

Documento di Analisi dei Requisiti Pixel Arena

Riferimento	
Versione	0.8
Data	08/01/2024
Destinatario	Carminc Gravino
Presentato da	NC10: Antonio Ferrentino, Emanuele Rosapepe, Francesco Perilli
Approvato da	

Sommario

Sommario	3
Team Composition	5
Revision History	6
1. Introduzione	7
1.1 Scopo del Sistema	8
1.2 Ambito del sistema	9
1.3 Obiettivi e criteri di successo del sistema	10
1.4 Definizioni, acronimi e abbreviazioni	11
1.5 Riferimenti	13
1.6 Panoramica	14
2. Sistema corrente	15
3. Sistema proposto	17
3.1 Panoramica	18
3.2 Requisiti funzionali	19
3.3 Requisiti non funzionali	21
3.4 Modello del Sistema	23
3.4.1 Scenari	24
3.4.2 Use-case	27
3.5 Modello ad oggetti e dinamici	32
3.5.1 Class Diagram	32
3.5.2 Sequence Diagram	33
3.5.3 Activity Diagram	37
3.6 Mock-ups	39

4. Glossario

44

Team Composition

Nome	Matricola	Contatti
Antonio Ferrentino	0512113367	a.ferrentino50@studenti.unisa.it
Emanuele Rosapepe	0512113418	e.rosapepe2@studenti.unisa.it
Francesco Perilli	0512113802	f.perilli2@studenti.unisa.it

Revision History

Data	Versione	Descrizione	Autori
3/11/2023	0.1	Prima stesura	Antonio Ferrentino
10/11/2023	0.2	Primo completamento	Antonio Ferrentino
11/11/2023	0.3	Aggiunta Use Case e Sequence Diagram	Antonio Ferrentino
12/11/2023	0.4	Ulteriori correzioni	Antonio Ferrentino
14/11/2023	0.5	Ulteriori correzioni	Francesco Perilli
15/11/2023	0.6	Aggiunta Class Diagram	Emanuele Rosapepe
16/11/2023	0.7	Aggiunta Activity Diagram e Mock-Up	Antonio Ferrentino
08/01/2024	0.8	Ulteriori correzioni e revisione	Antonio Ferrentino

1. Introduzione

1.1 Scopo del Sistema

Nel corso degli anni l'evoluzione del mondo videoludico è stata sia veloce che impressionante, spalancando le porte ad un'era d'oro per i videogiochi e per le aziende che ne trattano la loro creazione. La nostra idea è quella di portare alla luce un platform ortogonale con annessi combattimenti, il tutto rappresentato da una grafica pixelata: ciò potrebbe far sembrare il nostro gioco non molto dinamico rispetto ad altri giochi che si adattano perfettamente agli sviluppi sia hardware che software a cui abbiamo assistito in questi anni, unendo una grafica mozzafiato ad un'accuratezza nel gameplay. Ma per le dinamiche e per le sfide che il nostro gioco "Pixel Arena" ha intenzione di proporre, metterà sicuramente alla prova le abilità del videogiocatore, non negandogli anche il giusto divertimento.

1.2 Ambito del sistema

L'obiettivo del progetto è fornire un gioco con varie specifiche e peculiarità, tra cui:

- Presenza di una mappa di gioco, con la quale interagire e completare gli stage, e con la presenza di collisioni, soprattutto per definire i limiti della stessa;
- Dinamica di combattimento con nemici "intelligenti";
- Possibilità data al giocatore di curarsi tramite aree specifiche presenti nella mappa di gioco;
- Permettere al giocatore di poter salvare e rigiocare alla stessa partita avviata in qualsiasi momento, evitando di interrompere la propria esperienza di gioco;
- Possibilità di aumentare le caratteristiche del personaggio con la progressione del gioco.

1.3 Obiettivi e criteri di successo del sistema

Obiettivi principali del sistema:

- incentivare il giocatore a restare sempre attivo e a visionare tutta la mappa di gioco;
- Incentivare il giocatore a potenziare il proprio personaggio per riuscire a proseguire nel gioco;
- rendere i nemici dei degni avversari che infastidiscano il giocatore;
- Permettere al giocatore di utilizzare diverse strategie per affrontare al meglio il livello;

Ecco i criteri per la buona riuscita del progetto:

- interfaccia user friendly;
- Rispetto scadenze delle scadenze intermedie/di fine progetto **definite nello statement of work**
- Budget/Effort non superiore a $50 \cdot n$ ore dove n sono i membri del team
- Uso di sistemi di versioning - GitHub in particolare
- Utilizzo di un sistema di versioning, dove tutti i membri del team forniscono il loro contributo
- Utilizzo di tool di per la suddivisione dei task e attività (Trello o similare)
- Utilizzo di tool di comunicazione tracciabile (Slack)

1.4 Definizioni, acronimi e abbreviazioni

Acronimi	
SOW	Statement of Work
RAD	Requirement Analysis Document
UI	User Interface
AD	Activity Diagram
SC	Scenario
UC	Use Case
RF	Requisito funzionale
RNF	Requisito non funzionale
CD	Class Diagram
SD	Sequence Diagram
MU	Mock-up

Abbreviazioni	
PS	Punti Salute
HP	Healt Point
ATK	Attacco
SPD	Speed

Definizioni	
Mock-up	Realizzazione a scopo illustrativo di un sistema o un oggetto
User-friendly	In informatica, di facile utilizzo anche per chi non è esperto

Definizioni	
Use Case	È una tecnica usata nei processi di ingegneria del software per effettuare in maniera esaustiva e non ambigua, la raccolta dei requisiti al fine di produrre un software di qualità.
Class Diagram	Diagramma che consentono di descrivere tipi di entità con le loro caratteristiche ed eventuali relazioni tra tipi
Activity Diagram	Diagramma che permette di descrivere un processo attraverso dei grafi in cui i nodi rappresentano le attività e gli archi l'ordine con cui queste vengono eseguite
Sequence Diagram	Diagramma utilizzato per descrivere uno scenario. Uno scenario è una determinata sequenza di azioni in cui tutte le scelte sono state già effettuate.

