Professionale

Scuola Arti e Mestieri Trevano

Sezione informatica

Reaction Game 2.0

Titolo del progetto: Reaction Game 2.0

Alunni: Dyuman Bulloni, Erik Stalliviere

Classe: Info 3
Anno scolastico: 2017/2018

Docente responsabile: Francesco Mussi

Centro Profess**ional**e

SAMT – Sezione Informatica

ReactionGame 2.0

1	Introduzione	3
	1.1 Informazioni sul progetto	3
	1.2 Abstract	3
	1.3 Scopo	3
	Analisi	
	1.4 Analisi del dominio	
	1.5 Analisi dei costi e benefici	3
	Analisi e specifica dei requisiti	4
	1.6 Pianificazione	5
	1.7 Analisi dei mezzi	6
	1.7.1 Software	6
	1.7.2 Hardware	6
2	Progettazione	6
	2.1 Design dei dati e database	6
	2.2 Design delle interfacce	7
3	Implementazione	
	Cambiamenti codice modalità	19
	Collegamento Arduino → PHP → DB	19
	Parte fisica	
	Display 7 Segmenti	20
4	Test	26
	4.1 Risultati Test	
	4.2 Mancanze/limitazioni conosciute	27
5	Consuntivo	27
6	Conclusioni	28
	6.1 Sviluppi futuri	28
	6.2 Considerazioni personali	28
7	Bibliografia	28
	7.1 Sitografia	28
8	Allegati	28

Pagina 3 di 28



Introduzione

1.1 Informazioni sul progetto

Allievi:

Dyuman Bulloni, Responsabile Software, Responsabile Organizzazione

Erik Stalliviere, Responsabile Hardware

Docente Responsabile: Francesco Mussi. Scuola: Scuola d'Arti Mestieri Trevano.

Sezione: Informatica. Materia: Modulo 306.

Data Inizio Progetto: 16.03.2018 Data Consegna Progetto: 18.05.2018

1.2 Abstract

This documentation explains how continue the project "ReactionGame" and finish the main elements. In particular, the two seven segments displays, the new modalities and the php code will be show step by step.

ReactionGame 2.0

1.3 Scopo

Lo scopo di questo progetto consiste nel riprendere il punto di arrivo della prima versione del progetto ReactionGame (che aveva lo scopo di ricreare una BATAK machine, un gioco per allenare i propri riflessi, maggiori dettagli nella versione 1.0 del progetto) e portare a compimento tutti gli elementi mancati o non completati appunto in questa prima versione del lavoro. Le cose che con questo progetto dobbiamo portare a termine sono l'aggiunta dei due display sette segmenti per la visualizzazione di tempo e punteggio, una pagina di login che comprende una classifica dei punteggi per modalità, il completamento di tutte e 23 le modalità inizialmente previste e l'implementazione di una pedana di gioco.

Analisi

Analisi del dominio 1.4

Per maggiori informazioni, vedere il seguente paragrafo nella documentazione della versione 1.0 del progetto.

Analisi dei costi e benefici 1.5

1.5.1.1 Versione 1.0

Categoria	Costo
Telaio BATAK	700 CHF
12 Bottoni led	70 CHF
Display 7 segmenti	2 CHF x2
Display LCD	10 CHF
Cavi	10 CHF
Arduino MEGA	40 CHF
Ethernet Shield 2	15 CHF
Personale	94 ore * 4 persone * 50 CHF = 18 800
TOT	19 649 CHF

SAMT – Sezione Informatica

ReactionGame 2.0

Pagina 4 di 28

1.5.1.2 Versione 2.0

Categoria	Costo
Arduino UNO	20 CHF
Personale	72 ore * 2 persone * 50 CHF = 7 200
TOT PARZIALE	7 220 CHF
TOT V 1.0 + V 2.0	26 869 CHF

Analisi e specifica dei requisiti

ID: REQ-01				
Nome	Vecchie funzioni, Software			
Priorità	1			
Versione	2.0			
Note				
	Sotto requisiti			
Le modalità mancanti (5,6,9,10,20,21,22,23) devono essere implementate e integrate perfettame codice principale.				
Do La classifica dei punteggi per modalità deve essere implementata e funzionante.				

ID: REQ-02					
Nome	Vecchie funzioni, Hardware				
Priorità 1					
Versione 2.0					
Note					
	Sotto requisiti				
I due display 7 segmenti devono essere correttamente collegati e mostrare gli elementi descritti nel modalità (solitamente tempo e punteggio).					
Do La pedana deve essere analizzata, trovata la soluzione ideale da applicare e agire comp pratica.					

Professionale

SAMT – Sezione Informatica

ReactionGame 2.0

Pagina 5 di 28

ID: REQ-03			
Nome Nuove funzioni			
Priorità 1			
Versione	2.0		
Note			
Sotto requisiti			
Il collegamento tra Arduino e PC non sarà più tramite ethernet, ma sviluppato tramite un Fishin collegamento Wireless.			

