

# 调研与选型

## 开发工具

- **微信开发者工具**: 我们将要开发的圈地大作战是一个"微信小游戏", 所以会用到腾讯官方的"微信开发者工具"进行发布。事实上也有一些可视化小程序开发工具(微盟、微尘), 但这些工具一般是提供固定的组件和成熟的模板, 通过拖拽等操作完成搭建, 对于复杂的需求与设计就显得无力。而官方开发工具**自由度高**, 可以任意开发想要的功能。
- **小程序开发框架**
- **小游戏开发框架**

微信游戏可以分为**游戏类小程序**和**小游戏**两种, 前者使用小程序开发框架进行开发, 后者使用小游戏框架。下面是对这两种框架的调研和分析。

## 小程序开发框架

可行方案:

- 原生框架
- WePY
- mpvue
- Taro
- uni-app

## 原生框架

整个小程序框架系统分为两部分: 逻辑层和视图层。小程序提供了自己的视图层描述语言 WXML 和 WXSS, 以及基于 JavaScript 的逻辑层框架, 并在视图层与逻辑层间提供了数据传输和事件系统。

原生开发对Node、预编译器、webpack支持不好, 影响开发效率和工程构建流程。微信小程序本身对工程化几乎没有任何的支持, 要原生框架支持工程化的话, 需要自己动手搭建很多工程化上的东西。

没有提供原生的应用状态管理方式, 但是支持使用 Redux 或 MobX 进行状态管理。

## WePY

需要熟悉 vue 和 WePY 两种语法。

WePY 框架在开发过程中参考了 Vue 等现有框架的一些语法风格和功能特性，对原生小程序的开发模式进行了再次封装，更贴近于MVVM架构模式。

WePY 有如下特性：

- 支持组件化开发。原生小程序支持js模块化，但彼此独立，业务代码与交互事件仍需在页面处理。无法实现组件化的松耦合与复用的效果。
- 支持使用 npm 外部依赖。
- 目录结构清晰。在 WePY 中则使用了单文件模式，将原生小程序app实例的3个文件统一为 app.wpy，page页面的4个文件统一为 page.wpy。
- 便于工程化。wepy 有自己的 wepy-cli。
- 支持使用 Redux 或 MobX 进行状态管理。

## mpvue

需要熟悉 vue。

mpvue 是一个使用 [Vue.js](#) 开发小程序的前端框架。框架基于 `Vue.js`，修改了的运行时框架 runtime 和代码编译器 compiler 实现，使其可运行在小程序环境中，从而为小程序开发引入了 `Vue.js` 开发体验，在用法上面是高度和 Vue 一致的。

mpvue 有如下特性：

- 彻底的组件化开发能力：提高代码复用性。
- 支持使用 npm 外部依赖。
- 快捷的 webpack 构建机制：自定义构建策略、开发阶段 hotReload。
- 方便的 Vuex 数据管理方案：方便构建复杂应用与应用状态管理。

## Taro

需要熟悉React语法标准。

Taro 是一套遵循 [React](#) 语法规范的**多端开发**解决方案。

Taro 有如下特性：

- 二次开发。可以将原生微信小程序应用转换为 Taro 代码，进而通过 `taro build` 的命令将 Taro 代码转换为对应平台的代码，或者对转换后的 Taro 代码用 React 的方式进行二次开发。
- 支持使用 JSX 语法。让代码更具表现性。
- 支持使用 npm/yarn 安装管理第三方依赖。
- 支持使用 ES7/ES8 甚至更新的 ES 规范。
- 支持使用 CSS 预编译器。
- 支持使用 Redux 或 MobX 进行状态管理。

- 小程序 API 优化，异步 API Promise 化。

## uni-app

需要熟悉 vue。

uni-app 是一个使用 [Vue.js](#) 开发所有前端应用的多端框架。可发布到iOS、Android、H5、以及各种小程序（微信/支付宝/百度/头条/QQ/钉钉）等多个平台。

有如下特性：

- 支持使用 npm 外部依赖。
- 支持微信小程序自定义组件及SDK。
- 将常用组件和API进行跨平台封装。
- 内嵌mpvue，提供uni扩展组件及mpvue兼容组件。
- 支持条件编译，可在条件编译代码块中，随意调用各个平台新增的API及组件。

## 小游戏开发框架

可行方案：

- Cocos
- Gegjs
- Egret
- Laya

### Cocos

Cocos可以细分为**Cocos2d**、**Cocos2dx**、**Cocos Creator**、**Cocos Studio**、**Cocos Creator 3D**。

早在微信官方正式发布小游戏之前，Cocos Creator 团队就和微信开发团队合作完成了 Cocos Creator 对微信小游戏的支持。Cocos支持**C++**、**Lua**、**JavaScript**等开发语言。

