

MusiCube - 手势音游

项目期末报告

杨铭 - 5130379022

李晟 - 5130379017

张云翔 - 5130379012

2016 年 6 月 19 日

目录

1	项目概述	2
2	软件设计	3
2.1	架构设计	3
2.2	实现方法	6
2.3	制作难点	6
3	项目完成度	8
3.1	已实现功能	8
3.2	未来展望	11

1 项目概述

电子游戏，被认为是第九大艺术，在电子产品和网络愈发普及的今天，被越来越多的人所承认和喜爱。而作为游戏中一种独特的类型——音乐游戏（MUG），由最早的街机发展而来，因其与音乐的结合，受到不少玩家的喜爱。

而随着 VR，AR 等概念的兴起，电子游戏的玩法也将出现新的变革。MusiCube 就是一款突破性的音乐游戏，它使用了不同于之前音乐游戏键盘，鼠标或是手台的输入方式，利用 LeapMotion 的手势识别，让玩家可以更加自由地感受音乐游戏的快感。

在 MusiCube 中，所用的音符节拍将都在一个立体的魔方上出现，玩家可以通过手部的动作来操作魔方表面的音符，如戳击一个块，旋转一层。不仅如此，我们还提供图谱编辑器，玩家可以自由编辑自己的图谱，将来如果可能，还可以建立玩家社区，集排名，作图，讨论，自定义皮肤等功能于一体，丰富游戏的玩法，提升游戏的深度，增加玩家的粘性。

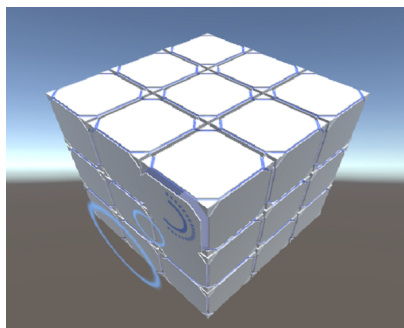


Figure 1:

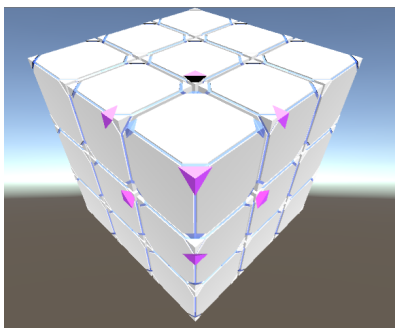


Figure 2:

软件类型 游戏 - 音乐游戏（MUG）

目标用户 PC 用户、游戏玩家

平台支持 PC - Windows7 及以上版本

硬件支持 LeapMotion

2 软件设计

2.1 架构设计

本软件主要采用 Unity3D 制作，通过 LeapMotion 或者键盘鼠标来进行用户交互。软件进入主界面后，可以选择进入选歌游玩界面或者选歌编辑界面。

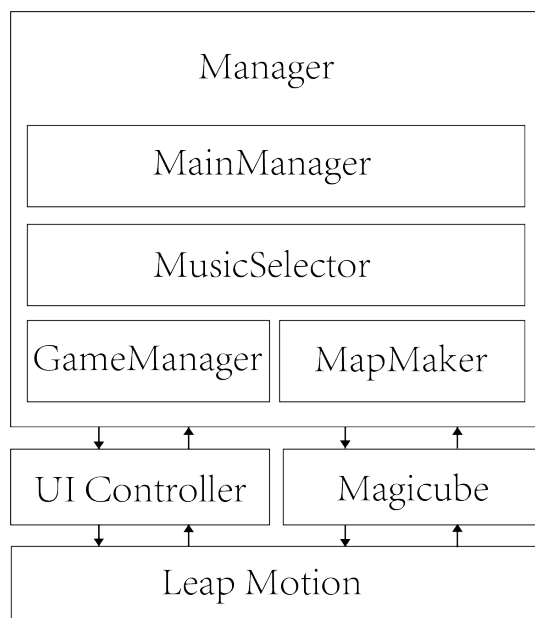


Figure 3:

