

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Dipartimento di Scienze Fisiche, Informatiche e Matematiche

Corso di Laurea in Informatica

Progetto e sviluppo in Swift di un'applicazione per dispositivi mobili basata su dati semantico-lessicali

RELATORE:

Ing. Riccardo Martoglia

LAUREANDA:

Diana Sighinolfi

A.A. 2014-2015

Obiettivi della tesi

Gli obiettivi che si intendevano raggiungere nell'ambito del progetto di tesi svolto sono:

- ▶ Apprendere e conoscere il linguaggio Swift
- ▶ Gestire ed elaborare dati provenienti da un database semantico-lessicale
- ▶ Realizzare un'applicazione iOS in Swift

Obiettivi della tesi

Gli obiettivi che si intendevano raggiungere nell'ambito del progetto di tesi svolto sono:

- ▶ Apprendere e conoscere il linguaggio Swift
- ▶ Gestire ed elaborare dati provenienti da un database semantico-lessicale
- ▶ Realizzare un'applicazione iOS in Swift

Obiettivi della tesi

Gli obiettivi che si intendevano raggiungere nell'ambito del progetto di tesi svolto sono:

- ▶ Apprendere e conoscere il linguaggio Swift
- ▶ Gestire ed elaborare dati provenienti da un database semantico-lessicale
- ▶ Realizzare un'applicazione iOS in Swift

Obiettivi della tesi

Gli obiettivi che si intendevano raggiungere nell'ambito del progetto di tesi svolto sono:

- ▶ Apprendere e conoscere il linguaggio Swift
- ▶ Gestire ed elaborare dati provenienti da un database semantico-lessicale
- ▶ Realizzare un'applicazione iOS in Swift

Attività svolte

Le attività svolte durante il tirocinio sono:

- ▶ Studio del linguaggio e delle tecnologie utilizzate in Swift
- ▶ Studio del database semantico MultiWordNet
- ▶ Software per l'elaborazione dei dati estrapolati da MultiWordNet
- ▶ Game app in Swift

Attività svolte

Le attività svolte durante il tirocinio sono:

- ▶ Studio del linguaggio e delle tecnologie utilizzate in Swift
- ▶ Studio del database semantico MultiWordNet
- ▶ Software per l'elaborazione dei dati estrapolati da MultiWordNet
- ▶ Game app in Swift

Attività svolte

Le attività svolte durante il tirocinio sono:

- ▶ Studio del linguaggio e delle tecnologie utilizzate in Swift
- ▶ Studio del database semantico MultiWordNet
- ▶ Software per l'elaborazione dei dati estrapolati da MultiWordNet
- ▶ Game app in Swift

Attività svolte

Le attività svolte durante il tirocinio sono:

- ▶ Studio del linguaggio e delle tecnologie utilizzate in Swift
- ▶ Studio del database semantico MultiWordNet
- ▶ Software per l'elaborazione dei dati estrapolati da MultiWordNet
- ▶ Game app in Swift

Attività svolte

Le attività svolte durante il tirocinio sono:

- ▶ Studio del linguaggio e delle tecnologie utilizzate in Swift
- ▶ Studio del database semantico MultiWordNet
- ▶ Software per l'elaborazione dei dati estrapolati da MultiWordNet
- ▶ Game app in Swift

Il linguaggio Swift

Swift è un linguaggio fresco, leggero e facile da apprendere, soprattutto per i neofiti. Basato sul C e sull'Objective-C, è tuttavia considerato quasi un linguaggio di *scripting* per la sua estrema flessibilità.

Le principali peculiarità del nuovo linguaggio sono:

- ▶ tipizzazione forte
- ▶ valori opzionali
- ▶ no *fallthrough* implicito
- ▶ *Automatic Reference Counting*

Il linguaggio Swift

Swift è un linguaggio fresco, leggero e facile da apprendere, soprattutto per i neofiti. Basato sul C e sull'Objective-C, è tuttavia considerato quasi un linguaggio di *scripting* per la sua estrema flessibilità.

Le principali peculiarità del nuovo linguaggio sono:

- ▶ tipizzazione forte
- ▶ valori opzionali
- ▶ no *fallthrough* implicito
- ▶ *Automatic Reference Counting*

Il linguaggio Swift

Swift è un linguaggio fresco, leggero e facile da apprendere, soprattutto per i neofiti. Basato sul C e sull'Objective-C, è tuttavia considerato quasi un linguaggio di *scripting* per la sua estrema flessibilità.

Le principali peculiarità del nuovo linguaggio sono:

- ▶ tipizzazione forte
- ▶ valori opzionali
- ▶ no *fallthrough* implicito
- ▶ *Automatic Reference Counting*

Il linguaggio Swift

Swift è un linguaggio fresco, leggero e facile da apprendere, soprattutto per i neofiti. Basato sul C e sull'Objective-C, è tuttavia considerato quasi un linguaggio di *scripting* per la sua estrema flessibilità.

Le principali peculiarità del nuovo linguaggio sono:

- ▶ tipizzazione forte
- ▶ valori opzionali
- ▶ no *fallthrough* implicito
- ▶ *Automatic Reference Counting*

Il linguaggio Swift

Swift è un linguaggio fresco, leggero e facile da apprendere, soprattutto per i neofiti. Basato sul C e sull'Objective-C, è tuttavia considerato quasi un linguaggio di *scripting* per la sua estrema flessibilità.

Le principali peculiarità del nuovo linguaggio sono:

- ▶ tipizzazione forte
- ▶ valori opzionali
- ▶ no *fallthrough* implicito
- ▶ *Automatic Reference Counting*

WordNet (1/2)

WordNet è un database semantico, strutturato a forma di grafo, sviluppato nell'università di Princeton.

