Arduino 设计——改进版别踩白块自动机

摘要：自动进行别踩白块的 arduino 设计已经较多且成熟。他们的通病是机械结构引起的得分上限，以及机器和游戏没有除了按下之外的交互。本设计在其设计上进行了创新改进。本设计通过使用舵机移动的方式，配以识别别踩白块游戏加速度的数学计算，抵消按下屏幕时的延迟，提高自动机得分。同时为了提高机器的实用性，增加了测定玩家生理极限得分的模式。目前尚存外形简陋，数学计算较为近似，双舵机多线程延迟难以处理的问题。但是经过验证，达到了预期目的。

目录

1 项目背景

本项目为基于 arduino 的别踩白块游戏相关设计。

1.1 项目背景及意义

通过调研目前的音游自动机，发现其最基本的结构就是：视觉识别-机械结构操作。在这种模式上，现阶段的音游自动机都在向适应更多的音游方向发展。而忽视了机械结构带来的分数极限。

这是因为，音游为了适应人的游戏体验，其所需的反应速度一定在人类的极限之下。而机械的反应速度显然优于人类。同时，音游自动机也不在乎对于游戏参数的自动识别，起因同样是，机械的反应速度已经足够了。

本项目即是为了弥补过去自动机在这两方面的缺点而设计的。

1.2 项目内容

本项目为了测试机械的极限反应速度，实现了一个基于 d3.js 的“别踩白块”游戏。

本项目的核心内容为识别游戏的黑块加速度，自动移动舵机，增加极限分数。

本项目的附加内容为，通过识别的黑块加速度，以及测量的使用者反应速度，反馈使用者游戏的生理极限分数。

1.3 本文结构

本文先介绍了项目的整体背景，然后介绍项目的代码架构。最后分析技术要点和进行测试结果展示。

附录中详细描述了本设计如何使用，以及相关库函数的知识。同时附上了主要代码。

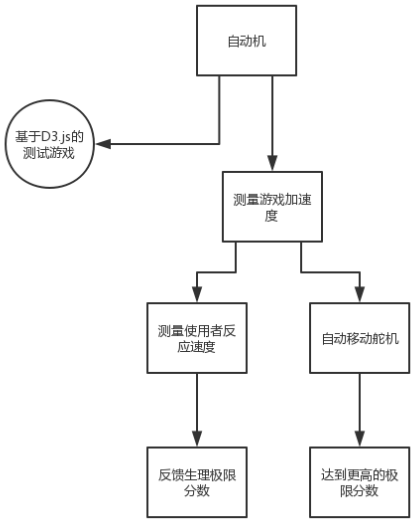
2 产品架构

产品分为 html 部分和 arduino 两大部分。html 部分的延时设计略有代码难度，arduino 部分的设计略有创新难度。

2.1. 整体需求

html 部分要求实现一个极限速度很高的动画游戏。arduino 部分要求实现一个能实现上述所有功能的机械结构以及相关代码。

2.2. 整体结构



2.3. 关键技术分析

关键技术分为三点：d3.js 的动画延时，双舵机的线程处理，推导加速度的数学过程以及误差分析。前两个技术细节将在下一节进行详细描述，本节将把重点放在数学过程上。
d3 是 javascript 的一个库，主要用于可视化处理，其对于动画的实现效果较强，但是实现动画的.transition 方法的延迟较难处理。

Arduino 的 servo 库对于舵机有很好的支持，只需要.write 方法就可以调整角度。但是此设计要求一个舵机变速转动，另一个舵机按需求转动。在 Arduino 的伪双线程处理下，延迟时间的处理就是很大的挑战。

因为别踩白块的游戏思路是速度线性变化，所以可以只用一次游戏就推导出其加速度，并使得机械装置根据其加速度进行调整适应。

加速度推导:
（所有单位换算都在 23.8 英寸的 1920*1080 屏幕下）

- 速度计算

已知
V=aT
当游戏结束时，即舵机无法在 30ms 即舵机无法在 30ms 的延时内按下。也即方块在 30ms 内下降了 70px。此时由于时间短暂，不考虑加速度。
所以方块此时的速度为 2.33px/ms。

