

Università degli studi di Modena e Reggio Emilia

Facoltà di Ingegneria "Enzo Ferrari"

Corso di laurea in Ingegneria Informatica

A.A. 2020/2021



Sommario

Specifica dei Requisiti del Software (SRS)	2
1. Introduzione	2
1.1. Obiettivo	2
1.2. Campo d'applicazione	2
1.3. Definizioni, acronimi e abbreviazioni	2
1.4. Riferimenti alla documentazione	2
1.5. Struttura del documento SRS	2
2. Descrizione generale	3
2.1. Inquadramento	3
2.1.1. Interfaccia Sistema/Utente	3
2.1.2. Interfaccia Hardware	3
2.1.3. Interfaccia Software	3
2.2. Funzioni principali software	3
2.3. Utenza	3
2.4. Vincoli generali	3
2.5. Ipotesi di partenza	3
3. Specifica dei requisiti	4
3.1. Requisiti di interfaccia esterna	4
3.2. Requisiti funzionali	4
3.2.1. Caratteristiche del sistema: Personaggio	4
3.2.2. Caratteristiche del sistema: Nemici	6
3.2.3. Caratteristiche del sistema: Partita	7
3.3. Vincoli di progettazione	8
3.4. Requisiti di prestazione	8
3.5. Altri requisiti	8
3.5.1. Requisiti logici del database	8
Unified Modelling Language (UML)	9
4. Class Diagram	9
4.1. Scenario di gioco	9
4.2. Design Patterns	10
4.2.1. State Pattern	10
4.2.2. Strategy Pattern	10
5. Use Case Diagram	11
6. Activity Diagram	12
7. State Diagrams	13
7.1. Scenario di gioco	13
7.2. Gestione delle animazioni	13
8. Sequence Diagram	14

Specifica dei Requisiti del Software (SRS)

1. Introduzione

La presente sezione ha lo scopo di rappresentare la visione globale dell'intero documento di Specifica dei Requisiti. La struttura del documento è basata sullo standard ANSI/IEEE 830 noto come SRS (Software Requirements Specifications).

1.1. Obiettivo

Questo documento ha lo scopo di spiegare, nel modo più preciso, non ambiguo e comprensibile, in modo che siano chiari sia al committente che ai progettisti, le funzioni del software per la realizzazione di un videogioco in 2D Scrolling Background.

Va tenuto presente che l'approccio seguito per lo sviluppo del software è di tipo prototipale, quindi nuovi requisiti potranno essere introdotti in seguito, oltre a tutti i vincoli che fino a questo momento non sono ancora stati individuati.

1.2. Campo d'applicazione

Il corso di laurea in ingegneria informatica prevede il corso di "Programmazione ad oggetti", il quale richiede di sviluppare un'applicazione, a scopo puramente didattico, da presentare durante la prova d'esame.

1.3. Definizioni, acronimi e abbreviazioni

In questo paragrafo sarà spiegato il significato di alcuni termini, degli acronimi e di eventuali abbreviazioni.

Videogioco 2D	Videogioco sviluppato attraverso tecniche di visione bidimensionale Scrolling Background.
Scrolling background	L'azione di gioco è ripresa da una vista laterale; la traslazione dello sfondo può avvenire sia orizzontalmente che verticalmente.
Drop	Termine videoludico che viene usato per riferirsi a oggetti lasciati cadere da un nemico dopo la sua uccisione.
Spawn point	Lo spawn point, ovvero " <i>punto di spawn</i> ", è il luogo in cui appaiono le entità.

1.4. Riferimenti alla documentazione

I requisiti illustrati sono stati dedotti e ricavati unicamente dal gruppo di lavoro. Durante lo sviluppo del software è stata rilasciata una versione beta in modo tale da testare, l'ambiente di gioco, e ricevere in tal modo feedback da parte della media utenza.

1.5. Struttura del documento SRS

Questo documento prosegue con una descrizione generale del software: in particolare verranno indicati vincoli generali, premesse e ipotesi di partenza, successivamente verranno elencati e descritti i requisiti dell'applicativo secondo un'organizzazione *by feature*, basata sulle sei dimensioni dei requisiti: una prima che riguarda le interfacce esterne del software, una seconda dove sono indicati i requisiti funzionali ed infine una terza parte sui requisiti non funzionali.

