Data: 11/11/2015 Classe: 3BI

# Esperienza sulle porte logiche



#### Professori:

- Massimo Papa
- Elena Talomo
- Giorgio Ghione

#### **GRUPPO:**

- -Mazilu Liviu
- -Popovici Gabriel

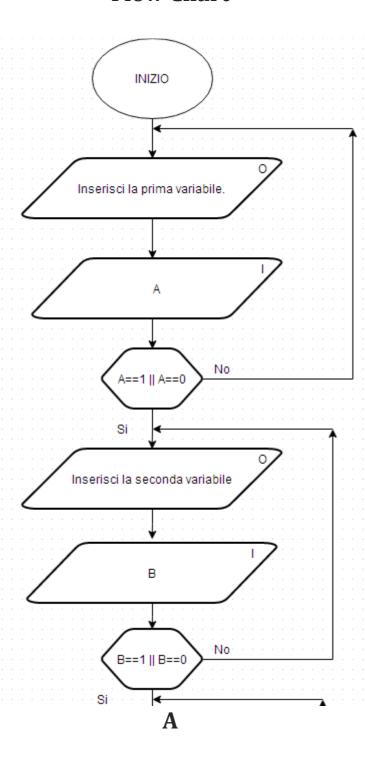
# **Sommario**

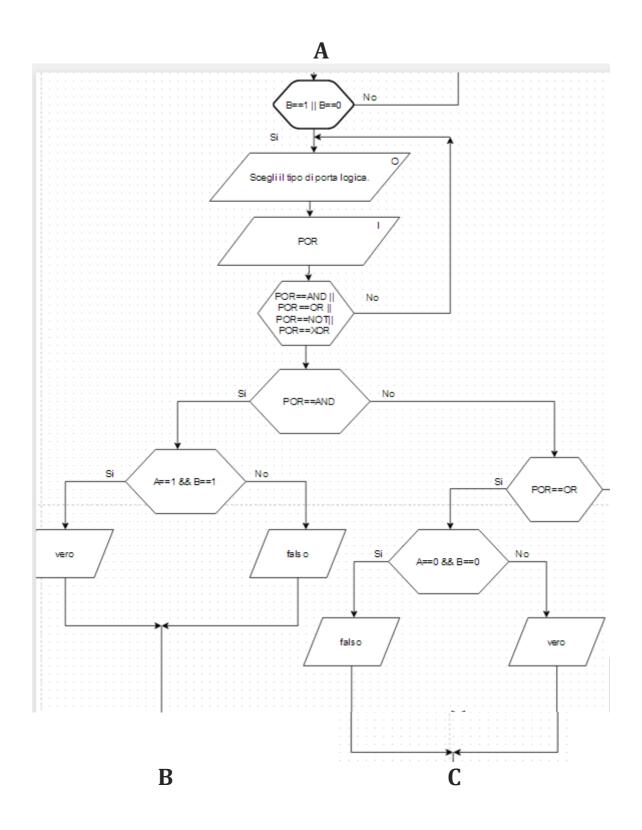
Oggetto dell'esperienza	<u></u> 3
Flow Chart	
Codifica	
Istruzioni per l'utente	
ScreenShot dell'applicazione	

### Oggetto dell'esperienza

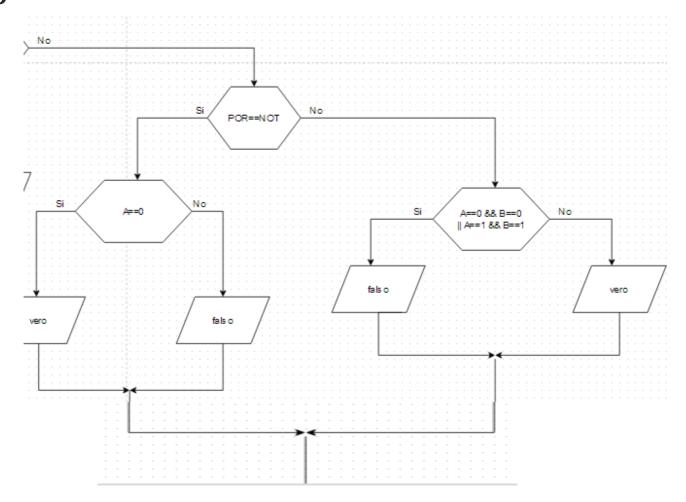
Quest'esperienza si è basata sul progettare e realizzare un programma in app inventor che abbia lo scopo di simulare il funzionamento delle porte logiche "AND", "OR", "NOT" e "XOR".

