



# Laboratorio di Applicazioni Mobili (LAM 2015)

**Luciano Bononi**

luciano.bononi@unibo.it

**Dipartimento di Informatica – Scienza e Ingegneria  
Alma Mater Studiorum - Università di Bologna**



# Progetti iOS

- ❖ 1) Sono graditi **progetti individuali** basati su proposte definite autonomamente dagli studenti, legate a interessi o progetti personali. Tuttavia, i **progetti vanno concordati** con i docenti del corso (Luciano Bononi per i progetti svolti in ambiente iOS, e Marco Di Felice e Luca Bedogni per i progetti in ambiente Android).
- ❖ Basta inviare una email e ricevere la conferma. **Non vale la regola del silenzio assenso.**



# Progetti iOS

- ❖ 2) I docenti, nell'ultimo mese del corso avranno illustrato alcuni possibili progetti "base" che sono considerati d'ufficio progetti approvati, per consentire a tutti il sostenimento anche dei primi appelli del corso. Ovviamente, per tutti i progetti elencati, sono possibili tutte le evoluzioni e estensioni che vi detta la vostra fantasia.
- ❖ **Non esistono limiti superiori.** E' possibile/gradito andare oltre quanto illustrato a lezione.



## Progetti iOS

- ❖ 3) E' necessario consegnare un progetto, individuale, in **almeno uno degli ambienti** (iOS e/o Android). Sono ovviamente graditi e valutati con extra-credits i progetti svolti in **entrambi gli ambienti**.
- ❖ **Tuttavia, all'esame orale occorre essere in grado di commentare o modificare i sorgenti delle demo svolte a lezione su entrambe le piattaforme (iOS e Android).**



# Progetti iOS

- ❖ 4) E' severamente **vietata la copiatura** di progetti. Nei casi di copiatura, o nei casi in cui lo studente non dimostri piena conoscenza delle metodologie e scelte usate, **il progetto sara' annullato.**



# Progetti iOS

- ❖ 5) i progetti devono essere consegnati entro le **date di scadenza** sotto indicate. La consegna del progetto avviene inviando una email all'indirizzo: **lam-projects@cs.unibo.it** con **subject: "progetto LAM2015 - iOS e/o Android - nome, cognome, matricola"** con allegato un file compresso che contenga una **cartella "ProgLAM2015\_matricola"** tutti i file sorgente e le sottocartelle del progetto (path relativo) e la documentazione (relazione). Cercate di **evitare file superiori a 2 MB**, se possibile. Nei casi di file particolarmente pesanti rendete il file compresso disponibile via web mediante una **URL privata** che ci comunicherete nella email.



## Progetti iOS

- ❖ 6) la **relazione** deve essere sintetica ma esaustiva e deve illustrare: scopo dell'applicazione, funzionalità previste, caratteristiche e requisiti (anche dell'ambiente di esecuzione e sviluppo), progettazione e scelte implementative, difficoltà e soluzioni, aspetti rilevanti per lo sviluppo, casi d'uso ed esempi di funzionamento tipici, eventuali idee per estensioni, conclusioni e commenti finali.



# Progetti iOS

- ❖ 7) La **prova orale** dell'esame si tiene di solito previo appuntamento fissato nella settimana successiva a quella di sottomissione del progetto. Riceverete una **convocazione via email** (per concordare la data e ora) al momento della chiusura delle sottomissioni.
- ❖ Alla prova orale dovrete venire dotati di una **breve presentazione**, di **circa 25 slide** in qualsiasi formato, che permetta di illustrare una sintesi equivalente alla vostra relazione. Inoltre, dovrete provvedere a fornire un **supporto di esecuzione del vostro progetto**. Non saranno penalizzate le realizzazioni dimostrate mediante simulatori, ma è importante che siate dotati dell'ambiente di sviluppo sul quale avete realizzato e testato il progetto. In sintesi, dobbiamo evitare di rischiare di perdere 30 minuti a progetto per capire come mai non compila sulla macchina X usata per il test.





