README(introduction)

0 入口

程序的入口是：**final_project/index.html**

建议： **F11 大屏观看**

操作悠闲一点

1. final_project/html.index

1.1 LOGIN 界面

1.1.1 默认账号密码有两个

banbao/123456

smjb/123456

1.1.2 点击[SIGN UP]可以实现自建账号

1.1.3 usename 输入框支持 enter 跳转至 password

1.1.4 password 输入框支持 enter 实现点击[SIGN IN]/[OK]

1.1.5 Tab/Shift+Tab 是浏览器自己支持的跳转

1.2 输入密码后登录进入页面

1.3 响应式布局(屏幕小的时候左上角的 cheer 是导航栏,点击可以跳转)

2. introduction.html(陈绮贞/cheer)

2.1 很简单，没啥介绍的

3. performance.html(演艺经历)

3.1 实现滚动鼠标向下翻页

4. album.html(专辑)

4.1 一直点"旋转的 CD"就完事了

5. life-story.html(生活-故事)

5.1 精华就在右边的小折扇

6. some_fun.html(周边)

6.1 有很多隐藏彩蛋

6.2 两个小游戏，注意不要点太快(**会崩溃的!!!**)

7. latest_news.html(最近消息)

7.1 轮播图支持点击跳转

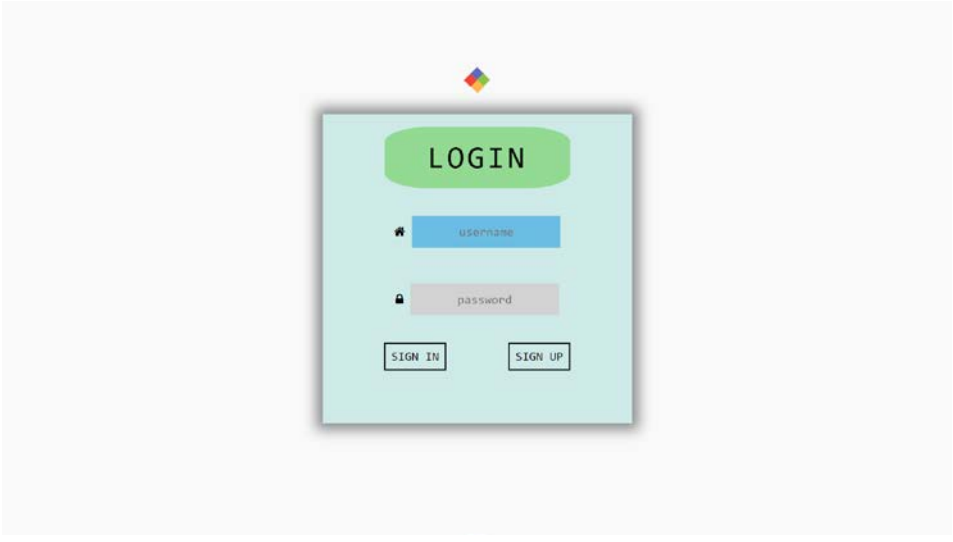
7.2 箭头点击会弹出消息

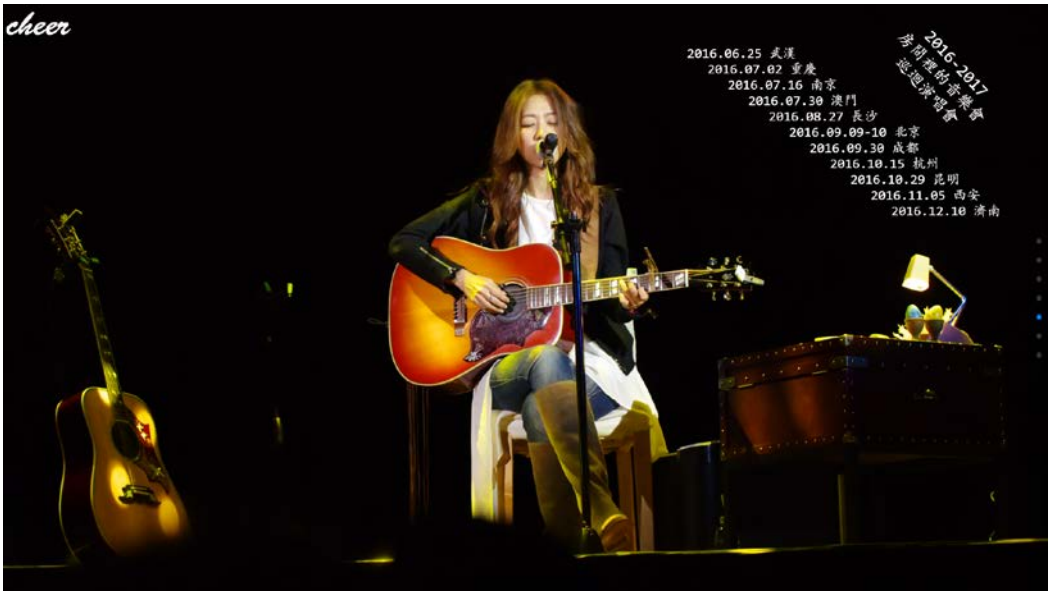
谢谢！

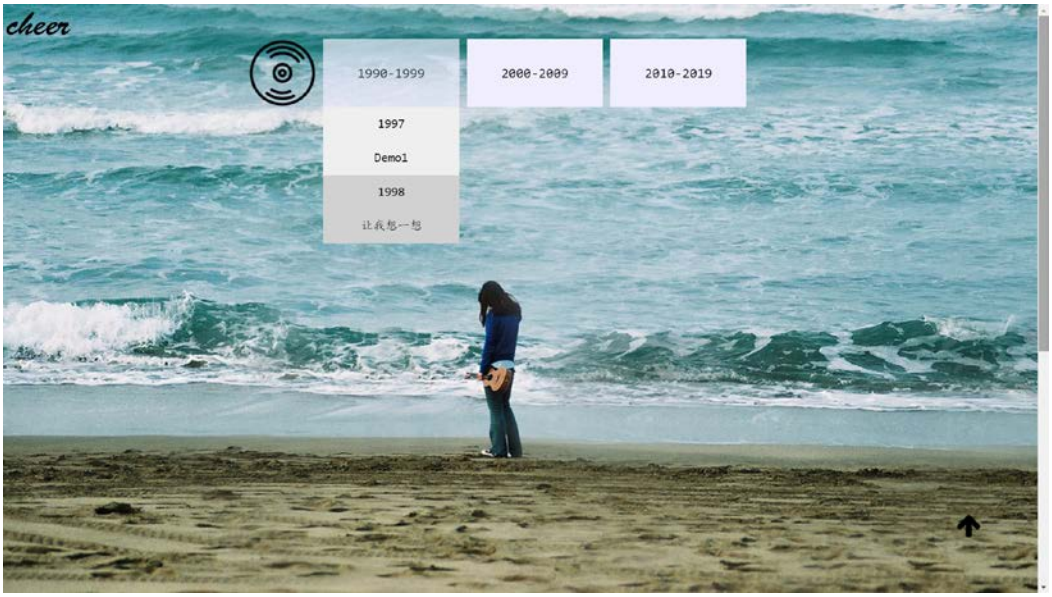
README(code)

0 前言

0.0 结果展示




cheer
2016.06.25 武漢
2016.07.02 重慶
2016.07.16 南京
2016.07.30 澳門
2016.08.27 長沙
2016.09.09-10 北京
2016.09.30 成都
2016.10.15 杭州
2016.10.29 昆明
2016.11.05 西安
2016.12.10 濟南
2016-2017
感謝有你
巡迴演唱會


cheer
1990-1999
2000-2009
2010-2019
1997
Demo1
1998
让我想一想


cheer
喜歡你
我喜欢上你时的内心活动
电影"喜欢你"主题曲
陈绮贞
我喜欢上你时的内心活动 詞曲:陈绮贞

cheer

cheer

Cheer

Game 1 Game 2

陈绮文

cheer

BACK

开始游戏

cheer

BACK

cheer

HOT!