1.6 Pianificazione

Professionale

SAMT – Sezione Informatica

ReactionGame 2.0

Pagina 6 di 28

1.7 Analisi dei mezzi

1.7.1 Software

Arduino 1.8.1.10
Fritzing 0.9.3b
Microsoft Word 2013
Gantt Project 2.8.5
PowerPoint 2013
HeidiSQL
Notepad++ 7.5
Google Chrome 64.0
SourceTree
WinRAR 5.50
Foxit Reader 8.3.1

1.7.2 Hardware

2 PC, uno per collaboratore Arduino MEGA. Arduino UNO Shield Ethernet 2 2 display 7 segmenti 1 display LCD 1 Pedana

Per gli elementi necessari alla costruzione del telaio, consultare la documentazione 1.0 del progetto.

2 Progettazione

2.1 Design dei dati e database

Per la visualizzazione del design utilizzato, consultare la documentazione 1.0 del progetto.



2.2 Design delle interfacce

Qui di seguito verranno mostrate e spiegate tutte le modalità attraverso un piccolo schema con relativa spiegazione.

Nella Batak i numeri a destra sono pari (0, 2, 4, 6, 8) e a sinistra i dispari (1, 3, 5, 7, 9) quello sopra "#", mentre quello sotto è "@". Il riquadro in basso corrisponde al display LCD, non così fondamentale da dover essere descritto.

Viene segnato il tempo

I bottoni si accendono tutti casualmente, il gioco varia la sua velocità in base ai riflessi del giocatore. Durata 1 min.

Figura 1: Modalià 1, Cumulativo 60[s], senior

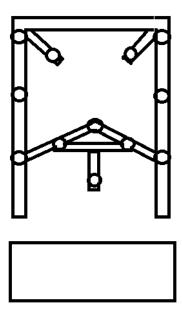
Viene segnato il punteggio

Identico alla modalità 1 solo che il tempo è di 5 min.

Figura 2: Modalità 2, Maratona 300[s], senior



Viene segnato il tempo



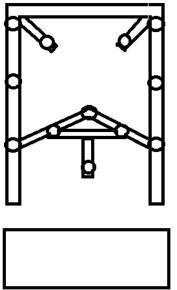
Colpi corretti

Il giocatore deve colpire, il più velocemente possibile, 50 pulsanti che si illuminano sequenzialmente. Il tempo è mostrato al decimo di secondo su un display.

Allo scadere di 100 secondi, la sequenza finisce ed il numero di colpi corretti viene mostrato sul secondo display.

Figura 3: Modalità 3, Corsa 50 pulsanti, senior

Viene segnato il tempo



Pulsanti rimanenti

In questo programma vengono utilizzati solo i pulsanti che si trovano agli angoli estremi (simulazione di un portiere).

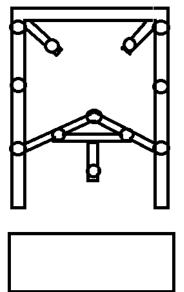
Si illuminano casualmente e restano illuminati per un secondo. Si conta alla rovescia da 100 fino a 0 e le due grandezze, numero restante e tempo sono mostrati sui due display.

Ogni volta che si preme un pulsante sbagliato, la sequenza rimanente viene accelerata.

Figura 4: Modalità 4, Stretching angolare (100 pulsanti), senior

Pagina 9 di 28

Viene segnato il tempo



Pulsanti rimanenti

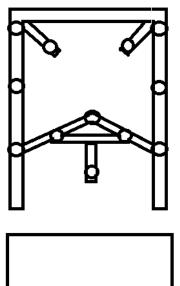
Variante del 4) utilizzando però tutti i pulsanti.

Durante la sequenza, quando il giocatore vede un pulsante che lampeggia (simulante

un tiro di palla sbagliato) NON DEVE colpirlo. Colpirlo per sbaglio genera un avviso verbale "Vuoto" ed una perdita di 5 punti. Similmente, se tutti e tre pulsanti centrali si illuminano contemporaneamente, il giocatore deve fare un salto indietro ed interrompere una barriera luminosa o premere una pedana con i piedi. Se questa azione non è fatta, si perdono 5 punti. Il tempo per reagire a questa situazione è di 1 secondo.

Figura 5: Modalità 5, senior

Livello attuale





Questo test si basa sul test di Léger o Beep test o Bleep test, utilizzato comunemente per misurare il massimo consumo di ossigeno VO₂max di un atleta.

Il test ha 10 livelli (30 pulsanti per livello) con la frequenza variante da 1.4 secondi per il primo livello giù fino a 0.5 secondi al decimo livello.

I livelli vengono mostrati su un display.

La sequenza viene interrotta se I giocatore sbaglia tre colpi consecutivi.

Al primo colpo sbagliato viene emesso un "Beep", al secondo due ed al terzo tre.

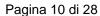
Se ad uno o due colpi sbagliati segue un colpo corretto, il conteggio degli errori ricomincia dall'inizio.

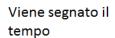
Il giocatore deve adattarsi alla velocità crescente della sequenza.