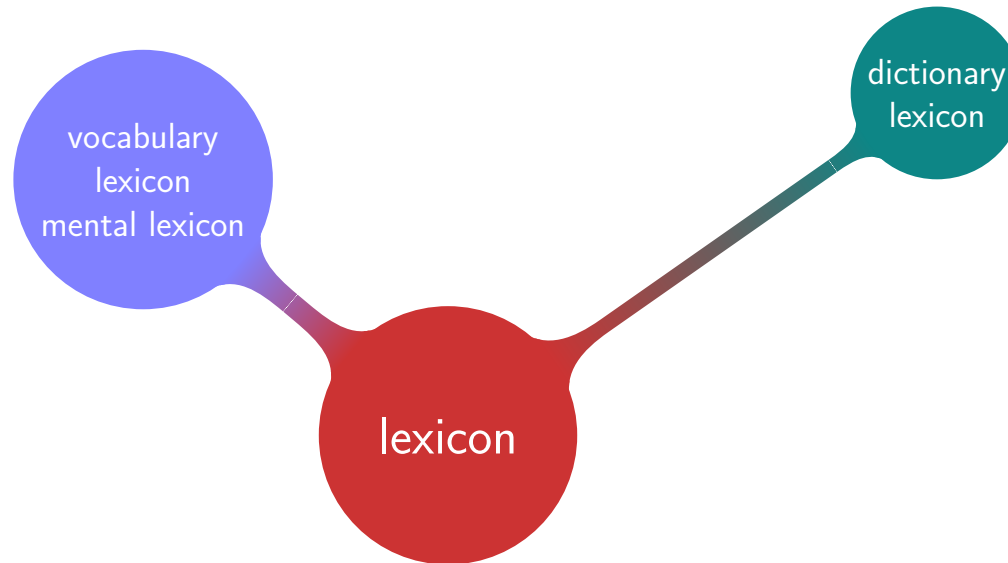
WordNet (1/2)

WordNet è un database semantico, strutturato a forma di grafo, sviluppato nell'università di Princeton.



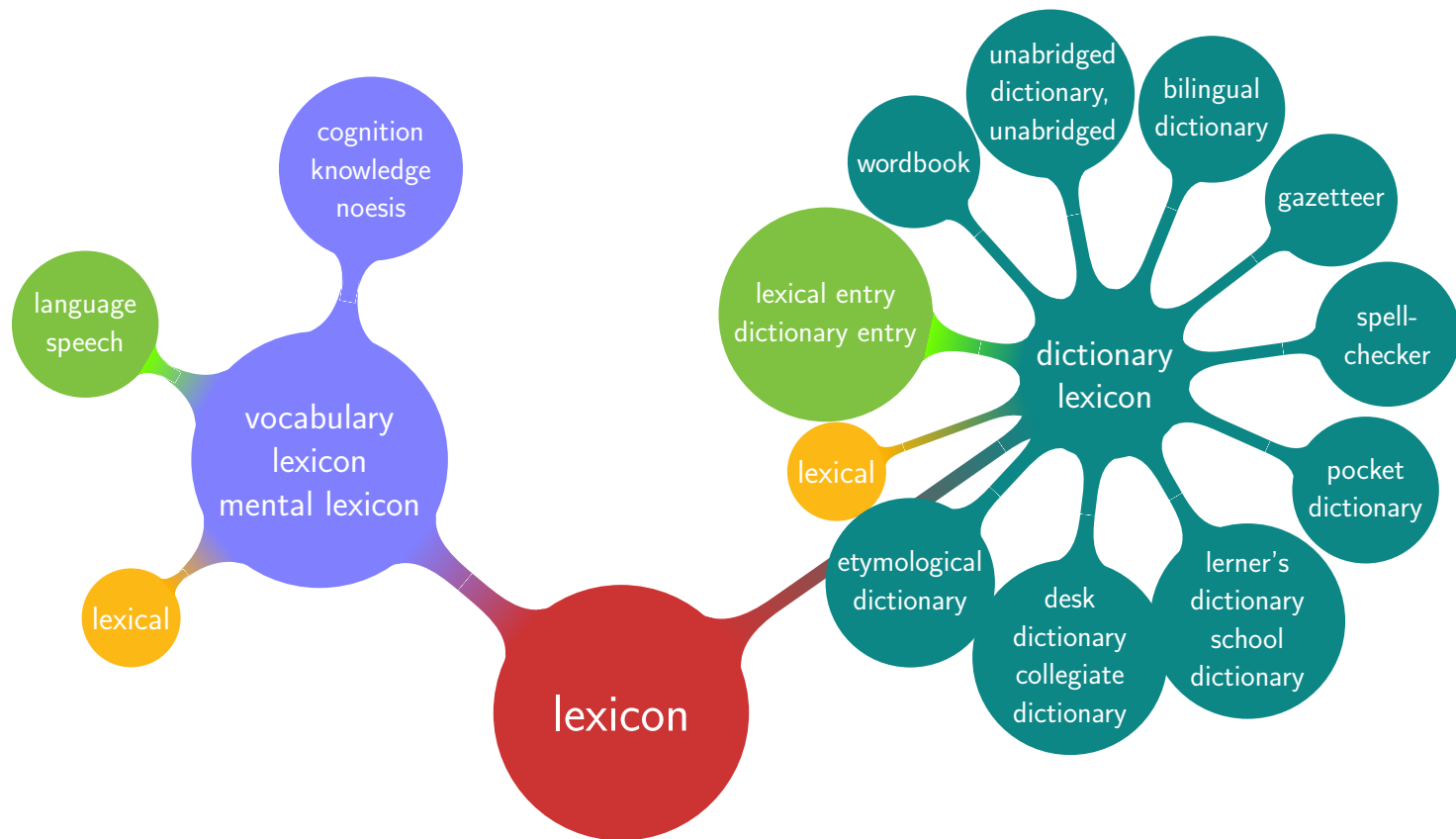
WordNet (1/2)

WordNet è un database semantico, strutturato a forma di grafo, sviluppato nell'università di Princeton.