2018年7月29日
五月天演唱会上,与五月天合唱"私奔到月球"

0.1 project 文件夹介绍

-> css/js/image/import 即字面意思
-> test 里面是写网页时用的一些测试 html
-> learnt 是一些 trick
-> html 是到的其他网页
-> README(你正在看的这个)
-> "password.jhj"里面放的文件是默认账号密码
-> 里面的".jhj"文件右键以记事本方式打开即可

0.2 网络导入的内容

-> final_project\js\jQuery.js
-> final_project\js\jquery.mockjax.js
-> final_project\js\swiper.jquery.min.js
-> final_project\js\swiper.min.js
-> final_project\import\font-awesome-4.7.0
   特殊符号
-> final_project\import\http_www.htmleaf.com\201901311628
   基于 canvas 的粒子组成文字动画特效
-> final_project\import\onepagescroll
   鼠标滚动支持滚动一页
-> final_project\js\jquery.easing.js
   jquery 的扩展包,可以实现 fadeOut/fadeIn/animate 等的更多动态效果
-> final_project\js\jquery.swatchbook.js
   后面的一个折扇的动态效果(来自网页)
-> final_project\js\ modernizr.custom.79639.js
   后面的一个折扇的动态效果(来自网页)

0.3 这边的建议是 F11 全屏观看

1 final_project\index.html(HOME)

1.1 LOGIN 部分

1.1.1 上面的 loading 动画

利用 animation 实现,就是将一个正方形分为"上左/上右/下右/下左"四个区域,然后小方块按照"上左->上右->下右->下左->上左"的路线循环运动,然后再将整个正方形旋转 45deg 就 OK 了。

1.1.2 LOGIN 部分

-> 头上的 icon

```
<link rel="shortcut icon" href="../image/icon/life-story_icon.jpeg"
    type="image/x-icon">
```

-> 整个框架用了 border-shadow

-> input 部分使用了":focus"/background-image

-> "小屋子/锁"使用了 font-awesome 特殊字体

-> "SIGN IN"/"SIGN UP"按钮使用了 stroke 属性实现动态效果

-> enter 跳转使用了 keydown()函数,判断 event 的 keyCode(enter 为 13)
捕捉到 enter 就跳转或者是模拟点击 trigger

-> username/password 验证部分
就是使用 jquery 获取 text 中的 val(),然后与存储的"账号/密码"比对即可

-> 模拟注册账号(鸡肋)
这个功能只是自己觉得好玩搞的,在实际应用之中一点用都没有,实际应用也不容许这样子出现
因为本身没有些服务器端程序,这里是通过 localStorage 来实现的(自己骗自己),记住输入的账号密码,然后加入账号密码库中。
因为功能用处不大,写的时候就只支持一个"账号/密码"(增加第二个时会把第一个覆盖)
实现方式,加入 key-value 对(key="aaa",value=username+" "+password),验证的时候通过 key 取出 value 然后通过 indexOf 将 username 和 password 分开

-> 跳转时使用了$("#id").focus()来使得 input 得到焦点

1.2 主体部分

-> 背景图片是 body 的 background-image

```css
.body_style {
    background-image: url(image/index_bg.jpg);
    background-repeat: no-repeat;
    /* 背景图垂直、水平均居中 */
    background-position: right center;
    /* 当内容高度大于图片高度时，背景图像的位置相对于viewport固定 */
    background-attachment: fixed;
    /* 让背景图基于容器大小伸缩 */
    background-size: cover;
    /* 设置背景颜色，背景图加载过程中会显示背景色 */
    background-color: #34151a;
}
```

-> 响应式布局,区分边界是 900px

-> 左上角的"cheer"是 nav(其他网页也是这样)
$(".cheer").on("click", function () {...});
从左边进入使用了 jquery 的 animate 函数,left 值的变化
回调函数使得"cheer"本身颜色的变化

-> 每个 label 的 hover 触发 transform,通过伪类元素实现

通过 before/after 宽度在 hover 前后的不同实现

玻璃擦除效果就是一个 before 元素的位置在 hover 前后不同实现
(hover 时有一个小方块从 label 上面滑过)