1.5 Riferimenti

- Libro “Object-Oriented Software Engineering: Conquering Complex and Changing Systems”, terza edizione, Bernd Bruegge & Allen Dutoit, 2014.
Autori: Bernd Bruegge, Allen H. Dutoit
- [Statement of Work](#)

1.6 Panoramica

Il presente documento è strutturato in tre macroaree, quali:

- **Introduzione:** sezione del documento dove viene illustrato lo scopo del sistema e l'ambito in cui il sistema deve operare. Sono presenti anche una lista di definizioni, acronimi ed abbreviazioni che verranno utilizzate all'interno di questo documento.
- **Sistema corrente:** sezione del documento che illustra il sistema attualmente funzionante, ma visto che il progetto di cui stiamo parlando è del tutto nuovo, si vuole illustrare le funzionalità di altri sistemi molto simili.. Questo ci permette di capire al meglio dove e come agire al meglio per il sistema proposto.
- **Sistema proposto:** Questa area del documento presenta delle sottosezioni all'interno delle quali è possibile leggere una panoramica del sistema proposto, i requisiti funzionali e non funzionali, il modello del sistema, il modello a oggetti e il modello dinamico.

2. Sistema corrente

Ripercorriamo le principali attività, caratteristiche e servizi che vari tipi di giochi, soprattutto quelli platform, tendono ad includere.

Menù principale

Il gioco una volta aperto mostra all'utente un menù principale con una grafica che rappresenta l'anima del gioco e inizia già a far capire cosa ci si può aspettare dallo stesso. Ovviamente, sono presenti i soliti bottoni di " Inizio Partita", "Carica Salvataggio" e "Chiudi Gioco", oltre che al bottone delle "Impostazioni" di gioco.

Nel gioco

L'utente può muoversi in tutte e 4 le direzioni, saltare, scavalcare, attaccare e fare tantissime altre azioni secondo le varie logiche di gioco differenti tra loro, il tutto attraverso i comandi selezionati dalle "Impostazioni", attaccare, e fare tantissime altre azioni secondo le varie logiche di gioco differenti tra loro. Noi abbiamo preso in esame le dinamiche di gioco riguardanti :

- il nascondersi dai nemici, che di solito viene effettuato attraverso un allontanamento da questi ultimi;
- la cura da parte del personaggio dei suoi HP, attraverso oggetti o luoghi situati nella mappa;
- il potenziamento del personaggio stesso, di solito effettuata attraverso raccolta di oggetti situati all'interno della mappa.

Inoltre, in tutti i giochi di solito c'è la presenza di uno o più boss.

Salvataggio

Le funzioni di salvataggio si differenziano in base al tipo di gioco: se parliamo di giochi online, i salvataggi vengono gestiti attraverso un database su cui vengono salvati tutti i dati del giocatore; al contrario, i giochi offline usano un file di testo su cui vengono segnate tutte le caratteristiche e i dati del giocatore.

3. Sistema proposto

3.1 Panoramica

Dall'analisi del sistema corrente vista precedentemente, il nostro sistema si preoccupa di implementare tutte le caratteristiche descritte, ovviamente in base alla nostra idea di videogioco su cui stiamo basando il nostro progetto:

- Menù principale;
- Azioni del personaggio;
- Interazioni del personaggio con la mappa e l'ambiente di gioco;
- Salvataggio della partita.

L'unico attore del nostro sistema, visto che è un gioco single player, sarà un unico utente che chiameremo per comodità "Giocatore".

3.2 Requisiti funzionali

RF_MENU

ID Requisito	Descrizione
RF_MENU1	Selezionare il proprio personaggio.
RF_MENU2	Possibilità di caricare la partita.

RF_GAME

ID Requisito	Descrizione
RF_GAME1	Muovere il personaggio principale in 4 direzioni.
RF_GAME2	Combattere contro i nemici.
RF_GAME3	Nascondersi dai nemici "intelligenti".
RF_GAME4	Postazioni per recuperare i punti salute.
RF_GAME5	Potenziare varie caratteristiche del proprio personaggio attraverso gemme (PS, ATK,SPD).
RF_GAME 6	É possibile mettere in pausa il gioco

RF_GAME7	All'esaurimento dei punti salute del personaggio l'utente dovrà ricominciare il livello dall'inizio.
RF_GAME8	Alla fine del livello il giocatore dovrà affrontare un Boss.
RF_GAME9	Raccolta di gemme che appaiono sulla mappa ad ogni livello.

RF_SAVE

ID Requisito	Descrizione
RF_SAVE1	Un file di testo che ci permette di salvare punteggi e caratteristiche del personaggio usate nella partita in corso.

3.3 Requisiti non funzionali

RNF_USE

Codice	Nome	Descrizione	Priorità
RNF_USE1	Interfaccia	Ad ogni iterazione che l'utente avrà col sistema, l'utente riceverà un feedback chiaro e preciso del cambiamento avvenuto.	Alta
RNF_USE2	User Friendly	Il sistema deve essere di facile utilizzo per tutti gli utenti .	Media
RNF_USE3	Immersione	Creare un'esperienza visiva e sonora che immerga completamente i giocatori nell'esperienza del gioco	Alta

RNF_PRES

Codice	Nome	Descrizione	Priorità
RNF_PRES1	Tempo di risposta	Il tempo di risposta alla pressione dei comandi deve essere minore di 1 secondo.	Alta

RNF_AFF

Codice	Nome	Descrizione	Priorità
RNF_AFF1	Continuità	Il sistema non deve subire rallentamenti e operare in modo accettabile anche in situazioni non previste e deve essere in grado di gestirle senza degenerare nel crash/freeze dell'applicazione.	Alta
RNF_AFF2	Sicurezza	Proteggere i dati degli utenti e dei salvataggi	Bassa

RNF_SUPP

Codice	Nome	Descrizione	Priorità
RNF_SUPP1	Manutenibilità	Il sistema sarà facilmente manutenibile grazie alla sua elevata modularità.	Alta
RNF_SUPP2	Aggiornamento	Il sistema permette la presenza di aggiornamenti, in previsione di future aggiunte di funzionalità.	Media

