
“乐在掌中” 电子乐团项目需求分析报告

团队：JavaMusician

组长：王浩

组员：林迪 曾韶彬 吴卓豪 段琳 沈元佶

目录

1. 概述	1
1.1 产品描述	1
1.2 用户群体	1
1.3 产品功能和特性.....	1
1.3.1 主要功能.....	2
1.3.2 附加功能.....	2
2. 项目开发工具及环境.....	3
2.1 开发语言	3
2.2 手机端开发平台.....	3
2.3 数据库管理系统.....	3
2.4 版本控制系统.....	4
2.5 持续集成环境.....	4
2.6 自动构建工具.....	5
2.7 自动测试工具.....	5
3. 团队组织	5
3.1 团队成员角色.....	6
3.2 团队沟通方式.....	7
4. 用户帮助文档	8
5. 项目计划	9
6. 项目风险应对措施	10
7. 用例	11
8. 用户界面设计	21

1. 概述

1.1 产品描述

“乐在掌中”电子乐团是一款基于手机平台并且集成多种电子乐器的应用软件。它使得用户可以利用这些集成的电子乐器进行音乐演奏、谱曲等音乐创作和娱乐活动。同时，“乐在掌中”电子乐团还可以录制用户创作和演奏的音乐，以方便用户日后在本应用软件或其他播放器上回放这些音乐。“乐在掌中”电子乐团作为一款多元化的音乐应用软件，它所包含的功能并不仅仅局限于音乐演奏功能。它还会包含其它一系列精彩且有趣的功能，例如自动谱曲、音乐小游戏等。

1.2 用户群体

“乐在掌中”电子乐团是一款适用性较广的音乐应用软件。它主要覆盖的用户群体有：

- 专业音乐制作人或歌手；
- 广大业余音乐发烧友；
- 音乐专业学生或教师；
- 对乐理知识知之甚少甚至一窍不通，但是对音乐充满热情，或者仅仅将音乐作为一种休闲娱乐的群体。

1.3 产品功能和特性

“乐在掌中”电子乐团包括以下四个主要功能以及两个附加功能。

1.3.1 主要功能

“乐在掌中”电子乐团的主要功能包括：

- 音乐制作及演奏功能。该应用集成了多种电子乐器，例如电子琴、吉他以及节拍器等；
- 音频录制以及播放功能。利用该功能，可以同步地将用户制作的乐曲录制下来，然后回放；
- 客户端之间联合演奏。用户可以利用多台手机或者 PC 机中的客户端，模拟多种乐器，进行联合演奏；
- 自动谱曲功能。用户可以在开始的时候输入一些简单的关于乐曲的参数，例如乐器种类，节拍等等，自动谱曲功能就会更具这些条件自动生成一首乐曲。

1.3.2 附加功能

除了以上所描述的主要功能外，我们还希望在这个基础上加上一些附加功能来拓展我们的产品。这些附加功能是：

- 语音谱曲功能。用户通过语言输入工具进行录音，程序自动为用户录入的音频进行解析并转换为乐曲；
- 音乐游戏比赛功能。用户可以选取自己制作的或者其他的曲目，通过网络连接，与其他安装了该应用的用户进行音乐演奏比赛。

2. 项目开发工具及环境

所有用于“乐在掌中”电子乐团的开发工具和环境以及选择这些工具的依据和优点都会在这一节中作较为详细的描述。

2.1 开发语言

我们所选用的开发语言是 Java。由于我们期望“乐在掌中”电子乐团是一款跨平台性较好（在手机和 PC 上、在不同的操作系统中均可以运行）的应用软件，而 Java 本身就是一种跨平台性极好的开发语言，而且使用范围广。Java 本身提供了多种强大的组件和库，为项目开发带来了极大的便利。因此，我们选用 Java 作为开发语言。

2.2 手机端开发平台

手机端的开发平台我们选定为 Android。由于 Android 使用的是 Java 开发语言并且是开源的，使得开发人员易上手。Android 使用的是虚拟架构，使得它的可移植性强，适合在不同型号的手机运行。

2.3 数据库管理系统

“乐在掌中”电子乐团所使用的数据库管理系统会因应不同的平台而不同，也就是说，手机端所采用的数据库管理系统与 PC 端所采用的数据库管理系统是不同的。

- 手机端的数据库管理系统

手机端所采用的数据库管理系统是 SQLite。SQLite 是轻量级

数据库，速度快而且效率高，适合在手机这中计算和存储资源有限的嵌入式设备上使用。

- **PC 端的数据库管理系统**

PC 端所采用的数据库管理系统是 MySQL。由于我们使用的开发语言是 Java，而且 MySQL 这套数据库管理系统的效率也高，适合在类似我们项目的中小型软件使用。

2.4 版本控制系统

我们在本次项目中所采用的版本控制系统是 Git。我们在 github.com 上创建我们的代码仓库，使用 Git 来远程控制的代码仓库的代码版本。由于我们可以结合 Git 和 github.com 来进行版本控制，这种分布式的代码版本管理模式使得所有开发人员都可以随时随地提交和下载代码。Git 本身所提供的其它功能，例如回滚、提交报错等，使得开发人员可以及时知道其编写的代码是否与现存的代码版本是否有冲突，从而提高开发效率和减少因版本问题所带来的风险。