## Progetti iOS

- ❖ 8) durante l'illustrazione del progetto viene svolto anche l'esame orale, con **domande su TUTTO IL PROGRAMMA DEL CORSO**. N.B. anche chi ha svolto il progetto in uno solo dei due ambienti (iOS o Android) deve essere in grado di rispondere a domande su entrambe le componenti del programma del corso. Si veda il punto 3.



## Progetti iOS

- ❖ 9) al termine della prova orale si ottiene **una proposta di voto**, e si può decidere se accettare o meno il voto proposto.
- ❖ Tuttavia, se il voto è insufficiente o non viene accettato, **non è possibile sostenere un nuovo orale nell'arco della stessa settimana (appello)**. Il progetto (se sufficiente e non copiato) rimane valido secondo quanto indicate al punto 10.



## Progetti iOS

- ❖ 10) il progetto consegnato nel corso delle scadenze sotto definite per l'A.A. 2014/2015 ha **validita' fino all'appello di Giugno 2016** (incluso).
- ❖ Chi sostiene l'esame dopo Giugno 2015 dovrà consegnare un nuovo progetto secondo le specifiche del corso 2015/2016. Quindi, chi consegna a Giugno 2015 ha un anno intero di validita' del progetto.
- ❖ Chi consegna nei mesi successivi ha un numero minore di opportunita' per sostenere la discussione.



# Progetti iOS

- ❖ 11) NON sono ammesse prove di esame senza consegna del progetto. Il progetto è parte sostanziale della valutazione. Il peso della valutazione del **progetto** e della relazione/presentazione costituisce il **75% della valutazione**. Il rimanente **25% è dato dalla prova orale** sugli argomenti del corso. Tuttavia, un ottimo progetto in assenza di un orale sufficiente non implica una proposta di voto sufficiente (o fino a 22/30). E' comunque necessario dimostrare di possedere le conoscenze sui temi svolti a lezione **sia su iOS che su Android all'orale** per ottenere una proposta di voto sufficiente.



# Progetti iOS

- ❖ Scadenze di consegna e finestre per il sostenimento dell'orale. Le finestre per gli orali possono essere estese o modificate su richiesta motivata, non oltre i 15 gg dalla consegna:
- ❖ **1) Giugno 2015**
- ❖ **Consegna entro il 24 Giugno 2015, ore 24.00**
- ❖ **N.B. scadenza di consegna solo per laureandi della sessione di Luglio: Giovedì 18 Giugno e discussione Venerdì 19 Giugno.**
- ❖ **Orali: indicativamente fine Giugno 2015 – inizio Luglio**



# Progetti iOS

- ❖ 2) Luglio 2015
  - ❖ Consegna entro il **22 Luglio 2015, ore 24.00**
  - ❖ Orali: indicativamente 23-24 Luglio 2015
- ❖ 3) Settembre 2015
  - ❖ Consegna entro il **9 Settembre 2015, ore 24.00**
  - ❖ Orali: indicativamente 10-11 Settembre 2015
- ❖ 4) Dicembre 2015
  - ❖ Consegna entro il **9 Dicembre 2015, ore 24.00**
  - ❖ Orali: indicativamente 10-11 Dicembre 2015
- ❖ 5) Gennaio 2016
  - ❖ Consegna entro il **13 Gennaio 2016, ore 24.00** (Termine ultimo per la consegna dei progetti)
  - ❖ Orali: indicativamente 14-15 Gennaio 2016