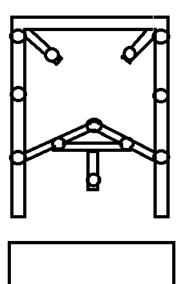
L'unico suono utilizzato in questa sequenza sono i "Beep" di awiso.

Figura 6: Modalità 6, Test di Legéro o Bleep test, senior







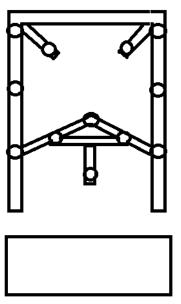


Colpi corretti

In questo programma 50 pulsanti sono attivati successivamente in modo casuale un display mostra il conteggio alla rovescia mentre l'altro display indica il numero di colpì corretti. Il tempo per colpire è di 1 secondo. Se il giocatore sbaglia il colpo o colpisce dopo che il tempo è scaduto, la sequenza accelera.

Figura 7: Modalità 7, 50 pulsanti temporizzati, senior

Viene segnato il tempo



Colpi corretti

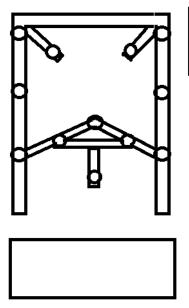
Una stafetta di 4 giocatori (30 secondi a testa). Un display mostra il tempo mentre l'altro i colpi corretti. (programma ideale per eventi o mostre)

Figura 8: Modalità 8, Staffetta 4 giocatori, senior



Pagina 11 di 28

Addendo 1 / Risultato risposte corrette



Addendo 2

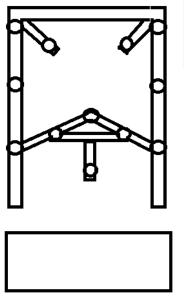
All'inizio di questo programma, premendo il pulsante "E", il sistema richiede di premere un pulsante numerico da 2 a 9 per impostare il tempo di reazione per rispondere alla somma richiesta.

Al giocatore vengono poste 8 semplici addizioni, i due addendi appaiono sui due display separati, il giocatore deve premere il pulsante con il numero corrispondente alla somma nel tempo preselezionato all'inizio.

Alla fine, il punteggio totale (su 8) viene indicato su un display, se il risultato è di 8 su 8, ci sarà un riconoscimento vocale ed un applauso. (ricordarsi che i numeri pari sono sulla destra mentre quelli dispari sono sulla sinistra)

Figura 9: Modalità 9, Reazione somma matematica

Viene segnato il tempo





All'inizio il giocatore deve selezionare il pulsante corrispondete alla tabellina che vuole esercitare (se vuole la tabellina del 3, premerà il pulsante con il numero corrispondente).

Il sistema proporrà verbalmente ed i modo casuale una moltiplicazione nella tabellina scelta (es: 3 per 5, 3 per 2 ecc,). Il giocatore premere il pulsante corrispondente al risultato. Se il risultato è di due cifre, dovrà premere due pulsanti in sequenza, prima le decine e poi le unità.

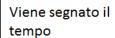
Il test propone 12 somme, il tempo è illimitato, lo scopo è di migliorare il proprio tempo nei tentativi successivi. Il tempo è indicato su un display.

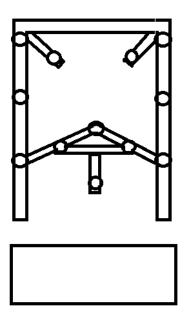
Figura 10: Modalità 10, Tabelline test velocità



Pagina 12 di 28

ReactionGame 2.0





Colpi corretti

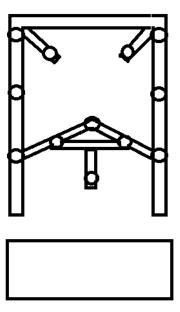
In questo programma, come nel 4), vengono utilizzati solo i pulsanti angolari.

Vengono attivati successivamente 25 pulsanti in modo causale. Ogni pulsante rimane acceso per 1 secondo. Un display mostra il conteggio alla rovescia 25 -> 0 mentre l'altro indica il numero di colpi corretti.

Se il giocatore preme un pulsante sbagliato o fuori tempo massimo, la sequenza viene accelerata.

Figura 11: Modalità 11, Stretching angolare (25 pulsanti)

Viene segnato il tempo



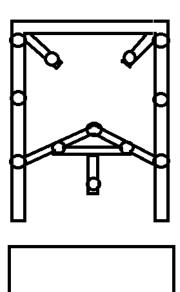
Colpi corretti

Come il programma 11) ma con 50

Figura 12: Modalità 12, Stretching angolare (50 pulsanti)



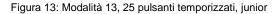
Viene segnato il tempo



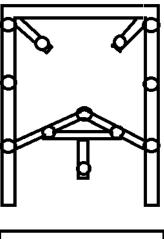
Colpi corretti

In questo programma 25 pulsanti sono attivati successivamente in modo casuale un display mostra il conteggio alla rovescia mentre l'altro display indica il numero di colpì corretti.

Il tempo per colpire è di 1 secondo. Se il giocatore sbaglia il colpo o colpisce dopo che il tempo è scaduto, la sequenza accelera.