2.5 持续集成环境

我们所采用的持续集成环境是 cruisecontrol.rb。我们使用 cruisecontrol.rb 来监视位于 github.com 上的代码库。通过 cruisecontrol.rb 持续地将远程代码库中的代码以及本地开发人员所编写的代码版本集成起来，从而使得开发人员可以随时了解到自己编写的代码是否可以集成到工程中。这样，我们便可以保证随时获得一个

可运行的代码版本。

2.6 自动构建工具

在本次项目中，我们使用 Ant 来作为自动构建工具。Ant 是一种基于 Java 的自动构建工具，既然我们选择 Java 来作为我们的开发语言，那么我们没有理由不选择 Ant 来作为我们项目的自动构建工具。通过使用 Ant，我们只需要一个简单的 build 脚本就可以完成自动构建工作。由于我们还是用了 `cruisecontrol.rb` 这种持续集成环境，所以需要一种自动构建工具来为持续集成提供支持。因此，最好的选择非 Ant 莫属。

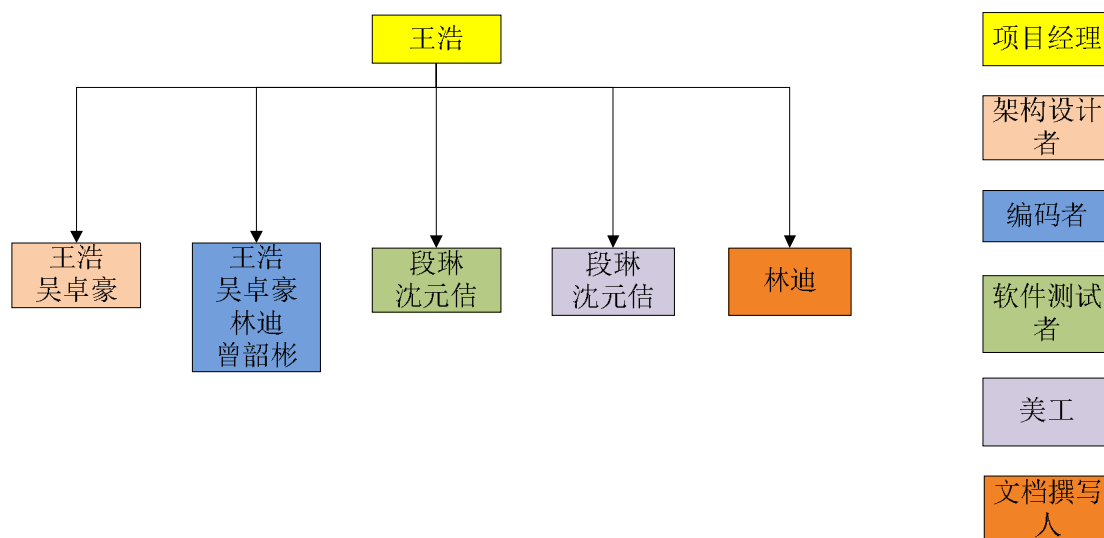
2.7 自动测试工具

我们使用 JUnit 作为我们的自动测试工具。选择使用 JUnit，是因为 JUnit 本身就是用于 Java 的代码测试框架，只要编写好测试脚本，便可以完全自动化进行测试，节省人工测试时间。而且，JUnit 可以配合 Ant 使用，在进行持续构建的过程中查看测试的结果，并以测试的结果作为构建是否成功的标准之一。

3. 团队组织

这个部分主要包括两项内容：团队成员的角色以及团队成员之间的交流方式。

3.1 团队成员角色



图片 1 团队成员角色

- 项目经理：由于王浩是整个团队唯一一位多次参与过学校、学院的项目的成员，具有较为丰富的项目管理经验。所以王浩适合作为本次项目的项目经理。
- 架构设计者：整个项目的最初设想是来源于王浩的，并且在一开始，整个项目的架构的初步构建是由王浩以及吴卓豪完成的。他们对整个项目架构的把握以及熟悉程度要超过团队的其他人，所以他们是作为架构设计者。
- 编码者：王浩、吴卓豪、林迪以及曾韶彬是项目组中编码能力较高的，而且整个项目的编码工作较为繁重，需要人数也自然要比其它工作多，所以有这四位成员负责整个项目的编码工作。
- 软件测试者：段琳和沈元佶两名成员从项目的初期便从事项目的所有测试工作。在整个项目的推进过程中，两名成员也是一直承担测试工作，所以她们是最为熟悉整个项目的测试流程，

因此有她们承担测试者的工作

- 美工：段琳和沈元佶两名成员是整个项目中美工较为出色的成员，而且熟悉 Photoshop 等多种图像处理软件。所以她们也会承担美工的工作。
- 文档撰写人：林迪是项目计划书主要撰写人，而且随着项目的进行，也一直从事项目各种文档的整合与编辑，比较熟悉项目各项文档的撰写要求。所以由他来负责项目文档的撰写。