- 引擎框架适配微信小游戏 API，纯游戏逻辑层面，开发者不需要任何额外的修改。
- Cocos Creator 编辑器提供了快捷的打包流程，直接发布为微信小游戏，并自动唤起小游戏的开发者工具。
- 自动加载远程资源，缓存资源以及缓存资源版本控制。
- 拥有更庞大的第三方库支持。

### Gegjs

[Gengine](#) 是一个基于 **Vue.js** 核心开发的前端框架，与Vue采用Typescript不同，Geg采用的是Es6开发，并修改了 Vue.js 的底层实现，包括了compile与platforms，并新增gxml部分。框架通过xml组织视图层

级，所以它与底层视图渲染的方式,如dom的渲染等并没有直接关系，也因为这个特点，虽然Gengine开发之初是为了使用Vue兼容微信小游戏而设计，但是理论上可以使用Geg.js开发任意特定平台的视图项目。

## Egret

Egret Engine是一个基于**TypeScript**语言开发的HTML5游戏引擎。

- 目前只支持白鹭引擎 5.1.x 版本发布为微信小游戏，推荐使用最新的 5.1.2 版本。
- 支持第三方扩展库。
- API文档很规范，自建工具链 workflow。
- 可以开发2D也可以开发3D，两方面性能一般。

## Laya

Layabox支持**JavaScript**、**TypeScript**、**ActionScript**三种开发语言。

- 支持开发2D、3D产品研发，并且两方面性能都还不错。
- 支持Canvas与WebGL模式。
- 支持同时发布为HTML5、Flash、APP（IOS、安卓）微信小游戏，QQ玩一玩多种版本。
- 早期工具不全，性能虽然好，但需要自己写工具。

## 最终选择方案：小游戏开发框架中的Cocos Creator

### 理由：

1. **为什么使用小游戏开发：**调研小程序开发框架之后发现，前者的开源项目很少包含游戏类的，小游戏的运行环境是小程序环境的扩展，基本思路也是封装必要的 WEB 接口提供给用户，尽可能追求和 WEB 同样的开发体验。小游戏在小程序环境的基础上提供了 WebGL 接口的封装，使得**渲染能力和性能**有了大幅度提升。因此这里我们直接使用小游戏开发。
2. **为什么选择Cocos：**Cocos拥有更庞大的第三方库支持，便于快捷开发。且有完整的学习文档，学习更简单一些，套用Unity 的操作方式，也更符合趋势。
3. **为什么是Cocos Creator：**（这里主要在Cocos2dX与Cocos Creator中选择）
  1. Cocos2dX可控性更强一些，使用C++开发，所以性能非常好。可以做一些比较**大型的游戏**。缺点是**学习起来比较费劲**。
  2. Cocos Creator是一款专门的游戏开发引擎，界面更简洁。拥有完整的游戏开发API，相较于Cocos2dx**易上手**，可以快速实现游戏场景和功能。
  3. Cocos Creator使用JS作为开发语言，所以性能**相对较差**。以**小游戏为主**，不太适合做太大型的游戏。
  4. Cocos Creator完全支持**发布到微信小游戏**，支持微信授权登录，发送邀请等API。
  5. Cocos Creator免费开源，支持游戏开发者进行创作，生态完整，论坛响应快。

# 已发布的微信小游戏

游戏	开发商	游戏引擎
欢乐斗地主	腾讯	cocos
四川麻将	腾讯	cocos
天天德州	腾讯	cocos
腾讯中国象棋	腾讯	cocos
爱消除乐园	腾讯	cocos
欢乐坦克大战	腾讯	cocos
欢乐消消消	华夏乐游	cocos
保卫萝卜迅玩版	飞鱼科技	cocos
全民大乐斗	腾讯	laya
悦动音符	腾讯	laya
广东麻将	腾讯	laya
贵州麻将	腾讯	laya
星途WeGoing	腾讯	laya
大家来找茬腾讯版	腾讯	phaser
拳皇命运KO不服	腾讯	createjs
跳一跳	腾讯	three.js
纪念碑谷2x给自己的成长语录	腾讯	three.js