-> nth-child(2n+1)选择不同效果

-> 响应式布局的另外一种实现的方式
$(window).resize(function () {...}

1.3 亮点

-> 鸡肋的 locolStorage

-> $(window).resize(function () {...}

2 final_project/html/introduction.html(陈绮贞/cheer)

2.1 酷炫的 loading 部分(使用网络上的)

-> 实现酷炫效果,,鼠标移动上去的时候可以实现弹开小球(基于 canvas)

2.2 主体部分

-> 排版用了 font-awesome

实现文字绕排效果,实现特殊字体

2.3 亮点

-> loading 的加载特效,(但是不是自己写的)

3 final_project/html/performance.html(演艺经历)

3.1 loading 部分

-> 就是使用 border 以及 after 先画出边框,然后 animation/rotate 就完事了

3.2 主体部分

-> 引入了 swiper 库实现了一次滚轮翻一面的效果(引入其他库)

-> 每次都是属于文字的那张 slide 出现时,文字才以 slideDown 的形式出现

  -> 这个就比较厉害了。

    简单地说就是用了一个函数,

    $(window).on("mousewheel",function () {...});

    每次滚轮滚动的时候触发这个事件

    使用 offset().top 来循环获得现在显现出来的是哪一个 Slide

    再使用 event.detail/event.wheelDelta 来得到向上滚还是向下滚

    从而判断出来下一个是什么,让下一个的 text 执行 slideDown,其他的 SlideUp

  -> 有一些小细节的保护措施

    -> 当已经是第一张 slide 显示出来的时候,上滚这个动作失效

      同理,当已经是最后 slide 显示出来的时候,下滚这个动作失效

      (这个在某种程度上说是为了适应 swiper 库的特性,

        因为是 swiper 实现不了 last->first 的跳转)

    -> 还有就是两次滚轮间隔太短,第二次滚轮事件失效(适应 swiper)

      利用系统的 Date 类记录前后两次滚动时间

      但是这个时间把握还有有点问题,我使用的是 300ms,但是这个是估计的,

      如果你就要在 300ms 左右操作的话,有一定几率触发 bug

-> 排版

-> 上面的文字竖排,width 的值设置得比较小

-> jquery 利用 index/padding-left 排版

3.3 亮点

-> `$(window).on("mousewheel",function () {...});`

-> 充足的资料

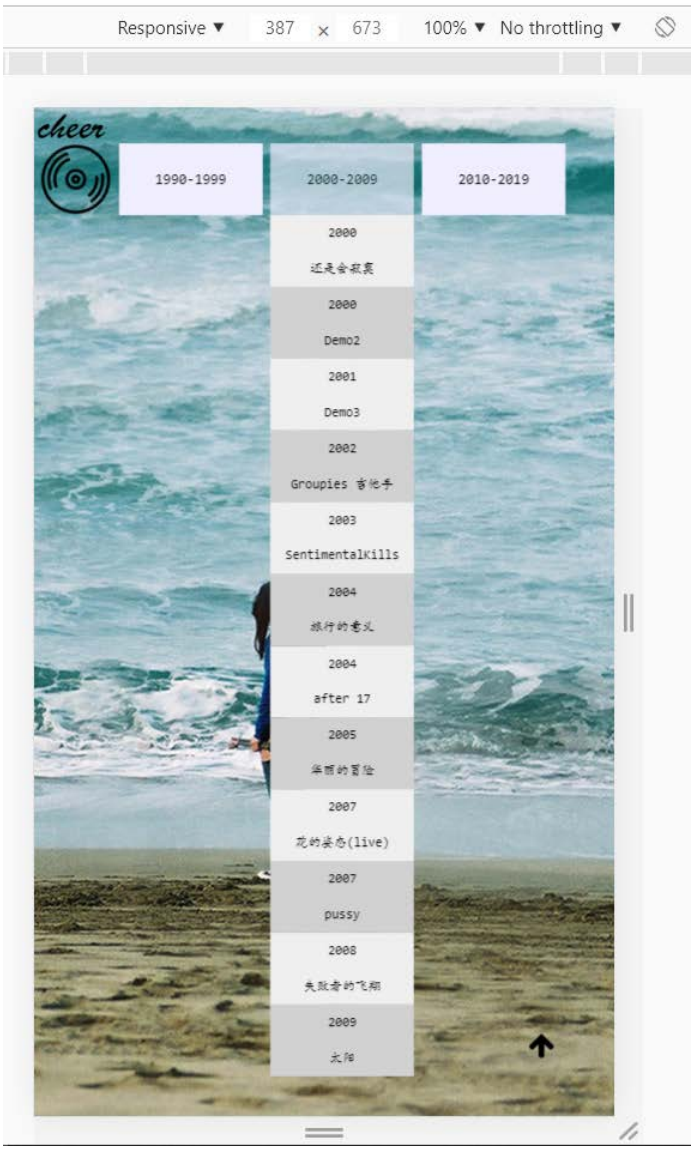
4 final_project/html/album.html(专辑)