3.2 团队沟通方式

团队成员之间的分歧的解决、项目有关问题的决定等目的主要是通过团队之间的沟通达到。

- 每周一次的全员会议。在这个每周一次的，项目的所有成员都会聚集起来，汇报一周内的工作进度以及遇到的困难。项目组的所有成员可以就相关问题发表意见，会议会有专门的记录人将会议的内容记录下来共会后各成员参考。会议全体成员会争取就一些项目上共同的分歧以及问题达成共识。如果一些问题没有达成共识，那么会后相关的成员可以自行组织小型会议来进行讨论；
- 成员之间的小型会议。成员之间的小型会议主要面向团队中负责同一项工作的成员，会议的主要内容主要针对相关工作中具体的问题。由于每周会议时间有限，而且不同成员负责的工作不同，所以难免有一些工作上的分歧以及困难不能再每周会议

上达成共识。我们考虑到负责同一项工作的成员平时是在一起工作的，那么会议安排的时间也可以较为灵活。面对面的沟通较为方便而且直接，解决问题的效率也会比每周会议高。因此，各项工作的分歧以及问题主要是通过成员之间的小型会议得到解决。

- Github 上的 issue tracker 以及 wiki。项目经理会在每周的会议后，将各成员下一周的任务上传到 issue tracker，以便团队中的每个成员可以随时清楚自己的职责以及项目的进度。同时，issue tracker 允许成员对彼此之间的工作进行评论，有助于工作水平的提高。wiki 为团队的成员提供一个网上交流的平台，成员可以利用 wiki 发布自己在工作中遇到的问题，也可以解答其他成员提出的问题，这样就算有个别成员因为有其他事情不得不在其他地方工作，成员之间的互相交流也不会被隔断。

4. 用户帮助文档

在本项目中，我们将会提供两种用户帮助文档，来帮助用户了解和正确使用我们的产品。

- 打印版的产品说明书。我们将会在打印版的说明书中列出所有的产品功能和特性，并且会对每个功能的使用方法以及注意事项做详细的说明；
- 集成在应用软件中的电子版帮助文档。用户可以在本应用软件

中找到电子版的帮助文档，索引到相应的帮助条目中了解软件的各项功能的使用方法以及注意事项。

5. 项目计划

Step ID	Assignee	Task	Effort Expected (days)
1	WangHao,LinDi	Project proposal	2
		Proposal	2
2		Kick off the project	14
	LinDi	Set up a team box for a CruiseControl	2
	LinDi	Learn the configuration of a CruiseControl	6
	LinDi	Deploy a CruiseControl server for the "乐在掌中" project on the virtual box	2
	LinDi	Initial a project on the Github	3
	WangHao	Add product backlog, sprint backlog, SRS., SDS on the wiki of the project	1
3		Design & Develop	112
	WangHao	Create a scaffold application	2
	WangHao	Refactor the scaffold code	2
	WangHao	General framework design	4
	WangHao	Android framework design	10
	WangHao	Android UI design	6
	WangHao	Android framework coding	3
	WangHao	Android Instrument coding	10
	WangHao	Android Record coding	10
	WangHao	Android Game coding	10
	WangHao,ShenYuanji,.		
	DuanLin	PC UI design	4
	LinDi	PC UI framework	3
	ZengShaobin,WuZhuohao		
	ZengShaobin,WuZhuohao	PC Instrument	15
	ZengShaobin,WuZhuohao		
	ZengShaobin,WuZhuohao	PC Game design	13
	LinDi	Auto create music	20

图表 1 项目计划

6. 项目风险应对措施

考虑到项目进度可能会落后于原定计划，所以我们特制订一下三项措施以应对这种情况：

- 抽调暂时没有工作或者工作量较轻的人员参与到这项工作中来。由于我们在进行每项工作时，都要求每位组员充分了解将要进行的工作，然后再进行分工，所以发生这种人员调配时，新参与到工作中的人员很快便可以很快跟上目前工作的进度而不需要额外太多的时间来适应；
- 召开一次会议，对现在进行的所有工作优先级进行排序。若落后于计划进度的工作所处于的优先级较高，则我们需要延缓或暂停优先级较低的工作，抽调部分人员来对落后与进度的高优先级工作进行攻关。待优先级较高的工作完成后，抽调的人员可以返回其原来的工作。如果落后于进度的工作优先级较低，则暂时先不做任何改动；
- 有时候工作落后于进度，很有可能是因为这项工作的难度太大，造成开发人员的工作进展缓慢。此时，参与这项工作的所有开发人员以及设计人员需要进行一次会议，集中讨论是否前期的可行性分析不够准确，这项工作的架构设计是否合理，所采用的技术是否效率太低，是否需要采用为我们熟知而且能提高工作效率和降低工作难度的技术等。如果会议得出的结论是，该项工作的难度确实太大，那么我们就需要考虑暂时搁置该项工作。如果会议找到了完成该项工作更好的方法，调整并完善了

