

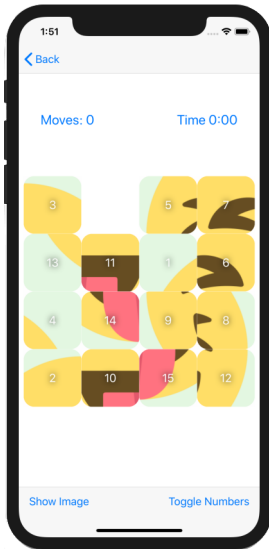
Laboratorio di Applicazioni Mobili

Applicazione in ambiente iOS

Giovanni Fazi
`giovanni.fazi@studio.unibo.it`

Alma Mater Studiorum - Università degli Studi di Bologna

Scopo del progetto



Progetto Base: il Model

- ▶ **Tile:** ogni tessera deve avere un identificatore univoco e coordinate x,y per posizionarsi sulla Board
- ▶ **Board:** la tavola di gioco, contiene un array di Tiles e si occupa sia di muovere i Tiles che controllare se il puzzle è stato risolto

Muovere una casella significa fare swap delle coordinate di un Tile con quelle della casella vuota. Solo i Tiles adiacenti alla casella vuota possono muoversi.

Progetto Base: Controller e Viste

- ▶ **GameSetup:** Vista che mostra una finestra di 3 immagini e permette di scegliere i parametri della partita
- ▶ **GameView:** Vista di gioco, viene mostrata la Board insieme ad un timer. La Board contiene l'immagine scelta disposta in tessere con ordine casuale.

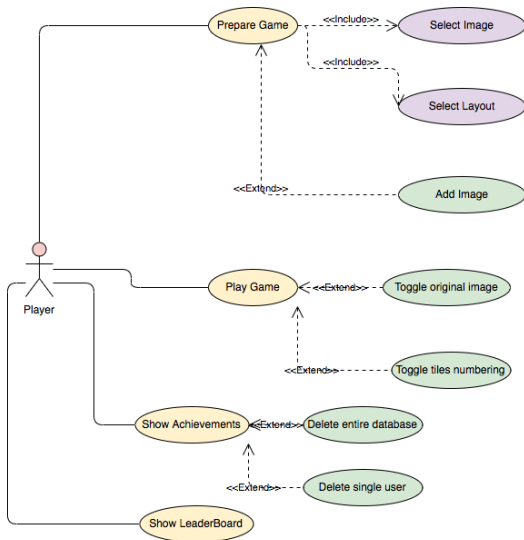
Per disordinare i Tiles vengono effettuate mosse casuali cercando di evitare backtracking.

Le tessere non sono altro che UIButtons disposti in griglia 4x4

Progetto Esteso: Features aggiunte

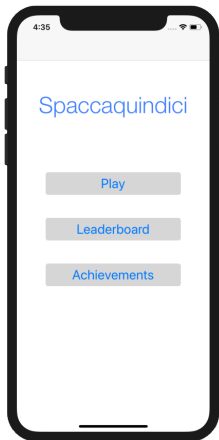
- ▶ È possibile giocare con tre griglie diverse (4x4 5x5 6x6)
- ▶ I dati di un giocatore sono salvati su un database. I campi memorizzati riguardano punteggio, partite migliori e sfide superate
- ▶ Esistono 24 sfide con livelli di difficoltà crescenti
- ▶ È possibile aggiungere immagini dalla galleria o acquisirle dalla fotocamera
- ▶ Durante una partita è possibile visualizzare la immagine risolta o i numeri delle caselle
- ▶ Alla fine di una partita è possibile condividere il punteggio tramite ActivityViewController (quindi su ogni social o applicazione supportata)

Progetto Esteso: Casi d'uso



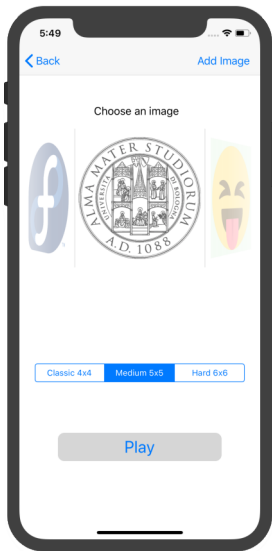
Menu

Per la gestione delle views è stato usato un Navigation Controller. Ci sono 7 views in totale, la prima è il Menu.



