

tesi di laurea

Strumenti per l'automazione del testing di applicazioni web Javascript-based

Anno Accademico 2005/2006

relatore

Ch.mo prof. Porfirio Tramontana

candidato

Salvatore Agnello

Matr. 41/2612

Sommario

- Introduzione.
- Problematiche.
- Obiettivi.
- Tecnologie utilizzate.
 - Passi del processo.
 - Programma interfaccia
- Esempio.

Introduzione (1).

Applicazioni web.

- Possono essere definite come l'insieme di servizi ed applicazioni che vengono utilizzate sul web.
- Gli utenti accedono tramite un browser.
- l'interazione con una applicazione web è indistinguibile dall'accesso a un sito Web statico.
- Le pagine visualizzate dal browser, in questo caso, saranno però generate dinamicamente dall'applicazione.

Introduzione (2).

Applicazioni web.

Si pongono come valida alternativa dalle tradizionali applicazioni Client - Server per vari motivi:

- facilità di distribuzione e aggiornamento.
- accesso multiplatforma.
- riduzione del costo di gestione.
- scalabilità.

Introduzione (3).

Testing per applicazioni web.

- L'attività di testing di una applicazione web è molto più complessa di quella tradizionale;
- Piccoli cambiamenti alla infrastruttura Web, possono influire drasticamente sulle sue funzionalità;

Testing per applicazioni web Javascript-based.

Per effettuare questa operazione si deve tener presente:

- La struttura base di una odierna applicazione web è basata sul concetto di: “**dinamicità dell'applicazione**”;
- Insieme di tecnologie che permettono di cambiare in modo dinamico la rappresentazione e il contenuto di un documento ed aumentare l'interattività dell'utente con l'applicazione.
- Tali meccanismi sono realizzabili tramite Javascript.

Problematiche.

Problematiche associate al testing di applicazioni web:

- **Diverse tipologie di rete;**
- **Perdita di informazioni;**
- **Diverse versioni del browser;**
- **Diverse tipologie di postazioni per l'accesso ai servizi.**

Obiettivi

Realizzare uno strumento con le seguenti caratteristiche:

- **Unicità**, deve essere uno strumento di tipo generale, che sia in grado di effettuare il testing di una qualsiasi applicazione web Javascript.
- **Parametrizzazione**, prelevare da fonte esterna tutte le informazioni necessarie per eseguire il testing sull'applicazione web.
- **Automaticità**, Tutte queste operazioni devono essere svolte in modo del tutto automatico, lasciando allo sviluppatore il solo onere di decidere che tipo di operazioni compiere.

Tecnologie utilizzate (1): Selenium

Sviluppato dal team ThoughtWorks, ha le seguenti caratteristiche:

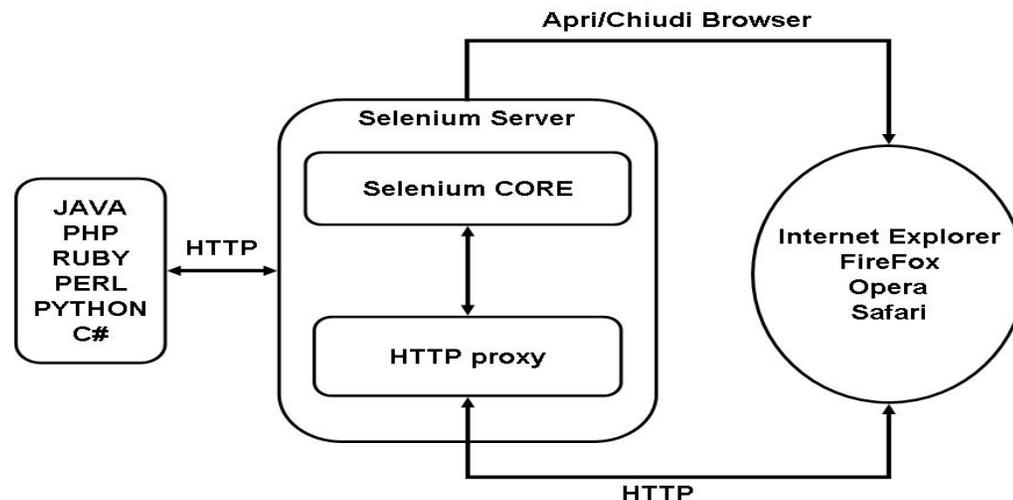
- Tool open source capace di eseguire test per web application;
- I test vengono eseguiti direttamente in un browser, proprio come se ci fosse un utente ad eseguire tali operazioni;
- Utilizzabile su di un gran numero di piattaforme.