### **Flow Chart**

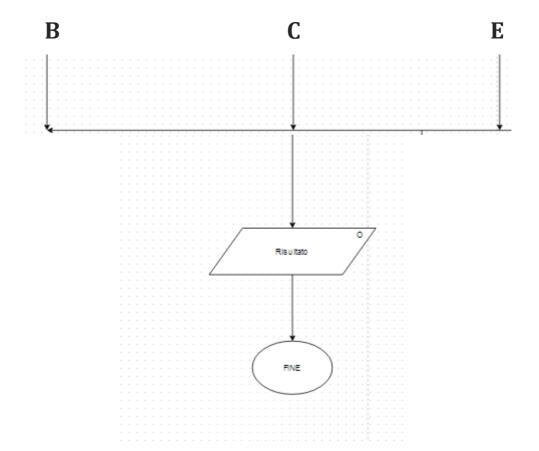




POR==OR > D



E



#### Codifica

Questa volta la codifica è stata più lunga, ma sorprendentemente più facile, perchè bisogna ripetere sempre gli stessi comandi in modo diverso. Partiamo dalle checkbox per AND,OR,NOT e (XOR).

when checknot .Changed

```
when checkand . Changed
                                                                 do set Canvas2 . Visible to false
do set Canvas2 . Visible to false
                                                                     if checknot • Checked •
   🔳 if
          checkand •
                       Checked *
                                                                          set checkand . Checked to false
         set checkor . Checked to false
                                                                           set checkor . Checked to false
         set checknot . Checked to false
                                                                           set checkxor . Checked to false
         set checkxor . Checked to false
                                                                           set Canvas1 . Visible to true
         set Canvas1 . Visible to to true
                                                                           set Canvas1 . BackgroundImage . to
                                                                                                              imgnot.png 1
         set Canvas1 . BackgroundImage to
                                            " imgand.png
                                                                           set Canvas2 . BackgroundImage . to
                                                                                                              " not.png '
                      . Visible To to true
         set TextVar2
                                                                           set TextVar2 . Visible to false
         set Canvas2 . Backgroundlmage . to
                                                                 when checkxor .Changed
when checkor . Changed
                                                                 do set Canvas2 . Visible to false
do set Canvas2 . Visible to false
                                                                           checkxor •
                                                                                        Checked •
    if 🗓
          checkor - Checked -
                                                                          set checkand . Checked to false
         set checkand . Checked to false
    then
                                                                          set checkor . Checked to false
         set checkxor . Checked to false
                                                                          set checknot . Checked to false
         set checknot . Checked to false
                                                                          set Canvas1 . Visible to true
         set Canvas1 . Visible to true
                                                                          set Canvas1 . Backgroundlmage to
                                                                                                             " (imgxor.png) "
         set Canvas1 . BackgroundImage . to
                                             imgor.png
                                                                                        Backgroundlmage • to
                                                                          set Canvas2 .
         set Canvas2 . BackgroundImage . to (
                                                                                        Visible to true
                                                                          set TextVar2 .
          set TextVar2 . Visible to true
```

Sono semplici istruzioni per far si che ogni volta che si seleziona un'altra porta logica, tutte le funzioni dell'altra si disabilitino.