3.4 Modello del Sistema

3.4.1 Scenari

NOME SCENARIO	SC.1: Movimento Personaggio
PARTECIPANTI	Utente: Pippo
UTENTE	SISTEMA

1.
Pippo preme il tasto W, D, A o S

2.
Il sistema muove il personaggio verso l'alto, destra, sinistra o basso

NOME SCENARIO	SC.2: Attacco Personaggio
PARTECIPANTI	Utente: Pippo
UTENTE	SISTEMA

1.
Pippo muove il personaggio vicino a un nemico

2.
Il sistema lo rileva e fa reagire i nemici facendogli attaccare il personaggio

3.
Pippo cerca di far evitare gli attacchi dei nemici al personaggio e a sua volta li attacca premendo il tasto K

4.
Il sistema genera gli attacchi e se questi vanno a segno diminuiscono i punti vita dei nemici

NOME SCENARIO	SC.3: SalvaPartitaeCont
PARTECIPANTI	Utente: Pippo
UTENTE	SISTEMA
1. Pippo finisce il livello	2. Il sistema presenta un pannello con due bottoni: "Salva partita e continua" o "Salva partita ed esci"
3. Pippo preme il bottone "Salva partita e continua" alla fine del livello	4. Il sistema effettua il salvataggio e porta il giocatore ad un nuovo livello

NOME SCENARIO	SC.4: RecuperoPuntiSalute
PARTECIPANTI	Utente: Pippo
UTENTE	SISTEMA
1. Pippo posiziona il personaggio su una postazione adatta per il recupero punti Salute	2. Il Sistema rigenera gradualmente i punti Salute del personaggio

NOME SCENARIO	SC.5: PotenzaPersonaggio
PARTECIPANTI	Utente: Pippo
UTENTE	SISTEMA
1. Pippo finisce il livello	2. Il Sistema presenta una schermata con le possibili modifiche attuabili
3. Pippo clicca sul potenziamento da applicare al personaggio	4. Il Sistema applica la modifica scelta da Pippo

NOME SCENARIO	SC.6: Generazione&RaccoltaGemme
PARTECIPANTI	Utente: Pippo
UTENTE	SISTEMA
1. Pippo inizia a giocare	2. Il Sistema dopo un tempo prestabilito, genera delle gemme in posizioni casuali sulla mappa
3. Pippo muove il giocatore sulla gemma	4. Il Sistema aggiunge la gemma al conteggio totale delle gemme collezionate dal personaggio.

3.4.2 Use-case

Identificativo UC_1 RIGENVITA	Recupero dei Punti Salute del personaggio principale	Data		03/11/20 23
		Vers.	0.1	
		Autore	Antonio Ferrentino	
Descrizione	Recupero dei Punti Salute del personaggio principale tramite postazioni specifiche presenti nello spazio di gioco			
Attore Principale	Giocatore			
Attori secondari	NA			
Entry Condition	Il giocatore si posiziona su una postazione			
Exit condition On success	Il giocatore esce dalla postazione AND Il sistema ha rigenerato punti vita proporzionali al tempo nel quale l'utente è stato sulla postazione			
Exit condition On failure	Il giocatore esce dalla postazione AND Il sistema non ha rigenerato punti vita proporzionali al tempo nel quale l'utente è stato sulla postazione			
Rilevanza/User Priority	Elevata			
Frequenza stimata	NA			
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Giocatore:	Con il personaggio principale si posiziona su una postazione a scelta nella mappa di gioco		
2	Sistema:	Recepisce l'input da parte del giocatore, attiva la postazione ed inizia a ricaricare i punti salute dello stesso in proporzione al tempo di sosta sulla postazione scelta		

3	Giocatore:	Quando vuole, l'utente può uscire dalla postazione
4	Sistema:	Il sistema smette di rigenerare i punti salute del personaggio principale, disattivando la postazione
Scenario/Flusso di eventi Alternativo:		
2.a1	Sistema:	Recepisce l'input da parte del giocatore ma quest'ultimo ha già il massimo dei Punti Salute, quindi il sistema non attiva la postazione
Scenario/Flusso di eventi di ERRORE:		
2.1	Sistema:	Recepisce l'input da parte del personaggio principale, attiva la postazione ma non inizia a ricaricare i punti salute
2.2	Sistema:	L'utente è già al massimo della vita, ma il sistema attiva la postazione per la ricarica dei Punti Salute

Identificativo UC_2ATTACCO	Mossa di attacco da parte del personaggio principale	Data		03/11/2023
		Vers.	0.1	
		Autore	Emanuele Rosapepe	
Descrizione	L'utente premendo il tasto “K” attiverà la funzione di attacco del personaggio, che può essere indirizzata verso i nemici per danneggiarli.			
Attore Principale	Giocatore			
Attori secondari	NA			

Entry Condition		Il giocatore preme il tasto "K"
Exit condition On success		Il giocatore rilascia il tasto "K" AND Il sistema ha attivato l'attacco del personaggio principale
Exit condition On failure		Il giocatore rilascia il tasto "K" AND Il sistema non ha attivato l'attacco del personaggio principale
Rilevanza/User Priority		Elevata
Frequenza stimata		NA
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Giocatore:	Preme il tasto "K" per attaccare
2	Sistema:	Recepisce l'input da parte del giocatore, attivando l'attacco del personaggio principale e diminuendo i punti vita del nemico attaccato se colpito
3	Giocatore:	Il giocatore continua ad attaccare o si allontana, tornando quindi al punto 1 se decide di attaccare nuovamente
4	Sistema:	Fa contrattaccare i nemici
Scenario/Flusso di eventi Alternativo:		
2.a1	Sistema:	Recepisce l'input da parte del giocatore, attivando l'attacco del personaggio principale, se questo non va a segno non diminuiscono i punti vita del nemico
4.a2	Sistema:	Fa inseguire il personaggio principale dai nemici
Scenario/Flusso di eventi di ERRORE:		
2.1	Sistema:	Recepisce l'input da parte del personaggio principale ma non viene attivato l'attacco