Windows	Mac OS X	Linux
Internet Explorer 6.0, 7.0	Safari 1.3+	Firefox 0.8 to 1.5
Firefox 0.8 to 1.5, 2.0+	Firefox 0.8 to 1.5	Mozilla Suite 1.6+, 1.7+
Mozilla Suite 1.6+, 1.7+	Camino 1.0a1	Konqueror
Seamonkey 1.0	Mozilla Suite 1.6+, 1.7+	
Opera 8	Seamonkey 1.0	

Tecnologie utilizzate (2).

Selenium si suddivide in tre parti, ognuna delle quali mette a disposizione dello sviluppatore strumenti e metodi per eseguire dei test automatici.

Selenium CORE, Selenium IDE, Selenium RC (Remote Control).



Tecnologie utilizzate: Selenium CORE

- Installazione all'interno di una delle cartelle del web server;
- Utilizzo del "TestRunner.html", struttura usata per eseguire i test.

Test Suite

Playback

Test Player	
open	/selenium-core/usage.html
click	link=Reference
waitForPageToLoad	30000
click	link=Usage
waitForPageToLoad	30000
click	link=Wiki
waitForPageToLoad	30000

Selenium TestRunner

Execute Tests

Fast Slow

Highlight elements

Elapsed: 00.00

Tests	Commands
■ run	■ passed
■ failed	■ failed
	■ incomplete

Tools

↑
Test Suite

↑
Current Test

↑
Control Panel

Selenium

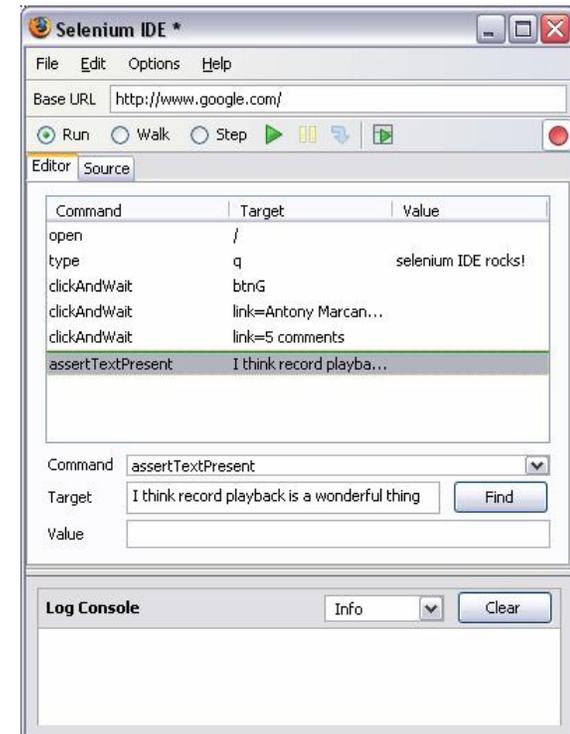
by ThoughtWorks and friends
For more information on Selenium, visit
<http://selenium.openqa.org>



Tecnologie utilizzate: Selenium IDE

Ambiente integrato di sviluppo implementato come estensione per FireFox, le caratteristiche principali sono:

- Semplicità di registrazione e playback;
- Selezione intelligente dei campi atti a contenere ID, nomi o XPath;
- Debug;
- Possibilità di settare i Breakpoint;
- Salvataggio dei test in diversi formati: HTML, Java, C#, Ruby, Perl, PHP, Python, ecc...;
- Supporto per file “.js”;
- Possibilità di importare anche tabelle scritte in Excel;



Tecnologie utilizzate: Selenium RC

Le caratteristiche principali del Remote Control:

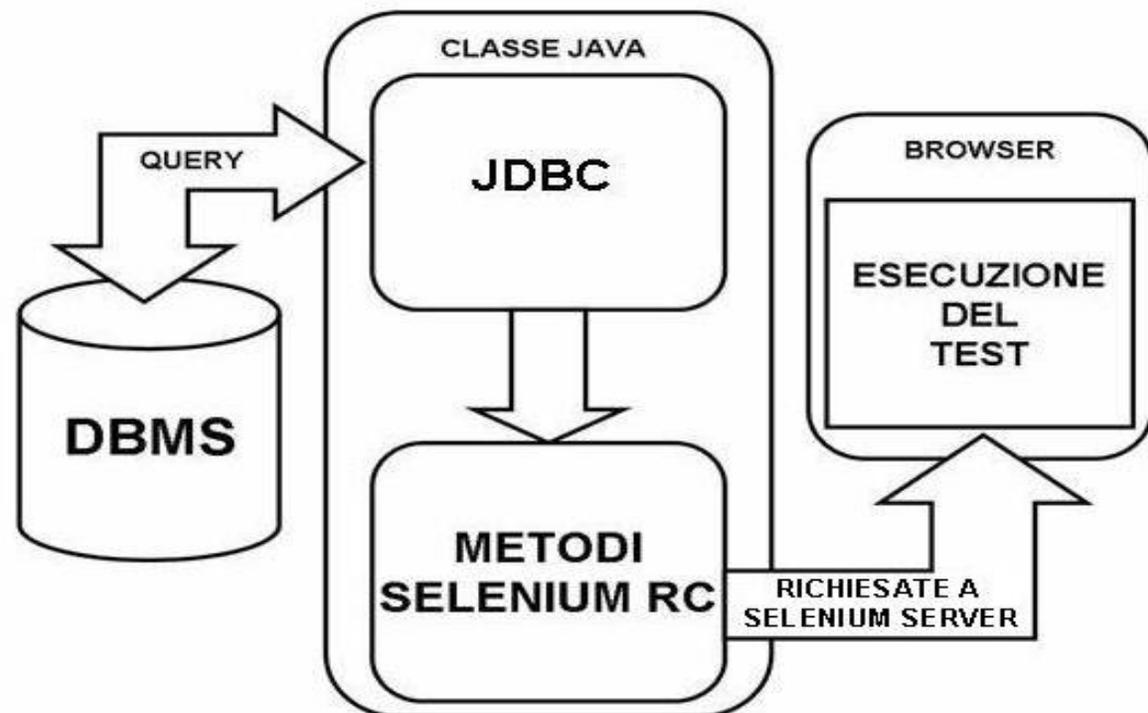
- Permette di scrivere test in diversi linguaggi di programmazione,;
- Fornisce il cosiddetto “Selenium Server”, per avviare/fermare/controllare i browser oggi disponibili;
- La versione RC lavora anch'essa utilizzando il CORE.

Le modalità di utilizzo di questa versione sono:

- ***Interactive Mode:***
generare test direttamente dalla linea di comando;
- ***Programming Mode:***
utilizzo di metodi per eseguire il testing automatico.

Soluzione proposta.

Realizzazione di una classe Java con la seguente struttura:

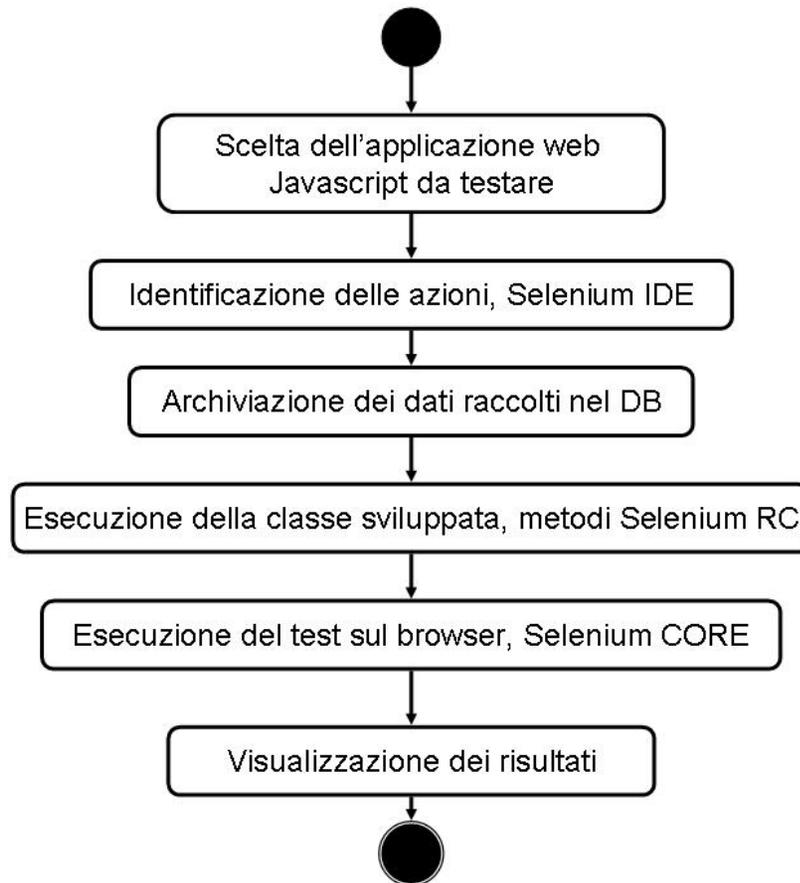


Passi del processo (1).

