**2016-2017学年第2学期**

**实 验 报 告**

* 课程名称:跨平台脚本开发技术
* 实验项目: 期末大作业
* 专业班级\_计算1402\_\_
* 学生学号\_31401128\_\_
* 学生姓名\_董佳铭\_\_
* 实验指导教师:郭鸣

**实验内容**

1. 大作业安排
   * 按 及格 中 良 优 计分
2. 每人完成一个介绍JS进展的技术报告,可以是翻译英文,或自己总结,要求如下
   * 内容范围
     + Javascript语言新技术动向:包括 async await, generator, web socket,Service Worker, Reflect API 等
     + Javascript框架 vue, cyclejs,infernojs, sveltejs,hyperapp... 介绍框架原理,使用案例.
     + Javascript 某个技术实现 Vdom,GraphQL
     + 技术报告范例参考[500 Lines or Less](http://www.aosabook.org/en/index.html)
   * 提交形式
     + 提交 code + markdown文档
   * 技术报告讲座 (可选,加分)
     + 预计 20分钟左右,下下周开始
     + 制作 ppt (markdown)
   * markdown[工具](https://www.npmjs.com/package/madoko-local) npm i -g madoko-local
   * 自我评估 5为最高
     + 原创性 (1-5)
     + 技术难度 (1-5)
     + 工作量 (1-5)
3. 大项目,分组完成,评分方式，以下为起始评分
   * 及格 完成书上项目，实现electron 发布为操作系统跨平台项目
     + 书上项目的代码有修改与改进
     + 改进建议见下面的条目
   * 中 完成书上项目，采用前端框架,实现应用跨平台的适配，Desktop, Tablet, Mobile
     + 对现有功能有较多改进
   * 良 完成自定义项目
     + 实现采用跨平台技术,包括服务端与浏览器端
   * 优秀 完成自定义项目.
     + 使用比较有特点JavaScript框架
     + 采用比较新颖的技术 例如FRP PWA等
4. 现有改进方式建议列表
   * 开发技术
     + UI 响应式设计
       - 采用BootStrap,Material Design等前端框架
       - 支持 Desktop,Tablet,Mobile
     + 服务端 NodeJs框架
       - Express、Koa等实现服务端
     + WebSocket socket.io等实时交互技术
5. 项目信息参考
   * 平台游戏 or [2048](http://gabrielecirulli.github.io/2048/)
     + 添加用户登录，用户积分排行
     + 积分排行存储到本地
     + 积分排行存储到服务器
     + 实现游戏手机平台，平板的支持
     + 在线关卡编辑器
     + 5 \* 5 2048
     + 立体的游戏,3d扫雷
   * Time travel 编译器compiler/debugger
     + vue devTools
     + Cyclejs Time traveler
   * 图形化 FSM 编译器
     + http://jsflap.com/
     + http://bgrawi.com/Fourier-Visualizations/
     + http://www.jflap.org/
     + http://www.cs.sjsu.edu/faculty/pollett/154.1.11s/
     + http://hackingoff.com/compilers
     + https://github.com/jahan-addison/posiy
   * JS语言创新 在js 中实现其他语言的语法特性
     + https://github.com/slurmulon/flap
     + https://github.com/sanctuary-js/sanctuary
     + http://www.sweetjs.org/
   * 健康/中医/传统武术方面项目 huawei store
     + 大家中医
     + 华为健康
     + 中医资料学习
     + https://github.com/fsx950223/MyWebFrameWork
   * IT技术学习平台
     + https://github.com/rusthon/Rusthon
     + webpackbin.com
     + https://hourofpython.trinket.io/ke3-shi4-hua4-python-jian3-jie4#/huan1-ying2/bian1-cheng2-yi1-xiao3-shi2
   * Jupyter in browser 无服务端的jupyter?
     + https://try.jupyter.org/
     + https://github.com/jvilk/BrowserFS
     + http://slides.com/bahmutov/browser-is-the-new-server-nodefest
     + https://github.com/kripken/emscripten
     + http://cyrille.rossant.net/numpy-browser-llvm/
     + http://www.rfk.id.au/blog/entry/pypy-js-first-steps/?utm\_source=tuicool&utm\_medium=referral
   * 可视化项目
     + 服务端数据存储, 网页端显示数据
     + echarts NVD3.js
     + superset https://github.com/airbnb/superset
     + https://vega.github.io/voyager2/
     + https://ipywidgets.readthedocs.io/en/latest/examples/Widget%20Basics.html
   * 参考 https://electron.atom.io/
   * git 管理技术文档,和项目代码,可以使用 github desktop
   * 验收大作业的时候查看git log
   * npm i -g git-stats
   * https://github.com/tomgi/git\_stats/graphs/contributors
6. 代码风格 standard https://standardjs.com/readme-zhcn.html
7. 项目分工表格
8. 1人项目权重为1.0
9. 2人项目权重和为 1.9

| **姓名** | **学号** | **班级** | **任务** | **权重** |
| --- | --- | --- | --- | --- |
| Name | No. | Class | Task | Factor |
| 董佳铭 31401128 | 计算1402 |  | 全部 | 1 |
|  |  |  |  |  |
|  |  |  |  |  |

1. 项目自我评估表

| **proto 继承** | **函数式编程** | **xhr** | **fetch api** | **json server** | **session Storage** | **响应式设计(Desktop/Tablet/Mobile)** | **Other** |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 5 | 1 | 0 | 0 | 0 | 0 | 0 |  |

1. 项目自评等级:(1-5)

3

1. 项目说明
   * 改进 判定 功能1

改进了俄罗斯方块的判定，在无法旋转的位置判定进行了修改，使得原本从竖立状态的直线无法穿过有障碍而变成水平状态的直线。

1. 解决技术要点说明
   * 解决 判定 问题1， 关键代码与步骤如下
2. function round() {
3. if (blockIndex == 4) {
4. return;
5. }
6. for (var i = 1; i < block.length; i++) {
7. var o = block[0];
8. var point = block[i];
9. //旋转后的位置不能与现有格子的方块冲突
10. var tempX = o.y + o.x - point.y;
11. var tempY = o.y - o.x + point.x;
12. if (isOverZone(tempX, tempY)) {
13. return;
14. }
15. }
16. clearBlock();
17. //可以旋转，设置新的旋转后的坐标
18. for (var i = 1; i < block.length; i++) {
19. var o = block[0];
20. var point = block[i];
21. //旋转后的位置不能与现有格子的方块冲突
22. var tempX = o.y + o.x - point.y;
23. var tempY = o.y - o.x + point.x;
24. block[i] = {
25. x: tempX,
26. y: tempY
27. };
28. }
29. drawBlock();