

Relazione al progetto: Naruto Hand Seals Fighting del corso di Sistemi e Applicazioni Multimediali

Michele Tamburini
mtamburi@cs.unibo.it

3 gennaio 2011

1 Abstract

Il progetto nasce con lo scopo di esplorare uno dei campi della *Augmented Reality*, che prevede l'uso delle mani nell'interazione uomo-macchina. La realizzazione ultima consiste di un gioco di combattimento in modalità picchiaduro che prende ambientazione e scenografia dall'ormai noto anime giapponese *Naruto*. Le mosse per così dire speciali, dovranno essere attivate non con semplici combinazioni di tasti, ma attraverso le posizioni ed i gesti delle mani dell'utente, acquisiti attraverso una webcam, che saranno poi analizzati da un motore di riconoscimento per verificarne l'accuratezza. L'implementazione del presente prevede il motore di riconoscimento e una semplice sezione di addestramento per l'utente, che lo coinvolga nell'apprendimento di alcune mosse.

Vengono utilizzate librerie *OpenCV* per l'analisi delle immagini, *Guichan* per l'interfaccia grafica, *SDL* ed il linguaggio C++.

2 Stato dell'arte

La ricerca di tecniche che favoriscano metodologie di utilizzo dei calcolatori attraverso forme di interazione sempre più innovative, quali ad esempio l'uso del proprio corpo, ha aperto la strada già da qualche tempo a nuove sfide per il mondo informatico. Il campo di ricerca dell' "*hand tracking*" vuole dare all'utilizzatore del calcolatore la possibilità di un controllo, completo o parziale, di questo, attraver-

so i complicati movimenti o stimoli che la mano umana è in grado di fornire. L'acquisizione di questi, sia essa agevolata dall'ausilio di sensori particolari, guanti o addirittura con la mano nuda, risulta essere la prima problematica da affrontare, comune a tutte le proposte fino ad ora osservate ([11] [16] [17] [20] [19]). Gli spunti proposti nella bibliografia, lontano dall'essere esaustiva sull'argomento, vogliono semplicemente offrire una panoramica ad ampio spettro di alcune soluzioni a tale problema.

3 Progetto: introduzione

3.1 Ambientazione

L'ormai celeberrimo anime "*Naruto*" ha portato anche in Italia una storia di fantasia dalle forti connotazioni orientali. Tra gli ingredienti maggiormente apprezzati dal grande pubblico risiede il fatto che i personaggi utilizzano movimenti delle mani estremamente complicati per lanciare mosse di arti marziali speciali. Ognuna di queste è composta da un numero variabile di "sigilli" (posizioni degli arti superiori) che corrispondono ai 12 segni zodiacali cinesi.

Nel seguente progetto ho inserito tutti e soli tali gesti, ovvero non si tengono in considerazione movimenti diversi (come il battito delle mani) che pur compaiono nell'anime. Lo scopo del giocatore sarà quello di riprodurre alcune posizioni che vedrà raffigurate sottoforma di immagini. Ciascuna di queste è denominata segno, o sigillo (signs, seals). Una

mossa (Move) è dunque costituita da un numero variabile di segni presi tra i 12 sopra descritti.

3.2 Il nostro problema nel dettaglio

In questo lavoro non ho certo la pretesa di affrontare un tema così imponente come quello dell'hand tracking. Il problema si riduce infatti ad una analisi di immagini seppur non priva di compromessi e difficoltà. A differenza di alcuni lavori di spicco che usano un database e oggetti esterni per l'inferenza della reale posizione della mano [19], in questo caso non è necessario avere una precisa cognizione del punto di partenza e di fine dell'arto. Il giocatore è libero di eseguire qualunque movimento e, una volta pronto, aziona l'acquisizione del gesto, che si traduce per il calcolatore con: l'acquisizione di alcuni frame di immagine, trasformazione di questi con tecniche di processazione dell'immagine e confronto con un template precedentemente preparato. Un punto che abbatte notevolmente la dimensione del problema, ed evita quindi l'uso di un database, si basa proprio su una caratteristica derivata dall'ambientazione: sebbene le combinazioni che possono avere luogo siano infinite (permutazioni di 12 elementi di lunghezza arbitraria) i diversi gesti da riconoscere rimangono comunque 12.

