小学生养成记

系 统 分 析 设 计

|  |  |  |
| --- | --- | --- |
| 文件状态：  [√] 草稿  [ ] 正式发布  [ ] 正在修改 | 文件标识： | Company-Project-RD-UR |
| 当前版本： | X.Y |
| 作 者： |  |
| 完成日期： | Year-Month-Day  南京师范大学 教育科学学院  2016年12月 |

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[系 统 分 析 设 计 1](#_Toc470643267)

[1 系统概述 4](#_Toc470643268)

[1.1 系统简介 4](#_Toc470643269)

[1.2 系统特色 4](#_Toc470643270)

[1.3 主要功能 4](#_Toc470643271)

[2 系统需求 5](#_Toc470643272)

[3 系统操作流程 6](#_Toc470643273)

[4 项目模拟（Mockplus） 6](#_Toc470643274)

[附录：参考文献 7](#_Toc470643275)

# 1 系统概述

## 1.1 系统简介

小学生养成记APP是面向即将进入小学的幼儿设计的辅助幼小衔接工作的教育类APP。

本APP的开发基于儿童学习的兴趣与心理特质，将儿童发展理论与游戏相结合，关注重心为幼儿进入小学时对新环境的适应能力，如上课纪律、作息习惯、自理能力、尊重师长等，而非学科学习能力，让儿童在游戏中享受幼儿园与小学教育的双向互动，通过累积记录的方式，形成儿童发展的阶梯。相关内容的呈现方式为分类的简短的视频动画与简单逻辑选择的结合。

## 1.2 系统特色

Android技术与数据库技术为用户提供了优质的体验，使孩子通过家长的手机就可以快速了解小学学习生活与幼儿园阶段的不同。可爱的人物、明亮的界面，吸引小孩子的注意力；同时，简单的操作，使孩子也能自主使用本APP。将丰富多彩的学校情景设计带入游戏，符合儿童角色游戏心理，让幼儿为达成虚拟毕业的目标而逐渐养成良好的学习品质，为小学阶段的学习做好准备。具体来说，小学生养成记APP的特色表现在以下几个方面：

1. 首次登录的欢迎动画：以可爱人物邀请用户开始使用APP，简单引入，告知学习内容，激发学习兴趣。
2. 主界面：地图式界面，分别设计了教学楼、医务楼、体育馆与食堂，分别放置不同类型的视频动画；另有成就楼，记录学习进度，可通过微信、微博及QQ进行分享。
3. 视频动画：使用帧动画与录制好的人声音频，配合标注拼音的文字，演示小学的学习生活片段，同时结合选择题，引导用户养成正确的习惯。
4. 成就奖励机制：每观看完一个动画，可点亮成就楼的一颗星星，可以随时将星星点亮进程分享到社交平台。

## 1.3 主要功能

1.3.1 视频动画观看

点击主页地图上的不同建筑物进入视频动画目录界面，点击选择观看。

1.3.2.行为选择

观看视频时，附带对应选择题，点击选项，呈现结果，若正确则继续进行，若不正确则在鼓励并且观看正确做法之后再次出现题目，重新选择。

1.3.3学习进度记录

每观看完一个短视频并正确完成选择题后，可点亮成就楼中的一颗星星，点亮所有星星则可虚拟“小学毕业”，观看特别毕业动画。

1.3.4. 学习成果分享

在成就楼中，可随时点击分享，通过微信、微博或QQ三种社交软件分享至自己的平台。

# 2 系统需求

|  |  |  |  |
| --- | --- | --- | --- |
| 系统组织与业务划分 | | | |
|  | 组织机构 | 主要业务 | 业务要点 |
| 2.1 | 教学楼、医务楼、体育场、餐饮楼 | 2.1.1 动画播放 | 可以观看视频学习正确行为习惯 |
| 2.1.2 行为选择 | 观看动画的过程中有题目让学生进行选择 |
| 2.1.5 奖励机制 | 观看完一个视频动画后即可获得一颗星星 |
| 2.2 | 成就楼 | 2.2.1 查看其它楼的学习进度 | 有进度条可以查看视频观看情况，同时查询星星获得情况 |

# 3 系统操作流程

# 4 项目模拟（Mockplus）



主界面

****

# 代码详解

## 5.1 帧动画——animation-list

新建一个帧动画：在res文件夹中新建一个Android Resource File，并选择Resource Type为Animator，以下是App中的代码：

**//帧动画声明**

private ImageView animationIV;

private AnimationDrawable animationDrawable;

**//帧动画播放****animationIV** = (ImageView) findViewById(R.id.***animationIV***);  
**animationIV**.setImageResource(R.drawable.***xueshengdonghua***);  
animationDrawable = (AnimationDrawable) **animationIV**.getDrawable();  
animationDrawable.start();  
  
**mediaPlayer** = MediaPlayer.*create*(**this**, R.raw.***hello***);  
**mediaPlayer**.start();

**帧动画的xml中利用animation-list控制每张图片停留时间，从而达到逐帧动画效果**

<**item android:drawable="@drawable/doorone" android:duration="150"**></**item**>

## 5.2 界面自动跳转（可以规定跳转时间）

**final** Intent intent = **new** Intent();  
Timer timer = **new** Timer();  
TimerTask task = **new** TimerTask() {  
 @Override  
 **public void** run() {  
 intent.setClass(MainActivity.**this**, Mainmap.**class**);  
 MainActivity.**this**.startActivity(intent);  
 }  
};  
timer.schedule(task, 1000 \* 7);

## 5.3 单击按钮返回主界面

**toHome**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 intent.setClass(stuVideo.**this**, Mainmap.**class**);  
 stuVideo.**this**.startActivity(intent);  
 }