Passi che il tester deve seguire:

1. **Ricerca dell'applicazione web da testare;**
2. **Identificazione delle azioni**, riconoscere quelle che sono le azioni compiute dall'utente nel corso di un normale utilizzo dell'applicazione;
3. **Caricamento della base dati**, nella quale mettere tutte le informazioni del punto precedente;
4. **Esecuzione della classe**, la quale sia in grado di eseguire in automatico il testing, basandosi sulle informazioni precedenti.

Passi del processo (2).



Interfaccia (1).

Connessione a Selenium Server, dalla riga di comando:

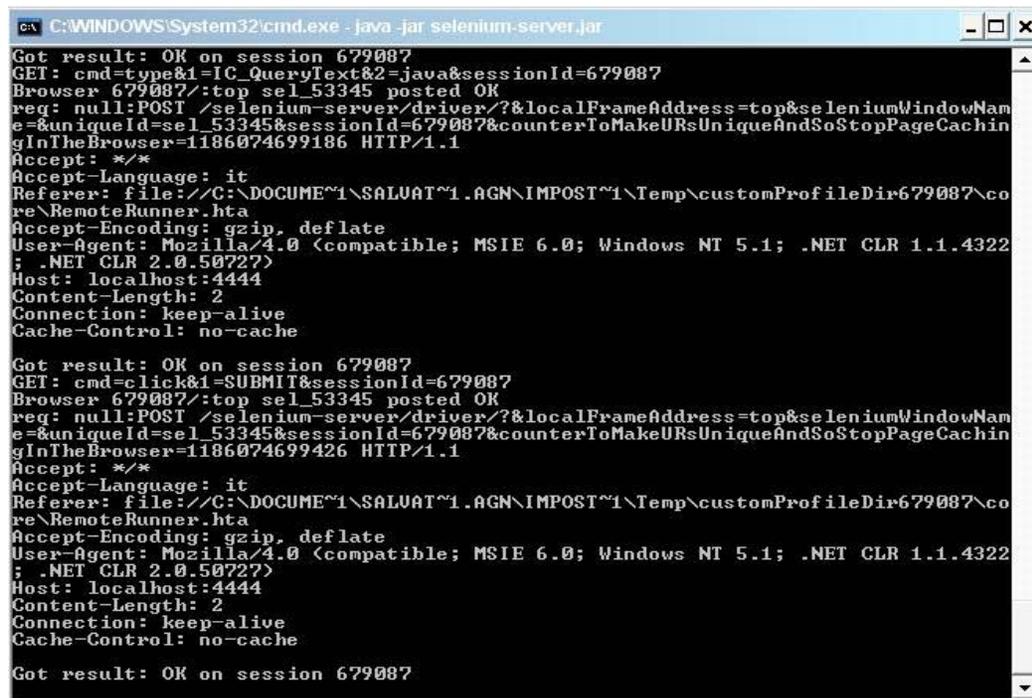
“java -jar selenium-server.jar”

```
application/xhtml+xml
2-ago-2007 19.01.54 org.mortbay.http.HttpServer doStart
INFO: Version Jetty/0.9.0
2-ago-2007 19.01.55 org.mortbay.util.Container start
INFO: Started HttpContext[/,/]
2-ago-2007 19.01.55 org.mortbay.util.Container start
INFO: Started HttpContext[/selenium-server,/selenium-server]
2-ago-2007 19.01.55 org.mortbay.util.Container start
INFO: Started HttpContext[/selenium-server/driver,/selenium-server/driver]
2-ago-2007 19.01.55 org.mortbay.http.SocketListener start
INFO: Started SocketListener on 0.0.0.0:4444
2-ago-2007 19.01.55 org.mortbay.util.Container start
INFO: Started org.mortbay.jetty.Server@a62fc3
```

Connessione avvenuta correttamente e con la configurazione sopra indicata.