3.3 Pensando in grande

Fondamentalmente questo gioco risiede nella categoria "picchiaduro". L'ingrediente della *Augmented Reality* emerge nel momento in cui l'utente abbandona la tastiera per interagire con le proprie mani attraverso la webcam ed eseguire dal vivo i gesti necessari per la mossa selezionata.

Per ciascuno gli verrà notificato il livello di accuratezza calcolato in una scala da 0 a 100. Al termine dell'ultimo viene poi restituito il voto medio.

4 Introduzione alla Libreria OpenCV

OpenCV è una libreria per l'analisi delle immagini estremamente versatile e completa. Nonostante la mia completa ignoranza a riguardo, mi sono convinto a sfruttarla fin dalle fasi di studio di fattibilità del progetto, spinto sia dal suggerimento avuto dalle lezioni del corso di Sistemi e Applicazioni Mul-

timediali [13] che dall'innumerevole ammontare di commenti positivi reperibili dalla rete.

Sono necessarie solo poche nozioni per prendere dimestichezza con questa. Prima fra tutte la gerarchia delle strutture dati (CvArr, CvMat), ed in particolare quella usata per la memorizzazione dell'immagine: `IplImage`. Quindi l'uso delle principali funzioni di analisi delle immagini: cominciando con l'applicazione dei filtri basilari (Laplaciano, Sobel, di Smoothing...) per arrivare al dominio delle frequenze, rilevazione dei contorni, utilizzo della webcam (o addirittura multi-cam). Estremamente apprezzabile è la possibilità di un approccio "lazy", per usare un termine informatico. In un certo senso è possibile iniziare la fase di programmazione senza preoccuparsi dell'uso preciso di una determinata funzione, utilizzandola dapprima con impostazioni di default, per poi scendere nei dettagli solo in un secondo momento.

Altro pregio della libreria è la facilità della visualizzazione del risultato grafico a schermo (attraverso le funzioni `cvNamedWindow()`, `cvShowImage()`) che consentono una veloce analisi del risultato delle operazioni all'interno di ogni singola parte del progetto.

Bisogna osservare tuttavia che, sebbene i primi capitoli del manuale "Learning OpenCV" [14], collocato probabilmente tra i più consigliati nella rete, siano davvero utili per approcciarsi alla libreria, lo stesso non si possa dire delle sezioni più specifiche, che non sostituiscono un libro di testo per l'analisi delle immagini. Durante tutto il periodo di sviluppo risulta però necessario armarsi del manuale ufficiale [9], disponibile anche online.

Un secondo punto a favore di tale libreria è la reperibilità di numerosi frammenti di codice ed esempi che si possono trovare in rete, ma che chiaramente vanno affrontati con la giusta ponderazione prima di essere utilizzati nel proprio lavoro.

5 Descrizione del progetto

Questa sezione contiene la descrizione della parte centrale del lavoro svolto.

5.1 Sezione di Acquisizione

Con il termine acquisizione ho inteso la cattura di frames attraverso la webcam. Webcam pensata in-

tegrata all'interno del pc portatile, anche se nulla esclude l'utilizzo di un dispositivo diverso, purché riconosciuto dal sistema, e impostato per essere quello di default.

L'oggetto Camera si compone semplicemente di una classe in grado di inizializzare una struttura di tipo Capture (della libreria OpenCV), e sfruttarla per reperire un'immagine di tipo IplImage*. Su questa avranno quindi luogo la fase di processazione e valutazione della macchina di riconoscimento (vedi di seguito).

5.2 Logica di Gioco

Per consentire lo sviluppo di un'applicazione che mostri i connotati per una stesura pressoché completa, è necessario immergerla nello scenario dell'ambientazione almeno dal punto di vista logico. Ho volutamente evitato tediosi dettagli, come ad esempio l'occorrenza nella storia di una particolare mossa o il personaggio che l'ha effettuata per la prima volta, che rendono magari l'anime estremamente interessante, ma che ai fini dell'attuale lavoro non sono poi così rilevanti.

Tuttavia ciascuna mossa viene catalogata in base al proprio tipo, rango di difficoltà e livello, in maniera il più possibile fedele all'anime. Per un compito così strutturato mi sono affidato alla versatilità del linguaggio XML, e l'uso della libreria "TinyXML" [7] per la sua manipolazione. Grazie a questi accorgimenti risulterà forse più agevole l'espansione del codice in futuro.