该项工作的架构，那么就按照优化的方法继续进行工作。

7. 用例

在这个部分，我们将针对之前提出的项目的主要功能，通过用例(use cases)的方式来进行描述，以求用户对我们的产品有全面的了解。

- 用例 1：使用手机端演奏系统模拟吉他演奏

主执行者：音乐演奏者（MP）

范围：演奏系统手机端（PSM）

层次：用户等级

前置条件：已打开手机端演奏系统，选择吉他演奏

最低保证：尝试载入吉他界面

成功保证：进入吉他弹奏界面，琴弦对应的琴音为没有任何和弦时的默认关系。

主成功场景：

- 1、MP 点击或触动对应的画面中的吉他琴弦。
- 2、PSM 获得点击位置。
- 3、PSM 通过点击位置找到对应的琴音。
- 4、PSM 播放对应的琴音。
- 5、MP 点击和弦按钮。
- 6、PSM 得到对应和弦。
- 7、PSM 判断是否是上一次选定的和弦。

8、PSM 根据选定的和弦改变琴弦与琴音的对应关系。

9、PSM 将对应和弦图标显示为选定（已按下）。

扩展：

5a、MP 点击和弦按钮时，有和弦按钮处于选中状态。

5a1、PSM 将此前已选中的和弦按钮设为未选中（弹起），
并继续步骤 6。

7a、PSM 认为和上次选择和弦一致。

7a1、PSM 将琴音和琴弦设置为默认对应关系。

- 用例 2：使用手机端演奏系统模拟节拍器

主执行者：音乐演奏者（MP）

范围：演奏系统手机端（PSM）

层次：用户等级

前置条件：已打开手机端演奏系统，选择节拍器

最低保证：尝试载入节拍器界面

成功保证：进入节拍器界面。

主成功场景：

1. MP 甩动手机。
2. PSM 获得甩动幅度。
3. PSM 通过甩动幅度判断是否发音。
4. PSM 播放对应的节拍音。

扩展：

5a、PSM 通过甩动幅度判断不发音。

5a1、继续步骤 1。

- 用例 3：使用 PC 端演奏系统演奏钢琴音乐

主执行者：音乐演奏者（MP）

范围：演奏系统 PC 端（PSPC）

层次：用户等级

前置条件：已打开 PC 端演奏系统，选择钢琴演奏

最低保证：尝试载入钢琴界面

成功保证：进入钢琴弹奏界面，显示钢琴界面琴键。

主成功场景：

- 1、MP 按下键盘按键。
- 2、PSPC 获得按下的按键。
- 3、PSPC 找出对应钢琴按键。
- 4、PSPC 将界面上的琴键显示为按下。
- 5、PSPC 通过对应关系找出对应琴音。
- 6、PSPC 发出对应琴音。
- 7、MP 放开键盘按键。
- 8、PSPC 将界面上的琴键显示为弹起。

扩展：

1a、MP 点击没有对应琴音的键盘按键。

1a1、PSPC 将不做出任何反应。

- 用例 4：乐曲录制

主执行者：音乐演奏者（MP）

范围：演奏系统（PS）

层次：用户

前置条件：进入某项乐器演奏系统

最低保证：尝试录入演奏

成功保证：显示录入系统界面

主成功场景：

- 1、MP 选择录制。
- 2、PS 开启录制功能。
- 3、MP 进行乐曲演奏。
- 4、PS 识别每一次演奏输入的内容。
- 5、PS 记录每次输入的内容与时间。

扩展：

4a、MP 再次选择演奏某项乐器。

4a1、PS 将记录清空，跳回步骤 1。

- 用例 5：语音谱曲

主执行者：音乐演奏者（MP）

范围：演奏系统（PS）

层次：用户

前置条件：进入演奏系统

最低保证：尝试载入语音谱曲页面

成功保证：载入语音谱曲页面

主成功场景：

- 1、MP 对音频输入系统演奏真实音乐。
- 2、PS 将音频信号解码。
- 3、PS 通过频率获得音阶音调。
- 4、PS 显示对应的乐谱。
- 5、MP 选择转换乐器。
- 6、PS 将记录的音阶音调传递给选择的乐器系统的录制部分的记录。
- 7、PS 进入选择的乐器的录制演奏部分（用例 3）。

- 用例 6：自动谱曲

主执行者：音乐演奏者（MP）

范围：演奏系统（PS）

层次：用户

前置条件：进入演奏系统

最低保证：尝试载入自动谱曲页面

成功保证：载入自动谱曲页面

主成功场景：

- 1、MP 输入参数。

2、PS 对参数进行分析。

3、PS 根据参数生成新的乐曲。

- 用例 7：乐曲回放

主执行者：音乐演奏者（MP）

范围：演奏系统（PS）

层次：用户

前置条件：进入某项乐器演奏系统

最低保证：尝试回放乐曲

成功保证：将读取并回放演奏内容

主成功场景：

1. MP 选择乐曲回放。
2. PS 读取记录。
3. PS 按记录重复内容。

- 用例 8：手机端音乐游戏 1

主执行者：游戏玩家（GP）

范围：游戏系统（GS）

层次：用户

前置条件：进入游戏系统

最低保证：尝试载入音乐游戏页面

成功保证：进入音乐游戏页面

主成功场景：

- 1、 GP 选择进入游戏系统。
- 2、 GS 开始计时。
- 3、 GP 选择背景音乐。
- 4、 GS 播放背景音乐。
- 5、 GS 根据背景音乐节拍产生音符。
- 6、 GP 触摸音符。
- 7、 GS 根据被触摸的音符记录得分。
- 8、 GS 判断游戏时间是否完毕。
- 9、 GS 将游戏页面切换到游戏记录页面。

