FASE DI INCEPTION

**Attori e Obiettivi**

Attori:

* UTENTE
* PROGRAMMATORE

Obiettivi utente:

1. Esegui programma utente (Scegliendolo dalla lista)
2. Gestisci Lista programmi (Carica/ Modifica/ Elimina)
3. Configura Emulazione (Velocità/Periferica)

Obiettivi programmatore:

1. Gestione Codice Sorgente (Scrivi/Salva/Carica)
2. Compila il Codice
3. Visualizza Stato Architettura
4. Esegui programma da programmatore (Scelta modalità di esecuzione)

**Descrizione in breve dei casi d’uso:**

* **EseguiProgrammaUtente:** L’utente seleziona dalla lista il programma che vuole eseguire e visualizza a schermo il programma in esecuzione.
* **CaricaInLista:** L’utente carica un programma dal suo computer all’applicazione. Il programma verrà aggiunto alla lista dei programmi eseguibili.
* **EliminaDallaLista:** L’utente seleziona dalla lista un programma e lo rimuove dalla lista.
* **ConfiguraEmulazione:** L’utente durante o anche prima dell’esecuzione di un programma potrà cliccare su di un pulsante per visualizzare e modificare i parametri di emulazione come la Velocità dell’esecuzione, le periferiche di ingresso e di uscita, la grandezza dello schermo etc.
* **SalvaCodiceSorgente:** Permette di salvare il codice scritto nell’editor di testo in un file di testo
* **CaricaCodiceSorgente:** Permette di visualizzare e modificare il codice scritto caricandolo da file
* **CompilaCodice:** Permette di compilare il codice scritto nell’editor producendo un file macchina comprensibile per l’emulatore.
* **VisualizzaStatoArchitettura:** Il programmatore potrà cliccare sull’immagine di una specifica componente hardware emulata per verificare il comportamento della stessa durante l’esecuzione di un programma
* **EseguiProgramma:** Il programmatore potrà eseguire il programma e visualizzare a schermo l’esecuzione. L’esecuzione potrà essere fatta sia in modalità standard che istruzione per istruzione.

**Casi d’uso da specificare per primi**:

1. Gestione Codice Sorgente (Scrivi/Salva): L’utente potrà scrivere in un editor di testo il codice da compilare successivamente e salvare il codice scritto su un file.
2. Compila il Codice (Evitando il controllo sintattico e solo istruzioni logico aritmetiche): Cliccando il pulsante compila il codice scritto verrà compilato creando un file di codice macchina.
3. Esegui programma programmatore (Semplicemente mostrare a video il risultato): Il codice compilato potrà essere eseguito e si visualizzerà a schermo il risultato dell’esecuzione.

**Requisiti di Qualità**:

Ogni singolo elemento dell’architettura emulata dovrà essere gestito in maniera a sé stante in modo da rendere l’architettura modificabile e riusabile per futuri emulatori. Il sistema deve essere avere delle prestazioni che permettano l’esecuzione dei programmi come se stessero effettivamente eseguendo su un’architettura NES.

**Rischi**:

**Architettura di Alto Livello**:

Architettura basata su repository centralizzato: Ci permetterà di mantenere i dati relativi ai programmi(giochi) da eseguire sull’emulatore. Gli agenti della nostra architettura potrebbero essere l’emulatore l’assemblatore e l’editor.

L’agente “emulatore” sarà basato a sua volta da un’architettura ad interprete (Il componente emulatore al suo interno avrà anche la parte di interfaccia). L’emulatore trae il programma dal repository che è stato caricato dall’assemblatore.

**Piano per la prima Iterazione:**

Sviluppo di un set minimo di istruzioni del processore emulata e sviluppo della memoria.

Sviluppo grafico dell’editor di testo e possibilità di scrivere e salvare il codice scritto.

Sviluppo della funzione di compilazione del codice per le istruzioni aritmetiche e salvataggio su file del codice macchina creato.

* Un breve Workshop dei requisiti
* Assegnazione dei nomi ad attori, obiettivi, casi d’uso
* Scrittura in formato breve della maggior parte dei casi d’uso
* Scrittura dettagliata di un primo 10% dei requisiti
* Requisiti di qualità più importanti
* Lista dei Rischi
* Prototipi (tecnici proof-of-concept, prototipi di UI)
* Proposta dell’architettura di Alto livello
* Piano per la prima Iterazione

CASI D’USO 1° ITERAZIONE:

|  |  |
| --- | --- |
| Nome del caso d’uso | EseguiProgramma |
| Portata | User-level |
| Attore primario | Utente Programmatore  Utente |
| Stakeholders e interessi | Utente Programmatore/Utente: selezionare il programma dalla **lista programmi** ed eseguirlo, visualizzando l’output |
| Pre-condizioni | **Programma** da eseguire caricato in **lista programmi** |
| Post-condizioni | Visualizzazione dell’esecuzione del programma; |
| Scenario Principale | 1. Seleziona **programma** 2. Avvia **programma** |
| Estensioni |  |

|  |  |
| --- | --- |
| Nome del caso d’uso | VisualizzaStatoArchitettura |
| Portata | User-level |
| Attore primario | Utente Programmatore |
| Stakeholders e interessi | Utente Programmatore/Utente: verificare l’evoluzione dello **stato dell’architettura** |
| Pre-condizioni |  |
| Post-condizioni | Visualizzazione dello **stato dell’architettura** |
| Scenario Principale | 1. Seleziona **componente** |
| Estensioni | 1.1 Se selezionata **CPU**  1. Mostra tutti i **registri**  1.2 Se selezionata la **memoria**  1. Seleziona range  2. Mostra porzione di **memoria** |