Gioco di carte con intelligenza artificiale

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi dei costi 4

2.3 Analisi e specifica dei requisiti 4

2.4 Use case 6

2.5 Pianificazione 7

2.6 Analisi dei mezzi 8

2.6.1 Software 8

2.6.2 Hardware 8

3 Progettazione 8

3.1 Design delle interfacce 8

3.1.1 Design pagina iniziale 8

3.1.2 Design scelta della telecamera 9

3.2 Design delle cartelle 9

3.3 Design delle classi 9

4 Implementazione 9

5 Test 9

5.1 Protocollo di test 9

5.2 Risultati test 10

6 Consuntivo 10

7 Conclusioni 10

7.1 Sviluppi futuri 10

7.2 Considerazioni personali 10

8 Bibliografia 10

8.1 Bibliografia per articoli di riviste: 10

8.2 Bibliografia per libri 11

8.3 Sitografia 11

9 Allegati 11

# Introduzione

## Informazioni sul progetto

* Alessandro Colugnat – Allievo
* Ugo Bernasconi – Docente Responsabile
* Scuola Arti e Mestieri Trevano, Informatica
* Data inizio: 08.01.2019
* Data fine: 10.04.2019

## Abstract

The job of this project is to create a robotic arm who plays card game called UNO and he plays with its own intelligence.

## Scopo

Lo scopo del progetto è quello di creare un’intelligenza artificiale che gioca a UNO e avrà una intelligenza che apprende autonomamente in maniera dinamica ciò servirà a far migliorare la IA ogni partita che verrà svolta, si deve implementare anche una videocamera che permette di riconoscere le carte di gioco e verificare che tutte le regole siano state applicate correttamente, per fare in modo che quando si gioca il computer riconosce le carte ed elabora la mossa che permette di battere l’avversario, verrà anche implementato un braccio robotico per poter fare in modo che il computer riesca a giocare le proprie carte a livello fisico.

Questa documentazione serve a introdurre gli utenti alla creazione del programma insieme anche al suo scopo del motivo per cui è stato creato.

Serve anche a introdurre su come è stata programmata l’intelligenza artificiale e come la macchina riesca ad auto apprendere le varie mosse che sono migliori per vincere e come fa a scegliere autonomamente quale strategia utilizzare durante la partita, la documentazione mostra anche come è stato reso possibile il riconoscimento delle carte tramite computer e come verificare che entrambi i giocatori riescano a seguire le regole impostate nel computer, fa vedere anche come la macchina riesca a gestire un braccio robotico che gestisce i movimenti scelti dalla intelligenza artificiale.

# Analisi

## Analisi del dominio

Questo progetto verrà utilizzato ogni volta che si vuole fare una partita a carte, un prodotto del genere esiste già e sono dei robot che giocano a scacchi oppure qualche altro gioco di carte, ma su internet non ho visto nessuno che ha costruito un robot con intelligenza artificiale che gioca a UNO.

## Analisi dei costi

|  |  |
| --- | --- |
| **Componenti** | **Prezzo** |
| 1 Lavoratore | 62 CHF/ ora |

Prezzo totale per 160 ore: 9920 CHF

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Creazione riconoscimento telecamera per carta in gioco |
| **Priorità** | 1 |
| **Note** | Si necessità di una buona videocamera con alta risoluzione per avere migliori prestazioni. |
| **Sotto requisiti** | |
| **001** | Si deve collegare la telecamera al computer |
| **002** | Prendere la carta e metterla davanti alla telecamera |
| **003** | Riconoscere il colore della carta |
| **004** | Riconoscere il numero della carta |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Creazione riconoscimento telecamera per carta in possesso |
| **Priorità** | 1 |
| **Note** | Si necessità di una buona videocamera con alta risoluzione per avere migliori prestazioni. |
| **Sotto requisiti** | |
| **001** | Si deve collegare la telecamera al computer (oppure la telecamera del pc) |
| **002** | Vedere il colore della carta pescata |
| **003** | Vedere il numero della carta pescata |
| **004** | Salvare la posizione della carta |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Creazione IA |
| **Priorità** | 1 |
| **Sotto requisiti** | |
| **001** | Controllare numero o colore delle carte IA siano uguale alla carta in gioco |
| **002** | Controllare se si possono fare altre combinazioni con le carte rimanenti |
| **003** | Scelta della mossa migliore tramite una percentuale di successo |
| **004** | Eseguire la mossa |