扩展：

3a、 GP 选择背景音乐时音乐列表为空。

3a1、 GS 将产生音符方式改为随机产生音符，继续步骤 6。

- 用例 9：手机端音乐游戏 2

主执行者：游戏玩家（GP）

范围：游戏系统（GS）

层次：用户

前置条件：进入游戏系统

最低保证：尝试载入音乐游戏页面

成功保证：进入音乐游戏页面

主成功场景：

- 1、 GP 选择进入游戏系统。
- 2、 GP 选择背景音乐。
- 3、 GS 播放背景音乐。
- 4、 GP 猜测得分。
- 5、 GS 在屏幕上产生音符。
- 6、 GP 触摸屏幕。
- 7、 GS 根据被触摸的位置产生波纹。
- 8、 GS 记录被波纹触及的音符，并记录得分。
- 9、 GS 将游戏页面切换到游戏记录页面。

扩展：

3a、 GP 选择背景音乐时音乐列表为空。

3a1、 GS 将产生音符方式改为随机产生音符，继续步骤 6。

- 用例 10：手机端音乐游戏 3

主执行者：游戏玩家（GP）

范围：游戏系统（GS）

层次：用户

前置条件：进入游戏系统

最低保证：尝试载入音乐游戏页面

成功保证：进入音乐游戏页面

主成功场景：

- 1、 GP 选择进入游戏系统。

- 2、 GS 开始计时。
- 3、 GP 选择背景音乐。
- 4、 GS 播放背景音乐。
- 5、 GS 根据背景音乐节拍产生音符。
- 6、 GP 触摸音符。
- 7、 GS 根据被触摸的音符记录得分。
- 8、 GS 判断游戏时间是否完毕。
- 9、 GS 将游戏页面切换到游戏记录页面。

扩展：

3a、 GP 选择背景音乐时音乐列表为空。

3a1、 GS 将产生音符方式改为随机产生音符，继续步骤 6。

- 用例 11：手机端音乐游戏 4

主执行者：游戏玩家（GP）

范围：游戏系统（GS）

层次：用户

前置条件：进入游戏系统

最低保证：尝试载入音乐游戏页面

成功保证：进入音乐游戏页面

主成功场景：

- 1、 GP 选择进入游戏系统。
- 2、 GP 选择背景音乐。

- 3、 GS 播放背景音乐。
- 4、 GS 在屏幕上产生音符。
- 5、 GP 触摸屏幕。
- 6、 GS 根据被触摸的位置移动弹球。
- 7、 GS 开始计时。
- 8、 当弹球移动到指定区域时，GS 记录游戏所用时间。
- 9、 GS 将游戏页面切换到游戏记录页面。

扩展：

3a、GP 选择背景音乐时音乐列表为空。

3a1、GS 将产生音符方式改为随机产生音符，继续步骤 6。

8a、弹球无法在指定时间内移动到指定区域。

8a1、不记录游戏所用时间，继续步骤 9。

- 用例 1（使用手机端演奏系统模拟吉他演奏）描述

主成功场景：

音乐演奏者打开手机端演奏系统，选择吉他演奏。演奏系统手机端载入吉他界面。音乐演奏者进入吉他弹奏界面，点击或触动对应的画面中的吉他琴弦，演奏系统手机端获得点击位置并通过点击位置找到对应的琴音，然后播放对应的琴音。音乐演奏者点击和弦按钮，演奏系统手机端得到对应和弦并且判断是否是上一次选定的和弦，根据选定的和弦改变琴弦与琴音的对应关系，将对应和弦图标显示为选定（已按下）。

