Data: 29/2/2016 Classe: 3BI

## **Calcolatrice Binaria/Esadecimale**



#### Professori:

- Massimo Papa
- Elena Talomo
- Giorgio Ghione

#### **GRUPPO:**

-Mazilu Liviu

### **Sommario**

- 1-Oggetto dell'esperienza
- **2-Codifica**
- 3-Screenshot dei blocchi
- **4-Screnshot dell'applicazione** 
  - 1. Screen1(menu):
  - 2. Screen 2: Binario
  - 3. Screen 3: Conversioni
  - 4. Screen 4:Errore nelle conversioni

### Oggetto dell'esperienza

Quest'esperienza si è basata sul progettare e realizzare un programma in app inventor che abbia lo scopo di poter fare operazioni matematiche in binario ed esadecimale.

Oltre alla consegna, che era solo di fare le operazioni di addizione e sottrazione, ho aggiunto la moltiplicazione, la divisione e la conversioni tra le varie basi, rispettivamente binario, decimale ed esadecimale.

#### **Codifica**

Per la codifica ho deciso di dividere l'app in 4 screen:

1 per la conversione tra le basi;

1 per I calcoli binari;

1 per I calcoli esadecimali;

1 per il menu.

I blocchi sono molti, e a volte complessi, poichè la sottrazione e la divisione mi ha dato molti problemi, e ho deciso di arrotondare la divisone (es: se 8/5=1.6 , la calcolatrice arrotonda a 2), la sottrazione non può dare numeri negativi, ma permette I calcoli tra di essi (es: se 300-500 = -200 , la calcolatrice farà il valore assoluto del risultato, dando 200 come output).

La divisione e moltiplicazione inoltre hanno bisogno di altre variabili a sè, poichè se lasciate come nell'addizione e sottrazione (cioè inizializzate a 0), l'output sarà sempre 0.

Passiamo agli screen dei blocchi:

#### Screenshot dei blocchi

```
when A . Click
                                                                          when 66 . Click
                                                                         do set output . Text v to O O
do set output . Text v to ( o join ( output v . Text v
                                                                                                              output v . Text v × 10 + 6
                                    " (A) "
                                                                          when b7 .Click
when B . Click
                                                                          do set output . Text to 60
do set output . Text to 0 join output
                                           . Text •
                                                                                                                                       + ( 7
                                                                                                              output v . Text v × (10)
                                    " B "
                                                                          when b8 . Click
when b0 . Click
                                                                          do set output . Text to
do set output . Text to
                                                                                                          0
                                                                                                                                        + ( 8
                                                                                                              output v . Text v × (10)
                                    output 🔻
                                            . Text - × (10)
                                                                          when b9 .Click
when b1 . Click
                                                                          do set output . Text to
do set output . Text to 📜 💿
                                                                                                              output - . Text - × (10)
                                    output v . Text v × (10)
                                                                          when C . Click
when b2 . Click
                                                                          do set output . Text . to O join Output . Text .
do set output . Text v to
                                    output v . Text v × (10)
                                                                                                              . C
                                                                          when cancella .Click
when b3 v .Click
                                                                          do set output . Text v to [ 0 ]
do set output . Text to
                                0
                                    output v . Text v × (10)
                                                                              call reset A -
when 64 7 .Click
                                                                         when D .Click
                                                                          do set output . Text v to 60 join 0 output v . Text v
do set output ▼ . Text ▼ to I
                                    output v . Text v × (10)
```

Questi blocchi servono solo a immettere l'input del primo dato per il calcolo,il per 10 serve a muovere di un posto il vecchio numero per far spazio al prossimo, premendo nuovamente un tasto.

Sotto procedura ho messo il reset delle variabili, cioè la riinizializzazione a 0 o 1.

```
when meno . Click
do set global AESADECIMALEDELPIU to absolute v
                                                 convert number | hex to base 10 v | get | global AESADECIMALEDELPIU v
                                                                                                            convert number (hex to base 10 v
                                                                                                                                       output 🔻
                                                                                                                                               lext
   set global Aesadecimale to get global AESADECIMALEDELPIU -
   set output . Text to 👭 0
   set global Variabile x to "meno"
when piu . Click
do set global AESADECIMALEDELPIU to 0
                                         convert number (hex to base 10 ) get global AESADECIMALEDELPIU
                                                                                                     convert number hex to base 10 -
                                                                                                                                output 🔻 Text 🔻
   set global AESADECIMALEDELPIU to convert number base 10 to hex bet global AESADECIMALEDELPIU to
   set global Aesadecimale v to get global AESADECIMALEDELPIU v
   set output . Text to 0
   set global Variabile_x v to ( più
     \mathbf{A}0
```

Questi blocchi fanno si che quando si schiaccia il bottone meno o più, si cancelli la label di output,e si faccia l'operazione desiderata tra il numero precedente (se c'è, sennò è inizializzato a 0) e quello corrente.

