音乐。

4.3 游戏

点击鼠标可以释放水果，两次释放至少间隔 0.5 秒。

两个相同的水果接触后会合成一个更高级的水果，并得到等同于新水果级别的分数。

同色的水果（樱桃与番茄、柠檬与菠萝）接触后会合成较大的那个水果，并得到等同于较大的那个水果级别的分数。

合成出大西瓜时，得 100 分。

若水果漫过红线且持续 5 秒，则游戏结束，此时可以点击右上角按钮重新开始游戏。

5 源代码

Github 仓库：github.com/chen-yd18/MergeForMelon

本地文件：见 src 文件夹，其中 watermelon.cpp 为主程序（main 函数在其中），其余文件提供必要的函数。

fruit.h/cpp：水果结构的定义和基本方法

fruit_detect.h/cpp：水果的碰撞检测

fruit_move.h/cpp：水果的移动

fruit_merge.h/cpp：水果的合并

button.h/cpp：按钮的定义和基本方法

animation.h/cpp：果汁动画、金光大西瓜动画的定义和基本方法

window_const.h：必要的常量定义

honor_code.h：满足课程的诚信要求

其余代码文件因原定功能过于简单，已废弃，功能直接转移到 watermelon.cpp。

6 调试和解决方法

6.1 游戏卡顿甚至卡死

每位组员都遇到过这个问题。查看控制台输出发现这是在主循环内加载 texture 造成的，主循环每秒执行 60 次，每次加载十个 texture，就是 600 次加载，极度消耗性能。

多位同学尝试过把 texture 定义为全局变量并在定义时加载，结果程序无法启动。这是因为 texture 的加载必须在 InitWindow()之后。将这些全局变量的赋值移到 main 函数中 InitWindow()之后即可。

6.2 涉及结构体的程序编译不通过

这是因为概念混淆。现有 JuiceAnimation 结构体的指针变量 anim，有两位组员在访问 type 成员时写了 JuiceAnimation.type，然后编译错误。正确的访问方式应该是 anim->type。

6.3 水果卡在墙里，未成功反弹

原逻辑为“若水果与墙碰撞，则将水果的速度反向”。这一逻辑错在：水果往往无法在 1 帧之内从墙内移到墙外，导致上述逻辑再度触发，速度再度反向，反而向墙里走了。

加上“若水果与墙碰撞，则直接改变水果位置，将其移出墙外”即可。

6.4 一个水果以极短的时间间隔连续碰撞两个水果后，可能会飞天

第一次碰撞后水果由于弹力获得一个很大的速度，它借这个速度冲入第二个水果内部，会获得巨大的弹力，直接上天。

但我们认为这是正常的。这很 cool！

6.5 一堆水果可能在高处以极快的速度散开并飞出银河系，导致 gameover

这是因为水果释放点离现有的水果太近，导致水果之间出现过大的重叠，形成巨大的弹力。这个弹力同时作用在两个水果上，导致下方的水果向下猛砸，再度与现有的其他水果重叠过大，引发连锁反应，全体飞出银河系。

将死亡线向下移动，即可增大水果释放点到现有水果的距离，从而解决问题。

6.6 以极快的频率点击鼠标会导致水果在空中碰撞、减速，最终 gameover

原版的移动端游戏不存在这个问题，因为人的手指不可能点击得这么快！

我们加上了一个限制：两个水果释放的时间间隔不能短于 0.5 秒（30 帧）。问题大大缓解。

6.7 屏幕高度不够，导致游戏难度过大

原版的移动端游戏也不存在这个问题，因为它是竖屏的，纵向分辨率很高。

为了照顾中低配置的笔记本电脑，我们无法增大窗口的高度。我们采用了替代方案：给游戏增加一些能降低难度的特性。

我们增加的特性包括：两个水果合并时，新水果有一定概率生成在上方水果的位置；同色水果（樱桃与番茄、柠檬与菠萝）可以合并，合并结果为较大的那一个水果。

7 心得体会

陈衍德：

这是一次超越规律的软件工程实践，让我惊喜于清华学生的能力和学习态度。

不管是在《软件工程》课程讲授中，还是在体育赛事管理系统的开发实践里，我都认识到理想和实际之间存在鸿沟。一般来说，随着迭代次数的增加，理想与实际的差距会成倍扩大。然而在《合成大西瓜》的开发过程中，理想与实际的差距保持在稳定的、较低的水平，我们甚至还开发出了预想之外的新特性。这与组员们精益求精的态度、强大的自学能力是分不开的。

软件工程理论还告诉我，当团队人数增加时，沟通复杂度会平方级增长。但我通过改变小组合作方式破解了沟通复杂性：每周例会给所有人下发任务，然后所有人通过微信与我单线联系，我操作 github 仓库。（感谢 github 极其糟糕的网络连接状况，导致半数以上组员无法使用它，迫使我想到了单线联系的沟通方法）

这还是一次不错的助教工作体验，为读研之后担任助教工作做了准备。

大作业布置之前，我完成了 start code 的编写工作：将《合成大西瓜》游戏切分成十几个模块并分别安排头文件和源文件，写好每个文件必备的函数声明和结构体定义，最后完成 Makefile 的编写，保证组员可以直接上手；