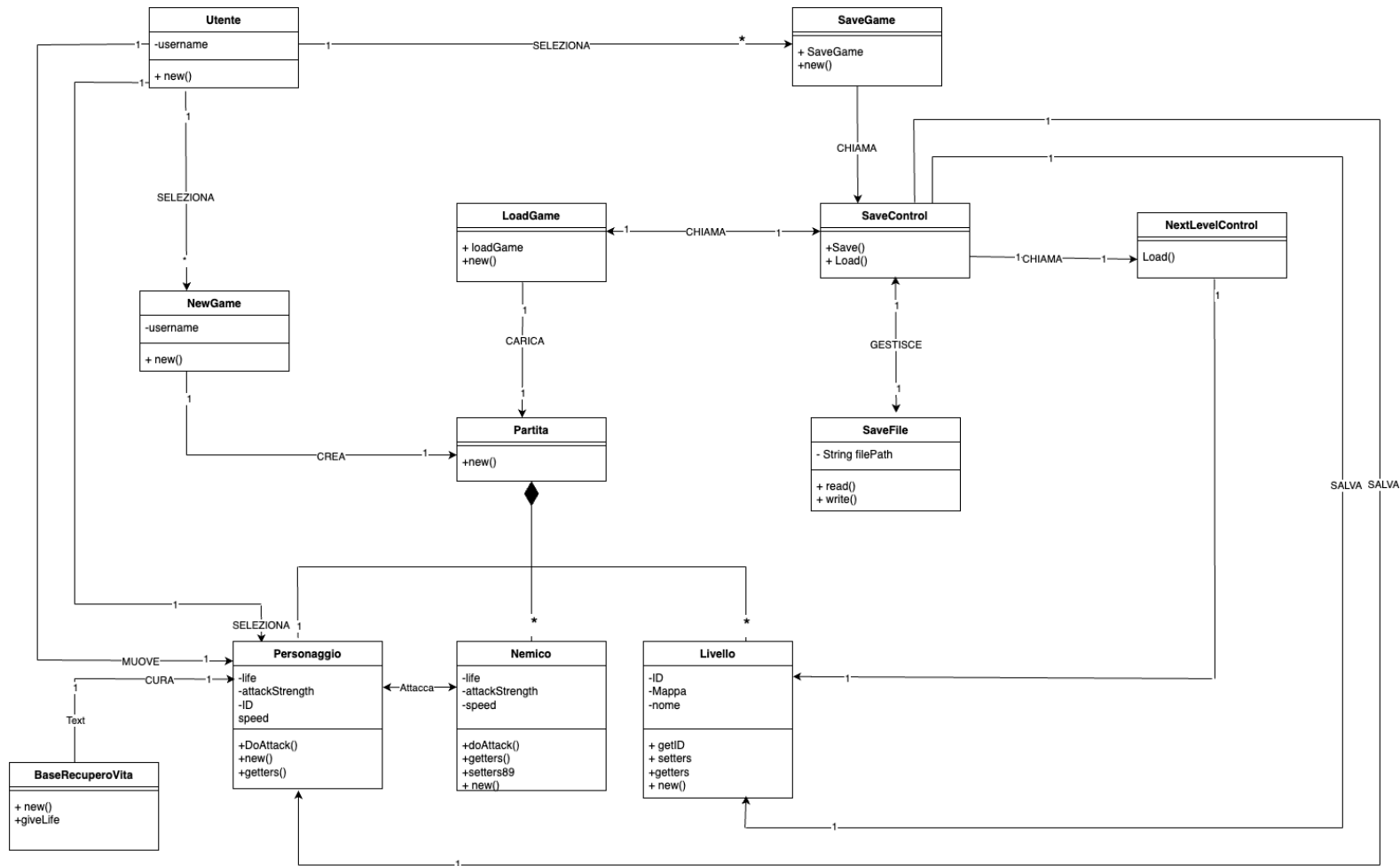
2.2	Sistema:	L'attacco va a segno ma non diminuiscono i punti vita del nemico attaccato
------------	-----------------	--

Identificativo UC_3SALVATAGGIO	Salvataggio dei progressi a fine partita	Data		03/11/2023
		Vers.	0.1	
		Autore	Francesco Perilli	
Descrizione	Alla fine del livello comparirà una schermata e il giocatore sceglierà se salvare e uscire o salvare e continuare			
Attore Principale	Giocatore			
Attori secondari	NA			
Entry Condition	Il giocatore ha finito il livello e procede verso quello successivo			
Exit condition On success	Il giocatore salva ed esce oppure salva e continua al livello successivo AND Il sistema ha salvato le statistiche attuali del personaggio e l'identificativo del livello appena superato			
Exit condition On failure	Il giocatore sceglie l'opzione di salvataggio AND Il sistema non memorizza le statistiche del personaggio e l'identificativo del livello			
Rilevanza/User Priority	Elevata			
Frequenza stimata	NA			

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Giocatore:	Il giocatore ha ucciso tutti i nemici e procede al livello successivo
2	Sistema:	Il sistema fa comparire un pop-up con le opzioni di salvataggio prima di iniziare il nuovo livello
3	Giocatore:	Il giocatore effettua la scelta di salvare e continuare al livello successivo
4	Sistema:	Il sistema memorizza statistiche attuali del personaggio e identificativo del livello appena superato e fa avanzare il giocatore al livello successivo
Scenario/Flusso di eventi Alternativo:		
2.a1	Giocatore:	Seleziona l'opzione di salvare ed uscire
4.a2	Sistema:	Il sistema memorizza statistiche attuali del personaggio e identificativo del livello appena superato e torna alla schermata del menu principale
Scenario/Flusso di eventi di ERRORE:		
2.1	Sistema:	Non riesce a memorizzare il salvataggio, fa comparire un pop-up con un messaggio di errore e riporta l'utente alla scelta del salvataggio

