

MUSHAPI

# 城市雷达开发文档

---

齐鲁软件大赛城市定向运动的社会化应用

Mush

2012/9/24

描述了本项目的设计思路 and 开发历程

# 目录

一、项目背景.....	2
二、需求分析.....	3
(一)、题目中提出的功能: .....	3
(二)、曲折的分析过程: .....	3
1、要简单易用.....	3
2、跟向运动划清界限.....	3
3、我们的卖点是什么吗(我们怎么吸引用户)? .....	3
4、要不要用户登录功能: .....	4
5、如何实现社交功能: .....	4
6、用户真的愿意去创建游戏(或者用户真的能创建出高质量的游戏)么? .....	4
7、UI 如何设计?(炫 or 简洁) .....	5
(三)、游戏流程示意.....	5
三、应用功能设计.....	6
(一)、基本功能.....	6
1、根据地理信息查找并显示活动.....	6
2、创建游戏.....	6
3、社交功能.....	6
4、现实游戏状态.....	6
(二)、创新功能.....	6
1、雷达模式.....	6
2、根据游戏点来生成游戏路线,方便用户使用。 .....	8
3、指南针.....	8
4、检查更新.....	8
5、侧滑式菜单.....	9
四、详细设计.....	10
(一)、应用主要功能.....	10
1、游戏点类和游戏路径类的设计的设计 .....	10
2、实时定位: .....	11
3、    游戏线程主要方法。使用 Handler 实现线程之间的通讯 .....	11
(二)、数据库管理.....	12
(三)、指南针和雷达模式.....	14
1、根据 GPS 坐标描绘用户与点的相对位置的主要算法.....	14
2、我们采用了 SurfaceView 来绘制雷达视图,其双缓存机制有效地避免了重绘时的 闪烁现象。 .....	15
(四)、开放平台链接.....	15
(五)、在线升级.....	15
五、测试情况.....	17
一、发现的问题.....	17
二、总结.....	17
六、其他.....	18

# 一、项目背景

当今许多人沉迷于网络游戏，而多数的网络游戏无非就是打怪升级，有的玩家为了升级甚至废寝忘食，有时候甚至会感觉是游戏把人耍的团团转而不是人在玩游戏。显然这种游戏主导人的模式不是我们期望的，于是我们设计了这个项目，利用虚拟游戏的模式将玩家带入到现实世界中来。

## 二、需求分析

### （一）、题目中提出的功能：

- 1、根据地理位置查找、显示活动。
- 2、加入/退出活动（团队活动，单人活动）。
- 3、活动生成（地图生成、路线生成、检查点生成、检查点项目生成）。
- 4、跟踪并实时显示使用者活动完成情况。
- 5、加入社交功能。

### （二）、曲折的分析过程：

#### 1、要简单易用

我认为一个出色的移动应用（特别是休闲娱乐性质的）是不需要冗长的用户手册的。如果我们设计的应用在实现设计功能的前提下能让用户一看到就会使用，那就说明我们的设计是符合用户习惯的。所以我们认为应尽量简化用户的操作，使应用简单易用。

#### 2、跟向运动划清界限

在应用中不应体现过多的竞技元素。我们要做的不是一个专业的定向运动软件，而是一个面向大众的休闲应用。

#### 3、我们的卖点是什么吗（我们怎么吸引用户）？

游戏！游戏！游戏！、

我们认为本应用的最大卖点就是游戏内容。我们应该去多多发掘一些好的游戏点来吸引不同类型的用户。例如：商场小吃、旅游景点、文化古迹、风景等等。如果我们的应用能让用户发现他身边的、之前他并没有注意到的、又非常有趣的地方，那么我们就赢了。

当然这需要大量的时间和精力来完成，如果我们将时间都放在了找游戏上，显然就是与大赛的目的背道而驰了，于是我们只实现了几个简单的小游戏来说明情况。

## 4、要不要用户登录功能：

我们认为最好不要将“登陆”作为用户使用该应用的必要条件。因为如果要实现一个登录功能，则必须获取用户的一些敏感信息（邮箱，密码）。站在用户的角度显然是不希望将此类信息交给一个陌生的应用。