4.1 loading 部分

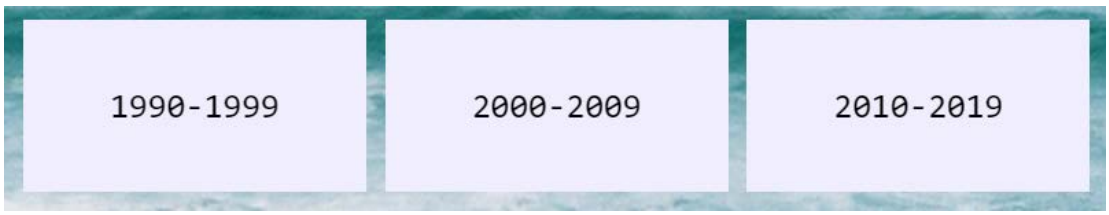
-> 主要就是好几个 div 的 animation/rotate,
然后调节一下大小不同齿轮的转动速率,看起来协调一些

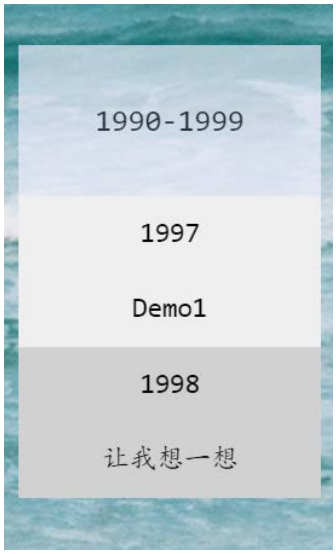
4.2 主体部分

-> 这边的建议是换成竖屏模式或者是调节浏览器的显现模式以获得最佳观感(如下)



-> 精华部分(旋转的 CD)


1990-1999
2000-2009
2010-2019


1990-1999
1997
Demo1
1998
让我想一想


CHEER'S WALKMAN
Demo 2 Cheer's Walkman
發行日期:2000年03月
發行公司:魔岩唱片
夜游 作词:陈绮贞 作曲:陈绮贞
Enemy 作词:陈绮贞 作曲:陈绮贞
温室花朵 作词:陈绮贞 作曲:陈绮贞
慢歌3 作词:陈绮贞 作曲:陈绮贞
微凉的你 作词:陈绮贞 作曲:林暐哲

-> 自上而下记作 CD,AA,BB,CC
-> $(".CD").on("click", function () {...});
click 函数的逻辑
if(存在 CC){
回收 CC(slideUp),return;
}
else{
if(存在 AA){
if(存在 BB){
BB(slideUp)->回调 AA(animate:left)
}
AA(animate);
}
}
-> BB 的具体某一个专辑点击触发命令
trigger(CD 的点击)
setTimeout(function(){CC(SlideDown)},2000);
//因为 trigger 函数没有回调函数
let height_now = [...];//用一个数组记录每一个 CC 下降的高度不同
-> 右下角的箭头点击可以返回最顶上
let time = setInterval(function(){...},1)以及learInterval(timer);
window.scrollTo(0, currentPosition);