WordNet (1/2)

WordNet è un database semantico, strutturato a forma di grafo, sviluppato nell'università di Princeton.



WordNet (2/2)

Il database lessicale è un *thesaurus* organizzato sulla base di *concetti* strutturati all'interno di insiemi:

- ▶ lemma
- ▶ sense
- ▶ synset

I *synset* sono collegati fra loro attraverso relazioni semantiche; le possibili relazioni previste dal dizionario sono:

- ▶ iponimia
- ▶ ipernimia
- ▶ meronimia
- ▶ implicazione
- ▶ similarità
- ▶ attributo

WordNet (2/2)

Il database lessicale è un *thesaurus* organizzato sulla base di *concetti* strutturati all'interno di insiemi:

- ▶ lemma
- ▶ sense
- ▶ synset

I *synset* sono collegati fra loro attraverso relazioni semantiche; le possibili relazioni previste dal dizionario sono:

- ▶ iponimia
- ▶ ipernimia
- ▶ meronimia
- ▶ implicazione
- ▶ similarità
- ▶ attributo

WordNet (2/2)

Il database lessicale è un *thesaurus* organizzato sulla base di *concetti* strutturati all'interno di insiemi:

- ▶ lemma
- ▶ sense
- ▶ synset

I *synset* sono collegati fra loro attraverso relazioni semantiche; le possibili relazioni previste dal dizionario sono:

- ▶ iponimia
- ▶ ipernimia
- ▶ meronimia
- ▶ implicazione
- ▶ similarità
- ▶ attributo

WordNet (2/2)

Il database lessicale è un *thesaurus* organizzato sulla base di *concetti* strutturati all'interno di insiemi:

- ▶ lemma
- ▶ sense
- ▶ synset

I *synset* sono collegati fra loro attraverso relazioni semantiche; le possibili relazioni previste dal dizionario sono:

- ▶ iponimia
- ▶ ipernimia
- ▶ meronimia
- ▶ implicazione
- ▶ similarità
- ▶ attributo

WordNet (2/2)

Il database lessicale è un *thesaurus* organizzato sulla base di *concetti* strutturati all'interno di insiemi:

- ▶ lemma
- ▶ sense
- ▶ synset

I *synset* sono collegati fra loro attraverso relazioni semantiche; le possibili relazioni previste dal dizionario sono:

- ▶ iponimia
- ▶ ipernimia
- ▶ meronimia
- ▶ implicazione
- ▶ similarità
- ▶ attributo

WordNet (2/2)

Il database lessicale è un *thesaurus* organizzato sulla base di *concetti* strutturati all'interno di insiemi:

- ▶ lemma
- ▶ sense
- ▶ synset

I *synset* sono collegati fra loro attraverso relazioni semantiche; le possibili relazioni previste dal dizionario sono:

- ▶ iponimia
- ▶ ipernimia
- ▶ meronimia
- ▶ implicazione
- ▶ similarità
- ▶ attributo

MultiWordNet

MultiWordNet è un database lessicale multilingue che nella versione online è realizzato in tre lingue:

- ▶ Inglese (WordNet dell'università di Princeton)
- ▶ Spagnolo (Università politecnica di Catalogna, Barcellona)
- ▶ Italiano (ITC-irst di Povo, Trento)

La versione distribuita, in formato XML, contiene solamente la versione in italiano allineata, per quanto possibile, con WordNet inglese dell'università di Princeton.

MultiWordNet

MultiWordNet è un database lessicale multilingue che nella versione online è realizzato in tre lingue:

- ▶ Inglese (WordNet dell'università di Princeton)
- ▶ Spagnolo (Università politecnica di Catalogna, Barcellona)
- ▶ Italiano (ITC-irst di Povo, Trento)

La versione distribuita, in formato XML, contiene solamente la versione in italiano allineata, per quanto possibile, con WordNet inglese dell'università di Princeton.

MultiWordNet

MultiWordNet è un database lessicale multilingue che nella versione online è realizzato in tre lingue:

- ▶ Inglese (WordNet dell'università di Princeton)
- ▶ Spagnolo (Università politecnica di Catalogna, Barcellona)
- ▶ Italiano (ITC-irst di Povo, Trento)

La versione distribuita, in formato XML, contiene solamente la versione in italiano allineata, per quanto possibile, con WordNet inglese dell'università di Princeton.

MultiWordNet

MultiWordNet è un database lessicale multilingue che nella versione online è realizzato in tre lingue:

- ▶ Inglese (WordNet dell'università di Princeton)
- ▶ Spagnolo (Università politecnica di Catalogna, Barcellona)
- ▶ Italiano (ITC-irst di Povo, Trento)

La versione distribuita, in formato XML, contiene solamente la versione in italiano allineata, per quanto possibile, con WordNet inglese dell'università di Princeton.

MultiWordNet

MultiWordNet è un database lessicale multilingue che nella versione online è realizzato in tre lingue:

- ▶ Inglese (WordNet dell'università di Princeton)
- ▶ Spagnolo (Università politecnica di Catalogna, Barcellona)
- ▶ Italiano (ITC-irst di Povo, Trento)

La versione distribuita, in formato XML, contiene solamente la versione in italiano allineata, per quanto possibile, con WordNet inglese dell'università di Princeton.