所以我们认为如果必须要求用户登录会大大降低用户体验，甚至一部分用户会因为不想注册而放弃使用该应用。

注：后来的实际操作过程中，由于我们使用了人人 API，所以才加入了使用人人登录的功能。

## 5、如何实现社交功能：

所谓社交功能无非就是为用户搭建一个沟通交流的平台，我们最初的设计是用一个服务器端程序实现用户的在线聊天，照片分享，游戏共享等功能（我们为此设计并实现了一个简单的服务器原型，该原型已经实现了在线聊天和游戏共享的功能）。后来考虑到单纯的在游戏玩家间的交流范围小效果不好，就算分享了照片用户的朋友也看不到。所以我们才正式确认了使用人人开放 API 的方案。

我们认为使用人人开放 API 较自行实现用户交流平台有以下几个优点：

- （1）、信息分享范围广。用户通过本应用分享的消息和照片可以被用户所有的好友看到，分享范围更广。
- （2）、技术简单，成本低。如做 web 服务还需租用服务器，成本高。
- （3）、利于宣传和推广。

同样的使用新浪微博和腾讯微博开放平台也能达到同样的目的，最好是同时提供几个不同平台的社交分享功能，让用户可以选择自己最常用的平台作为信息的分享平台，但是由于时间限制，我们并未做到这一点。

## 6、用户真的愿意去创建游戏（或者用户真的能创建出高质量的游戏）么？

我们不认为用户会热衷于创建游戏，至少大多数用户很难创建出高质量的游戏。

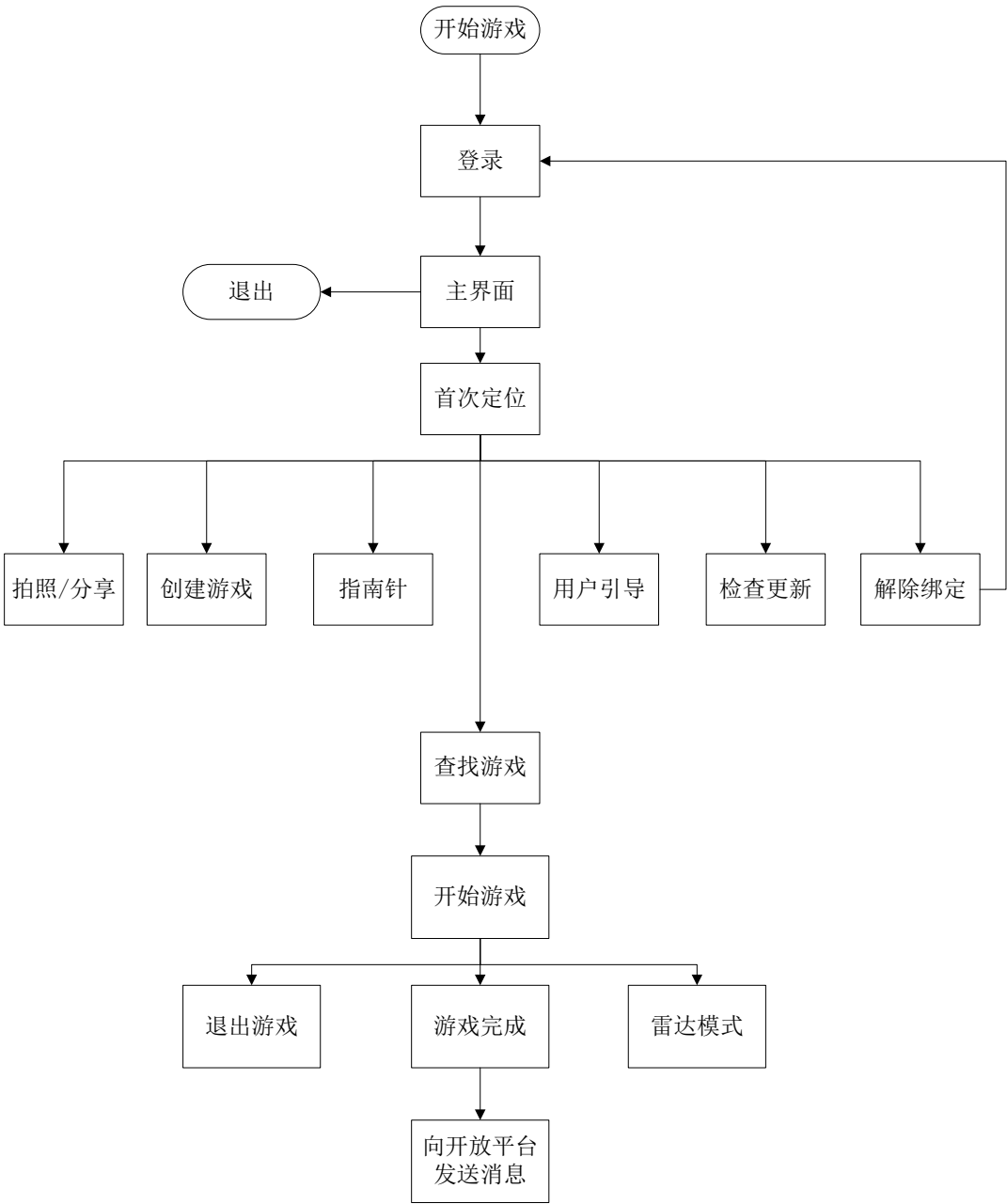
假设用户下载了我们的应用，那么他下载本应用的主要目的会是为了去创建游戏给别人玩么？显然不会。用户下载本应用是为了去游玩游戏的，是为了在生活着寻找乐趣的。