Viene segnato il tempo



Colpi corretti

In questo programma 50 pulsanti sono attivati successivamente in modo casuale un display mostra il conteggio alla rovescia mentre l'altro display indica il numero di colpì corretti.

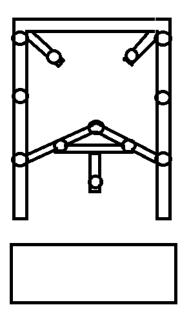
Il tempo per colpire è di 1 secondo. Se il giocatore sbaglia il colpo o colpisce dopo che il tempo è scaduto, la sequenza accelera.



Figura 14: Modalità 14, 50 pulsanti temporizzati, junior



Viene segnato il tempo



Numero pulsanti premuti correttamente

Questo programma prevede un'accensione casuale dei pulsanti per un periodo di 30 secondi

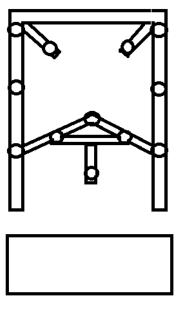
I pulsanti si accendono sequenzialmente e rimangono accesi finché non sono premuti. È il giocatore che determina la velocità del gioco:

Vengono conteggiati quanti pulsanti sono stati premuti correttamente nel periodo di tempo

Tempo e numero vengono visualizzati sui due display superiori.

Figura 15: Modalità 15, Cumulativo 30[s], junior

Viene segnato il tempo



Numero pulsanti premuti correttamente

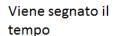
Come il 15) ma per 60 secondi

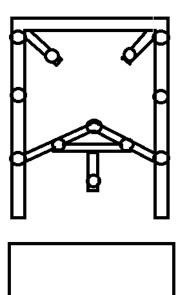
Figura 16: Modalità 16, Cumulativo 60[s], junior



Pagina 15 di 28

ReactionGame 2.0

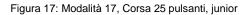




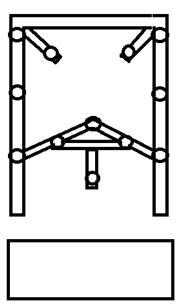
Numero pulsanti premuti correttamente

Il giocatore deve colpire, il più velocemente possibile, 50 pulsanti che si illuminano sequenzialmente. Il tempo è mostrato al decimo di secondo su un display

Allo scadere di 100 secondi, la sequenza finisce ed il numero di colpi corretti viene mostrato sul secondo display.



Viene segnato il tempo



Numero pulsanti premuti corretamente

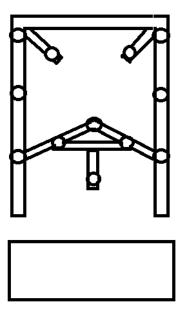
Come il 17) ma con 50 pulsanti.

Figura 18: Modalità 18, Corsa 50 pulsanti, junior



Pagina 16 di 28

Viene segnato il tempo

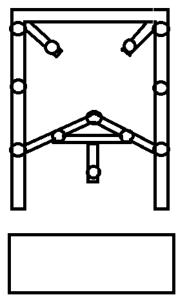


Numero pulsanti premuti correttamente

Come il programma 15) ma il tempo è di 180 secondi, ideale per esercizi aerobici.

Figura 19: Modalità 19, Maratona 180[s], junior

Livello attuale



Viene segnato il punteggio

Questo programma si basa sul gioco Simon, popolare qualche anno fa. Lo scopo del gioco è di premere i pulsanti riproducendo la medesima sequenza mostrata senza fare errori.

All'inizio, nel primo livello, vengono illuminati casualmente quattro pulsanti seguiti da due "Beeps". Il giocatore deve quindi riprodurre correttamente la sequenza.

Se il giocatore ha completato il livello, nel successivo, viene riproposta la medesima sequenza aggiungendo un pulsante. Ogni sequenza termina sempre con due "Reen"

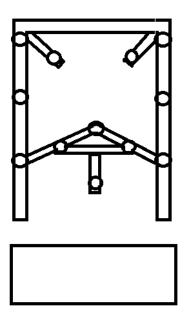
Questa progressione continua fino a che il giocatore commette un errore o fino a quando completa tutti i

Il livello ed il punteggio accumulato vengono indicati sui due display.

Figura 20: Modalità 20, Semplice gioco Simon



Punteggio massimo



Punteggio attuale

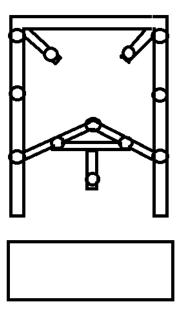
Questo programma accende casualmente fino a 6 pulsanti tenendoli accesi per un tempo scelto dal giocatore (selezionando dapprima il tasto "E" e poi il tasto con il numero corrispondente al tempo scelto da 1 a 8 secondi).

Allo scadere del tempo il sistema emette due "Beep" ed il giocatore deve premere in qualsiasi ordine gli stessi pulsanti che si erano accesi.

Per ogni schema corretto il punteggio viene incrementato, un display indica il punteggio massimo possibile, l'altro, quello raggiunto.