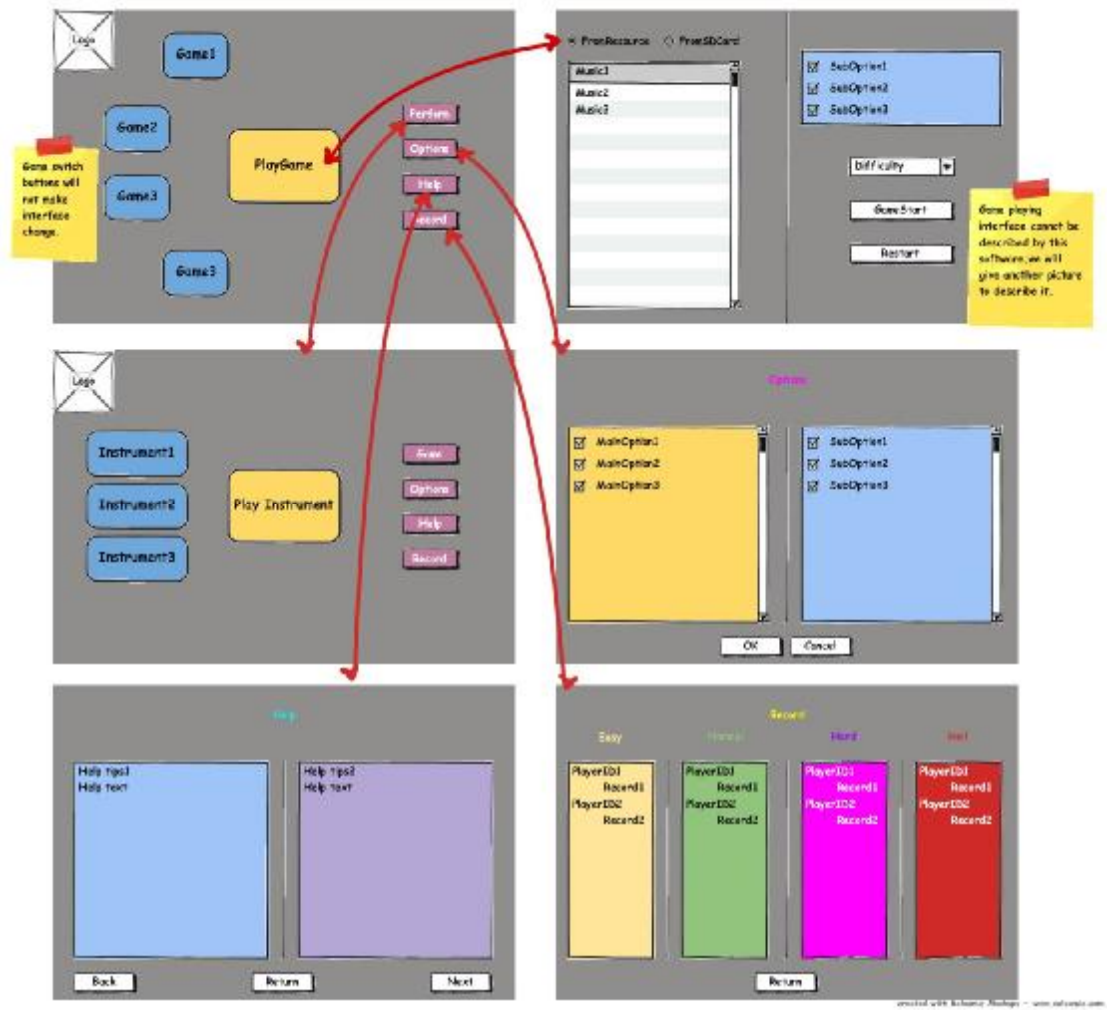
扩展：

当音乐演奏者点击和弦按钮时，有和弦按钮处于选中状态时，演奏系统手机端将此前已选中的和弦按钮设为未选中（弹起）；当演奏系统手机端认为和上次选择和弦一致时，演奏系统手机端将琴音和琴弦设置为默认对应关系。