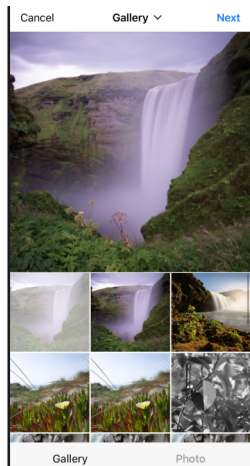
- ▶ **Play:** mostra la GameSetupView dove si definiscono parametri della partita
- ▶ **Leaderboard:** mostra una UITableView con i migliori punteggi
- ▶ **Achievements:** permette di visualizzare gli utenti registrati e le sfide di ognuno

Game Setup View Controller

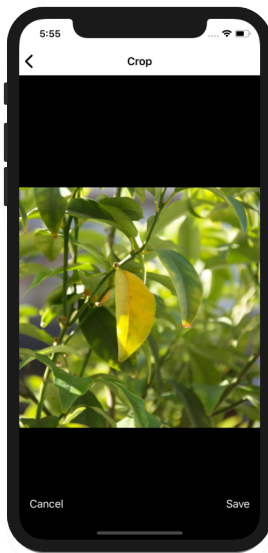
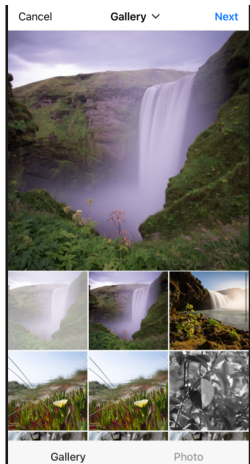


- ▶ La **pickerView** è stata ruotata in orizzontale, inoltre ha un effetto di scroll infinito.
- ▶ Il layout della Board viene scelto tramite **SegmentedControl**
- ▶ La aggiunta di immagini viene gestita dalla libreria **YPIImagePicker** che gestisce anche il ridimensionamento
- ▶ Le immagini aggiunte sono persistenti e sono memorizzate nella galleria

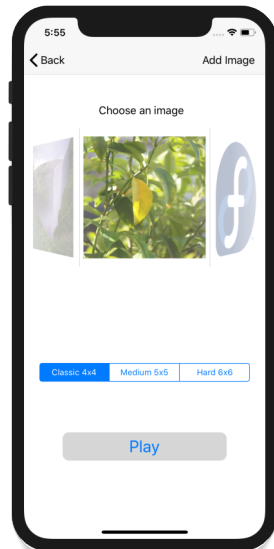
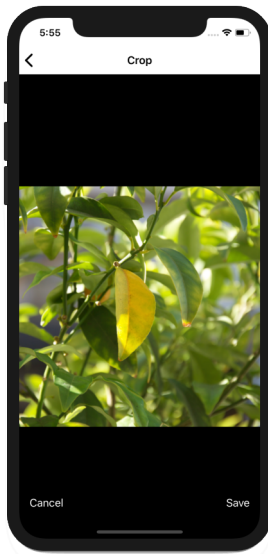
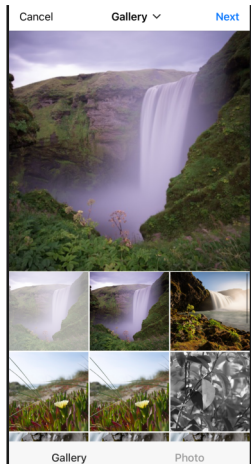
Game Setup: esempio aggiunta immagine



Game Setup: esempio aggiunta immagine



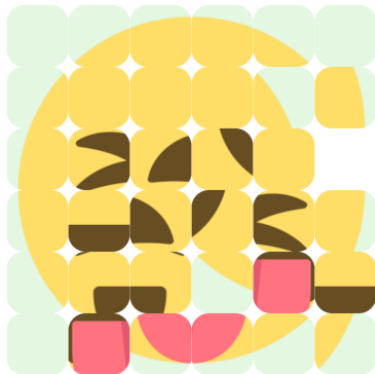
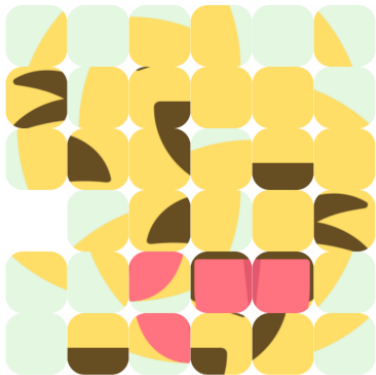
Game Setup: esempio aggiunta immagine



Game View

Le caselle sono UIButtons che vengono generati e posizionati tramite codice. Toccandoli il controller provvede ad aggiornare il Model.

Per aumentare la "casualità" dello shuffle si evita di invertire l'ultima mossa random.