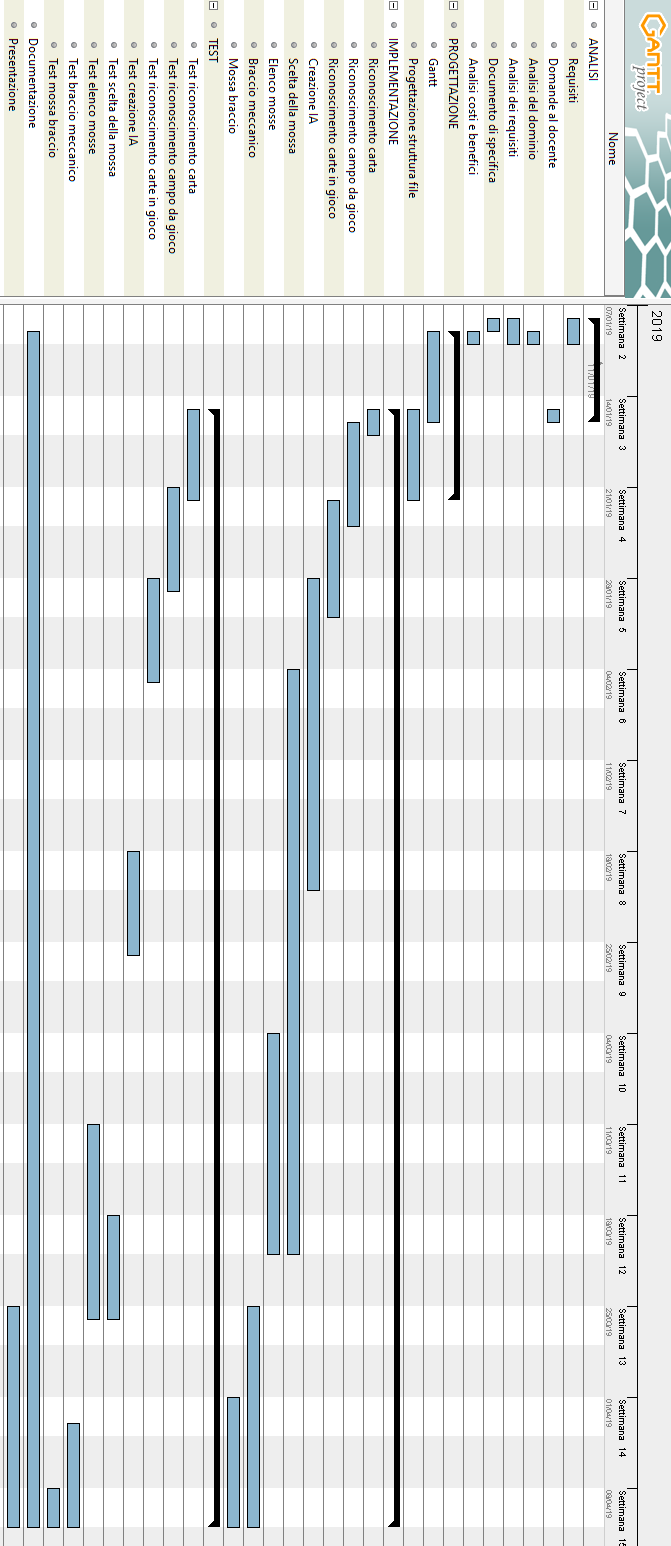
|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Movimento braccio elettronico per mossa della carta |
| **Priorità** | 1 |
| **Note** | Si necessità di un braccio con minimo 5DOF |
| **Sotto requisiti** | |
| **001** | Aspettare la scelta della mossa dell’IA |
| **002** | Avere le coordinate della posizione della carta |
| **003** | Muovere il braccio nella posizione corretta e prendere la carta |
| **004** | Trasportare carta nella sezione della carte in gioco |
| **005** | Lasciare la carta e tornare in posizione iniziale |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Movimento braccio elettronico per pescare la carta |
| **Priorità** | 1 |
| **Note** | Si necessità di un braccio con minimo 5DOF |
| **Sotto requisiti** | |
| **001** | Prendere le coordinate del mazzo di carte |
| **002** | Avvicinare il braccio elettronico al mazzo |
| **003** | Appiccicare la carta alla mano |
| **004** | Riconoscere la carta con la telecamera |
| **005** | Mettere la carta in posizione |
| **006** | Salvare le coordinate della carta nuova |