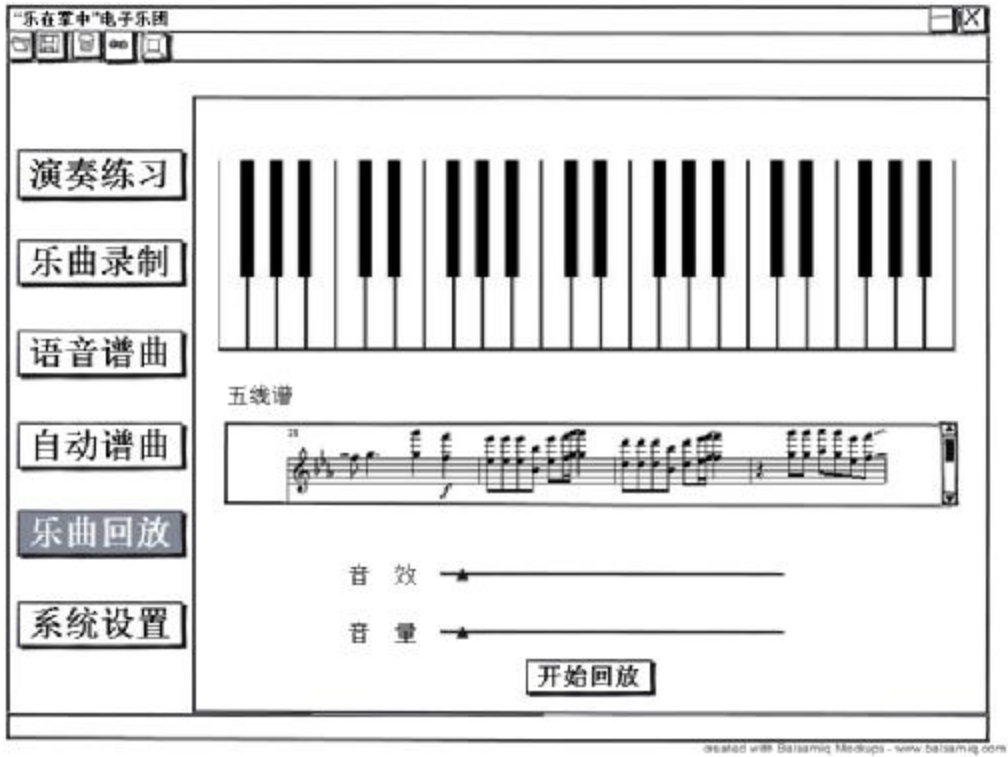
8. 用户界面设计

在用户界面设计这个部分，我们将会展示手机端以及 PC 端的各个功能的用户界面（包括我们的设计草图），使用户对我们的产品的用户界面有大致地了解。

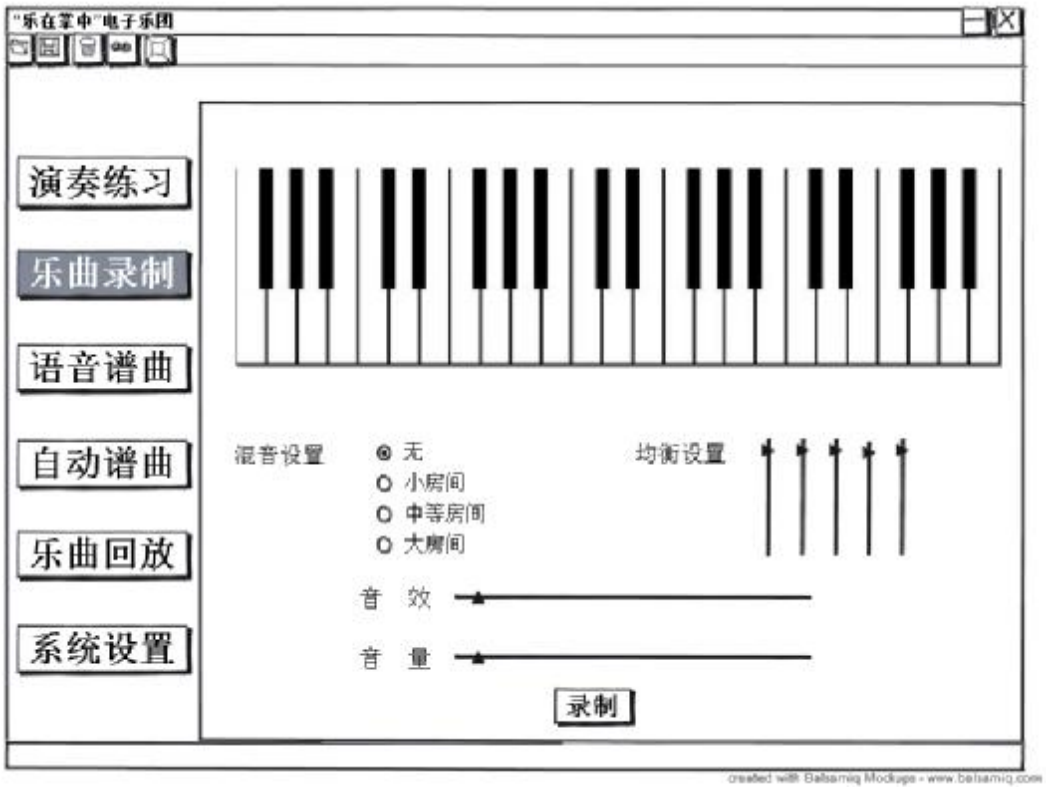
- 设计草图



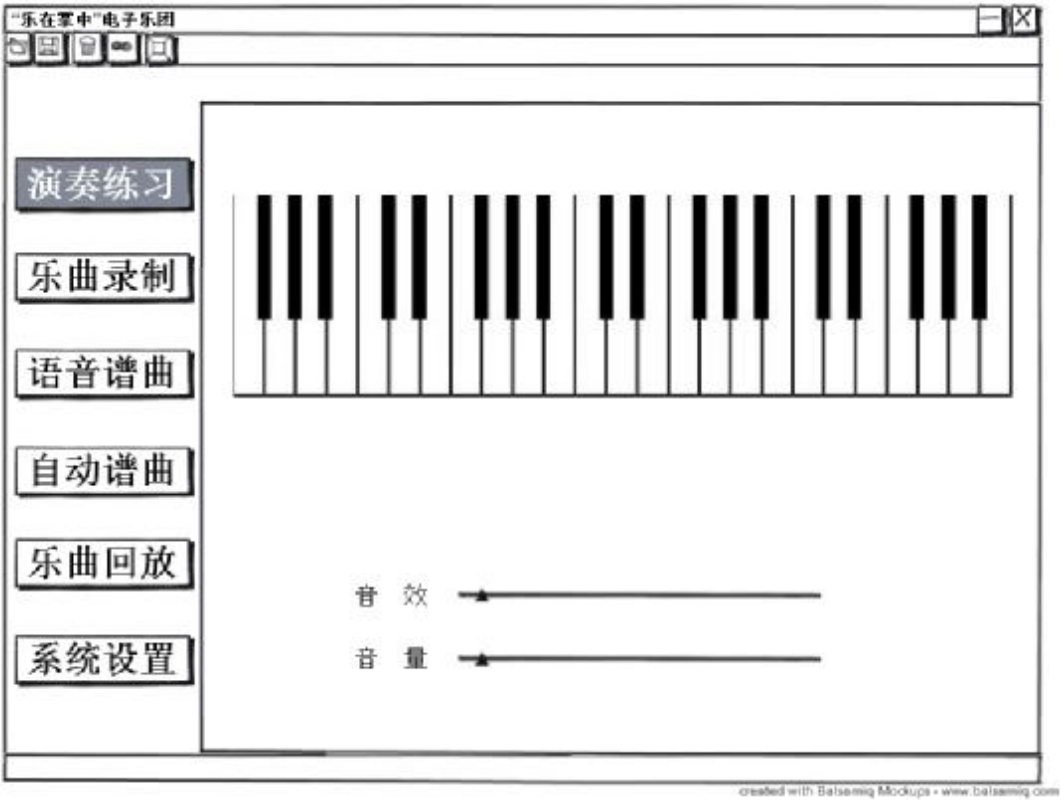
图片 2 整体设计草图