MainManager 场景管理类，负责管理主菜单和各个场景见的切换

MusicSelector 选歌界面的管理类，包括两个实力，分别管理选歌游玩和选歌编辑

MapMaker 包含一个 Magicube，为编辑图谱提供许多接口，并可以根据 BPM （每分钟节拍数）划分时间轴和制作图谱

GameManager 游戏管理类，负责接收 Magicube 传来的 Note 数据并进行判定

UI Controller 在各个场景都有，负责用户和 UI 的交互

Magicube 游戏主题魔方的主题，管理 27 个方块负责播放游戏动画和音乐

LeapMotion 用户进行输入的设备

流程图

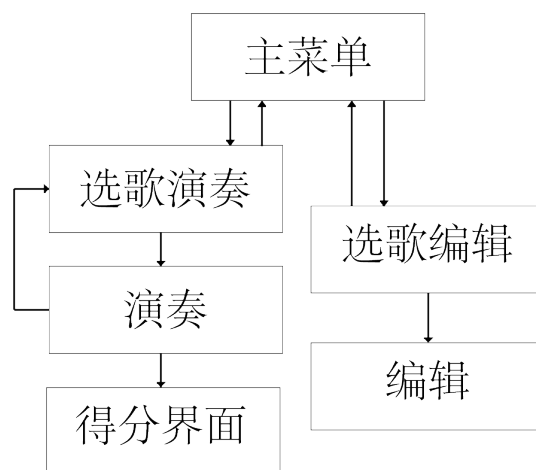


Figure 4:

游戏内有五个界面，按照图中流程可以互相转换。

主菜单

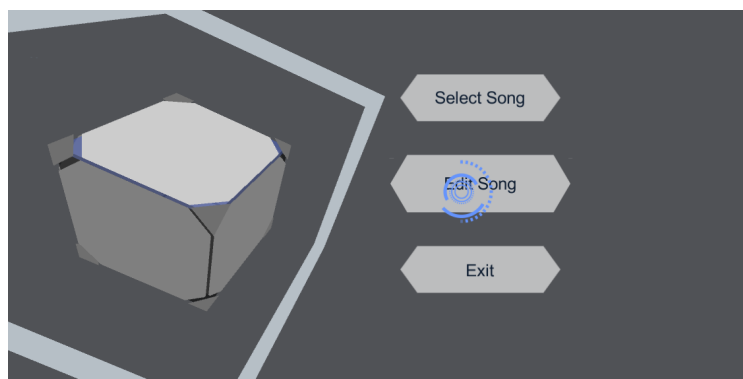


Figure 5:

选歌

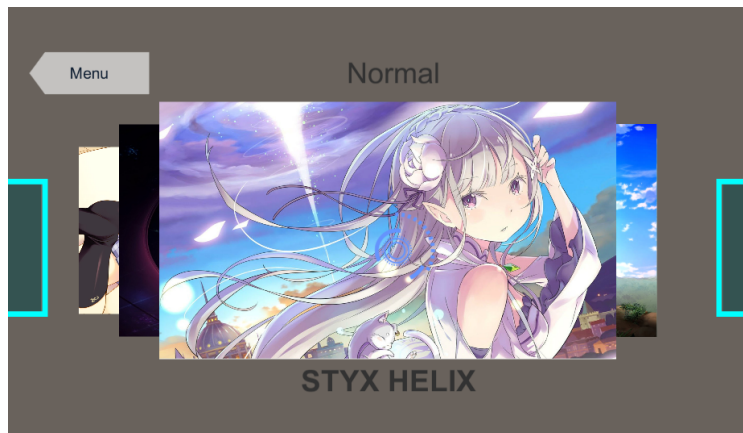


Figure 6:

编辑

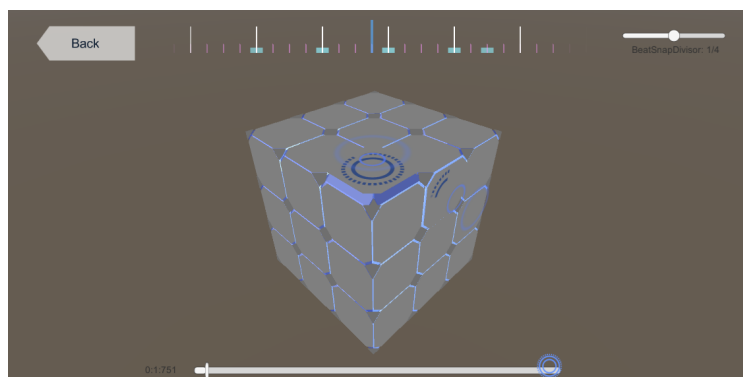


Figure 7:

演奏

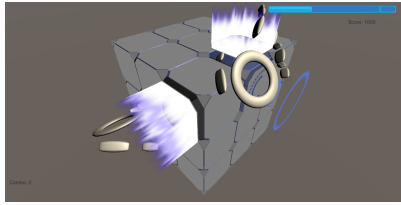


Figure 8:

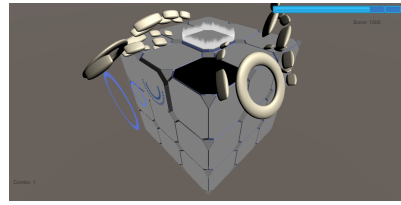


Figure 9:

数据格式 歌曲和图谱文件放在 **Songs**文件夹内，每一个文件夹是一个单独的 **Song**，每一个 **Song**文件夹内有一个 **mci**文件和若干个 **mcb**文件。其中 **mci**文件记录了这首歌曲的一些信息，每个 **mcb**文件则是一个单独的难度，文件头部记录了此难度的信息，接下来是具体的物件信息。

```
diffname:Easy
ar:8
od:6.5
bpm:123
offset:898
0, 0, 2, 3
0, 0, 0, 1
0, 121, 0, 1
0, 242, 0, 1
0, 363, 0, 1
0, 484, 0, 1
0, 605, 0, 1
```

Figure 10:

数据部分有多行四列，每一行是一个单独的物件，对应着游戏内的一次需要用户操作的音符。而四列从左到右分别为：物件类型编号、时间戳（毫秒）、方块或层数编号、方向编号

2.2 实现方法

2.3 制作难点

Unity3D 内置音频支持问题 由于希望能够用户自制图谱，所以游戏需要从外部读取文件（包括音频文件）。但是经过查询文档和测试发现，Unity3D 内置的音频支持在 Windows 平台下，只能从外部读取 **.ogg**格式的文件。

解决方案 一般使用 C# 进行音频读取或处理时都采用 **NAudio**库，但是 Unity3D 支持的 **net**框架与 **NAudio**不兼容。我们使用了一个第三方修改后重新编译的 **NAudio**的 **il** 文件并自己实现了一个使用 **NAudio**接口的音频读取器 **AudioLoader**

现阶段采用的方式是只支持`.ogg`格式的音频，以后准备使用 **NAudio**库来支持大部分格式的音频文件，并将数据加载进 Unity3D 内置的音频支持类中。

界面美观和游戏画面 界面和游戏画面一直是我們头疼的问题，因为没有相关专业的同学帮助，我们花费了不少精力在做动画和美化 UI 上。

LeapMotion 与 UI 的交互 由于 LeapMotion 的 API 特性，不易使用 Unity3D 原生 UI，我们根据自己的需求重新编写了一套 UI 交互的脚本。交互有两种方法：

- 将手指位置映射到 Canvas 上，代替鼠标与其他原生 UI 交互，并且可以通过手势检测完成“点击按钮”的操作
- 使用实体作为 UI，根据碰撞检测实现交互

LeapMotion 精度问题 虽然 LeapMotion 一直吹捧自己的高精度、低延迟，在经过它的驱动 Orion 升级后，LeapMotion 的性能得到了长足的提升。

精度 虽然精度有待提高，但是对于一个 3X3 一共 27 个面的魔方来说，能够在 4 立方英尺工作的 LeapMotion 基本足够了

延迟 一般音乐游戏的最佳判定在正负 20 到 30 毫秒左右，而要使玩家获取较为舒服的游戏体验，游戏帧数可能需要达到 240 或更高。由于不易测量，我们使用官方提供的参数——最大 290 帧，也基本满足我们的需求。

但是在实际测试中，我们发现 LeapMotion 有不少问题：

1. 误识别（如将脸识别为手，中指识别为食指）
2. 无法识别
3. 识别固定关节时会將抖动幅度放大

3 项目完成度

3.1 已实现功能

我们通过一个学期的努力，完成了一个可以通过 LeapMotion 控制游玩和制作图谱的音乐游戏。

代码 项目为 Unity3D 工程，主要资源和代码在项目目录的 Assets 文件夹下，在 Assets 文件夹下：

Scenes 所有游戏场景的场景文件和控制代码

Plugins 第三方动态链接库

UI UI 控件、动画和控制脚本

LeapMotion、LeapMotionVR LeapMotion 及其扩展库

MusiCube 方块弹出、击打以及旋转动画

Resources UI 的动画资源文件

StreamingAssets 歌曲存放

实现了游戏中“魔方”类 **MagiCube** 的大部分逻辑控制代码，并完成了一个控制它的编辑器类 **MapMaker** 以及游戏控制类 **GameManager**。通过这两个类基本可以实现游戏和编辑的两种主要功能。