Show Warnings

I numeri si convertono da base x a base 10, si fanno I calcoli e poi ritornano alla base x poichè è più semplice.

La variabile x è una variabile di stato e serve a far riconoscere all'"uguale" la condizione e I calcoli che deve fare.

Come già scritto in precedenza, ho messo il valore assoluto al meno poichè app inventor non ammette numeri negativi in sè, a meno che non si crei un algoritmo.

```
when diviso .Click
           🙆 if
                                            get global variabile per diviso = 1
           then set global Aesadecimale to
                                                                                                               convert number hex to base 10 v get global Aesadecimale v
                                                                                                                                                                                                                                                                       / ( convert number (hex to base 10 -
                           set global Aesadecimale to convert number base 10 to hex get global Aesadecimale
                            set output . Text to 0
                            set global Variabile_x to ( "diviso "
                        set global Aesadecimale to convert number thex to base 10 to convert number thex to base 10 to convert number the convert number the
                            set global Aesadecimale to
                                                                                                         convert number (base 10 to hex *) get global Aesadecimale *
                            set output . Text to 0
                            set global Variabile_x to ( " diviso "
           set global variabile per diviso to 1
when per .Click
           if 💿
                                get global variabile_per_diviso = 1
                            set global Aesadecimale to
                                                                                                                        convert number hex to base 10 v get global Aesadecimale v
                                                                                                                                                                                                                                                                                × Convert number (hex to base 10 -
                            set global Aesadecimale to convert number base 10 to hex to get global Aesadecimale
                             set global Variabile_x to free per "
                             set output . Text to 0
                         set global Aesadecimale to convert number hex to base 10 to
                            set global Aesadecimale to convert number base 10 to hex get global Aesadecimale
                            set output . Text to 0
                            set global Variabile_x to ( " per "
            set global variabile_per_diviso to [1]
```

Uguale a prima, ma le variabili sono inizializzate a 1 poichè facendo altrimenti il risultato darrebbe sempre 0.

La variabile "variabile per diviso" è una variabile per determinare se c'era un numero precedentemente calcolato o no (es:se si fa 5/1, la variabile rimane 0, se si fa 5/1, poi di nuovo il risultato / 3, prende come il numero prima il risultato).

```
when uguale .Click
    set global Besadecimale to cutput
    set global Aesadecimale to convert number (hex to base 10 v ) get global Aesadecimale
    set global Besadecimale To
                                  convert number (hex to base 10 v ) get (global Besadecimale
                 get global Variabile x 🔻
           set global Risultato esadecimale to
                                                0
                                                      get global Aesadecimale
                                                                                     get global Besadecimale
           set global Risultato esadecimale to convert number base 10 to hex
                                                                                  get global Risultato esadecimale
           set output . Text to get global Risultato esadecimale
    o if
                get (global Variabile_x ▼
           set global Risultato esadecimale to
                                                absolute 🔻
                                                                get global Aesadecimale •
                                                                                             get global Besadecimale
           set global Risultato_esadecimale to convert number base 10 to hex by get global Risultato_esadecimale
          set output . Text to get global Risultato esadecimale
    🧿 if
                get global Variabile_x - = -
    then
           set global Risultato_esadecimale to
                                                      get global Aesadecimale
                                                                                      get global Besadecimale
           set global Risultato esadecimale to convert number (base 10 to hex to global Risultato esadecimale
           set output *
                         Text ▼ to
                                      get global Risultato_esadecimale
    o if
                get global Variabile x •
                                               " diviso
           set global Risultato esadecimale to
                                                   get global Aesadecimale •
                                                                             / get global Besadecimale -
           set output . Text to convert number base 10 to hex .
                                                                        format as decimal number
                                                                                                  get global Risultato_esadecimale
```

Questi blocchi fanno si che quando si schiaccia il bottone uguale, si riconverte il numero in base 10, si fanno I rispettivi calcoli a seconda dell'operazione, si riconverta il risultato nella base desiderata, e che il risultato venga messo in output.