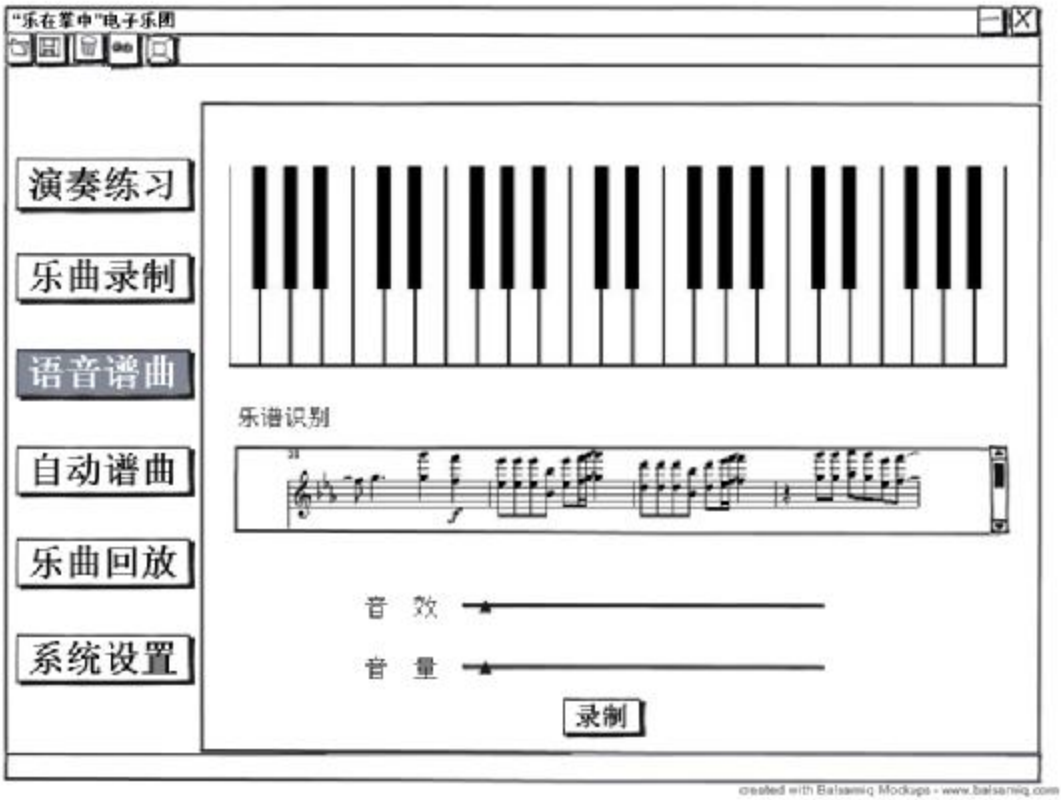
图片 3 乐曲回放功能设计草图



图片 4 乐曲录制设计草图



图片 5 演奏练习设计草图

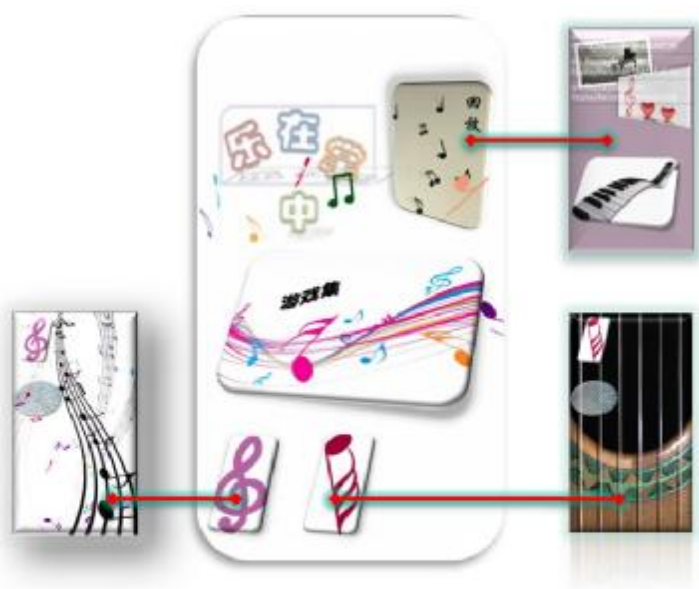


图片 6 语音谱曲设计草图



图片 7 自动谱曲设计草图

● 手机端用户界面



图片 8 手机端早期界面



图片 9 手机端已完成的界面

• PC 端多风格界面图



图片 10 PC 端已完成界面