4.3 亮点

-> jquery/界面的设计
-> 文字材料的获取(收集资料,制作出来搞了好久)
-> loading 界面

5 final_project/html/life-story.html(生活故事)

5.1 直接就是主体部分

5.1.1 最抓人的就是这个折扇部分了(这个部分是来自于网络)

-> 点击第一页会有合上/展开的效果
点击中间随便一页都会有将那一页置于中间的效果
javascript 实现也比较简单
-> 点击第一页时，设计一个初始位置
点击时实现合上(rotate(0deg))/初始位置的切换 toggle
-> 其他部分实现
设定间隔 A，B，C
举例说明，点击 5
5 居中(0deg)
5 后面的数字离 5 等间隔 A 排列(这个间隔 A 与初始位置间隔 A 相同)
5 前面的第一个数字(即 4)距离 5 为一个间隔 B
4 之前的数字等间隔排列，间隔为 C
-> 响应式布局
注意背景颜色和位置
-> 点击对应的数字可以实现慢慢滚动到对应的事件(改进了锚定,锚定执行的时候太快了)
改进了 toTop()函数，能够使得滚到任意指定位置

5.2 亮点

5.2.1 jquery 的充分运用

-> 第一方面这里的图片都是利用 jquery 添加的
-> 这里的图片长宽比不是直接调定的，而是通过生成一个 Image 对象来获得宽高比从而计算出 height 的

```
/*保证图片长宽高与原始相同*/
function image_harmony() {
    let divs = $("#show_cheer_life > .show_cheer_life_inbox");
    let left = '../image/life-story/';
    let right = '.jpeg';
    for (let i = 0; i < divs.length; ++i) {
        /*取得url*/
        let img_url = left + (i + 1) + right;
        let img = new Image();
        img.src = img_url
        // 加载完成执行
        img.onload = function () {
            let _width = parseInt(img.width);
            let _height = parseInt(img.height);
            let div = divs.eq(i).children(".show_cheer_life_inbox_img");
            let now_width = parseInt(div.css("width"));
            let height_to_be = now_width / _width * _height;
            div.css("height", height_to_be);
        }
    }
}
```

-> 当窗口的大小变化时，通过同样的方式来实现了长宽比恒定，从而使得图片在感官上获得较好体验

```
/*窗口大小变化事件*/
$(window).on("resize", function () {
    right_limit();
    image_harmony();
});
```

6 final_project/html/some_fun.html(周边)

6.0 这里分为三个部分，主界面以及两个小游戏

6.1 主界面

6.1.1 loading 部分

-> 这里是之前做的，比较简单，就是用了一张"gif"图片

6.1.2 主体部分

-> 表面上很简单，其实有挺多彩蛋的

-> 彩蛋 1，这里的空白背板支持画画，而且每一次动笔都是不同的颜色
(由于位置问题，每次窗口大小变化都会触发画板清零事件)
实现：
-> mousedown 触发 beginPath/lineto/strokrStyle(重设)事件
-> mousemove 触发 moveTo 事件
-> mouseup 触发清零事件(保证位置，dpi)

```
oC.onmouseup = function () {
    document.onmousemove = null;
    document.onmouseup = null;
};
```

-> 彩蛋 2

-> 右键点击会出现一串跟随鼠标的"台球"
实现：
-> 整个屏幕屏蔽初始的右键单击事件
-> document.oncontextmenu = function () {return false; }
-> 重新注册右键单击事件 mousedown(e.witch==3)事件
-> fadeToggle()实现菜单的出现与隐藏
-> 出现时，animation/setInterval 实现
-> 第一个台球直接设置 position 为鼠标点
之后 setInterval 依次设置后面鼠标的 position
数组越界时 clearTimeout