L'applicazione (1/7)

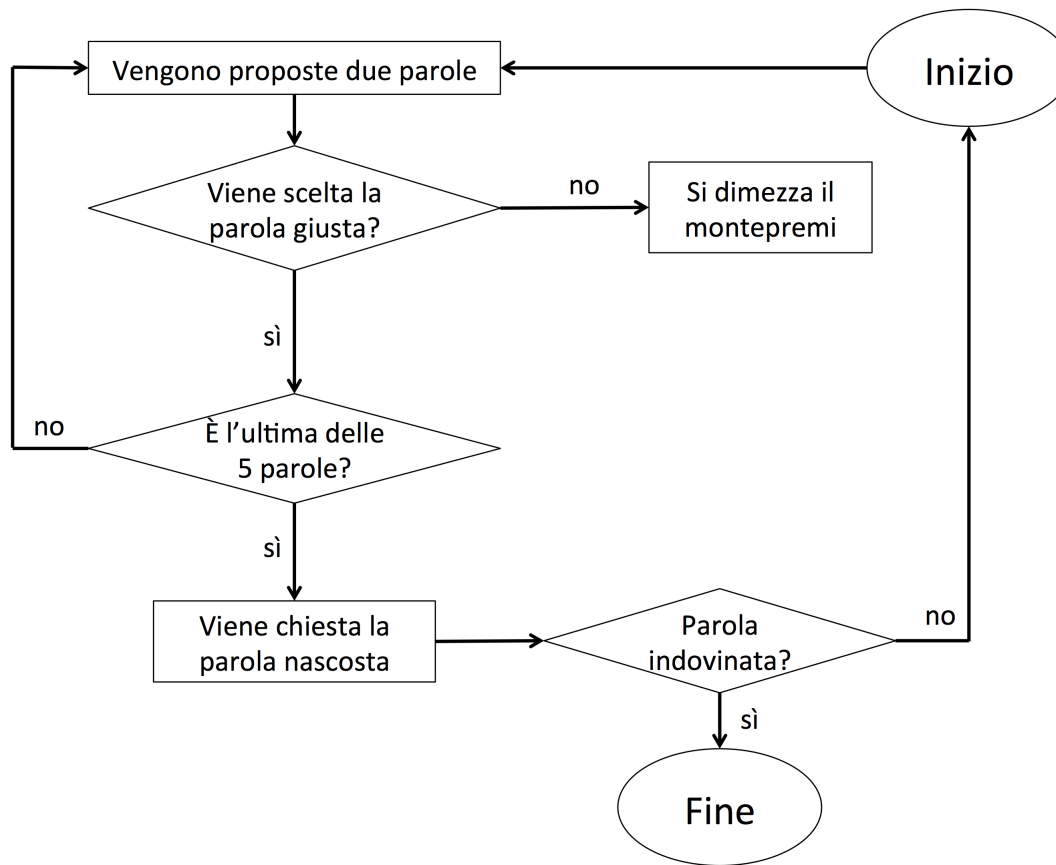
Il gioco

Il software sviluppato è una *game application* che ripropone il gioco de “*La Ghigliottina*” presente nel programma televisivo *L'eredità*.

L'applicazione (1/7)

Il gioco

Il software sviluppato è una *game application* che ripropone il gioco de “*La Ghigliottina*” presente nel programma televisivo *L’eredità*.



L'applicazione (2/7)

Schemi di gioco

Obiettivo proposto

Generare in maniera automatica gli elementi necessari per la creazione degli schemi di gioco.

Lo schema di gioco si compone di:

- ▶ parola nascosta
- ▶ 5 parole-indizio
- ▶ 5 parole-intruso

L'applicazione (2/7)

Schemi di gioco

Obiettivo proposto

Generare in maniera automatica gli elementi necessari per la creazione degli schemi di gioco.

Lo schema di gioco si compone di:

- ▶ parola nascosta
- ▶ 5 parole-indizio
- ▶ 5 parole-intruso

L'applicazione (2/7)

Schemi di gioco

Obiettivo proposto

Generare in maniera automatica gli elementi necessari per la creazione degli schemi di gioco.

Lo schema di gioco si compone di:

- ▶ parola nascosta
- ▶ 5 parole-indizio
- ▶ 5 parole-intruso

L'applicazione (2/7)

Schemi di gioco

Obiettivo proposto

Generare in maniera automatica gli elementi necessari per la creazione degli schemi di gioco.

Lo schema di gioco si compone di:

- ▶ parola nascosta
- ▶ 5 parole-indizio
- ▶ 5 parole-intruso

L'applicazione (3/7)

Elaborazione dei dati



I dati messi a disposizione da MultiWordNet sono distribuiti in formato XML (\approx 1 milione di voci). Per il loro utilizzo si è deciso di elaborarli attraverso un software realizzato *ad hoc* in PHP che permette la conversione dei dati XML in un database relazionale SQLite.

L'applicazione (4/7)