Figura 21: Modalità 21, Flash test

Punteggio massimo



Punteggio attuale

Complemento al 21) in questo caso il giocatore deve premere i pulsanti che NON sono stati accesi.

Figura 22: Modalità 22, Anti flash test

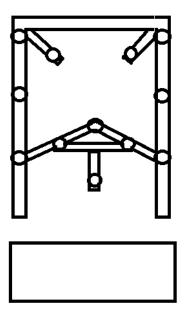


SAMT – Sezione Informatica

ReactionGame 2.0

Pagina 18 di 28

Viene segnato il tempo



Viene segnato il punteggio

Il giocatore inizia premendo il pulsante "#" ed in seguito i pulsanti che vuole utilizzare (massimo 11) terminando di nuovo con il tasto "#". I pulsanti si accendono quindi casualmente e sequenzialmente e rimangono accesi finché non sono premuti.

sono premuti.

Si il giocatore che determina la velocità del gioco:

Vengono conteggiati quanti pulsanti sono stati
premuti correttamente nel periodo di tempo.

Tempo e numero vengono visualizzati sui due
display superiori.

Lo schema viene riproposto per 10 volte variando la sequenza.

Figura 23: Modalità 23, Reazione veloce

Professionale

SAMT - Sezione Informatica

ReactionGame 2.0

Pagina 19 di 28

3 Implementazione

Cambiamenti codice modalità

La conoscenza e l'esperienza necessaria alla creazione delle modalità è semplicemente il fatto che sia stata la stessa persona a farsi carico della riuscita di tutte le modalità.

Le categorie descritte nella prima versione del progetto sono rimaste, con le seguenti aggiunte: Alla Categoria 3 è stata aggiunta la modalità 5, visto il comportamento molto simile di quest'ultima con le restanti della categoria. È stata gestita in modo che se il numero dei bottoni indicato alla chiamata della funzione sia 11, esegua i dovuti cambiamenti che permettano di giocare la modalità 5. Nello specifico, la possibilità che ci siano bottoni che lampeggino e che quindi non vadano premuti e l'accensione dei 3 bottoni centrali (che formano un triangolo) che indicano la necessità di premere la pedana, che come verrà descritto in seguito, è stata temporaneamente sostituita con il pulsante "#".

Le restanti modalità, nonostante per alcune avessimo anche codice relativamente funzionante, che con un po' di tempo saremmo riusciti a sistemare (cosa fatta con modalità 21 e 22), per altre abbiamo ritenuto più adatto rifarle completamente da 0, causa una logica disordinata ed evitabile (modalità 9 e 23). La modalità 6, 10 e 20, così come le 4 sopracitate, sono state tutte implementate in modo singolo, con una categoria a loro completamente dedicata.

Collegamento Arduino → PHP → DB

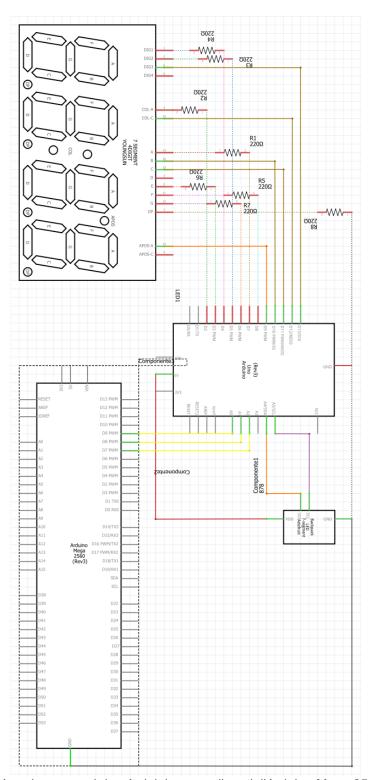
Il collegamento tra questi ultimi non è stato modificato dalla prima versione del progetto (e quindi trovata descritta nel dettaglio nella documentazione di quest'ultima), con l'eccezione di un piccolo miglioramento nella gestione della pagina php principale (index.php) e per il fatto che la connessione da Arduino al Server (PC) sia una sola, per punteggio e modalità, invece che dividere i due aggiornamenti rendendo il codice inutilmente più macchinoso. La classifica è stata aggiunta direttamente nella pagina di gioco della persona che ha eseguito l'accesso. Questa visualizza appunto l'ordine decrescente dei punteggi avuti per la modalità giocata, con tanto di riga colorata di rosso per la partita appena terminata oppure di verde, per qualsiasi altra partita giocata dalla persona loggata per quella modalità.