A questo si è poi aggiunto un forte apporto ingegneristico per rendere fin dal principio tale sezione manutenibile senza troppi sforzi. In figura 1 è riportato il diagramma UML della logica di gioco.

5.3 Stack di Gioco

Non è difficile osservare come questa sezione, così come pure l'interfaccia grafica, non sono state l'oggetto di cure maniacali. In particolare ho voluto creare strutture architetture valide che dessero il massimo riuso di codice, ma allo stesso tempo sufficientemente versatili in termini di espandibilità.

Il risultato di tali ragionamenti è sintetizzato dal diagramma 2. La macchina di gioco (GameMachine) altro non è che un semplice contenitore di elementi, come ad esempio i menu o la sezione di ad-

destramento (TrainingSection), ciascuno dei quali deve provvedere la funzione che ricoprirà il ruolo di ciclo principale. Infatti la GameMachine dispone tutti i moduli contenuti in una struttura a stack ed eseguirà sempre la funzione di quello in cima.

A ciascuno di questi, sia esso un menu, o un modulo differente, viene affidato un controller, che deve essere in grado di gestire eventi conformi all'interfaccia SDL.

5.4 L'interfaccia Grafica

5.5 La macchina per il Riconoscimento

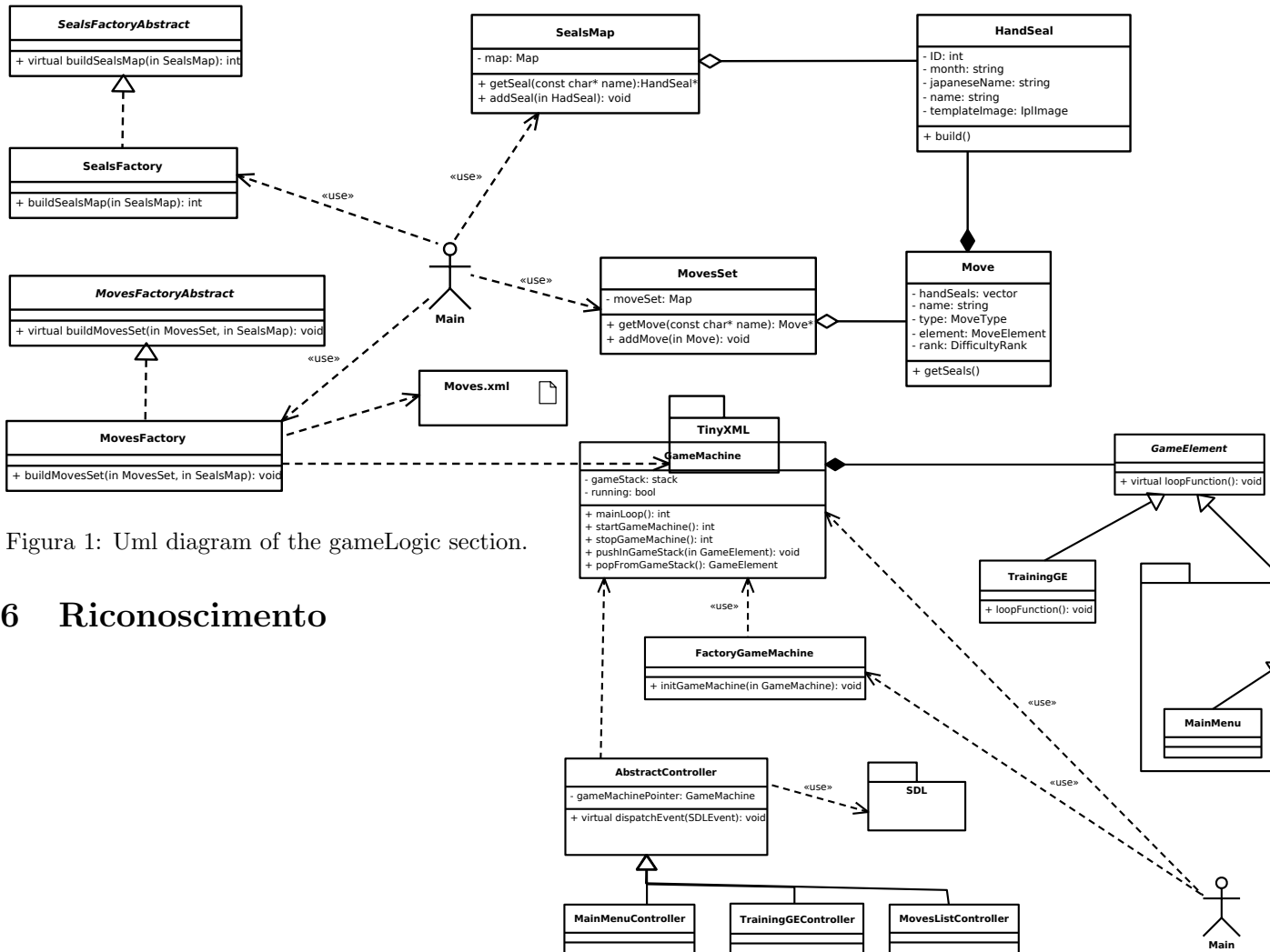


Figura 1: Uml diagram of the gameLogic section.

6 Riconoscimento

Figura 2: Uml diagram of the gameLogic section.

6.1 “The Recognition Engine”