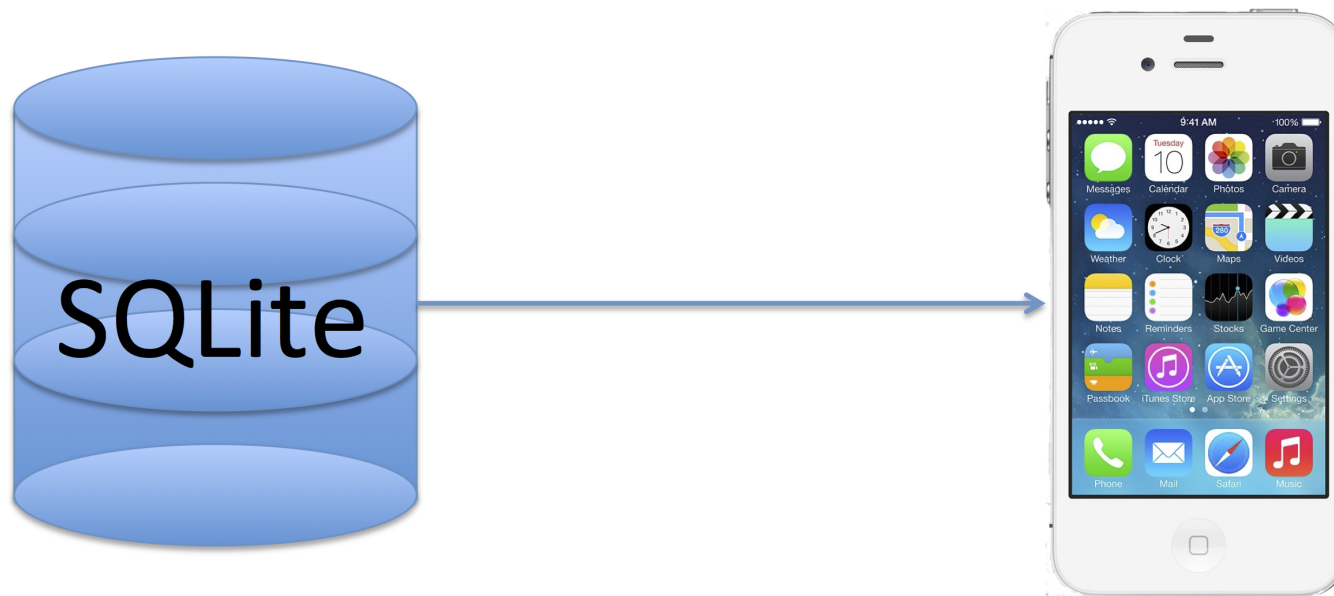
Elaborazione dei dati

Il software utilizza un parser XML per estrapolare i dati dal database e genera il codice SQL necessario per la creazione e il popolamento della base di dati destinata all'applicativo.

L'applicazione (4/7)

Elaborazione dei dati

Il software utilizza un parser XML per estrapolare i dati dal database e genera il codice SQL necessario per la creazione e il popolamento della base di dati destinata all'applicativo.



Per interfacciare il database SQLite con l'applicazione si è utilizzato un wrapper in Swift.

L'applicazione (5/7)

Algoritmo di generazione schemi

Algoritmo di generazione automatica degli schemi di gioco:

L'applicazione (5/7)

Algoritmo di generazione schemi

Algoritmo di generazione automatica degli schemi di gioco:



L'applicazione (5/7)

Algoritmo di generazione schemi

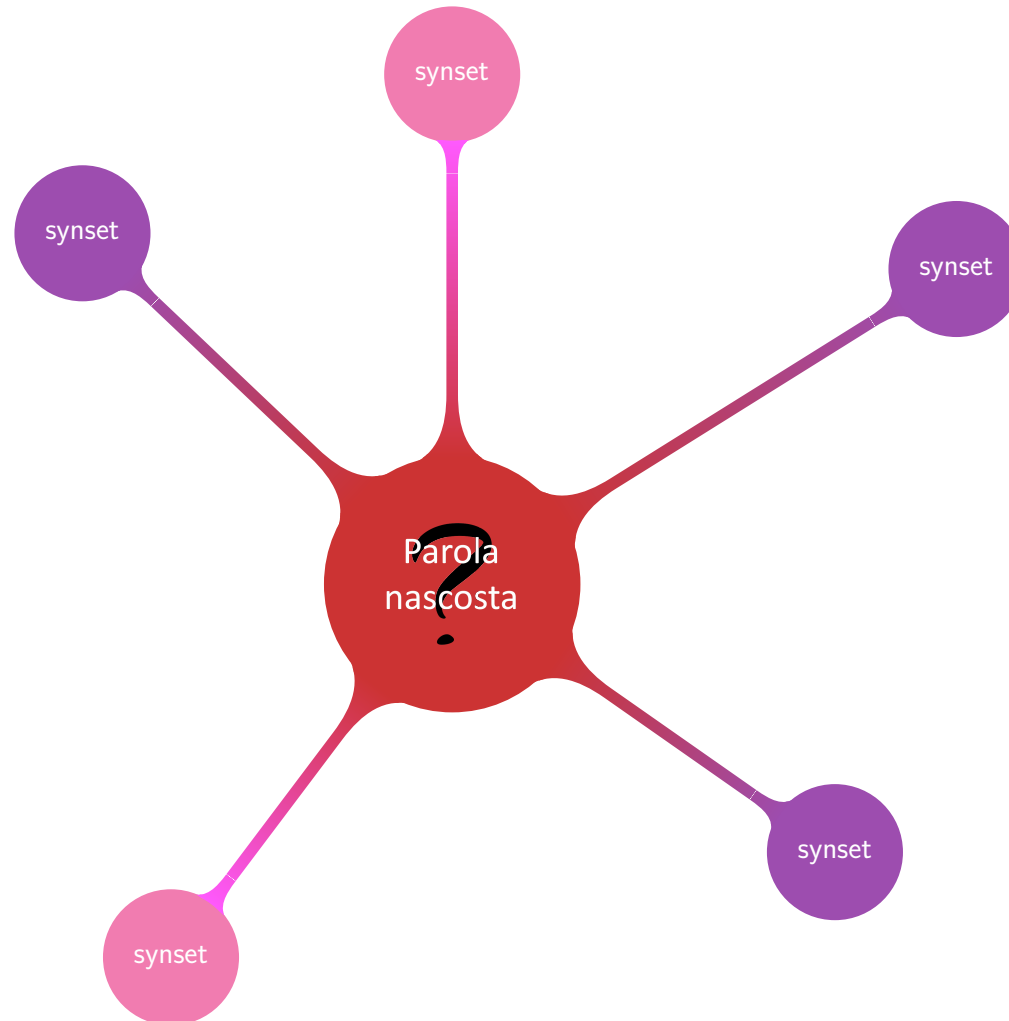
Algoritmo di generazione automatica degli schemi di gioco:



L'applicazione (5/7)

