PLAStiblob

Bernardi Lorenzo  
Bortolamiol Alessandro  
Dal Colle Gabriele  
Scognamiglio Simone

# Introduzione

# Files del progetto

### menu-levels.lua

Questo file è la landing page del menu dove l’utente arriverà appena cliccato il bottone ‘play’ nella schermata di home. Questa pagina ha il dovere di controllare a che livello è arrivato l’utente all’interno del gioco tramite un controllo del database *Sqlite3.*   
Prima di tutti, quindi, controlliamo se è la prima volta che l’utente entra nel gioco in modo da sapere se bisogna creare un nuovo database o dovremmo utilizzare uno già esistente. Questo si farà con questa istruzione:

local checkifdbexists = [[SELECT \* from levels]]

    local dbexists = db:exec( checkifdbexists )

    print(dbexists)

Il print potrà ritornare due valori:

* **0:** l’esecuzione della query è andata a buon fine, ciò significa che la tabella è già istanziato all’interno della memoria e non serve crearlo
* **1:** in questo caso, la query non è andata a buon fine. Ciò significa che la tabella *levels* non esiste in memoria, andrà quindi creato.

Questa parte di codice invece è il fulcro della ppagina, infatti costituisce gran parte del funzionamento di essa. All’interno della variabile *livellicompletati* ci sarà il numero dei livelli completati dall’utente che è venuto precedentemente preso dalla tabella levels del database. In base a questo numero renderò disponibili al click o meno le varie immagini del display del cellulare.

* local levels={}
* for i=1, 8 do
* local impath
* --controllo se ho già passato il livello nell'identificativo su cui è posizionata 1
* if (tonumber(livellicompletati) >= i) then
* --assegno al percorso dell'immagine l'immagine corrispondente al livello in modalità SBLOCCATA
* impath = "immagini/menu/livelli/"..i..".png"
* else
* --assegno al percorso dell'immagine l'immagine corrispondente al livello in modalità BLOCCATA
* impath = "immagini/menu/livelli/locked"..i..".png"
* end
* print("percorso: ", impath)
* levels[i] = display.newImageRect( menu, impath, 200, 200 )
* levels[i].anchorX = 0
* levels[i].anchorY = 0
* levels[i].x = checkImagePositionX(i)
* levels[i].y = checkImagePositionY(i)
* end