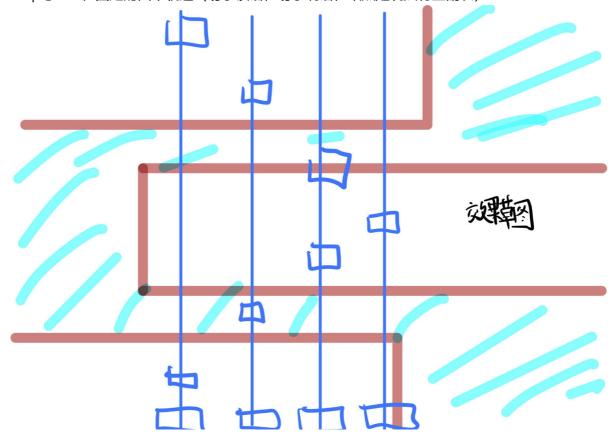
# QT homework

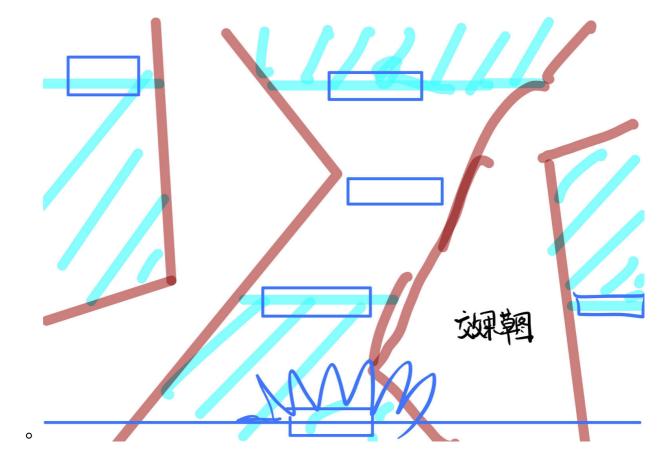
# 一个想法

# 关于玩法

- 左手控制键盘, 打击tap/hold
- 右手控制鼠标/键盘,保证人物处于蓝色区域(加分),无色区域,不在红色区域(扣血)
- 方案a: tap与hold在固定的四个轨道 (易于读谱, 易于制谱, 缺点是玩法有些割裂)



• 方案b: tap与hold在区域里 (不易于制谱.....)



### 关于铺面

- 由于编辑可视化制铺工具十分困难,且鸡肋()。为了对音方便并增加可扩展性,我们考虑使用arcade 铺面进行转谱
- 具体做法

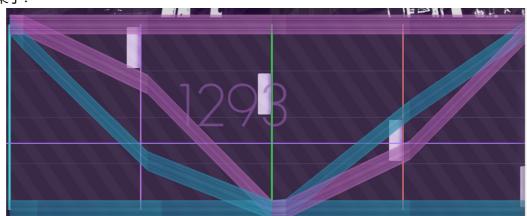
#### **Timing**

- Timing代码如下
- timing(t,bpm,beats);
  - t(ms): Timing起始位置,数字为整数
  - 。 bpm(拍/分钟): 节奏速度, 数字为不可省略小数点后两位的一个数
  - 。 **beats**(四分音个数(拍)):表示每多少个四分音符(拍)为一小节(出现一条小节线),数字为不可省略小数点后两位的一个数,比如4.00就是4/4拍,代表四拍一小节
    - 需要注意的是每个谱面一定要有一个t=0的Timing! 而且其bpm数值不可为负数!

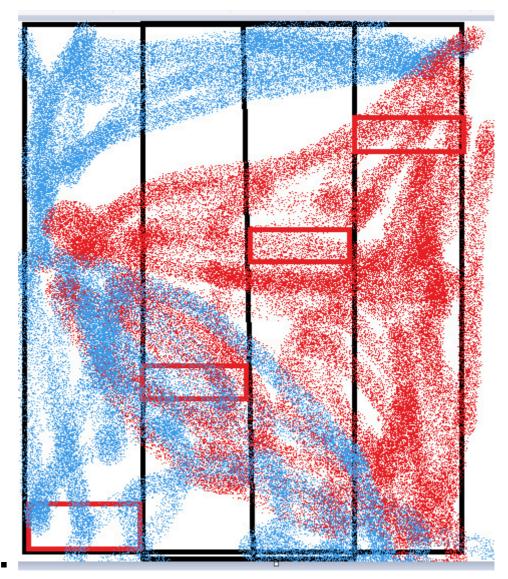
#### 地面Note & 地面Hold

- 地面Note & 地面Hold代码如下
  - o (t,lane);
  - o hold(t1,t2,lane);
    - t(ms):地面Note所在时间,数字为整数
    - t1,t2(ms):地面Hold物件开始/结束的时间,数字为整数,t1 < t2
    - lane(1-4): 物件所在轨道,一般情况下轨道的编号从左到右依次为1,2,3,4,绝大部分情况下也只需要用到1~4轨道