La parte binaria è uguale, solo con variabili diverse e solo 0 e 1 come numeri a disposizione.

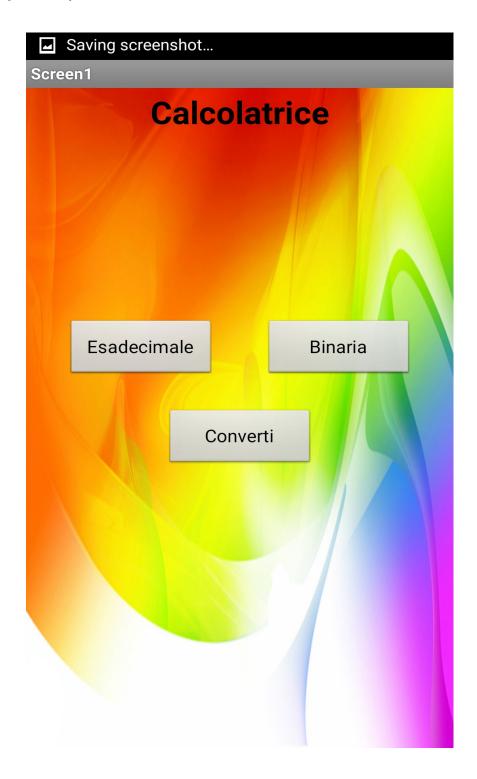
```
when Button1 . Click
do 🔯 if
                not is empty ( text_base_10 * ). Text * and * ( text_base_16 * ). Text * and * ( text_base_2 * ). Text *
         set text_base_16 - . Text - to convert number base 10 to hex - text_base_10 - . Text -
          set (text_base_2 * ). Text * ) to ( convert number (base 10 to binary * ) (text_base_10 * ). Text *
                not is empty text_base_16 · Text · and ·
                                                                   is empty (_text_base_2 - ). Text - | and - | is empty (_text_base_10 - ). Text -
    then set text_base_10 v . Text v to A convert number hex to base 10 v text_base_16 v
          set global hex_to_base_10 * to convert number hex to base 10 * text_base_10 * . Text *
         set (text_base_2 v ). Text v ) to ( convert number (base 10 to binary v ) get (global hex_to_base_10 v
                not is empty ( text_base_2 * ) . Text * is empty ( text_base_16 * ) . Text * and * ( text_base_10 * ) . Text *
    then set text_base_10 v . Text v to convert number (binary to base 10 v ) text_base_2 v . Text v
          set global binary_to_hex • to convert number binary to base 10 • text_base_2 • . Text •
          set text_base_16 * . Text * to convert number base 10 to hex * get global binary_to_hex *
               is empty ( text_base_2 - Text - and -
                                                           is empty (_text_base_16 * ). Text * | and * | is empty (_text_base_10 * ). Text *
          call Notifier1 .ShowMessageDialog
                                                Devi inserire UNA variabile! "
                                                Errore! "
                                               " (Ok.) "
                                  buttonText
                not is empty text base 2
                                                                   not is empty ( text_base_16 * ). Text * or * ( not ) is empty ( text_base_10 * ). Text *
          call Notifier2 .ShowMessageDialog
                                                Devi inserire UNA SOLA variabile!
                                                Errore! "
                                  buttonText
                                               " Ok. "
```

Come quarto e ultimo screen ho la conversione tra le varie basi, semplicemente si ha 3 checkbox, se ne compila 1, si schiaccia il bottone "converti!" e nelle altre 2 apparirà lo stesso numero nelle rispettive basi.

(l'errore è un errore di ai companion e non intralcia il programma).

# Screenshot dell'applicazione

# Screen1(menu):

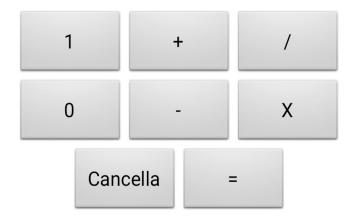


### **Screen 2: Binario**



**Calcolatrice binaria** 

# 10

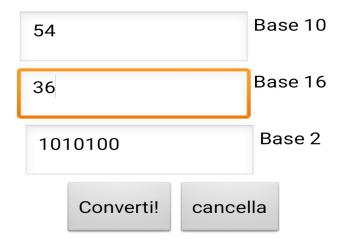


Torna al menu

### **Screen 3: Conversioni**



# Conversioni



torna al menu

Screen 4:Errore nelle conversioni



C'è anche un altro tipo di errore se non si inserisce nessuna variabile.