# Progetti iOS - 2015

- ❖ 1) valgono **tutti i progetti presentati per la parte Android.**
- ❖ 2) - iOS1: **progetto "spaccaquindici"**: realizzare un'applicazione che mostri una finestra di scelta di almeno 3 immagini.
- ❖ Una volta selezionata un'immagine, una nuova vista mostrerà l'immagine divisa in 16 caselle, di cui una vuota, disordinate.
- ❖ L'utente deve toccare le caselle che intende spostare nella posizione vuota, attivando un timer che conta il tempo trascorso, e che verrà fermato a indicare il tempo necessario per l'ordinamento completo dell'immagine.
- ❖ Si noti che per ottenere un'immagine riordinabile è necessario partire dall'immagine ordinata, disordinandola secondo passi randomizzati sequenziali.
- ❖ Il progetto dovrà comprendere almeno un navigation controller. Non ci sono limiti superiori alla complicazione (es. inserire nome del giocatore, gestire un DB di utenti, aumentare la matrice 4x4 a 5x5 ecc., acquisire immagini da flickr, pubblicare i tempi ricordo su facebook, inviare tweet, ecc.)



# Progetti iOS

- ❖ iOS2: **progetto "forza-N"**, simile al noto "forza-4" o tic-tac-toe (occorre mettere in fila un numero N di simboli giocando uno contro l'altro), oppure un drawer di figure geometriche che consenta di disegnare geometrie, modificarne la dimensione, colore e riempimento, salvando l'immagine in formato grafico.
- ❖ Il progetto dovrà comprendere almeno un navigation controller. Non ci sono limiti superiori alla complicazione (es. rendere possibile la partita uno contro l'altro anche da remoto, inserire nome del giocatore e tenere traccia del numero di vittorie, gestire un DB di utenti, pubblicare i tempi recordo su facebook, inviare tweet, ecc.)





# Progetti iOS

- ❖ iOS3: progetto "**briscola**": realizzare il gioco della briscola utente vs. computer. Non si richiede una strategia accurata di gioco lato PC ma il PC non può giocare carte a caso. Ad esempio, è opportuno che il PC abbia almeno una minima nozione di logica del gioco almeno per i punti seguenti: usare la briscola più bassa per prendere una mano, usare una carta più alta dello stesso seme per "strozzare" una mano senza usare briscola, tenere traccia dei carichi già giocati e giocare le carte dei semi corrispondenti quando si è i primi a giocare (per evitare strozzature).
- ❖ Il progetto dovrà comprendere almeno un navigation controller. Non ci sono limiti superiori alla complicazione (es. inserire nome del giocatore, gestire un DB di utenti, variare nella versione a 3 e 4 giocatori (a coppie) dei quali uno sia il PC, pubblicare i tempi record su facebook, inviare tweet, ecc.)



# Progetti iOS

- ❖ iOS4: progetto **campo minato**: viene mostrata una griglia di caselle  $M \times N$ , delle quali  $K$  scelte a caso ad ogni riavvio,  $K = 0,33 \cdot (M \cdot N)$  contengono una bomba. L'utente avvia il gioco (parte un timer) e tocca una casella e la casella si rivela nel suo contenuto: se la casella selezionata è una bomba il gioco termina (partita persa), mentre se la casella  $c$  selezionata è libera, tutte le 8 caselle libere adiacenti a  $c$  ( $c_1, c_2, \dots, c_8$ ) mostrano un valore intero che corrisponde al numero di bombe adiacenti ad ogni casella (ovvero con quante bombe è in contatto la casella, mostrano la bomba nel caso essa vi sia contenuta (bomba scoperta ma non esplosa). Dalle informazioni sui contatori l'utente deve cercare di toccare tutte e solo le caselle libere, arrivando a scoprire tutte le bombe. Il gioco termina quando tutte le bombe sono scoperte e il timer mostra il tempo necessario a risolvere con successo il gioco. Il progetto dovrà comprendere almeno un navigation controller. Non ci sono limiti superiori alla complicazione (es. inserire nome del giocatore, gestire un DB di utenti, aumentare la matrice  $M \times N$  ecc., pubblicare i tempi recordo su facebook, inviare tweet, ecc.)



# iOS considerations

Questions?