3.5 Modello ad oggetti e dinamici

3.5.1 Class Diagram



3.5.2 Sequence Diagram

SD_ Save OGGETTI

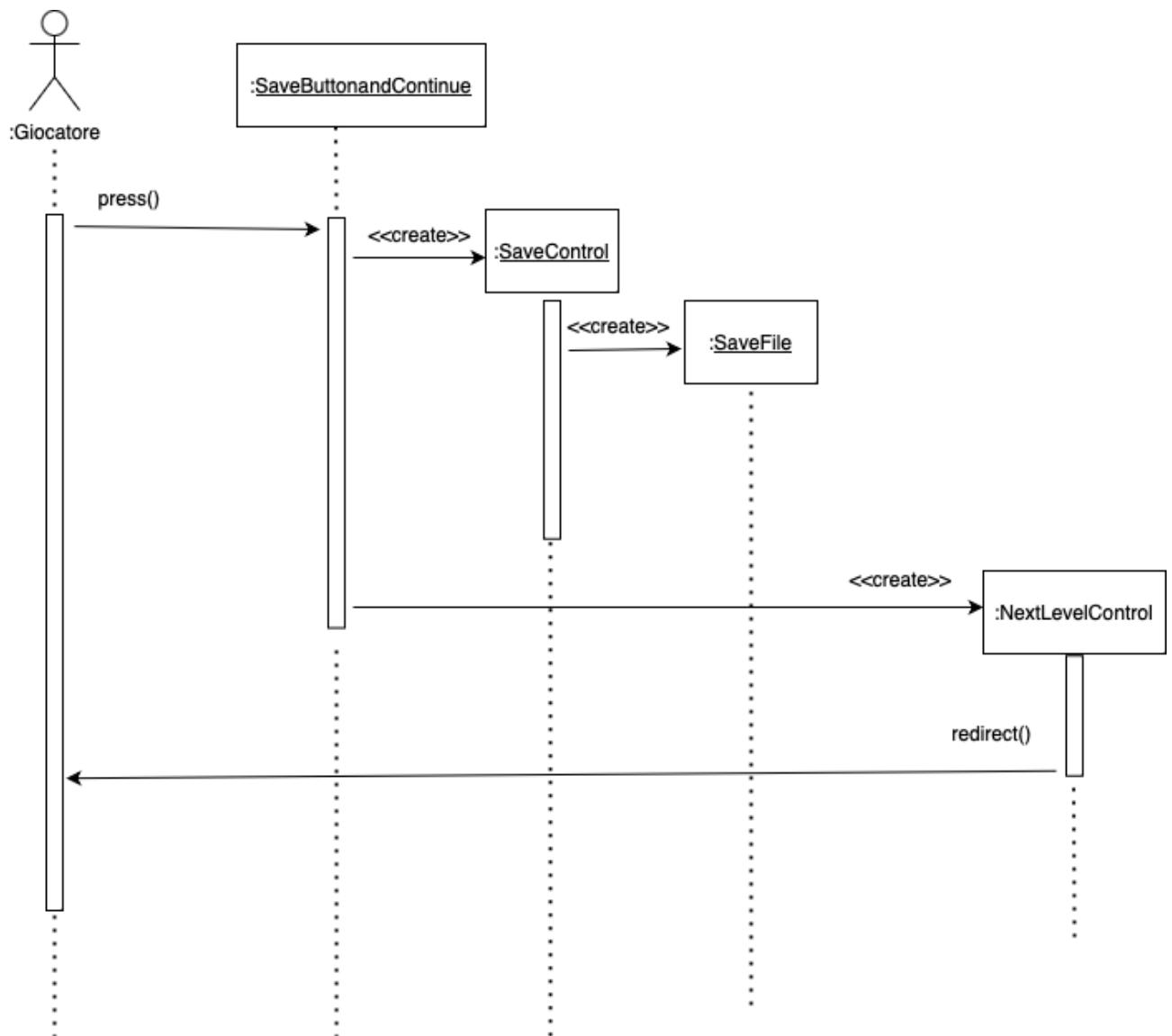
Nome	Tipologia	Descrizione
SaveButtonAndContinue	Boundary	Bottone per salvare e passare al livello successivo
SaveButtonAndExit	Boundary	Bottone per salvare ed uscire dal gioco
SaveControl	Control	Permette al sistema di salvare le statistiche del giocatore in un file di testo
SaveFile	Entity	File di testo che contiene tutte le statistiche del Main Character
NextLevelControl	Control	Permette al sistema di caricare un nuovo livello

Nota: c'è solo la presenza di una tabella per i due SD del salvataggio perché nell'insieme i due SD rappresentano una stessa azione, ma con diverse finalità.