2. Descrizione generale

2.1. Inquadramento

“*Reckless Frank*” è un software di gioco, di carattere puramente ludico, sviluppato sfruttando la libreria Slick2D. Il gioco si caratterizza come un semplice sparatutto in scrolling 2D, caratterizzato da:

2.1.1. *Interfaccia Sistema/Utente*

Non è necessario avere alcun tipo di conoscenza specifica o esperienza affinché sia possibile utilizzare il programma

2.1.2. *Interfaccia Hardware*

Per garantire un approccio semplice al gioco, non è previsto l’uso di particolari controller al di fuori di quelli normalmente disponibili alla media utenza, ovvero mouse e tastiera.

2.1.3. *Interfaccia Software*

Il programma deve essere eseguibile su più piattaforme, come Windows, Mac, Linux.

2.2. Funzioni principali software

Le macro funzionalità di “*Reckless Frank*” sono:

- Creazione di una nuova partita
- Creazione di una classifica locale dei punteggi dei giocatori
- Terminazione di una partita in corso
- Riavvio di una partita in corso

Queste sono solo alcune delle funzioni principali che verranno meglio specificate, assieme ad altre, nel paragrafo riguardo i requisiti funzionali.

2.3. Utenza

L’utenza non deve possedere particolari conoscenze in ambito informatico.

Ogni utente che avvia l’applicazione, lo fa in veste di *Giocatore* e ha potere d’accesso limitato poiché sarà in grado di avviare una partita o visualizzare i dati di partite già giocate.

2.4. Vincoli generali

L’applicazione non richiede vincoli e non richiede l’accesso a internet o collegamenti con altri sistemi.

2.5. Ipotesi di partenza

Non vi sono particolari meccanismi che devono essere conosciuti dall’utente per poter usare correttamente il software.

3. Specifica dei requisiti

3.1. Requisiti di interfaccia esterna

- **Interfaccia utente:** l'interfaccia utente è realizzata in Java ed è ottimizzata per una risoluzione dello schermo di 1280x720.
- **Interfaccia Hardware:** la dotazione minima del computer affinché il programma venga eseguito senza problemi:
 - Pentium 4 (o equivalente)
 - 512 Mb di memoria RAM
- **Interfaccia Software:** L'applicazione deve girare su un computer su cui è installata una Java Virtual Machine (JVM), almeno aggiornata alla Standard Edition 6.

3.2. Requisiti funzionali

In questo paragrafo sono spiegati tutti i requisiti funzionali, per ogni requisito saranno indicate l'importanza e la stabilità utilizzando una scala da 1 a 5, dove 1 è il minimo e 5 è il massimo.

3.2.1. Caratteristiche del sistema: Personaggio

Numero del requisito	RF001
Descrizione	Frank - Movimento
Input	Tastiera
Processo	Il Frank deve essere in grado di muoversi all'interno della scena di gioco, sia a destra che sinistra, entro i limiti dello schermo. <ul style="list-style-type: none">• Freccia destra: movimento verso destra.• Freccia sinistra: movimento verso sinistra.
Output	Frank si muove
Importanza	5
Stabilità	5

Numero del requisito	RF002
Descrizione	Frank – Salto
Input	Tastiera
Processo	Frank deve essere in grado di saltare, abbastanza in alto da poter scavalcare un gorilla nemico. <ul style="list-style-type: none">• Freccia su: salto Quando Frank si trova a mezz'aria può comunque spostarsi a destra e sinistra, ma non può eseguire un ulteriore salto.
Output	Frank salta
Importanza	5
Stabilità	5

Numero del requisito	RF003
Descrizione	Frank – Lancio munizione
Input	Tastiera
Processo	<p>Frank ha a disposizione delle noci di cocco come munizioni da usare per difendersi dai nemici.</p> <ul style="list-style-type: none"> Barra spaziatrice: lancio munizione <p>Frank può lanciare una noce di cocco alla volta. Tuttavia, il numero munizioni su cui fare affidamento è limitato, non è possibile quindi lanciare una noce di cocco se questo numero è uguale a zero.</p>
Output	Frank lancia una noce di cocco
Importanza	5
Stabilità	4