Parte fisica

Per quanto riguarda la parte fisica del progetto, le uniche cose cambiate sono la presenza di due display 7 segmenti. La pedana, tanto discussa in corso di progetto sul come realizzarla, alla fine non è mai stata implementata. L'idea iniziale consisteva in un semplice gioco di molle che avrebbe permesso di chiudere un circuito e far così leggere il segnale. Si è poi arrivati all'ipotesi di rendere il tutto wireless o Bluetooth, per non usare dei cavi, scomodi nel caso si volesse variare la posizione della pedana, con un Fishino che permettesse di appunto inviare il segnale della pedana (che alla fine non è nient'altro che un output booleano) all'Arduino MEGA. Nonostante stessimo procedendo in questa direzione, dopo qualche settimana persa a implementare questa idea, è poi stato convenuto dal Docente Barchi che sarebbe stato molto meglio utilizzare delle resistenze flessibili per la creazione della pedana. Anche in questo caso, dopo un po' di tempo percorso in questa direzione, si è arrivati alla conclusione della fragilità delle suddette resistenze, che avrebbero rischiato di non resistere a diversi tipi di peso. Ormai il tempo a disposizione era agli sgoccioli e una volta tornati all'idea precedente, con l'utilizzo di un Fishino, abbiamo potuto constatare che il tempo per far arrivare il pezzo principale mancava e che le idee sulla pedana erano diventate troppe e sconclusionate per poterne effettivamente tirare fuori qualcosa in una settimana. È stata così sostituita, almeno in questa versione del progetto, in un semplice pulsante da premere (il "#", per la precisione).



Display 7 Segmenti



In questo schema si può vedere come i due Arduini sono collegati, l'Arduino Mega 2560 comunica attraverso tre pin (7,8 e 9) i quali possono inviare i dati analogici. Abbiamo collegato anche il "GND" in comune. Il collegamento al modulo Adafruit con i pin "A4" (SDA), "A5" (SCL), 5[V] (VCC) e "GND", i quali servono per fare una comunicazione I2C.

SAMT - Sezione Informatica



ReactionGame 2.0

Pagina 21 di 28

Inoltre nello schema si può vedere anche un sette segmenti a 4 digits, per il quale non ho trovato il pezzo corretto. Infatti questo ha 16 pin e non 14, ma ho indicato i collegamenti corretti (come se fosse un sette segmenti a 14 pin). Ho utilizzato sette resistenze per i rispettivi segmenti.

```
1 int start = 3;
 2 int finish = 5;
 3
 4 int button = 8;
 5 int increment = 6;
 6
 7 void setup() {
 8
     pinMode(start, OUTPUT);
 9
     pinMode(finish, OUTPUT);
10
11
     pinMode (increment, OUTPUT);
12
     pinMode (button, INPUT);
13
     analogWrite(start, 1023);
14
15
     analogWrite(finish, 0);
16
     start = millis()/1000;
17 }
18
19 void loop() {
20
21
     if (digitalRead (button) == HIGH) {
       analogWrite(increment, 1023);
22
23
     }
24
     else{
25
       analogWrite(increment, 0);
26
27
     if((millis()/1000 - start) >= 20){
       analogWrite(finish, 1023);
28
29
       analogWrite(start, 0);
       delay(3000);
30
31
32
33 }
```

Figura 24: Codice Arduino 2560 ProvaMaster

Questo codice è stato usato per l'Arduino Mega 2560. Il codice è relativamente semplice, si creano 4 variabili con le quali si fa la comunicazione e la ricezione della pressione del bottone. Nel loop viene fatto il controllo del tempo passato per far finire il programma e la gestione dell'incremento del punteggio.

SAMT – Sezione Informatica



ReactionGame 2.0

Pagina 22 di 28

```
1 #include <Wire.h> // Enable this line if using Arduino Uno, Mega, etc.
 2 #include <Adafruit GFX.h>
 3 #include "Adafruit LEDBackpack.h"
 5 Adafruit_7segment matrix = Adafruit_7segment();
 7 int pinFinish = A0;
 8 int pinStart = A1;
 9
10 int increment = A2;
11
12 const int dig4 = 13;
13 const int dig3 = 12;
14 const int dig2 = 11;
15 const int dig1 = 10;
16
17 int dueP = 9;
18
19 int tempo[] = {5,4,2,3,6,7,8};
20
21 int punteggio = 0;
22
23 int start;
24 int ora;
```

Figura 25: Creazione variabili Arduino Uno

Queste sono le variabili globali dell'Arduino Uno con il quale si comandano i sette segmenti. Gli "#include" servono per importare le librerie necessarie della "Adafruit" per comandare il sette segmenti attraverso il relativo modulo. I tre pin (pinFinish, pinStart e increment) servono per ricevere i dati dal Mega. I restanti servono per le modalità e il sette segmenti.



```
26 void setup() {
     pinMode (dig4, OUTPUT);
27
     pinMode(dig3, OUTPUT);
28
29
     pinMode (dig2, OUTPUT);
30
     pinMode(dig1, OUTPUT);
31
32
     for(int i = 0; i < sizeof(tempo); i++) {</pre>
       pinMode(tempo[i], OUTPUT);
33
34
     }
     pinMode(dueP, OUTPUT);
35
36
37
     digitalWrite(dig4, HIGH);
     digitalWrite(dig3, HIGH);
38
39
     digitalWrite(dig2, HIGH);
40
     digitalWrite(dig1, HIGH);
41
42
     pinMode (pinStart, INPUT);
     pinMode (pinFinish, INPUT);
43
44
     pinMode (increment, INPUT);
45
46
     matrix.begin(0x70);
47 }
48
49 void loop() {
50
     if (analogRead (pinStart) > 1000 && analogRead (pinFinish) < 30) {
51
       start = millis()/1000;
52
       while (true) {
53
         boolean finished = false;
54
55
         if (analogRead (pinFinish) > 1000) {
56
            finished = true;
57
         }
58
59
         timer(start, finished);
         if (analogRead(pinStart) < 30 && analogRead(pinFinish) > 1000) {
60
           punteggio = 0;
61
62
           start = 0;
63
           ora = 0;
64
           timer(start, true);
65
           break;
66
         }
67
68
69 }
```