SD_Cura OGGETTI

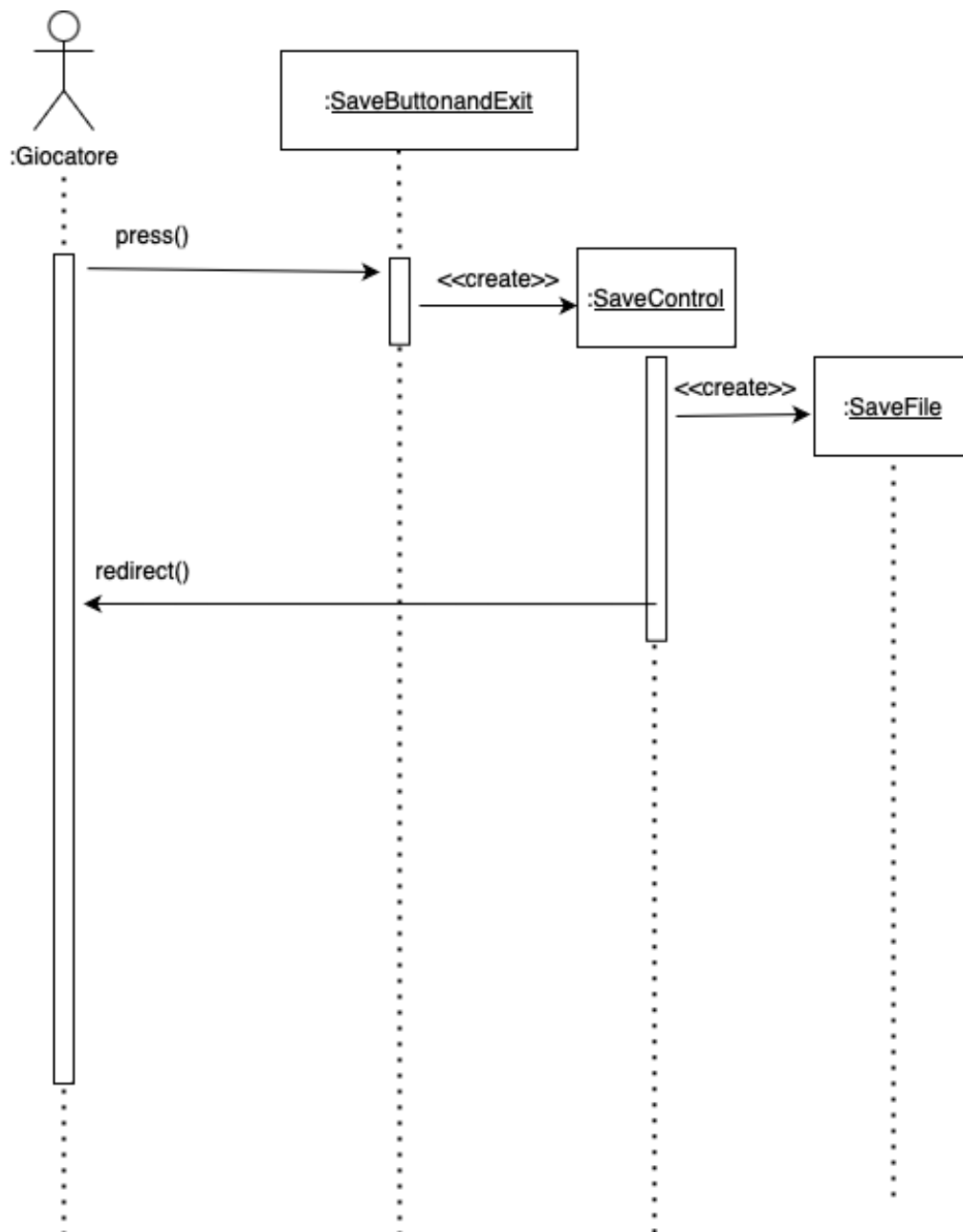
Nome	Tipologia	Descrizione
InputListener	Boundary	Registra il tasto premuto dal Giocatore
InputManager	Control	Associa il tasto passato dall'InputListener ad un azione
Personaggio	Entity	Permette al giocatore di muoversi nella mappa
AreaMappa	Boundary	Specifica area di mappa dove il giocatore può muoversi
HPHealer	Control	Permette al sistema di rigenerare i punti vita del Personaggio

SD_SaveAndContinue



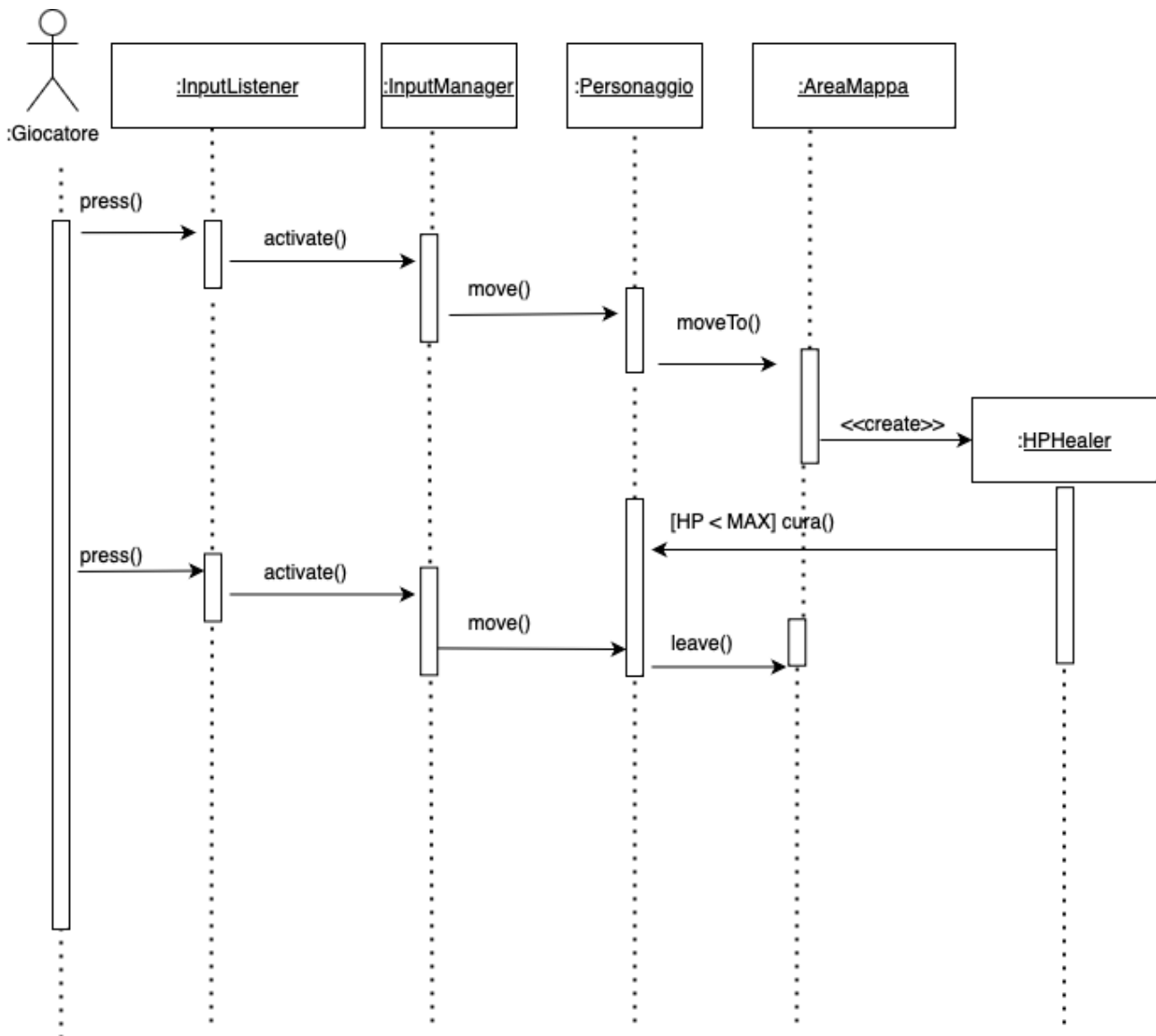
Fa riferimento allo [UC_3SALVATAGGIO](#): specificatamente, si riferisce al flusso principale.