Questa volta abbiamo utilizzato un canvas semplicemente perchè è più ordinato nella sezione design e non si deve fare fatica a trovare dove sono le vari immagini.

```
when BtnIstruz Click
do open another screen screenName "Screen2"

when BtnTab Click
do set Canvas2 Visible to true
```

```
when BtnReset ▼ .Click
    set checkand . Checked to
                                    false
    set checkor . Checked to
                                    false
    set checknot . Checked .
                                     false
    set checkxor ▼ . Checked ▼
                                     false
    set TextVar1 ▼
                   . Text ▼ to
    set TextVar2
                   . Text ▼
    set [blout . Text . to ]
    set TextVar2 ▼ . Visible ▼ to
                                  true
    set Canvas2 -
                   . Visible v to
                                   false
    set Canvas1 -
                    Visible •
                                   false
```

Nella seconda parte della codifica ci sono i bottoni per il reset ( fa diventare tutte le immagini invisibili, deseleziona le porte logiche e cancella le caselle di testo), Btnlstruz serve per andare allo screen in cui ci sono le vari istruzioni su come usare il programma, BtnTab è il bottone che serve a rendere visibile l'eventuale tabella della porta selezionata, e infine la variabile Output semplicemente dice se dalla combinazione di 0 e 1 immessa, il risultato è vero (1) o falso (0).

```
when BtnStart .Click
            checknot -
                          Checked -
    if
    then
                                                        and 🔻
                                    . Text ▼ ≠ ▼ (
                                                                  TextVar1 →
                                                                               Text - # 1
                         TextVar1 ▼
                 call Errore . ShowMessageDialog
                                                     La variabile è errata,si accetta solo 0 e 1!
                                                     Oops! "
                                             title
                                       buttonText
                                                    " Ok. "
```

```
else if TextVar1 · Text 	 # 1 Or TextVar2 · Te
```

Qui abbiamo creato le notifiche per gli errori.

Partendo dalla seconda immagine, i blocchi fanno si che se l'utente immette qualsiasi variabile che sia diversa da 0 o 1,gli/le appaia un errore con scritto "le variabili sono errate, si accetta solo 0 e 1!".

Nel primo screen, il concetto è lo stesso, ma la porta logica NOT ha solo 1 variabile, quindi abbiamo reso invisibile la variabile 2 e fatto si che l'errore valga solo per la variabile 1.

```
checkand . Checked .
then
     if
                                                and 🔻 🧯
                             Text - = - (0)
                                                       TextVar2 ▼
                                                                    Text - = - (0)
                  TextVar1 ▼
          set (blout . Text .
                              to 📜 🔾
     then
     if
                                                and 🔻
                             Text • = • (1)
                  TextVar1 ▼
                                                         TextVar2 ▼
                                                                    Text ▼
                                                                           then
          set [blout v . Text v
                              to [1
     if
                                                and 🔻
                  TextVar1
                              Text •
                                     = 7 (0
                                                         TextVar2 ▼
                                                                    Text ▼
          set [blout •]. Text •
     then
     if.
                                               and 🔻
                  TextVar1 •
                             Text - = - (1)
                                                         TextVar2 ▼
                                                                    Text ▼
          set [blout . Text to [0]
```

Qui i blocchi servono a verificare l'input dell'utente e dare l'output giusto. Questi blocchi specifici servono per la porta AND.

Le prossime immagini saranno la stessa cosa, fatta per le altre porte,quindi non ci sarà commento.

```
if
       checkor 🔻
                   Checked •
then
     if
                                                and 🔻
                              Text • = • (1)
                   TextVar1 ▼
                                                         TextVar2 ▼
                                                                     Text - = - (
           set (blout . Text to 11
     then
                                                and 🔻
                                     TextVar1 ▼
                              Text ▼
                                          0
                                                          TextVar2 ▼
                                                                     Text ▼
     then
           set [blout . Text .
                              to [1
     if 🔳
                                                and 🔻
                                           1
                              Text ▼ = ▼
                   TextVar1 ▼
                                                          TextVar2 ▼
                                                                     Text ▼
                              to 📜 🚹
     then
           set Iblout . Text .
                                                and 🔻
                   TextVar1 ▼
                              Text ▼ = ▼ (
                                          0
                                                          TextVar2 ▼
                                                                                  0
                                                                     Text ▼
           set Iblout
                      . Text ▼
                              to 0
```

```
then call Call . Speak
message  join  "il risultato è "

"Falso "

then call Call . Speak
message  join  "il risultato è "

then call Call . Speak
message  join  "il risultato è "
"Vero "
```