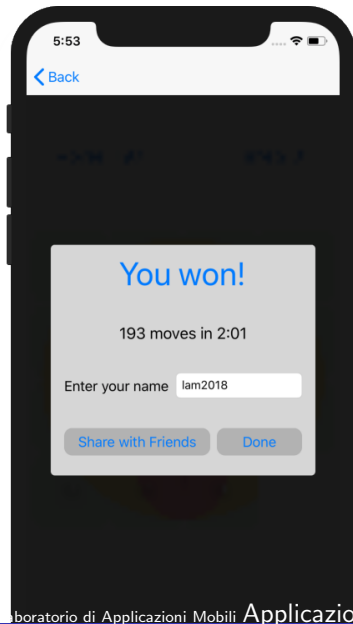


Game View

- ▶ **ViewDidLoad:** in questo metodo si calcola la posizione della Board e si creano i Buttons assegnando ad ognuno la giusta porzione della immagine di gioco. Dopo aver assegnato le targetAction si effettua lo shuffle.
- ▶ **ButtonDidTouch:** oltre a gestire il movimento di una casella il metodo si occupa della gestione del timer e conteggio mosse. Ad ogni mossa controlla se il puzzle è stato risolto.
- ▶ **makePopUpAppear:** mostra un popup con i dati della partita e permette di registrare un utente nel database.
- ▶ **ToggleNumbers e ShowImage:** Sono due action associate ai relativi Buttons. Mostrano rispettivamente la numerazione attuale delle caselle e l'immagine risolta

Game View

- ▶ **ShareScore**: metodo che tramite `UIActivityViewController` permette di condividere i risultati su ogni service supportato.
- ▶ Inserendo un username esistente si aggiornano i punteggi, altrimenti si crea un nuovo utente.



Leaderboard

Top 5 scores	
giovi	350
lam2018	250
test	200
lam	150
----	0
Best time on 4x4 puzzle	
giovi	1:0
Best time on 5x5 puzzle	
test	3:25
Best time on 6x6 puzzle	
lam	5:53
Less moves on 4x4 puzzle	
giovi	101
Less moves on 5x5 puzzle	
test	349
Less moves on 6x6 puzzle	
lam	607

- ▶ Una **UITableView** mostra i risultati migliori.
- ▶ Le query al database sono effettuate tramite le API **Core Data** e filtrate con **NSPredicate**.
- ▶ È presente una top 5 dei punteggi migliori. Ogni partita aumenta i punti in base al numero sfide superate in essa.
- ▶ Sono poi mostrati anche i record di mosse e tempo per ogni puzzle.

Achievements

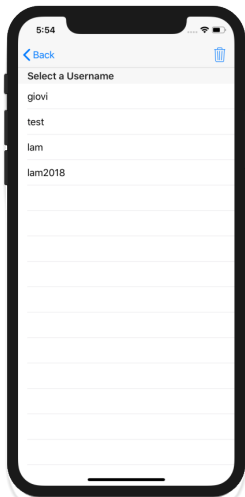
Sono definiti nel file `Achivements.swift`:

- ▶ **getAchievements**: metodo che prende in input i valori di tempo e numero mosse di una partita e restituisce la corrispondente stringa da salvare nel database.
- ▶ **getPoints**: calcola i punti di una partita. un contatore viene aumentato di 50 punti per ogni sfida superata in una partita.
- ▶ Le descrizioni delle sfide sono salvate in un array di stringhe.
- ▶ Ad ogni sfida è associato un numero intero univoco.

Le sfide sono 24 e consistono nel risolvere un puzzle con determinati vincoli di mosse o tempo.

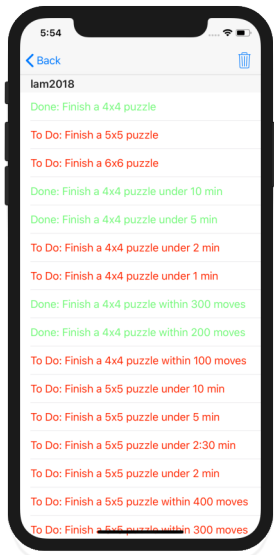
Achievements

Gli obiettivi sono rappresentati su due UITableViews



- ▶ La prima View mostra tutti gli utenti registrati e permette di cancellare l'intero database previo messaggio di warning.
- ▶ Toccando un username si apre la seconda View.

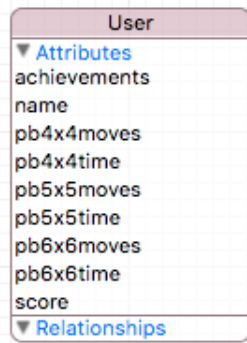
Achievements



- ▶ Nella seconda **UITableView** vengono mostrate le sfide specifiche del giocatore.
- ▶ Le sfide con testo rosso e prefisso "To do:" sono quelle da superare. Quelle superate hanno colore verde e prefisso "Done:"

Core data

Per registrare gli utenti viene usato il framework **Core Data**.
Tale scelta è stata fatta per sperimentare la API e per la semplicità di uso di essa.



- ▶ Gli attributi sono stringhe o interi.
- ▶ Per memorizzare gli achievements si usa una stringa contenente "0" o "1" in corrispondenza della n-esima sfida.

Idee per future estensioni

- ▶ Aggiungere layout particolari (esempio 3x6) con relative sfide
- ▶ Creare una leaderboard con punteggi online
- ▶ Aggiungere una modalità player vs player dove due (o più) giocatori competono per risolvere lo stesso puzzle. Il pairing si può fare tramite internet.
- ▶ Aggiungere una sezione "Tutorial" per i principianti.

Fine presentazione

Grazie per l' attenzione.