当然并不排除这样一种情况，用户可能会希望创建一个游戏来与好友分享。这也是我们的作品所欠缺的一点。但是考虑到实现该功能的技术难度，以及时间等因素我们还是放弃了该功能。

# 7、UI 如何设计？（炫 or 简洁）

我们在主要界面的布局上纠结了很久，我认为出色的 UI 应具备两点：不要把用户从来用不到的按钮放到用户眼前，一定要让用户很方便的找到需要的功能。

## （三）、游戏流程示意



图

## 三、应用功能设计

### （一）、基本功能

#### 1、根据地理信息查找并显示活动

定位→城市名→查找数据→显示→用户选择→开始活动

#### 2、创建游戏

用户可以使用本应用自行创建游戏。

填入游戏基本信息→添加检查点→添加检查点→。。。。结束

#### 3、社交功能

使用人人开放 API 中的“发布新鲜事”、“发布状态”、“发布照片”等功能与社交平台无缝整合，给用户提供方便实用的社交功能。

#### 4、现实游戏状态

使用电子地图和 GPS 功能来现实用户的位置、活动的完成情况。

### （二）、创新功能

#### 1、雷达模式

为了增加游戏的趣味性，我们设计了一个雷达模式，即用类似雷达图的形式来显示活动和检查点。



所谓雷达模式就是利用 GPS 坐标计算用户于游戏点的相对位置并按照一定的比例模仿雷达的样式绘制在屏幕上，增加趣味性的同时能更精确的应应用户与游戏点的相对位置，方便用户使用。同时用户可是在屏幕的下方获取其详细的运动状态。



## 2、根据游戏点来生成游戏路线，方便用户使用。



当用户进行选择并开始一个游戏时，程序会调用百度地图 API 根据游戏点生成一个合理的路径，引导用户来完成游戏。

## 3、指南针

我们开发了单独的指南针插件方便用户在高楼林立的城市中找到前进的方向。

## 4、检查更新

应用可以在线检查更新。

## 5、侧滑式菜单

考虑到用户使用本应用时，大部分的时间都是在参照地图进行运动，所以我们采用了侧滑式菜单的方式将菜单隐藏在屏幕的右侧。用户可以很方便地使用菜单键或直接从屏幕边缘拖拽出菜单，完成菜单选择操作时菜单会自动隐藏。



# 四、详细设计

## (一)、应用主要功能

我们将应用的主要功能放到包 team.mkhw1.CityRun 中。主要由王珂完成。

### 1、游戏点类和游戏路径类的设计的设计

GamePoint	
- pathId	: int
+ index	: int = -1
- gpName	: String = null
- question	: String = null
- answer	: String = null
- msg	: String = null
- status	: int = C.game.gamePointStatus_NotFini
- finishFlag	: boolean = false
+ MAX_ACCURACY	: int = 10
+ MIN_ACCURACY	: int = 1
+ MIDDLE_ACCURACY	: int = 5
* <<Constructor>>GamePoint (int lat, int lon)	
* <<Constructor>>GamePoint (GeoPoint gp)	
+ getGpName ()	: String
+ setGpName (String gpName)	: void
+ setQuestion (String question)	: void
+ getQuestion ()	: String
+ setAnswer (String answer)	: void
+ checkAnswer (String answer)	: boolean
+ getAnswer ()	: String
+ getStatus ()	: int
+ setStatus (int status)	: void
+ setMSG (String msg)	: void
+ getMSG ()	: String
+ setFinishFlag (boolean flag)	: void
+ isFinished ()	: boolean
+ onPoint (GeoPoint tagPoint)	: boolean
+ toGPDT ()	: GamePointDT
+ distance (GeoPoint tagPoint)	: double

