

Final - Dev Exam - 15 Dic 2015

Nome e Cognome.....

Numero di matricola..... Github user:

1- Santa Little Helper: Xmas Game for Kids

Il tuo team sta sviluppando un gioco per bambini con tema natalizio. Il gioco consiste nell'aiutare Santa a processare le richieste dei bambini. L'app dovrebbe funzionare secondo le seguenti specifiche:

 Santa little helper

Carlo wants a TOY. Shall I pack a banana?

- yes
- no

 Santa little helper

Packed 2

Le domande e le opzioni sono presentate

Alla fine e' visualizzato il totale di risposte giuste

La sessione di gioco visualizza una domanda alla volta, con le opzioni (risposte) possibili (figura a sinistra). Per ogni domanda, l'utente deve rispondere scegliendo una delle risposte. Se la risposta è giusta, l'utente guadagna un punto, zero altrimenti. Alla fine, il totale di punti e' visualizzato (figura a destra).

Primo Sprint

1. All'apertura della pagina viene visualizzato all'utente la prima richiesta.
2. Al cliccare su una risposta, i punti guadagnati sono calcolati, e il gioco si sposta alla prossima domanda.
3. Alla fine, il totale di punti guadagnati e' visualizzato.

In questa fase, le domande si trovano nell'array "requests" e hanno il formato:

```
var requests = [  
{  
  question : "Carlo wants a TOY. Shall I pack a banana?", // domanda di Santa  
  options : ["yes", "no"], // opzioni visualizzate all'utente  
  answer : "no" // risposta giusta  
}, ...  
]
```

Final - Dev Exam - 15 Dic 2015

Il vostro team ha deciso di strutturare la gestione delle domande - per facilitare il lavoro degli sviluppatori e tester - in questo modo:

<pre>var SantaModel = { <i>Initializes the model with a list of requests, and sets the first as the current one</i> init : function(list){ }, <i>It moves "current" to the next request</i> next : function () { }, <i>Returns the current request. If all requests have been processed (there is no current one), it returns null</i> getCurrentRequest : function () { }, <i>Packs the given item if it fulfils the current request. returns 1 if the given item fulfils the request (= answer) returns 0 if the given item does not fulfil the request</i> pack : function(item) { } };</pre>	<p>Example of usage:</p> <pre>SantaModel.init(requests); SantaModel.getCurrentRequest() -> { question : "Carlo wants a TOY. Shall I pack a banana?", options : ["yes", "no"], answer : "no" } SantaModel.pack("yes") -> 0 SantaModel.next() SantaModel.getCurrentRequest() -> { question : "Julia wants a doll. Shall I pack a barbie?", options : ["yes", "no"], answer : "yes" }</pre>
---	---

I tuo compito:

- 1- Implementa il gioco **Santa Little Helper** (vi suggeriamo di utilizzare js patterns)
- 2- Implementa casi di test per le funzioni di **SantaModel**: **getCurrentRequest** e **pack**.

Guidelines

- Fai Fork del repository su <https://github.com/mbaezpy/trentose2015-final.git>
- Crea un branch **sprint1**, e fai commit della tua soluzione.

Directory structure:

- index.html -> il documento principale
- app.js -> qui dovete scrivere la logica della vostra app
- data.js -> qui si trovano i dati di prova. Non modificare questo file.
- test
 - SpecRunner.html -> to run the tests
 - spec/app.spec.js -> here you should write your test cases

Secondo Sprint

A qualche punto durante l'esame annunceremo cambiamenti dei requisiti. Questi requisiti saranno su un nuovo branch **sprint2** e dovrai fare commit della tua soluzione sullo stesso branch (sprint2).

Non dovrai sviluppare casi di test per questo branch. Dettagli su README.md