6.2 Game1 移动方块 (final_project/html/game_1.html)

-> 这个比较简单
-> 初始化整张图片
-> 用一个一维数组来记录 16 个方块
-> 依次设置背景图片，隐藏最后一张的背景图片
-> 随机将空白部分与周围的一个方块交换位置 500 次
由此生成一个一定有解的拼图
(其实这个地方完全可以不模拟交换背景图片，
而只是交换数组中元素，最后再设置图片)
-> 点击一张图片，直接给所有的 cube 注册"click"事件即可
-> 首先判断是不是和空白块相邻，不相邻则无效点击
-> 若相邻则交换位置

-> 每次执行交换顺序之后，判断是否还原(id 与背景图片是否匹配)

-> 这里的响应式布局也是通过 window 的 resize 事件触发的

6.3 Game 翻转方块(final_project/html/game_2.html)

-> 这个就比较复杂

-> 首先有几个要点

-> 主要是通过 rotate(addClass("current"))实现的

//因为 animate 不支持 transform 的缓变

-> rotate 设置为 backface-visibility:hidden;

但是这里会引发几个问题

-> 反转过去之后显示的就是背景色，这无法与消去之后区分，

因此这里用了两层 frame，第二层作为跟随动作存在(显示背景)

-> backface-visibility:hidden;之后无法响应 click 事件

而下层的 frame 由于 z-index 被上层的背景挡住也无法响应 click 事件

(解决方法下面讲)

-> 实现总体思路

-> 初始化

-> 首先设置一个数组 a[16];a[i]=i/2;

-> 打乱顺序

-> 随机产生两个 0-15 的数字 500 次，交换数组所对应两个位置的数字

-> 游戏开始

-> 点击的事件无法被响应，在这里就在每次点击时获取鼠标的位置

let x = e.pageX; let y = e.pageY;

若点击位置再 frame 内，则依次循环判断点击位置在哪一块 div 内

接着再执行 addClass 事件

-> 点击后实现翻转,并且 addClass("current")

每次求$(".current").length

1 个就只执行翻转

2 个就进行判断

若背景图片相同就执行 fadeOut，回调 removeClass

(同时设置跟随的 frame 执行 visibility:hidden)

否则就直接 removeClass

-> 问题:fadeOut 之后位置会被下一个填补

//因为用的是 float:left 排版

-> 解决方案就是在 fadeOut 之后 removeClass 之后

设置 visibility 为 hidden，再直接 show()

```
setTimeout(function () {
    back_divs.eq(index_1).css("visibility", "hidden");
    back_divs.eq(index_2).css("visibility", "hidden");
    my_rotate_divs.fadeOut(1000, function () {
        my_rotate_divs.removeClass("my_rotate");
        my_rotate_divs.css("visibility", "hidden").show();//也可以通过修改opacity的值实现
    });
}, 1000);
```

-> 也可以直接通过 animate({"opacity",0},1000);实现

-> 还有一个问题就是时间的把控，

很多不能通过简单的回调解决，或者说回调写起来很麻烦
这里主要是通过 setTimeout 来解决

-> 判断胜利条件用一个 int 来记录(==16 则胜利)

6.4 亮点

-> Game 的设计

-> Game_2 中的解决方案以及时间的把握

-> 主界面中的两个彩蛋

7 final_project/html/latest_news.html(最新消息)

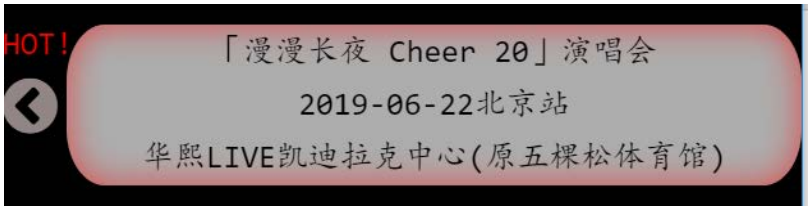
7.1 loading

-> 就是一个小球 animate

animation-timing-function: cubic-bezier(0.5,0.05,1,0.5);

7.2 主界面

7.2.1 mock 技术



这里的最新消息通过 mock 实现(实际中可以通过从服务器读取)

(上面左边的图标是通过 font-awesome 实现，

变化通过回调函数中的 addClass/removeClass 实现)

7.2.2 swiper(外部导入)

没啥好说的

7.3 没啥亮点