Interfaccia (2).

Finestra di comunicazione nella quale verranno visualizzate tutte le richieste al server.



```
c:\WINDOWS\System32\cmd.exe - java -jar selenium-server.jar
Got result: OK on session 679087
GET: cmd=type&i=IC_QueryText&2=java&sessionId=679087
Browser 679087/:top sel_53345 posted OK
req: null:POST /selenium-server/driver/?&localFrameAddress=top&seleniumWindowName=&uniqueId=sel_53345&sessionId=679087&counterToMakeURsUniqueAndSoStopPageCachingInTheBrowser=1186074699186 HTTP/1.1
Accept: */*
Accept-Language: it
Referer: file:///C:/DOCUME~1\SALVAT~1\AGN\IMPOST~1\Temp\customProfileDir679087\core\RemoteRunner.hta
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)
Host: localhost:4444
Content-Length: 2
Connection: keep-alive
Cache-Control: no-cache

Got result: OK on session 679087
GET: cmd=click&i=SUBMIT&sessionId=679087
Browser 679087/:top sel_53345 posted OK
req: null:POST /selenium-server/driver/?&localFrameAddress=top&seleniumWindowName=&uniqueId=sel_53345&sessionId=679087&counterToMakeURsUniqueAndSoStopPageCachingInTheBrowser=1186074699426 HTTP/1.1
Accept: */*
Accept-Language: it
Referer: file:///C:/DOCUME~1\SALVAT~1\AGN\IMPOST~1\Temp\customProfileDir679087\core\RemoteRunner.hta
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)
Host: localhost:4444
Content-Length: 2
Connection: keep-alive
Cache-Control: no-cache

Got result: OK on session 679087
```

Interfaccia (3): DefaultSelenium

Appartiene ai “**Costructor**” forniti dal tool, attraverso di esso che i metodi definiti nella classe possono essere inoltrate come richieste al server. La sua sintassi è:

```
DefaultSelenium(java.lang.String serverHost,  
                int serverPort,  
                java.lang.String browserStartCommand,  
                java.lang.String browserURL)
```

Nella classe:

.....

.....

```
selenium = new DefaultSelenium("localhost", 4444, "*firefox",target[m] );
```

.....

.....

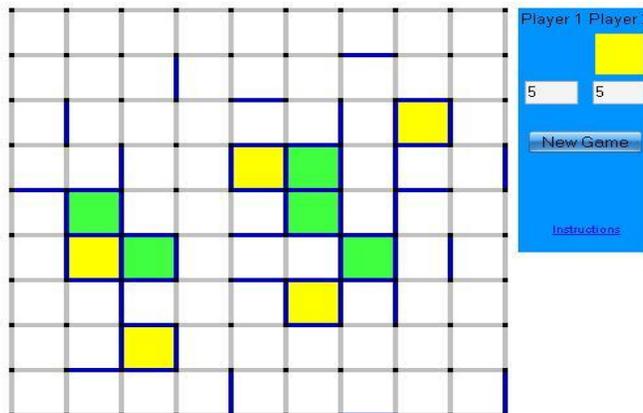
Esempio (2): L'applicazione JS.

- L'applicazione presa in esame è quella relativa ad un gioco completamente scritto in Javascript, "Squares",
- Verranno testate anche tutte le funzionalità di contorno a tale applicazione

[Home](#) ▶ [Games](#) ▶ [Squares](#)

Test your strategic ability with this classic example of the game Squares. Be careful, it's very addicting!

[Download the Required Image Files](#)



[Sign up for the JavaScript Weekly](#)

[internet.com Privacy Policy](#)

We'll send this script to you!

(just click "Send it!" once!)

[\[Contact Us \]](#)
[\[JavaScript Forum \]](#)
[\[Submit Your Script \]](#)

Visit the IBM Web Services Learning Center on DevX
> [Click Here](#)

JupiterOnlineMedia.

[internet.com](#) | [EARTHWEB](#) | [dev](#) | [graphics.com](#)

Search:

Jupitermedia Corporation has two divisions: [Jupiterimages](#) and [JupiterOnlineMedia](#)

[Jupitermedia Corporate Info](#)

Copyright 2007 Jupitermedia Corporation All Rights Reserved.
[Legal Notices](#), [Licensing](#), [Reprints](#), & [Permissions](#), [Privacy Policy](#).