## Use case

I casi d’uso rappresentano l’interazione tra i vari attori e le funzionalità del prodotto.

## Pianificazione



Nella parte dell’implementazione ho scelto di mettere più importanza al riconoscimento delle carte e della intelligenza artificiale perché sono le parti principali del progetto, invece il braccio meccanico è secondario.

## Analisi dei mezzi

### Software

* Visual studio 2017
* EmGu CV / AForge.NET

### Hardware

* 1 PC portatile (Windows 10)
* Braccio con 5 DOF Arduino
* Raspberry 3B
* Telecamera

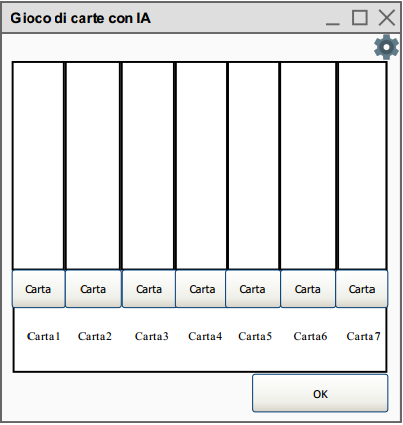
# Progettazione

## Design delle interfacce

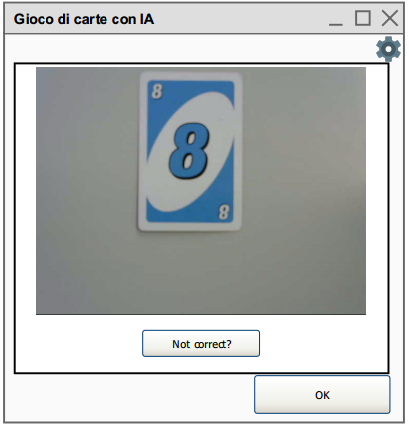
### Design scelta della telecamera



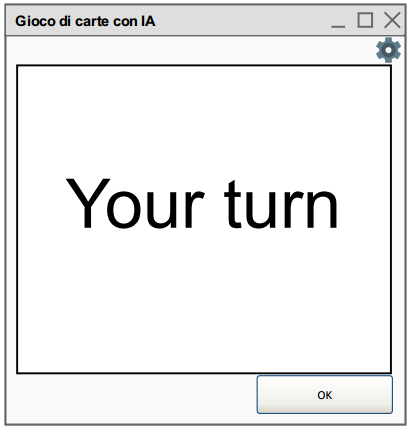
### Design della scelta delle prime carte



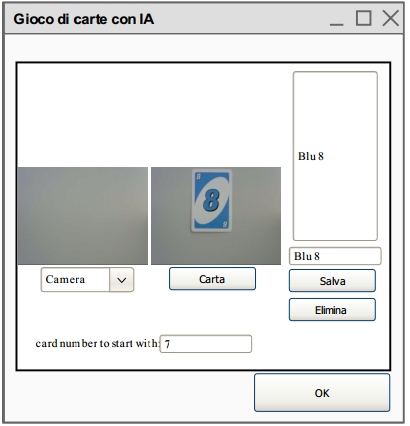
### Design della pagina di gioco dell’IA