Figura 26: Metodi "setup" e "loop" Arduino Uno

Le prime 17 righe (27 - 44) servono ad istanziare le variabili per il sette segmenti senza modulo, poi le tre istanze sotto sono per i pin di ricezione del Master. Il "matrix.begin(0x70)" serve a "selezionare" la porta di uscita per il collegamento I2C con sette Segmenti con modulo.

Mentre nel loop c'è il controllo se è stata scelta una modalità e parte la partita con il timer e un "while(true)" dove si controlla se ha ricevuto il segnale dal Master di bloccarsi e poi di finire.

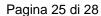


Pagina 24 di 28

```
116 void stampa(int nrDisplay, int arrayP[]) {
117
118
      switch (nrDisplay) {
119
120
      case 0:
121
        digitalWrite(arrayP[4], HIGH);
122
        digitalWrite(arrayP[6], HIGH);
123
        digitalWrite(arrayP[2], HIGH);
124
        digitalWrite(arrayP[0], HIGH);
125
        digitalWrite(arrayP[1], HIGH);
126
        digitalWrite(arrayP[5], HIGH);
127
        digitalWrite(arrayP[3], LOW);
128
        break;
129
130
      case 1:
131
        digitalWrite(arrayP[4], LOW);
132
        digitalWrite(arrayP[6], HIGH);
133
        digitalWrite(arrayP[2], HIGH);
134
        digitalWrite(arrayP[0], LOW);
135
        digitalWrite(arrayP[1], LOW);
136
        digitalWrite(arrayP[5], LOW);
137
        digitalWrite(arrayP[3], LOW);
138
        break;
```

Figura 27: Metodo stampa

etodo che serve a fare la stampa del numero per il digit selezionato. L'immagine mostra solo il codice necessario per i numeri 0 e 1, come esempio, i restanti sono stati ritenuti superflui e prolissi da aggiungere.





```
79 void timer(int inizio, boolean finish) {
 80
      digitalWrite(dueP, HIGH);
 81
 82
 83
        if (!finish) {
 84
          ora = (millis()/1000);
 85
 86
        for(int dig = 4; dig > 0; dig--) {
 87
          numero(dig);
 88
 89
          matrix.print(punteggio);
 90
          matrix.writeDisplay();
 91
 92
          if(dig == 4) {
 93
            stampa((ora -start)%10, tempo);
 94
 95
          else if(dig == 3){
 96
            stampa(((ora -start)/10)%6, tempo);
 97
98
          else if (dig == 2) {
99
            stampa(((ora -start)/60)%10, tempo);
100
          }
101
          else{
102
            stampa((((ora -start)/60)/10)%6, tempo);
103
104
105
          delayMicroseconds (5500);
106
107
          if (analogRead (increment) > 1000) {
108
            punteggio++;
109
110
111
          spegni();
112
        }
113
```

Figura 28: Metodo timer

Metodo che serve a selezionare il digit e il numero corretto, sempre nel metodo controlla se è stato incrementato il punteggio attraverso l'input.

Professionale

SAMT – Sezione Informatica

ReactionGame 2.0

Pagina 26 di 28

4 Test

Test Case:	TC-001 Nome :	Funzionamento bottoni telaio	
Riferimento:			
Descrizione:	Controllare che tutti i bottoni funzionino e che vengano rilevati tramite il loro numero corrispettivo.		
Prerequisiti:	 Telaio BATAK con gli 11 bottoni funzionanti. Display LCD collegato al dispositivo Arduino all'interno del telaio. 		
Procedura:	 Premere "@" per far partire la scelta delle modalità. Premere bottone per bottone, verificando che il numero della modalità selezionata sia quello premuto. Il numero della modalità usa un Sistema di Shifting, ergo, il numero appena premuto diventerà l'unità, mentre quello che prima era unità diventa decina. 		
Risultati attesi:	Il numero dell'unità del display LCD dovrebbe essere uguale a quello del bottone appena premuto.		

Test Case:	TC-002	Nome:	Funzionamento led telaio	
Riferimento:				
Descrizione:	Controllare che tutti i led funzionino e che corrispondano al bottone corretto.			
Prerequisiti:	Telaio BATAK con gli 11 bottoni funzionanti.			
Procedura:	 Premere "@" per far partire la scelta delle modalità e selezionare la modalità 23. Premere bottone per bottone, verificando che una volta premuto 			
	quest'ul	quest'ultimo si accenda, e che venendo ripremuto si spenga.		
Risultati attesi:	I led dovrebbero tutti corrispondere al bottone premuto.			