[Web Hosting](#) | [Newsletters](#) | [Tech Jobs](#) | [Shopping](#) | [E-mail Offers](#)

Esempio (2): Il database.

Rappresenta tutte le operazioni che sono state svolte sull'applicazione Web durante la fase di testing:

COMANDI	TARGET	VALORI
open	http://javascript.internet.com/games/squares.html	
click	hx6hy5	
click	vx7vy5	
click	hx7hy8	
click	vx7vy8	
click	hx6hy8	
click	vx8vy8	
click	hx7hy9	
click	//input[@value='Highlight All']	
type	document.scriptbot.email	pasta
click	//input[@value='Send it!']	
assertEquals	Oops, something is wrong....\n\nInvalid email address.	
click	vx8vy7	
click	vx7vy6	
click	vx7vy6	
click	vx8vy6	
click	hx7hy6	
type	IC_QueryText	sugo
click	SUBMIT	
waitForPageToLoad		

Esempio (3): Esecuzione del test.

La classe esegue il testing sull'applicazione attraverso l'interfaccia del CORE:



```
Got result: OK on session 50185
GET: cmd=getAlert&sessionId=50185
Browser 50185/:top sel_16385 posted OK,Oops, something is wrong...
Invalid email address. Your email address is missing an '@' sign and a '.' in t
he domain name (like '.com'). Please check your address then submit again.
req: null:POST /selenium-server/driver/?&localFrameAddress=top&sessionId=sel_16385&sessionId=50185&counterToMakeURsUniqueAndSoStopPageCaching
InTheBrowser=1183916059058 HTTP/1.1
Accept: */*
Accept-Language: it
Referer: file:///C:/DOCUME~1/SALVAT~1/AGN/IMPOST~1/Temp/customProfileDir50185\cor
e\RemoteRunner.hta
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322
; .NET CLR 2.0.50727)
Host: localhost:4444
Content-Length: 188
Connection: keep-alive
Cache-Control: no-cache

Got result: OK,Oops, something is wrong...
Invalid email address. Your email address is missing an '@' sign and a '.' in t
he domain name (like '.com'). Please check your address then submit again. on s
ession 50185
```

Esempio (3): Report degli esperimenti.

Visualizzazione del file log, attraverso una pagina html, al termine della fase di testing:

```
Selenium Log Console ○Error ○Warn ○Info
info: Log window displayed
info: Adding loadListener for [object HTMLIFrameElement], function () { return __method.apply(object, args.concat($A(arguments))); }
debug: currentTest.start()
debug: currentTest.continueTest() - acquire the next command
debug: currentTest.continueTestAtCurrentCommand()
debug: currentTest.resume() - actually execute
debug: runScheduledPollers
debug: runScheduledPollers DONE
info: Executing: |open | /keiki_page/javascript_games/checkers_/checkers.html | |
debug: Command found, going to execute open
debug: modifyWindow 1183133054762:true
debug: modifySeparateTestWindowToDetectPageLoads: already polling this window: selenium1183133054762
debug: getCurrentWindow newPageLoaded = false
debug: openLocation newPageLoaded = false
debug: runScheduledPollers
debug: runScheduledPollers DONE
debug: Page load detected
debug: Page load location=http://www.tammyyee.com/keiki_page/javascript_games/checkers_/checkers.html
debug: runScheduledPollers
debug: runScheduledPollers DONE
debug: condition satisfied; let's continueTest()
debug: Command complete
debug: currentTest.continueTest() - acquire the next command
debug: currentTest.continueTestAtCurrentCommand()
debug: currentTest.resume() - actually execute
debug: runScheduledPollers
debug: runScheduledPollers DONE
info: Executing: |click | space62 | |
debug: Command found, going to execute click
debug: modifyWindow 1183133054762:undefined
debug: modifySeparateTestWindowToDetectPageLoads: already polling this window: selenium1183133054762
debug: getCurrentWindow newPageLoaded = false
debug: modifyWindow 1183133054762:true
```

Conclusioni.

Lo strumento realizzato risulta essere molto efficiente, semplice da utilizzare in quanto richiede solo di caricare il database, versatile in quanto può essere utilizzato su piattaforme diverse e utilizzare i browser più comuni per il testing automatico.

I miglioramenti che possono essere apportati a tale strumento riguardano per lo più l'aggiunta di nuovi metodi di comparazione al fine di aumentare le sue potenzialità.