大作业开始时，我给组员们演示了如何编译运行项目、如何查 raylib 文档等常用操作；

大作业开发过程中，组员遇到了一些困难，我进行了一些 debug 和讲解工作，帮助他们纠正了一些概念理解上的错误。

段祎然：

这次作业非常幸运地能在一位经验丰富的组长带领下，和三位同样热情满满、友好高效的队友一起完成。第一节课后，衍德哥哥突然问我“你也在上计程设呀”，我惊觉开给大一的计程设课上竟然有一位软件学院大三的 dalao。原来答疑坊志愿者衍德哥哥希望能更好地帮助在答疑坊提问的同学，自己也选了课。衍德哥哥是一位负责的志愿者（在上学期软院计网课上他也耐心帮助不会做大作业的我），也是一位负责的组长。相比我们，他有着丰富的开发经验，这些经验在这次大作业中，我认为主要体现在技术选型的迅速确定（图形库使用轻量级的 Raylib 而不是重量级的 Qt）、科学的架构、明晰合理的分锅和进度安排。在组长的

超强分工下，我们得以自如地开展工作，不必担心和其他组员的代码发生较大冲突，在自己部分的 coding 时也不必深入了解他人的具体工作，大大提高了工作效率。在一周一轮的不断迭代下，我们的大作业进度稳步推进，这是我在清华的三年来第一次在 ddl 前没有觉得头冷的大作业（捂脸）。看到小游戏逐渐成型，心里有一种说不出的成就感，这种成就感也更能促进我在下周的迭代中更用心、开心地写代码和 debug。

在这次大作业中，我更加巩固了 c 语言基础，快速上手了 Raylib，学习到了一个较大型程序的文件组织方式，并且学习到了如何科学地安排作业进度——这不仅对编程类的课程有帮助，我最近总在感叹，如果自己是大一时在这样一位组长的带领下完成一次大作业，找到这样“匀速推进进度”的感觉，可能大学的生活也会更好过一些吧（sigh）。

ddl 前最后几天，我在这个大作业上花的时间也有所增加，主要花在了~~玩游戏~~测试上。我们做的游戏，我觉得比原版的还要有趣。这次大作业真是太有趣了。

杨健藩：

这次大作业我主要负责的是美工（图片处理）和一些图片展示、按钮实现相关代码的改写。

经过这次大作业，我初步学会了如何去了解、学习和使用一个以前从来没有接触过的库（raylib.h），就拿想要把封面图片在窗口里画出来这个功能来说，当时也没有学过可视化的操作，所以这个过程就需要仔细去阅读库里的函数功能，并不断地尝试各种函数组合。同时，我还深刻地理解了优化的重要性，记得我刚开始把 imageloading 和格式转换都放在了循环里面，然后一打开文件就会不断地上传图片，当需要上传的图片多了以后，就开始变的卡顿，最后直接闪退了，而优化之后，把图片上传和格式转换都放在了循环之外，这个问题就被解决了。除此之外，我觉得最重要的是，我训练了 debug 的思维，在做大作业的过程中，我出现过许多可能比较低级的 bug，但是通过不断地 debug，我现在可以比较熟练得通过 error 反馈去寻找 bug 所在。

总之，这次大作业让我了解到了一个游戏从零到比较成熟的完整流程，并学到了一些可视化的操作，但最重要的，还是培养了我的编程和 debug 的思维，让我学会如何学习不会的东西，如何寻找自己代码的漏洞。这次大作业收获真的很大！

最后不得不提的是衍德哥哥真的太 carry 了！不仅设计了整个游戏的框架，在这个过程中也帮助了我很多，给我了讲许多易混淆的知识，并在 debug 方面极大地帮助了我，真的 tql！！！

杨祖怡：

从来没有想到这样一款很火热的游戏就这样在我们团队中一点点生长出来了。从最初对框架摸不着头脑，弄不清思路，到后来不仅实现了预想效果，还不断出现新点子，让我们的游戏有了许多比原版更有趣的地方。攻克难关的过程是极具挑战的，一起解决的过程是非常愉悦的。设计这款游戏让课内的知识不再只是僵硬地储存在我的脑海里，而变成了真实有用的工具。学与用，学与玩，在这个过程中得到了最好的结合。

陈益良：

学程设前：程设是什么？学程设后：程设是什么？!!! 得知大作业要写游戏后，完全不知道从何下手，不相信自己学的这点东西能写出能玩的游戏。

写程序的过程也非常迷惑且痛苦，一边学习一边写大作业的过程更迷惑且痛苦。慢慢的一点一点理解了基本的框架，虽然代价是键盘上很多的头发。

尤其是过程中还遇到了很多很多的问题，比如 github 永远也上不去，代码只能手动发出和合并，以及写程序中的各种 bug，印象尤其深的是做特效的时候完全不知道如何对图形进行缩放和旋转。

不过最终做出来的东西还是很妙的，而且感觉自己做的游戏比原版好玩（而且比原版增加了很多功能哈哈哈哈），调试和发现 bug 的过程也非常有趣（其实后来所谓“bug”都是增加游戏趣味性的特效——比如水果在天上飞来飞去之类的）

大作业写完了，非常快乐！收获了一个好玩的游戏！更快乐！