### Design pagina di gioco del giocatore

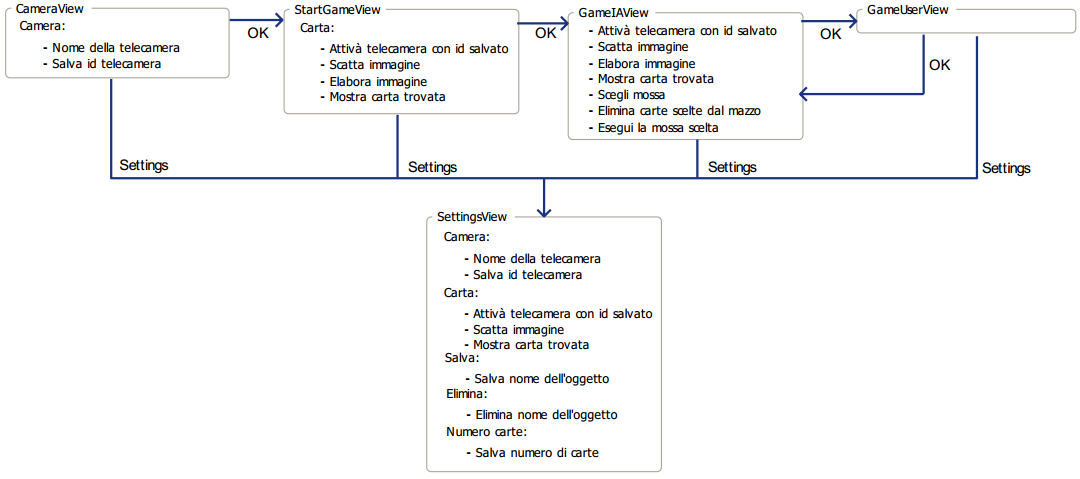


### Design pagina delle impostazioni



## Design delle cartelle

## Design delle classi



# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente/print screen di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | Primo | **Nome:** | Funzionamento python |
| **Descrizione:** | Provare a testare le funzioni e metodi di python | | |
| **Prerequisiti:** | Per far funzionare python si deve avere una piattaforma per far compilare il codice di python | | |
| **Procedura:** | 1. Provare a far stampare una scritta 2. Testare una somma di numeri o di testo 3. Creare un’array e verificare i possibili funzionamenti 4. Provare a mettere i tipi di costruttori condizionali 5. Testare un ciclo 6. Provare a creare funzioni | | |
| **Risultati attesi:** | Ogni singolo punto ha funzionato ci sono stati errori ma li sono risolti | | |

## Risultati test

|  |  |
| --- | --- |
| **Test** | **Risultati** |
| Provare a far stampare una scritta | Provato a stampare “Hello world” all’inizio non andava perché non avevamo messo le parentesi ma poi ci siamo corretti e abbiamo messo le parentesi e tutto ha funzionato |
| Testare una somma di numeri o di testo | Non ci sono stati problemi nella somma |
| Creare un’array e verificare i possibili funzionamenti | È andato tutto bene nel funzionamento degli array |
| Provare a mettere i tipi di costruttori condizionali | Non ci sono stati problemi |
| Testare un ciclo | Il ciclo ha avuto un problema perché eravamo abituati in un altro linguaggio ma poi abbiamo risolto il problema facendo una ricerca su internet |

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap 1.7) (ad esempio Gannt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc

# Bibliografia

## Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo,

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

* <https://www.codeproject.com/Articles/125478/Versatile-WebCam-C-library> 16.01.2019
* <http://hemant-srivastava.blogspot.com/2012/11/image-color-detector-in-c.html> 16.01.2019

# Allegati

* Diari di lavoro