Algoritmo di generazione schemi

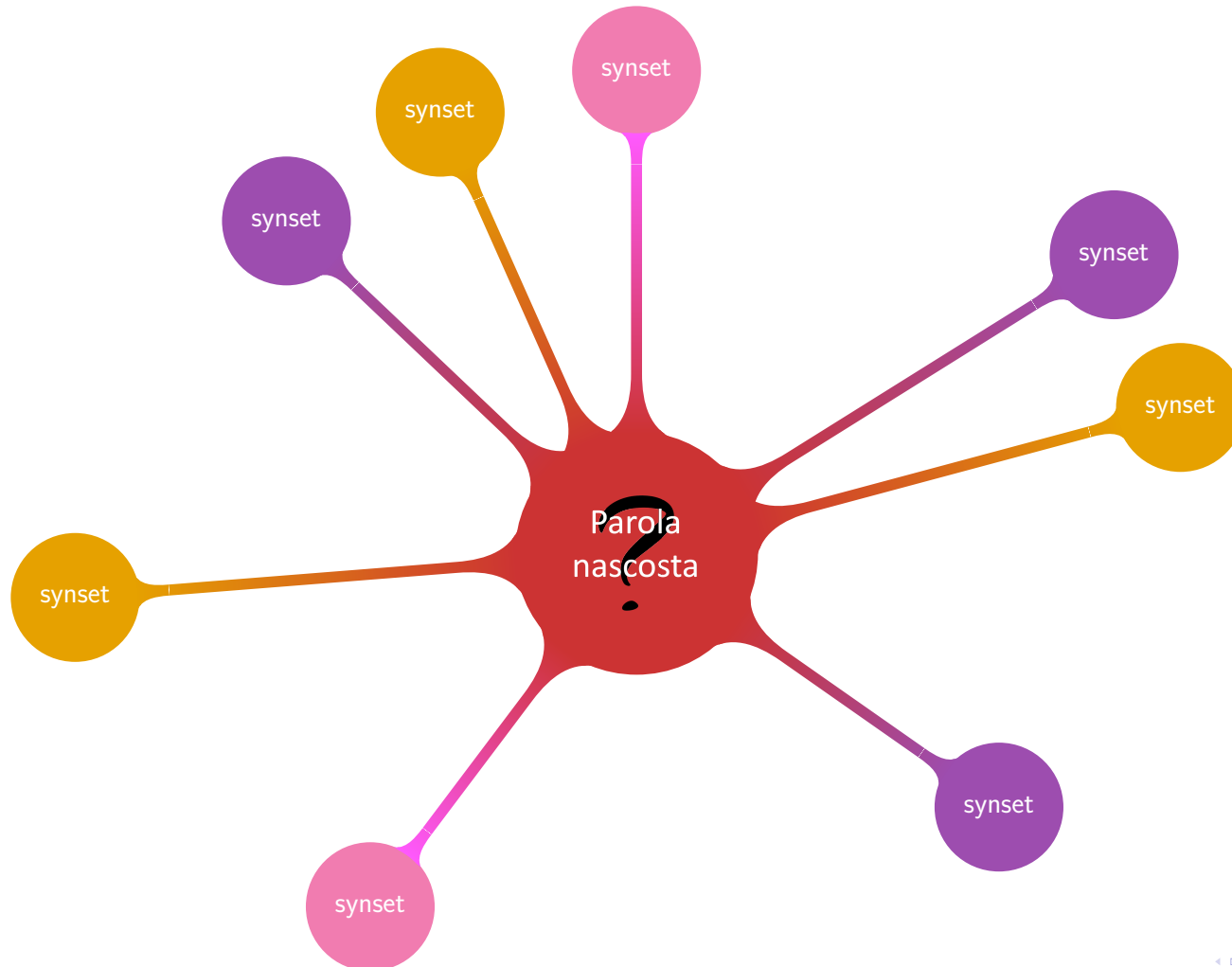
Algoritmo di generazione automatica degli schemi di gioco:



L'applicazione (5/7)

Algoritmo di generazione schemi

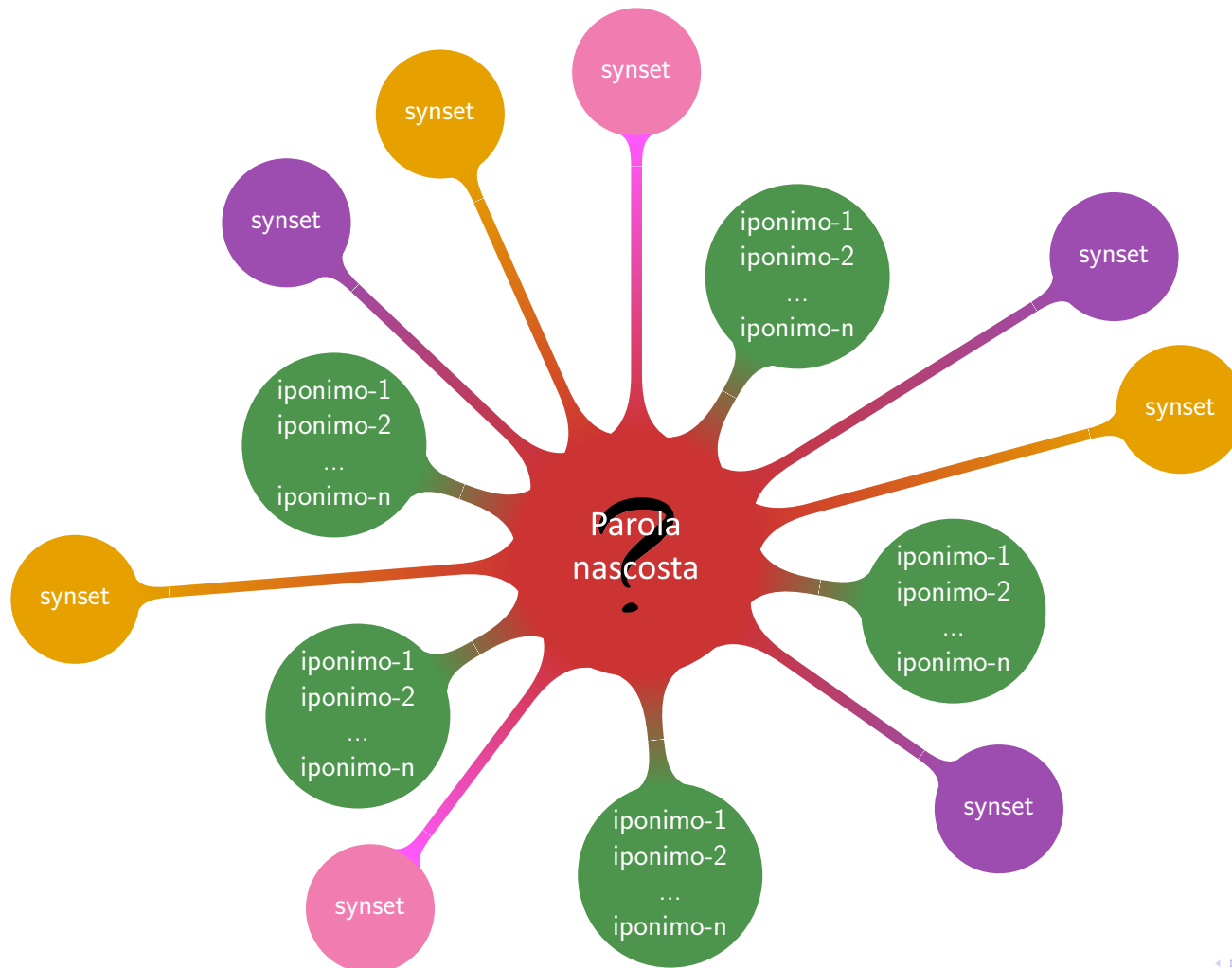
Algoritmo di generazione automatica degli schemi di gioco:



L'applicazione (5/7)

Algoritmo di generazione schemi

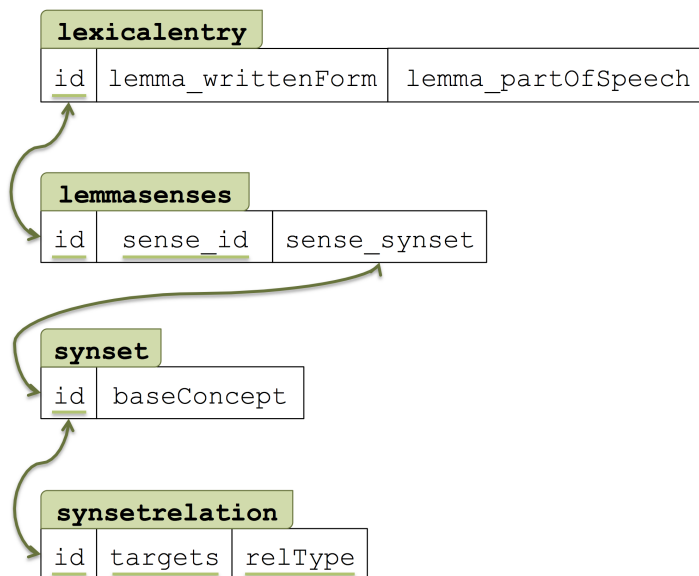
Algoritmo di generazione automatica degli schemi di gioco:



L'applicazione (6/7)

Algoritmo di generazione schemi

L'implementazione dell'algoritmo precedentemente descritto è stata realizzata principalmente in linguaggio SQL dopo la trasformazione del grafo in tabelle relazionali secondo il seguente schema:



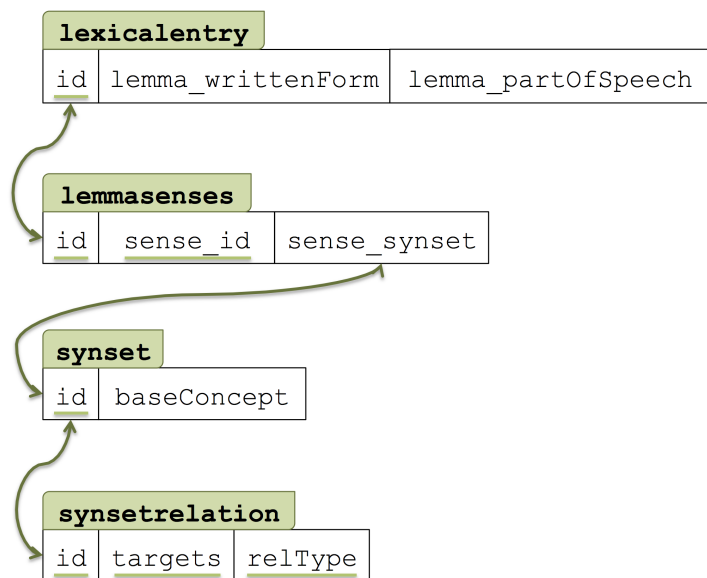
Recupero parola nascosta

```
let db = SQLiteDatabase.sharedInstance()
var array_tuple: [(writtenForm:String,
                    synset:String,
                    targets:String)] = []
let parole = db.query( "select lex.lemma_writtenForm,
                          lem.sense_synset,
                          s1.targets
                        from lexicalentry lex,
                          lemmasenses lem,
                          synsetrelation s1
                        where s1.reltype='hypo'
                          and s1.id = lem.sense_synset
                          and lex.id = lem.id
                          and lex.lemma_writtenForm <> 'GAP!'
                          and lex.lemma_writtenForm <> 'PSEUDOGAP!'
                          and s1.id <> s1.targets and 6<= (
                            select count (distinct s2.targets)
                            from synsetrelation s2
                            where s1.id=s2.id
                            and s2.reltype='hypo')")
for i in 0...parole.count-1{
    let riga_parola = parole[i].data
    let tupla = (
        writtenForm:riga_parola["lemma_writtenForm"]!.asString(),
        synset:riga_parola["sense_synset"]!.asString(),
        targets:riga_parola["targets"]!.asString()
    )
    array_tuple.append(tupla)
}
var i = Int(arc4random_uniform(UInt32(array_tuple.count))+1)
var parola_nascosta = array_tuple[i].writtenForm
```

L'applicazione (6/7)