6.2 I moduli

6.3 Le strategie

7 Alcuni Test

8 Osservazioni

9 Conclusioni

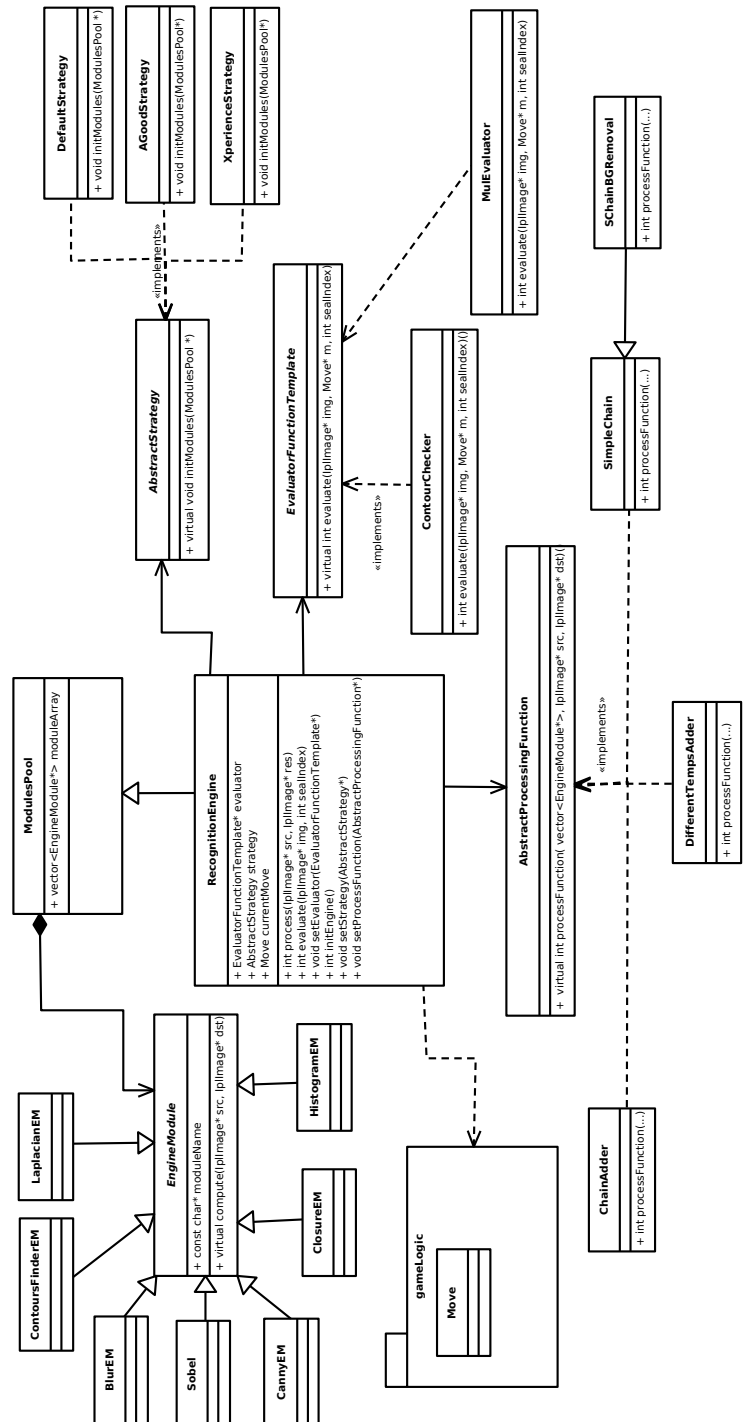


Figura 3: Uml diagram

Riferimenti bibliografici

- [1] Cmake: official wiki. <http://www.cmake.org/Wiki/CMake>.
- [2] Guichan wiki. <http://code.google.com/p/guichan/>.
- [3] Leafninja website. <http://www.leafninja.com/>.
- [4] Opencv: Official wiki. <http://opencv.willowgarage.com/wiki/>.
- [5] Sdl simple directmedia layer forums. <http://forums.libsdl.org/viewtopic.php?t=6442&sid=7e7ea3a081a8906de37de3c2f2548b76>.
- [6] Stackoverflow: [blog. http://stackoverflow.com/questions/393954/how-to-convert-an-opencv-iplimage-to-an-sdl-surface](http://stackoverflow.com/questions/393954/how-to-convert-an-opencv-iplimage-to-an-sdl-surface).
- [7] Tinyxml: xml manager library. <http://sourceforge.net/projects/tinyxml/>.
- [8] The basics of background substraction, Mar 2009. <http://blog.damiles.com/?p=127>.
- [9] *OpenCV Reference Manual v2.1*, Mar, 18 2010.
- [10] Flavio Bernardotti. *Computer Vision Encyclopaedic Forum*. 15100 Alessandria.
- [11] Ramon Mas Cristina Manresa, Javier Varona and Francisco J. Perales. Hand Tracking and Gesture Recognition for Human-Computer Interaction. In *Electronic Letters on Computer Vision and Image Analysis*, volume 5, pages 96–104. 2005.
- [12] Erdem Yörük, Ender Konuko İlu, Bülent San-
kur, Jérôme Darbon. Shape-based hand recog-
nition. *IEEE TRANSACTIONS ON IMAGE
PROCESSING*, 15(7), June 2006.
- [13] Marco Roccetti Paola Salomoni Stefano Fer-