今天微信小游戏引擎占比

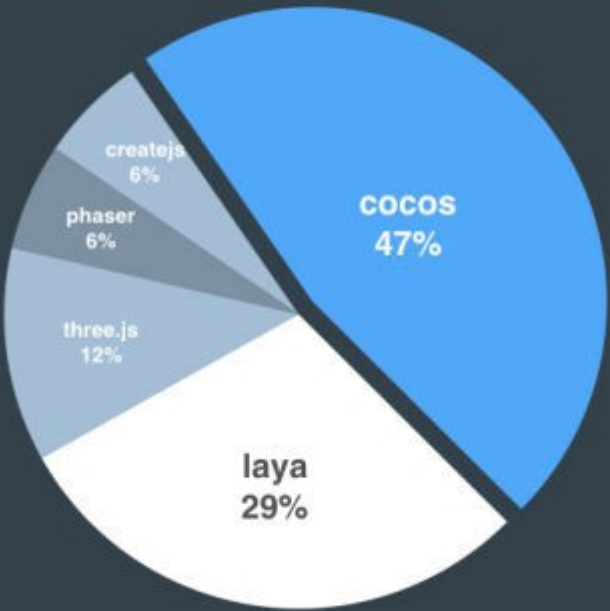
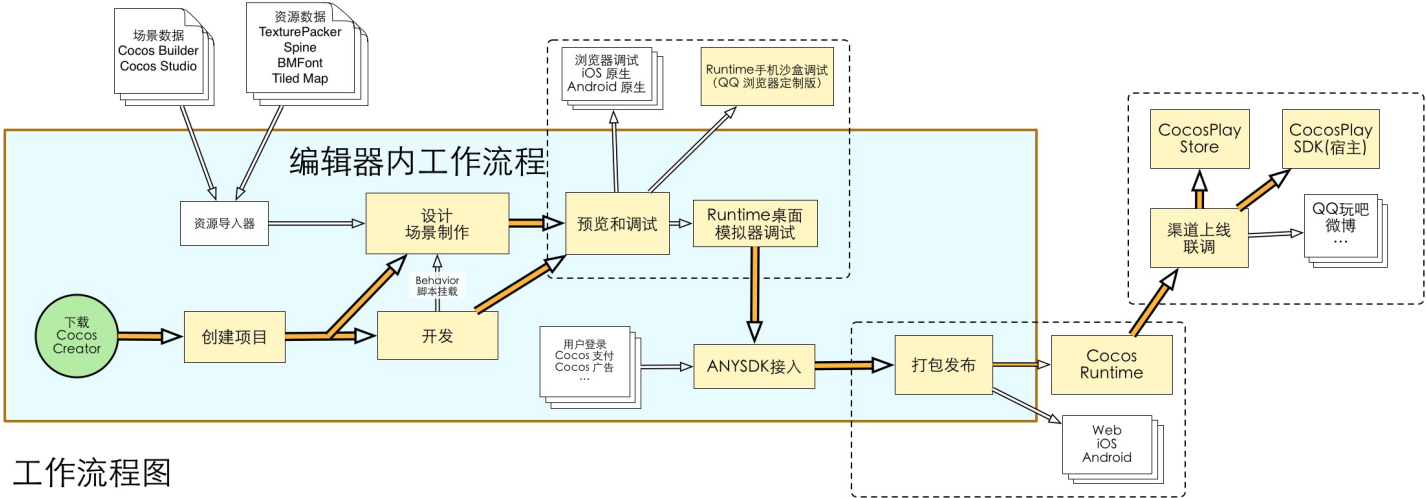


图1: 微信小程序开发工具使用情况



工作流程图

图2: Cocos工作流程

## 服务器端资源和框架

因为圈地大作战需要实现多人联网对战的功能, 所以需要服务器资源

## 可行方案:

- **KBEngine**

KBEngine 是一款开源 mmog 服务端引擎, 使用统一协议能够轻松与前端对接, 能轻松使用 unity3d、ogre、cocos2d、html5 等作为前端表现。

底层框架由 C++编写, 逻辑层使用 Python(支持热更新)。

- **Skynet**

Skynet 是一个基于C跟lua的开源服务端并发框架, 结构精简, C层没有多余的一堆三方库, 只有紧凑的核心结构, 提供最核心的消息处理框架; Lua层用来写游戏逻辑, 降低了开发门槛。

- **Matchvs**

Matchvs游戏云, 简称“Matchvs”, 一款商业化游戏服务器引擎。

它将一整套联网功能集成为一个高可用的SDK, 通过SDK API, 允许开发者根据自身的游戏逻辑定制开发。

## 最终选择方案: Matchvs

### 理由:

1. Matchvs是国内的服务器引擎, 有大量中文文档和资料, 上手快
2. Matchvs直接提供了cocos和微信小游戏的解决方案, 为工程提供了方便快捷的实现渠道

## 其他资料

### AI设计方案:

1. [游戏AI决策方法](#)
2. [简单的贪吃蛇AI实现](#)
3. [贪吃蛇AI](#)

## 开源代码

1. [圈地大作战开源代码](#)
2. [利用matchvs实现的斗地主游戏](#)