Algoritmo di generazione schemi

L'implementazione dell'algoritmo precedentemente descritto è stata realizzata principalmente in linguaggio SQL dopo la trasformazione del grafo in tabelle relazionali secondo il seguente schema:



Recupero parola nascosta

```
let db = SQLiteDatabase.sharedInstance()
var array_tuple: [(writtenForm:String,
                    synset:String,
                    targets:String)] = []
let parole = db.query( "select lex.lemma_writtenForm,
                        lem.sense_synset,
                        s1.targets
                        from lexicalentry lex,
                        lemmasenses lem,
                        synsetrelation s1
                        where s1.reltype='hypo'
                        and s1.id = lem.sense_synset
                        and lex.id = lem.id
                        and lex.lemma_writtenForm <> 'GAP!'
                        and lex.lemma_writtenForm <> 'PSEUDOGAP!'
                        and s1.id <> s1.targets and 6<= (
                        select count (distinct s2.targets)
                        from synsetrelation s2
                        where s1.id=s2.id
                        and s2.reltype='hypo')")
for i in 0...parole.count-1{
    let riga_parola = parole[i].data
    let tupla = (
        writtenForm:riga_parola["lemma_writtenForm"]!.asString(),
        synset:riga_parola["sense_synset"]!.asString(),
        targets:riga_parola["targets"]!.asString()
    )
    array_tuple.append(tupla)
}
var i = Int(arc4random_uniform(UInt32(array_tuple.count))+1)
var parola_nascosta = array_tuple[i].writtenForm
```

L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



L'applicazione (7/7)

Interfaccia grafica

Il gioco prevede un'interfaccia molto semplice per l'utente:

- ▶ Parole tra cui scegliere in alto a sinistra
- ▶ Montepremi aggiornato in alto a destra
- ▶ Lista delle *parole-indizio* al centro colorate in base alla selezione
- ▶ Campo di testo per scrivere la propria soluzione in basso
- ▶ Pulsante “Parola nascosta” per verificare la soluzione



Conclusione e sviluppi futuri

In conclusione sono stati realizzati i seguenti software:

- ▶ Software per la conversione dei dati dal formato XML a quello SQL
- ▶ Game application in Swift con calcolo automatico degli schemi di gioco

Sviluppi futuri:

- ▶ migliorare l'*efficienza* dei tempi di risposta del sistema
- ▶ rendere l'interfaccia grafica più *user friendly*
- ▶ introduzione dei punti esperienza
- ▶ sfide con amici

Conclusione e sviluppi futuri

In conclusione sono stati realizzati i seguenti software:

- ▶ Software per la conversione dei dati dal formato XML a quello SQL
- ▶ Game application in Swift con calcolo automatico degli schemi di gioco

Sviluppi futuri:

- ▶ migliorare l'*efficienza* dei tempi di risposta del sistema
- ▶ rendere l'interfaccia grafica più *user friendly*
- ▶ introduzione dei punti esperienza
- ▶ sfide con amici

Conclusione e sviluppi futuri

In conclusione sono stati realizzati i seguenti software:

- ▶ Software per la conversione dei dati dal formato XML a quello SQL
- ▶ Game application in Swift con calcolo automatico degli schemi di gioco

Sviluppi futuri:

- ▶ migliorare l'*efficienza* dei tempi di risposta del sistema
- ▶ rendere l'interfaccia grafica più *user friendly*
- ▶ introduzione dei punti esperienza
- ▶ sfide con amici

Conclusione e sviluppi futuri

In conclusione sono stati realizzati i seguenti software:

- ▶ Software per la conversione dei dati dal formato XML a quello SQL
- ▶ Game application in Swift con calcolo automatico degli schemi di gioco

Sviluppi futuri:

- ▶ migliorare l'*efficienza* dei tempi di risposta del sistema
- ▶ rendere l'interfaccia grafica più *user friendly*
- ▶ introduzione dei punti esperienza
- ▶ sfide con amici

Conclusione e sviluppi futuri

In conclusione sono stati realizzati i seguenti software:

- ▶ Software per la conversione dei dati dal formato XML a quello SQL
- ▶ Game application in Swift con calcolo automatico degli schemi di gioco

Sviluppi futuri:

- ▶ migliorare l'*efficienza* dei tempi di risposta del sistema
- ▶ rendere l'interfaccia grafica più *user friendly*
- ▶ introduzione dei punti esperienza
- ▶ sfide con amici

Conclusione e sviluppi futuri

In conclusione sono stati realizzati i seguenti software:

- ▶ Software per la conversione dei dati dal formato XML a quello SQL
- ▶ Game application in Swift con calcolo automatico degli schemi di gioco

Sviluppi futuri:

- ▶ migliorare l'*efficienza* dei tempi di risposta del sistema
- ▶ rendere l'interfaccia grafica più *user friendly*
- ▶ introduzione dei punti esperienza
- ▶ sfide con amici

Conclusione e sviluppi futuri

In conclusione sono stati realizzati i seguenti software:

- ▶ Software per la conversione dei dati dal formato XML a quello SQL
- ▶ Game application in Swift con calcolo automatico degli schemi di gioco

Sviluppi futuri:

- ▶ migliorare l'*efficienza* dei tempi di risposta del sistema
- ▶ rendere l'interfaccia grafica più *user friendly*
- ▶ introduzione dei punti esperienza
- ▶ sfide con amici

Conclusione e sviluppi futuri

In conclusione sono stati realizzati i seguenti software:

- ▶ Software per la conversione dei dati dal formato XML a quello SQL
- ▶ Game application in Swift con calcolo automatico degli schemi di gioco

Sviluppi futuri:

- ▶ migliorare l'*efficienza* dei tempi di risposta del sistema
- ▶ rendere l'interfaccia grafica più *user friendly*
- ▶ introduzione dei punti esperienza
- ▶ sfide con amici

Grazie per l'attenzione