- Arc & Arctap代码如下
  - o arc(t1,t2,x1,x2,easing,y1,y2,color,hitsound,skylineBoolean);
    - **t1,t2**(ms): Arc物件开始/结束的时间,数字为整数,
      - t1可以等于t2
    - x1,x2: 代表Arc物件开始/结束时的横坐标
    - easing(b,s,si,so): Arc滑动方式。b = bezier, s = straight, si = sine in, so = sine out。
    - 当t1=t2时都是直的
    - y1,y2: 代表Arc物件开始/结束时的纵坐标
      - 在本游戏中全为0!
    - **color**: Arc颜色, 0蓝, 1红, 2绿。
    - hitsound: 这不重要
    - **skylineBoolean**: (false,true): 判定这一段Arc是不是天空Note的判定线 (黑线)。 false为普通Arc, true为天空Note的判定线,但是只要有Arctap本参数就无意义 (会将此Arc的类型强制转换为天空Note的判定线)
      - 当skylineBoolean=true,并且该Arc上有天空Note (Arctap)时,代码如下
      - arc(t1,t2,x1,x2,easing,y1,y2,color,hitsound,true)
        [arctap(tn1),arctap(tn2),.....,arctap(tnm)];
      - tn1,tn2,.....,tnm(ms): m个天空物件在这条判定线上的时间点,数字为整数,且不能超出t1和t2的区间
  - 。 ※※在本游戏中※※
    - 对于任意时间一个截面从左至右:
      - 检测这一段的arc
        - 遇到blue -> 寻找另一个blue -> pop掉两者,两者之间涂上蓝色
        - red同理,两者之间涂上红色
        - 允许重叠,若出现奇数对,忽略......
    - 举个栗子:



■ 应当表现为



■ 代码为:

```
(134100,1);
arc(134100,134400,-0.50,-0.50,s,0.00,0.00,0,none,false);
arc(134100,134250,1.50,1.50,s,0.00,0.00,1,none,false);
arc(134100,134250,1.50,-0.50,si,0.00,0.00,1,none,false);
arc(134100,134250,1.50,-0.50,b,0.00,0.00,0,none,false);
(134175,2);
arc(134250,134400,-0.50,1.50,si,0.00,0.00,1,none,false);
(134250,3);
arc(134250,134400,-0.50,1.50,so,0.00,0.00,0,none,false);
arc(134250,134400,1.50,1.50,so,0.00,0.00,1,none,false);
(134325,4);
```

#### Timinggroup

```
timinggroup(){
   //正常aff语句
};
```

每一个timinggroup语句块中的语句(物件)使用其内部单独的timing语句(并且至少包含一个timing语句),因此可以实现同时刻不同note流速

- timinggroup语句块中的timing语句不会产生小节线,小节线是由所有timinggroup语句块外面的timing语句决定的
- 一张谱面理论可以存在无限多个timinggroup语句块,也可以仅由t=0的timing和无数timinggroup组成
- 实现方法:
  - 。 简而言之, 就是加载的时候加载n张铺面, 叠加在一起。

#### noinput

```
timinggroup(noinput){
    //正常aff语句
};
```

- 此时本timinggroup内的物件只有显示效果,没有打击效果和物量,不会判定为击中
- noinput中的实体Arc和hold在经过判定线后依然会消失而不会直接穿过
- 关于arcaea铺面格式: https://wiki.arcaea.cn/index.php?title=谱面格式&action=edit
- 关于arcade下载: https://www.schwarzer.wang/mug#arcade
- 关于arcade教程: https://www.bilibili.com/video/BV177411b7QH

#### 一些机制

- 判定:
  - 。 进入红色区域超过一定时间才扣血, 但是回复少量能量
    - 与时间线性增加
  - 。 进入蓝色区域恢复一定血量, 回复少量能量
    - 同时有蓝与白不进入不扣血
  - o tap/hold击打回复少量血量,一定能量
    - 不击打扣血,不回复能量
  - (能量机制再说((((()))
- 人物移动控制
  - 类似osu! catch (
  - 。 左移到屏幕最右回到最左
  - · 方案一: 四个按键: 左右平移, 左右大跳/自动归位到最近的整格
  - · 方案二:两个按键:按下左右自动瞬移到最近的整格/半整格,长按平移
    - 或者可以考虑长按视作点的叠加,频率与bpm相对应
    - 长按左+点按右,长按左+长按右,
  - 方案三: (osu! catch) 三个按键: 左右平移, 一个按键加速
  - 。 铺面设计尽量做到全难度四指可玩
    - 铺面两指可玩,不同时落键与catch
    - 困难铺面落键与catch不在同一侧手
    - 发狂铺面 ^\_^
  - · 增加enter,扣除能量释放霸体
  - 困难:按键反射难以建立

- 结合剧情? .....
- 变速及铺面效果
  - o timing
    - 正向, 负向, 0
  - o timinggroup
  - nopinput
  - 。 特效?
- 铺面解锁
  - 。 达到一定的级别?

## 其他细节

- 开始界面
  - 新手教程!!!!
  - 。 开始游玩
  - 。 看板娘qwq
  - 。 设置
    - 调节延迟
    - 调节音量,背景亮度
    - etc.....
- 进入游戏
  - 。 选曲
    - 不同难度
    - 隐藏?
  - mod (简单模式? 严判模式?)
- 游玩
  - 。 暂停界面
    - 重新开始
    - 继续
    - 返回
  - 。 判定
    - 打击效果
    - 打击区间
- 结算界面
  - o acc
  - 。 成绩
    - 两种满分: ap/无伤/无伤ap
  - 。 返回
  - 。 看板娘qwq
- 剧情

# 其他

• 我们的项目github: https://github.com/HGAQ/QT-homework-