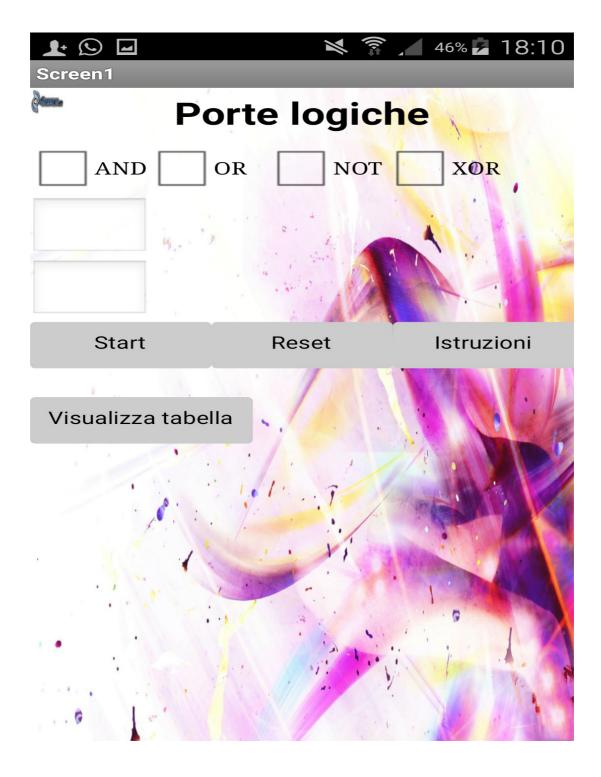
Infine si riceva l'output sia come scritto (0 o 1) che vocale (vero o falso).

### Istruzioni per l'utente

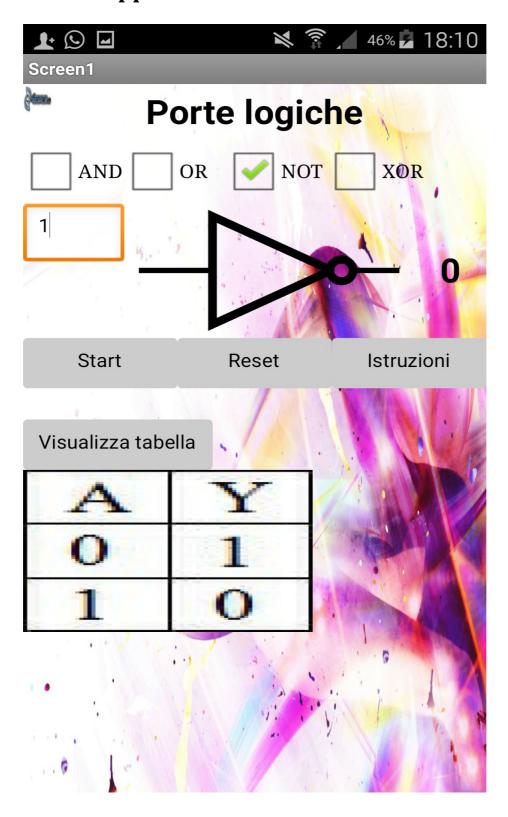
Quest'istruzione serve letteralmente solo a far tornare l'utente all'interfaccia del programma, poiché tutte le istruzioni per l'utente le abbiamo fatte sotto forma di label, quindi non serivano blocchi per esse.

# ScreenShot dell'applicazione

### 1 - Schermata iniziale



# 2 - Test dell'applicazione



# 3 - Errore dell'applicazione



#### 4 - Schermata delle istruzioni