现阶段完成了 **MagiCube** 的方块弹出动画和玩家触碰的反馈动画和控制。

同时他们还完成了游戏主界面——选歌界面、主菜单和得分界面，并进行了美化。

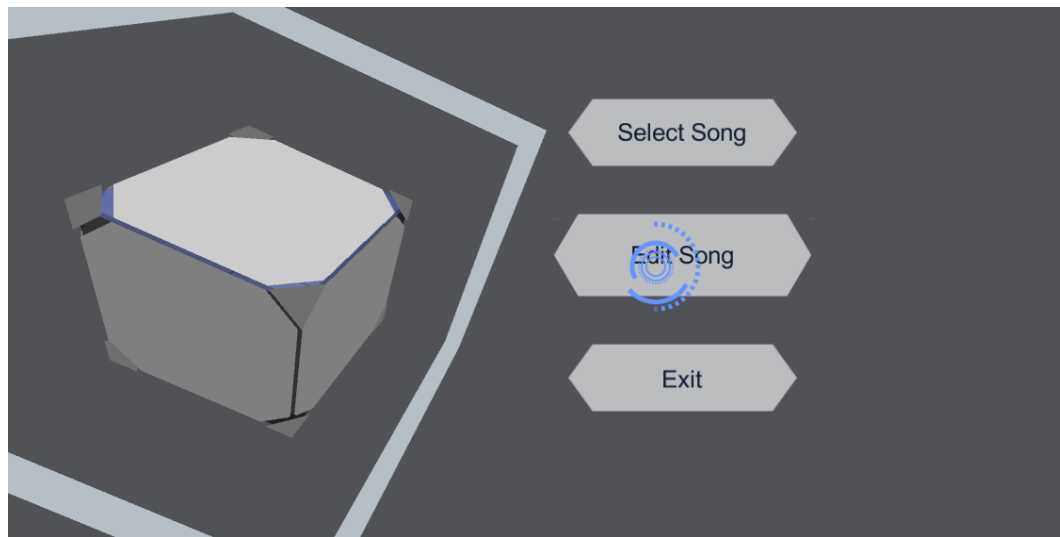


Figure 11:

模型/动画 完成了“魔方”的模型和动画制作，动画包括

1. 编辑时选择
2. 编辑时取消选择
3. 游戏时弹出
4. 游戏时击打 - Perfect 判定
5. 游戏时击打 - Good 判定
6. 游戏时击打 - Normal 判定
7. 游戏时击打 - Miss 判定

动画的实现是通过代码和动画曲线实现的，可以在 Unity3D 内灵活调整动画参数，在编辑器和游戏运行时都可以改变动画效果，并提供了单帧播放的接口。

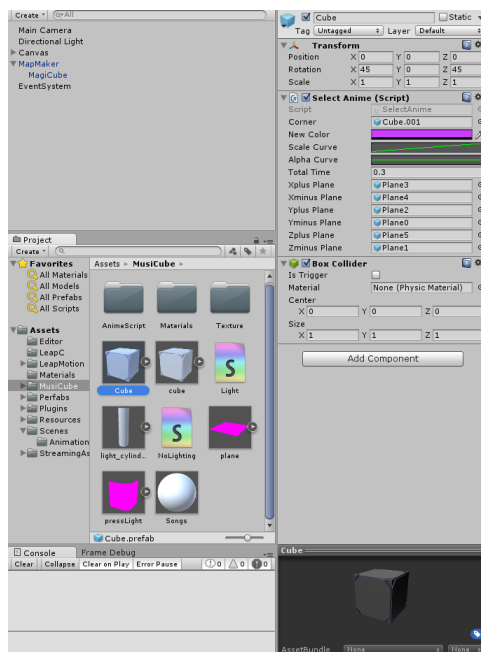


Figure 12:

3.2 未来展望

在一个学期的努力下，我们已经实现了一个基础的能够配合音乐节拍游玩的游戏。对于 Musicube，我们仍有一下的展望

1. 完成游戏内旋转的判定
2. 提供更多游戏方式
3. 增加拖动音频文件开始编辑的功能
4. 打包歌曲及铺面功能
5. 构建玩家社区，实现上传图谱、共享图谱等功能
6. 进一步美化 UI