SAMT – Sezione Informatica

ReactionGame 2.0

Pagina 27 di 28

Test Case:	TC-003	Nome:	Funzionamento Aggiornamento Classifica	
Riferimento:	REQ-01			
Descrizione:	Controllare che I punteggi del giocatore vengano inviati a php e mostrati in tabella.			
Prerequisiti:	Telaio B	Telaio BATAK.		
	 Etherne 	t2 collegato all	'Arduino MEGA e Cavo Ethernet.	
	Server /	Apache e Serv	er MySQL funzionanti (consigliato XAMPP).	
Procedura:	1. Collega	re tramite cavo	Ethernet il PC e Ethernet2.	
	2. Dare al	nostro comput	er l'ip fisso 192.168.5.17.	
	3. Accende	ere I due serve	Apache e MySQL.	
			PC, nella cartella htdocs, la cartella reactiongame e la orm, trovabili sul GitHub del progetto sotto 2.0/codice.	
	5. Entrare	5. Entrare nella pagina locale localhost/reactiongameform/index.php.		
	6. Creare	 Far partire una modalità qualsiasi dalla BATAK ed eseguire qualche punto per controllare che la scrittura dei punteggi vada a buon fine. 		
	Dovrebl			
Risultati attesi:	Il punteggio e la modalità corrispondono a quelli realmente avuti giocando.			

4.1 Risultati Test

Test	Risultato
Funzionamento bottoni telaio	Positivo
Funzionamento led telaio	Positivo
Funzionamento Aggiornamento Classifica	Positivo

4.2 Mancanze/limitazioni conosciute

La pedana e il collegamento Wireless invece che Ethernet, inizialmente entrambi previsti per questa seconda versione del progetto, sono stati scartati per mancanza dei pezzi primari da utilizzare (pedana e Fishino). Volendo potremmo definire come mancanza anche la scarsa giocabilità di alcune modalità, che non essendo state rianalizzate assieme ai responsabili risultano a dir poco impossibili, abbiamo trovato. Tuttavia abbiamo ritenuto questo fattore non importante e per questo non gli è stato dedicato neanche del tempo.

5 Consuntivo

SAMT – Sezione Informatica



ReactionGame 2.0

Pagina 28 di 28

6 Conclusioni

Il prodotto ha dato i risultati sperati, con il completamento di tutti i requisiti primari e del progetto in sé. Ha molte possibilità di sviluppo, come il miglioramento delle cose abbandonate in corso del lavoro, e sarebbe potuto sicuramente andare meglio nel discorso 7 segmenti, visto l'enorme mole di tempo che ci ha fatto perdere.

6.1 Sviluppi futuri

Nuove modalità, migliorare la giocabilità di molte di quelle già presenti, una pedana effettiva e il collegamento al pc per l'aggiornamento punteggio sono gli esempi più lampanti di miglioramento che si potrebbe dare ulteriormente al progetto. Vanno poi aggiunte anche alcuni elementi che già nella scorsa versione del lavoro erano considerati non ottimali ma che sono stati ignorati per questa seconda versione. L'esempio migliore sono i led dei bottoni, che illuminano fin troppo poco, rendendo tedioso notare anche solo il fatto che siano illuminati, specialmente con certi tipi di luminosità. Ultimo ma non per importanza,

6.2 Considerazioni personali

Questo progetto ci ha mostrato esattamente come un solo elemento (due display 7 segmenti) possa risultare così problematico da bloccare tutto il procedimento del lavoro, finendo per spingere il completamento di quest'ultimo agli ultimi attimi, cosa non poco stressante e che per ora non siamo ancora riusciti ad evitare. Ciò nondimeno, il lavoro ci ha permesso di portare a compimento un progetto nato parecchi mesi fa assieme a due colleghi che per questa seconda fase del progetto purtroppo non ci hanno potuto seguire. Il senso di soddisfazione nel vedere un lavoro completato come si deve, dopo tanti sforzi, serate e perfino un weekend rimasti a lavorarci su come ore supplementari pensiamo possa dimostrarci come tenessimo anche noi stessi al lavoro, di come lo considerassimo già allora come una nostra creazione, qualcosa di cui siamo pienamente responsabili, sia nei pregi che nei difetti.

7 Bibliografia

7.1 Sitografia

- https://www.arduino.cc/en/Guide/
 - Ethernet2
 - o Fishino
 - o ITC
- Arduino.
- https://www.w3schools.com/
 - Javascript
 - o Php
- W3Schools

http://www.hobbytronics.co.uk/arduino-4digit-7segment, codice prova sette segmenti senza modulo Adafruit https://cdn-shop.adafruit.com/datasheets/865datasheet.pdf, datasheet sette segmenti senza modulo

8 Allegati

Allegato A: I3_Diari_ReactionGame2.pdf

Allegato B: I3_Documentazione_ReactionGame.pdf