GamePath	
- type	: String
- pathLocation	: String = ""
- pathName	: String = ""
- pathMSG	: String = ""
- pahtid	: int
- path	: ArrayList<GamePoint>= new ArrayList<GamePoint>()
+ <<Constructor>>GamePath ()	
+ <<Constructor>>GamePath (ArrayList<GamePoint> path)	
+ <<Constructor>>GamePath (List<GamePointDT> pointDTs, GamePathDT path)	
+ <<Constructor>>GamePath (GamePoint gamePoint[])	
+	addGamePoint (GamePoint gamepoint) : void
+	getPath () : GamePoint[]
+	getPahtOverlays () : Collection<GamePointOverlay>
+	getnow () : GamePoint
+	size () : int
+	getPahtid () : int
+	setPahtid (int pahtid) : void
+	getPathName () : String
+	setPathName (String pathName) : void
+	getPathLocation () : String
+	getPath2ArrayList () : ArrayList<GamePoint>
+	getPath2List () : List<GamePoint>
+	getPath2GeoList () : ArrayList<GeoPoint>
+	setPathCity (String pathCity) : void
+	getPathtype () : String
+	setPathtype (String pathtype) : void
+	getPathMSG () : String
+	setPathMSG (String pathMSG) : void

## 2、实时定位：

定义了接口 CRLocationListeneri

```
public interface CRLocationListeneri {
    public void onLocationChanged(Location location);
}
```

在 LocationListener 中通过回调该接口将 Location 信息传递到各个线程。

## 3、 游戏线程主要方法。使用 Handler 实现线程之间的通讯

```
private CRLocationListeneri myCRLocationListeneri = new CRLocationListeneri(){

    ArrayList<GeoPoint> algeopoints = new ArrayList<GeoPoint>(1);
    PathOverlay pho;
```

```

public void onLocationChanged(Location location) {
    // TODO Auto-generated method stub
    GamePoint gpWgs84 = new GamePoint(
        (int) (location.getLatitude() * 1e6),
        (int) (location.getLongitude() * 1e6));
    GeoPoint gpBaidu = CoordinateConvert
        .bundleDecode(CoordinateConvert
            .fromWgs84ToBaidu(gpWgs84));

    //调试代码
    //gpBaidu = gpWgs84;

    algeopoints.add(gpBaidu);
    if (mMapView.getOverlays().contains(pho)) {
        mMapView.getOverlays().remove(pho);
    }
    pho = new PathOverlay(algeopoints);
    // 绘制路径
    mMapView.getOverlays().add(pho);

    if (null == gamePath.getnow()) {
        // 这里表示游戏完成!
        CRLocationlistener.removeUpdates(myCRLocationListeneri);
        isRunning = false;
        //将游戏完成的消息通知主线程
        hostHandler.sendMessage(C.handler.finish_game);
        return;
    }
    if (gamePath.getnow().onPoint(gpBaidu)) {
        if (dialogIsShow == false) {
            // 弹出检查点问题
            checkAnswerDialog();
        }
    }
}

}};

```

## (二)、数据库管理

数据库相关操作在包 team.mkhw1.CityRun.db 中，由王珂完成。

DBManager	
- TAG : String	= "DBManger"
+ db : SQLiteDatabase	
+ <<Constructor>> DBManager (Context context)	
+ addgamePoints (List<GamePointDT> gamePoints, String tableName)	: void
+ addgamePath (GamePathDT gamePath, String tableName)	: void
+ queryPaths (String cityName)	: List<GamePathDT>
+ queryPoints (int pathId, String tableName)	: List<GamePointDT>
+ queryTheCursor ()	: Cursor
+ closeDB ()	: void