retti. *Slide e appunti di lezione del cor-
so di Sistemi ed Applicazioni Multimediali*,
Feb-May 2009-10. [http://www.cs.unibo.it/
~roccetti/multimedia-09-10.html](http://www.cs.unibo.it/~roccetti/multimedia-09-10.html).
- [14] Adrian Kaehler Gary Bradski. *Learning
OpenCV*. O'Really, 2008.
- [15] Intel Corporation. *Open Source Computer
Vision Library*. <http://developer.intel.com>.
- [16] Reinhard Klein Markus Schlattmann. Si-
multaneous 4 gestures 6 dof real-time two-
hand tracking without any markers. In *ACM
Symposium on Virtual Reality Software and
Technology*.
- [17] Adrien Bernhardt Marie-Paule Cani Paul
G. Kry1, Adeline Pihuit. Handnaviga-
tor: Hands-on interaction for desktop vir-
tual reality. In *ACM Symposium on Vir-
tual Reality Software and Technology*, pa-
ges 53–60, 2008. [http://hal.inria.fr/
inria-00336348/fr/](http://hal.inria.fr/inria-00336348/fr/).
- [18] Aaron Saenz. Mit's ridiculously colorful
glove is latest hand tracking interface. In
human computer interfaces. May, 28 2010.
[http://singularityhub.com/2010/05/28/mits-
ridiculously-colorful-glove-
is-the-latest-hand-tracking-i
nterfacevideo/](http://singularityhub.com/2010/05/28/mits-ridiculously-colorful-glove-is-the-latest-hand-tracking-interfacevideo/).
- [19] Robert Y. Wang. Real-time hand-tracking as
a user input device. Monterey, CA, October
19-22 2008.
- [20] A. Argyros X. Zabulis, H. Baltzakis. Vision-
based hand gesture recognition for human-
computer interaction. In *The Universal Access
Handbook*, 2009.