- 误差计算:

考虑极限误差, 即 30ms 时下一个方块已经到达上方, 即方块在 30ms 内下降了 700px。速度为 23.3px/ms。但这种极限情况下不能进行近似匀速运动，只能以实际情况进行计算。另一个边界条件是，上一个方块的速度是 2.33px/ms。
即
v1=2.33
v2=23.3
d1=d2
由
v=tx
解出 x 即可。
解得此时速度为 16.33px/ms。即误差为 800%。但是不能忽略的是 x=0.2334px/ms^2。即游戏刚开始 1ms，普通的人已经战败了（取反应速度为 330ms）。
考虑到别踩白块的 x 在 1/20000 左右。在这种情况下误差可以忽略。
此设计的 html 游戏的 x=1/2000。
重复上述计算。
d=700px
v=2.33
解得 v'=2.47
此时最大误差为 6%。同时，因为可测量结果是叠加的时间，而不是瞬时的速度。所以调整舵机加速时的误差会更小

- 使用者得分

简单计算得到，你的应得分数为 t_{score}=70px * t /(4.66 px/ms *$t_{反应}$)

利用此公式即可达到计算你生理最高分的目的。

调节光敏探头的转动速度:

- 转动速度计算

舵机臂长 70px

h=70px*sinα

可转动角度为±30°

即可上下移动的 h 为 70px

即最大测试速度为方块在 15ms 内下降了 140px。

即方块的速度为 4.66px/ms

测量范围增大一倍。

当舵机角度为负时

v=tx

d=v*30ms=tx*30ms

d=h=70px*sin(30°)-70px*sinα

α=arcsin((35px-tx*30ms)/70px)

当舵机角度为正时

d=h+35px=70px*sin(30°)+35px

α=arcsin((30ms*tx-35px)/70px)

3 关键细节实现

- 动画延时细节实现:

d3 库提供 transition 方法，使得 svg 在两个不同状态之间平滑转化。transition 方法支持在链式写法中加入延时。

```
count += 2000 / c;
c = count / 5000;
    rec[i].transition()
        .duration(2000 / c)
        .ease("line")
        .delay(count)
        .attr("y", 600);
time[i] = count;
```

因为不同的下降下来的方块是需要分开处理的，所以设计一个变量来记录延时，将延时叠加，同时也可以根据记录的延时提供最后的分数。

而此处的延时会影响之后的所有步骤，包括检测鼠标按下，以及弹出分数。这两个被动画延时的部分，使用 settimeout 方法解决。具体解决方案见代码。

- 舵机双线程解决方案:

在此设计中需要让一个舵机按规律变速转动，一个舵机在光敏电阻检测到黑块时按下。此设计应该使用双线程来实现。故使用 MsTimer2 来实现。

MsTimer2 提供 set 函数来使得每隔一段时间便中断程序来执行某个函数。也就是说，这是软件层面的双线程，即延迟实际上会相加。

考虑到光敏电阻检测要求非常频繁，否则将极大的影响极限分数。故将光敏电阻检测servo()函数放入中断中。同时为了保持 30ms 的间隔不变以进行对比，将中断间隔设置为20ms，舵机延迟设置为 10ms.但是此举使得舵机的按压距离十分短。为了解决此问题，将鼠标外壳卸下，减小按下所需压力。

```
int doublecheck(){
  MsTimer2::stop();
  MsTimer2::set(20,servo);
  MsTimer2::start();
  if (pushbotton())
     {
      for (int i=0;i<60;i++)
      moveotherservo();
     }
}
```

主线程中的舵机使用 for 循环和 delay 进行变速转动。此处带来的误差较小，但是中断可能会影响此处误差，但是这与 servo 函数带来的误差相比可以忽略，详见下节。

4 测试及成果展示

在 a=1/5000 的情况下，游戏得分从 14000 增加至 17000，增加了约 1/7，未达到预期的 100%。

可能的原因如下:

- 由于黑块的出现的离散的，所以不能达到预期分数。考虑到 17000 的下一个阶段分数小于 18000，即实际误差不是由此项引起的。
- 移动第二个舵机的时间影响到了按下鼠标。取两次中断的延迟为 10ms，误差即高达 25%。当两次中断的延迟为 30ms 时，误差便已经极大。误差很有可能是由此项引起的。
- 舵机移动距离太短，可能未按下鼠标。

得分	次数	占比
14000	2	10%
17000	7	70%
6000	1	10%
0	0	10%

为了检测此项，进行了 10 轮游戏。可以看到，舵机未按下鼠标是有可能的。0 和 6000 分数的两轮，可以确定是因为未按下鼠标而打断游戏。

其它各项功能均达到目标。实现较为成功，但是双舵机的转动和多线程设计尚需改进。

5 总结与展望

本设计是基于 arduino 的一款音游周边产品。其改良了以往音游自动机缺乏读取游戏参数和缺乏和玩家交流的缺点，增加了测量玩家生理极限得分，以及增加了改进自动

机机械结构限制的最高分数部分。

缺点：

本设计只能针对自己完成的 html 游戏，还需要将探头和舵机分别粘至屏幕和鼠标上。设计简陋。

双线程的延迟处理不合理，导致了理论增加 100%分数的设计只增加了 1/7 的分数，同时带来了 10%的失败风险。

改进：

优秀的时序设计是解决此设计遇到的双线程处理问题的根本手段。本程序对于时间的要求十分精确，arduino 的伪双线程运作带来了 ms 级别的时间估计误差。这是待解决的问题。

同时，更好的包装设计也是此设计走向完美的改进之一。

附录：

本设计使用说明：

此设计可连接充电宝，电脑 usb 供电口等电压为 7-12V 的电源供电。

使用时，开机可见 LED 灯亮，按压黑色按钮。LED 灯会在 1s 后熄灭，熄灭 1s 后亮起。亮起后松开黑色按钮。此时反应速度已录入。

录入反应速度后，将含探头的舵机置于基准线上，将不含探头的舵机置于鼠标上，必要时可加以固定。此时含探头的舵机不会运动。

待游戏结束后，将分数通过旋钮录入。此时你的生理极限分数已经显示在 LED 屏幕上。录入分数后，将旋钮顺时针旋转至底。重新开始游戏，含探头舵机开始转动以提高分数，可以注意到，舵机转动带来的分数提高约为 1/7。

此时重启设计可以开始下一轮使用。

d3.js 说明

d3 的全称为（Data-Driven Documents），其为一个数据驱动的 javascript 库。其主要特性是可以将元素绑定数据，从而修改元素。一般用于数据可视化方面。

本设计未用到其数据驱动的特性，使用其的目的只是为了熟悉 d3 的时序设计。其动画对于时序的安排不是很完善。

主要用到了.transition 方法来进行 svg 元素的过渡。

使用到的 arduino 库说明

本设计使用了 MsTimer2, LiquidCrystal, Servo 三个库：分别是对应中断，液晶屏，舵机

的使用。
MsTimer2: 此库使用到了 set，start，stop 函数，分别对应中断的设置，开始和终止。
LiquidCrystal: 此库使用到了 lcd, print, clear, setCursor 方法，分别是，初始化，显示，清屏和定位。
Servo：此库使用到了 write 方法，为设定角度。

Bilibili 视频简介:

https://www.bilibili.com/video/av31798531/

GitHub 地址:

https://github.com/evenlasting/arduino/

主要代码:

- Html:

```
<script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>

<html>
<head>
    <meta charset="gb2312">
    <title>HelloWorld</title>
</head>
<body>
    <script type="text/javascript">
        var svg = d3.select("body")
            .append("svg")
            .attr("width", 500)
            .attr("height", 550);
        var line = svg.append("line")
            .attr("x1", 0)
            .attr("x2", 500)
            .attr("y1", 500)
            .attr("y2", 500)
            .attr("stroke", "black")
            .attr("stroke-width",2);

        var count = 0;  //延时的时间数
        var c = 1; //分母
        var ni = 0;//delay count
```

```
        var dtime = new Array();//delay array
        var time = new Array();
        var click = new Array();
        var rec = new Array();
        for (var i = 0; i <= 100; i++)
            rec[i] = svg
                .append("rect")
                .attr("id", i)
                .attr("x", 250)
                .attr("y", 300)
                .attr("fill", "black")
                .attr("width", 40)
                .attr("style", "cursor:pointer") //change the mouse
                .attr("height", 70)
                .on("mousedown", function () { if(this.getAttribute("y") < 500 && this.getAttribute("y") > 430) click[this.getAttribute("id")] = 1;
/*click[rec.attr("num")] = 1; */ });
        for (var i = 0; i <= 100; i++) {
            var x = 250, y = 30;
            //click :means that push and move out
            count += 2000 / c;
            c = count / 5000;
                rec[i].transition()
                    .duration(2000 / c)
                    .ease("line")
                    .delay(count)
                .attr("y", 600);
            setTimeout("if (!pushnum()) ans()", count);
            time[i] = count;
            //console.log(count, c);
        }
        //setTimeout("ans()",count);
       // rec[0].
        function ans() {
            for (var i = 0; i <= 100; i++) {
                if (click[i] != 1) break;
            }
            alert("your score:"+((i>0)?time[i-1]:0));
        }
        function pushnum() {
            console.log("click:",click);
            console.log(dtime, ni);
            var ans = 0;
            for (var i = 0; i <= 100; i++) {
                if (click[i] == 1) ans++;
            }
```

```
            console.log("ans:",ans);
            if (ni>1&&dtime[ni-1] == ans) return false;
            dtime[ni++] = ans;
            return true;
        }


    </script>
    <button onclick="click()">黑块来了</button>
</body>
</html>
```

- Arduino

```
void setup() {                                  //init
  lcd.begin(16,2);
  Serial.begin(9600);
  myservo1.attach(9);
  other.attach(A3);
  pinMode(LED_BUILTIN,OUTPUT);
  pinMode(6,INPUT);
  digitalWrite(LED_BUILTIN,LOW);
  int tpush=check();
  MsTimer2::set(40,LED);
  MsTimer2::start();
}

void LED(){                                     //interrupt1
  int bscore=analogRead(A1);
  int score=map(bscore,23,1018,2000,15000);
  //Serial.println(score);
  lcd.clear();
  lcd.print("auto s:");
  lcd.print(score);
  lcd.setCursor(0,2);
  lcd.print("your s:");
  long yours=(long)change*score/tpush;
  lcd.print(yours);
}

void servo(){                                   //servo1
  int x=analogRead(A0);
  if (x>1000) {myservo1.write(0);
  delay(15);
  myservo1.write(90);
```

```
  delay(100);}
  //Serial.println(x);
}

void loop() {                                  //check for the last mode
  // put your main code here, to run repeatedly:
  servo();
  doublecheck();
}

int doublecheck(){                             //check for the last mode

  if (begindoublecheck())
     {
        MsTimer2::stop();
        MsTimer2::set(20,servo);
        MsTimer2::start();
        moveotherservo();
     }
}

int begindoublecheck(){                        //check the A1
  static int rem=INF;
  int x=analogRead(A1);
  if (rem>change&&x>change) {acce=rem;rem=x;}
  if (rem<change&&x<change) return 1;
  return 0;
}

int check(){                                   //check the reaction rate
  digitalWrite(LED_BUILTIN,HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN,LOW);
  delay(1000);
  digitalWrite(LED_BUILTIN,HIGH);
  int time=millis();
  while (digitalRead(6));
  return millis()-time;
}

void moveotherservo(){                         //move the other servo

for (int i=0;i<=60;i++)
{
  if (i<30)
```

```
  other.write((int)asin(((-30*acce*i+35)/70)*180/3.14));
  else
  other.write((int)(asin(30*acce*i-35)/70*180/3.14));
  delay(1);
}
};
```