SD_SaveAndExit



Fa riferimento allo [UC_3SALVATAGGIO](#): specificatamente, si riferisce al flusso alternativo.

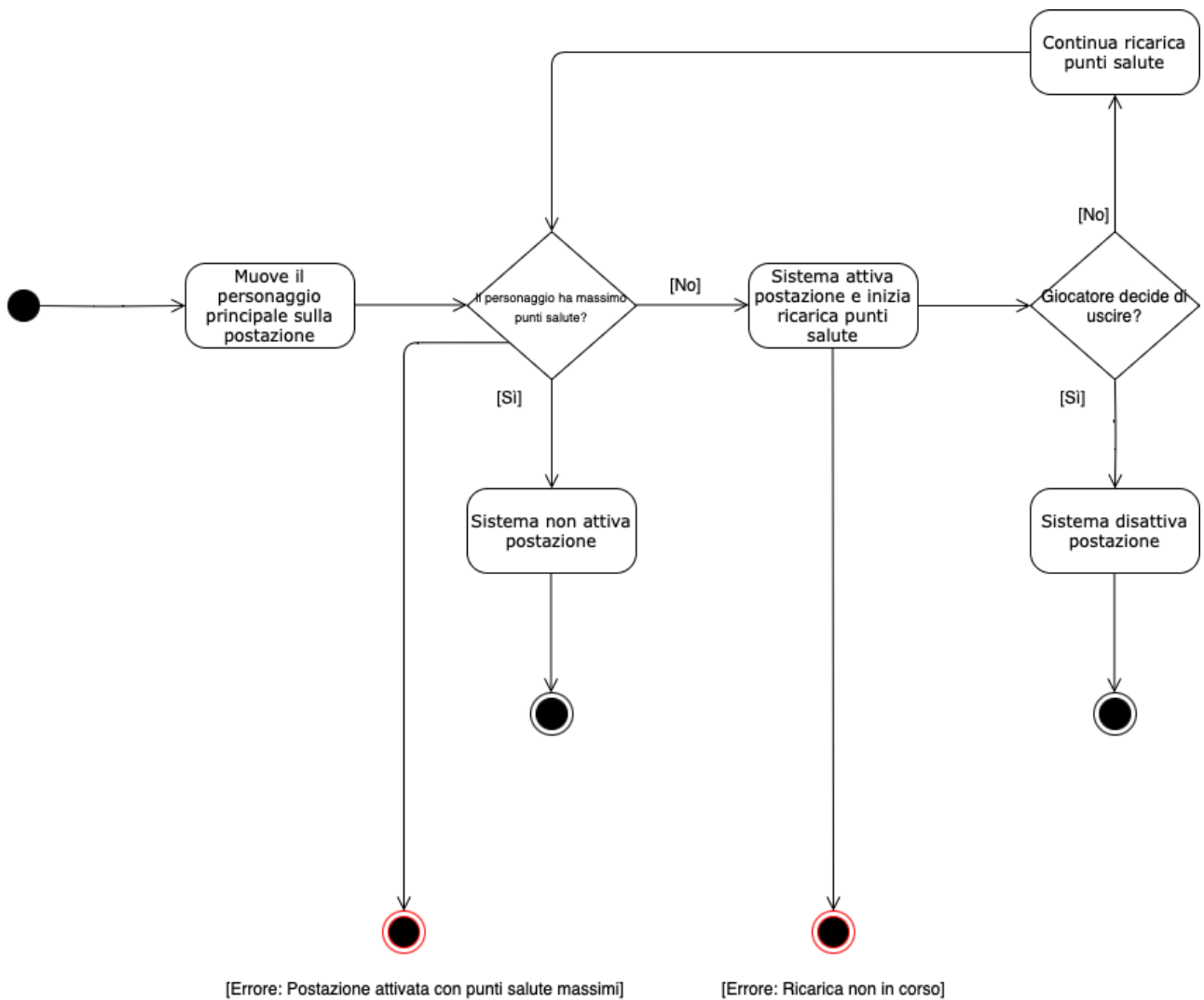
SD_Cura



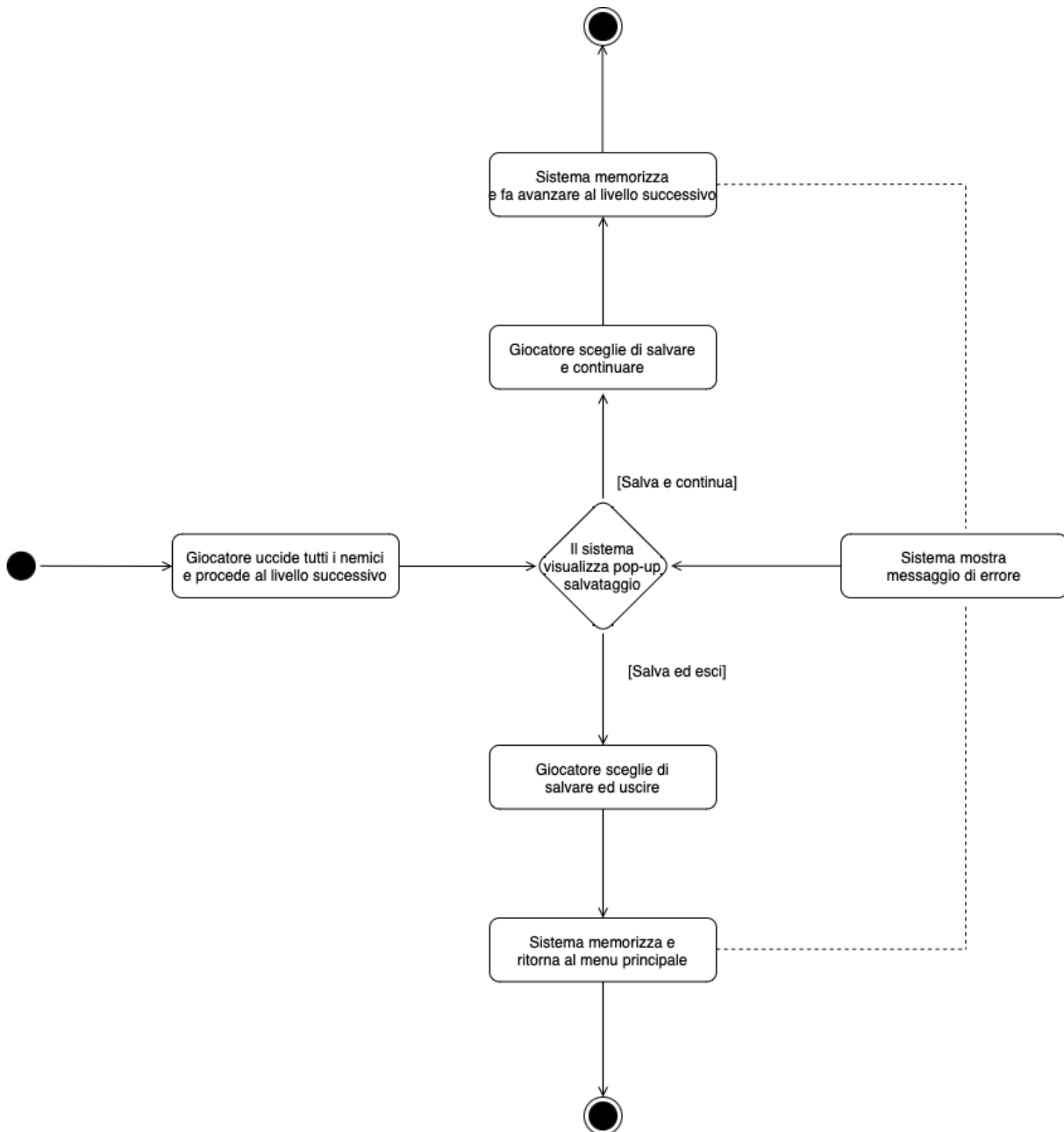
Fa riferimento allo [UC_1RIGENVITA](#).

3.5.3 Activity Diagram

AD_Cura



AD_Salva



3.6 Mock-ups

MU_1 Personaggio



MU_2 Base Ricarica Vita



MU_3 Menù principale

Pixel Arena


Load Game

New Game

Exit

MU_4 Scelta personaggio

Select a character



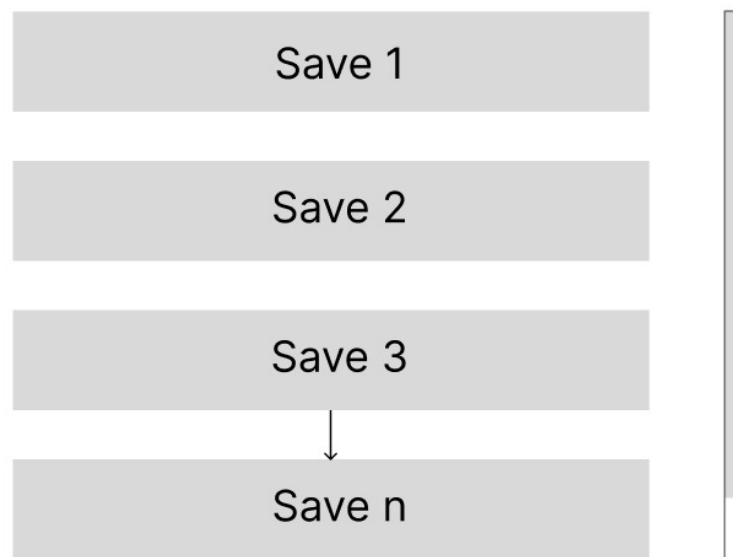
Char 1

Char 2

Char 3

MU_5 Scelta salvataggio

Choose your save



4. Glossario

Termine	Definizione
Platform	Il videogioco a piattaforme è un sottogenere dei videogiochi d'azione dove la meccanica di gioco implica principalmente l'attraversamento di livelli costituiti da piattaforme, spesso disposte su più piani.
Boss	Un boss è, nel mondo dei videogiochi, un avversario particolarmente difficile da sconfiggere, e di solito rappresenta l'avversario finale per finire il livello o l'intero gioco.
Giocatore	Il giocatore, al femminile giocatrice, è il partecipante a un gioco, come dedizione momentanea o come praticante assiduo. Nell'ambito dei videogiochi si usa anche il termine videogiocatore.
Pixel	Con il termine pixel, o px, si fa riferimento all'unità minima di cui si compone un'immagine digitale.
Game design	Il game design è il processo attraverso cui si definisce la struttura del gioco, con particolare riferimento ai supporti digitali e ai videogiochi. Il game design è indipendente dall'effettiva realizzazione del gioco, è un processo di ideazione e progettazione.
Requisito Funzionale	Un requisito funzionale è una dichiarazione o una specifica che descrive le funzioni o le azioni che un sistema software.
Requisito non funzionale	Un requisito non funzionale è una specifica che descrive le qualità o le caratteristiche del sistema software che vanno oltre le sue funzionalità di base.
Collisioni	In ambito videoludico si dice che un'entità implementa le collisioni quando questa non può essere attraversata da un'altra entità. Ad esempio, una parete di un edificio.