## 5.4 单击实现弹窗效果——Dialog

**//对某具体事件设置监听**

**private void** initListenner() {  
 **hospital**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 showPopForShare();  
 }  
 });

**//调用弹出事件**

**protected void** showPopForShare() {  
 View view = LayoutInflater.*from*(**this**).inflate(R.layout.***pop\_share***, **null**);  
 **final** Dialog dialog = **new** Dialog(achieve.**this**, R.style.***transparentFrameWindowStyle***);  
 dialog.setContentView(view, **new** ActionBar.LayoutParams(ViewGroup.LayoutParams.***MATCH\_PARENT***, ViewGroup.LayoutParams.***WRAP\_CONTENT***));  
  
 Window window = dialog.getWindow();  
 window.setGravity(Gravity.***CENTER***);

**//设置显示动画**window.setWindowAnimations(R.style.***main\_menu\_animstyle***);  
 WindowManager.LayoutParams w1 = window.getAttributes();  
**//保证按钮可以水平满屏**  
 w1.**width**ViewGroup.LayoutParams.***MATCH\_PARENT***;  
 w1.**height** = ViewGroup.LayoutParams.***WRAP\_CONTENT***;  
 window.setAttributes(w1);  
**//设置显示位置**dialog.onWindowAttributesChanged(w1);  
**//设置点击外围解散**dialog.setCanceledOnTouchOutside(**true**);  
 dialog.show();

## 5.5 创建多线程消息的函数：postDelayed

**//创建一个Handler对象**

Handler handler = **new** Handler();

**//创建一个Runnable对象**

handler.postDelayed(**new** Runnable() {  
 **public void** run() {**animationDrawable**.stop();  
 showBottomPop();

**//使用PostDelayed方法，两秒后调用此Runnable对象**

handler.postDelayed(runnable, 2000);

## 传递参数——星星获得&进度条变化

新建一个名为ScoreRecordHelper的类，通过调用这个类传递各项参数

public class ScoreRecordHelper {

private static final String FILE\_NAME = "score";

private static ScoreRecordHelper helper;

private SharedPreferences mSp;

public enum Platform{ // 医务楼， 体育楼，餐饮楼， 教学楼

MEDICAL, SPORT, RESTAURANT, TEACHING}

private ScoreRecordHelper(){

mSp =StudyApplication.getApplication().getSharedPreferences(FILE\_NAME, Context.MODE\_APPEND);

}

public static ScoreRecordHelper getHelper(){

if (helper == null){

synchronized (ScoreRecordHelper.class){

if (helper == null){

helper = new ScoreRecordHelper();

}

}

}

return helper;

}

**// 记录某一栋楼的成就值score : 要记录的数值**

public void recordScore(Platform platform, int score){

SharedPreferences.Editor editor = mSp.edit();

editor.putInt(platform.name(), score);

editor.commit();

}

**// 某栋楼成就值+1**

public void recordScore(Platform platform){

int score = getScore(platform) + 1;

recordScore(platform, score);

}

**// 取得某一栋楼的成就值**

public int getScore(Platform platform){

return mSp.getInt(platform.name(), 0);

}

**//清除某一栋楼的成就值**

public void clearScore(Platform platform){

recordScore(platform, 0);

}

## 5.7 ViewPager滑动标签页

一个Activity里包含多个可滑动的标签页，每个标签页可以有独立的布局及响应，ViewPager应该和Fragment一起使用时，此时ViewPager的适配器是FragmentPagerAdapter。

public class stuVideo\_Teaching extends Activity {

private ViewPager vp;

private MyViewPagerAdapter vpAdapter;

private List<View> views;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.act\_videos);

initViews();

}

@Override

protected void onDestroy() {

super.onDestroy();}

private void initViews() {

LayoutInflater inflater = LayoutInflater.from(this);

views = new ArrayList<View>();

**// 初始化引导图片列表**

views.add(inflater.inflate(R.layout.stuvideo, null));

views.add(inflater.inflate(R.layout.stuvideo, null));

views.add(inflater.inflate(R.layout.stuvideo, null));

views.add(inflater.inflate(R.layout.stuvideo, null));

**// 初始化Adapter**

vpAdapter = new MyViewPagerAdapter(views, this);

vp = (ViewPager) findViewById(R.id.viewpager);

vp.setAdapter(vpAdapter);

}

}

public class MyViewPagerAdapter extends PagerAdapter {

**// 界面列表**

private List<View> views;

private Activity activity;

public MyViewPagerAdapter(List<View> views, Activity activity) {

this.views = views;

this.activity = activity;

}

**// 销毁arg1位置的界面**

@Override

public void destroyItem(View arg0, int arg1, Object arg2) {

((ViewPager) arg0).removeView(views.get(arg1));

}

@Override

public void finishUpdate(View arg0) {

}

**// 获得当前界面**

@Override

public int getCount() {

if (views != null) {

return views.size();}

return 0;

}

**// 初始化arg1位置的界面**

@Override

public Object instantiateItem(final View arg0, final int arg1) {

((ViewPager) arg0).addView(views.get(arg1), 0);

**// 点击按钮左划**

Button toLeft = (Button) arg0.findViewById(R.id.toLeft);

toLeft.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if (arg1 != 0) {

((ViewPager) arg0).setCurrentItem(arg1 - 1);

}

}

});

**// 点击按钮右划**

Button toRight = (Button) arg0.findViewById(R.id.toRight);

toRight.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if (arg1 != views.size() - 1) {

((ViewPager) arg0).setCurrentItem(arg1 + 1);