### (三)、指南针和雷达模式

分别位于包 team.mkhw1.CityRun.Comopass 和 team.mkhw1.CityRun.Radar 中， 由王珂和牟俊秋完成。

#### 1、根据 GPS 坐标描绘用户与点的相对位置的主要算法

```
public void doDraw(Canvas canvas, GeoPoint cLocation) {
    Log.v(TAG, "doDraw");
    Log.v(TAG, "我的坐标 " + cLocation.toString());
    // 先清屏再画
    drawBackGround(canvas);
    Paint paint = new Paint();// 画笔
    int i = 0;

    for (GamePoint point : this.points) {
        i++;
        double lat = (double) point.getLatitudeE6() / 1000000;
        double lon = (double) point.getLongitudeE6() / 1000000;

        double radius = 0;// 图上的半径px
        double alpha = 0;// 图上角度
        float pic_xB;
        float pic_yB;

        radius = scale
            * Calculate.getDistance(
                ((double) cLocation.getLatitudeE6() / 1000000),
                ((double) cLocation.getLongitudeE6() / 1000000),
                lat, lon);
        alpha = Calculate.getAngle(
            ((double) cLocation.getLatitudeE6() / 1000000),
            ((double) cLocation.getLongitudeE6() / 1000000), lat, lon);
        alpha *= Math.PI / 180;// 转换成弧度

        pic_xB = (float) (centerX + radius * (Math.sin(alpha)));
        pic_yB = (float) (centerY - radius * (Math.cos(alpha)));

        paint.setStyle(Paint.Style.FILL);
        paint.setColor(Color.RED);
        canvas.drawCircle(pic_xB, pic_yB, 5, paint);
        paint.setColor(Color.rgb(0, 0, 102));
        paint.setTextSize(18);
```

```

        canvas.drawText(point.getGpName(), pic_xB, pic_yB, paint);
    }

}

```

2、我们采用了 **SurfaceView** 来绘制雷达视图，其双缓存机制有效地避免了重绘时的闪烁现象。

## （四）、开放平台链接

该功能在包 `com.renren.api.function` 和 `com.renren.api.activity` 中，参考了人人 API，由牟俊秋完成。

实现了帐号绑定、推送新鲜事、拍照上传、发送消息等功能。

## （五）、在线升级

该功能位于包 `team.mkhw1.CityRun.Update` 中，由徐永强完成。  
主要方法

```

public void checkUpdate() {
    hasNewVersion = false;
    new Thread() {

        @Override
        public void run() {
            try {

                String verjson = NetHelper.httpStringGet(UPDATE_CHECKURL);
                JSONArray array = new JSONArray(verjson);

                if (array.length() > 0) {
                    JSONObject obj = array.getJSONObject(0);
                    try {
                        newVersionCode = Integer.parseInt(obj
                            .getString("verCode"));
                        newVersion = obj.getString("verName");
                        updateInfo = "";
                        if (newVersionCode > curVersionCode) {

```



```
        hasNewVersion = true;
    }
} catch (Exception e) {
    newVersionCode = -1;
    newVersion = "";
    updateInfo = "";
}
}
} catch (Exception e) {

}
updateHandler.sendMessage(UPDATE_CHECKCOMPLETED);
};
}.start();

}
```

# 五、测试情况

## 一、发现的问题

- 1、ListView 拖动时会变黑。
- 2、拍照上传后又一个黑屏出现。
- 3、某些情况下，当程序不可见时会意外停止。
- 4、当创建的游戏或选择开始的游戏只有一个游戏点时会抛出空指针异常。
- 5、第一次运行程序，第一次找游戏时会出现未找到任何游戏，第二次后正常。
- 6、雷达模式误差较大。
- 7、2G 网络下地图加载较慢。
- 8、室内时无法定位。
- 9、部分内存低的手机在游戏过程中开启引导界面会产生内存溢出错误。
- 10、不能在线更新游戏数据库。

需要说明的是前六项问题已经在 1.1 版中做出了修正，第九项也做出相应的改进（但仍存在不足）。7、8 两项尚未解决。

其中 7 可以采用离线地图包的形式来解决，8 可以利用基站定位的方式来解决。

10 是我们下一步计划完成的功能。

## 二、总结

应用可以流畅运行，实现了设计的功能，用户体验良好。

## 六、其他

我们开发的应用通过了人人网和安卓市场的审核，已经上线发布并进行了一次更新。  
详细情况参见[人人公共主页](#)和[安卓市场](#)。