Numero del requisito	RF004
Descrizione	Frank – Munizioni e ricarica
Input	Frank tocca una palma
Processo	<p>Frank ha a disposizione un massimo di 15 noci di cocco, ogni volta che il personaggio tocca una palma il numero. Il numero di munizioni a disposizione verrà decrementato ogni volta nel momento in cui il personaggio lancia una noce di cocco.</p>
Output	Ricarica munizioni
Importanza	4
Stabilità	4

Numero del requisito	RF005
Descrizione	Frank – Banana
Input	Il personaggio cammina sopra l'oggetto
Processo	<p>All'inizio della partita, Frank è in possesso della banana.</p> <p>Il personaggio può raccogliere la banana droppata dal gorilla nemico che la possedeva, una volta eliminato.</p>
Output	Banana compare sulla testa del personaggio in ogni momento in cui questo ne è in possesso.
Importanza	5
Stabilità	4

3.2.2. Caratteristiche del sistema: Nemici

Numero del requisito	RF006
Descrizione	Gorilla - Movimento
Input	n/a
Processo	<p>Ogni gorilla nemico deve essere in grado di muoversi all'interno della scena di gioco autonomamente, sia a destra che sinistra, entro i limiti dello scenario di gioco.</p> <ul style="list-style-type: none"> • Il gorilla con la banana si deve spostare sempre verso lo spawn point più vicino. • Ogni gorilla senza banana deve muoversi sempre verso la posizione della banana, indipendentemente che questa si trovi a terra o in possesso del personaggio o un altro gorilla.
Output	Il nemico si sposta.
Importanza	5
Stabilità	5

Numero del requisito	RF007
Descrizione	Gorilla - Banana
Input	Il gorilla tocca Frank mentre tiene la banana oppure tocca la banana mentre questa si trova a terra.
Processo	<p>Ogni nemico è in grado di raccogliere la banana da terra o di rubarla a Frank (ma non a un altro gorilla). Un gorilla con la banana si dirige verso lo spawn point più vicino.</p>
Output	Banana compare sulla testa del nemico che l'ha raccolta, finché questo ne è in possesso.
Importanza	5
Stabilità	5

Numero del requisito	RF008
Descrizione	Gorilla - Barra della salute
Input	n/a
Processo	<p>Ogni gorilla deve essere suscettibile ai danni inflitti dalle noci di cocco lanciate da Frank.</p> <p>Ogni nemico possiede un certo numero di punti salute, questo valore cala ogni volta il gorilla viene colpito.</p> <p>Quando i punti salute di un gorilla scendono a zero (o meno) questo muore e, eventualmente, dropa la banana.</p>
Output	I punti salute di un nemico sono rappresentati graficamente da una barra colorata posta sopra la sua testa.
Importanza	4
Stabilità	4

3.2.3. Caratteristiche del sistema: Partita

Numero del requisito	RF009
Descrizione	Creazione scenario
Input	n/a
Processo	Ogni volta che viene avviata la partita è necessario realizzare una mappa di gioco sulla quale si svolgerà la partita. La mappa deve essere generata in maniera automatica e casuale.
Output	n/a
Importanza	5
Stabilità	4

Numero del requisito	RF010
Descrizione	Partita – Game over
Input	Un gorilla con la banana raggiunge uno spawnpoint.
Processo	La partita termina e si registra il punteggio del giocatore.
Output	Messaggio di partita terminata e nuovo dialog per inserimento nome del giocatore.
Importanza	5
Stabilità	5

Numero del requisito	RF011
Descrizione	Partita - Pausa
Input	Tastiera
Processo	Durante una partita un giocatore deve essere in grado di poter invocare un menù di gioco tramite il tasto ESC; mentre il menu è attivo il gioco deve entrare nello stato di pausa. Il menù di gioco deve almeno implementare le seguenti funzionalità: <ul style="list-style-type: none"> • Abbandona partita • Ritorno alla partita • Riavvia partita
Output	
Importanza	4
Stabilità	5

Numero del requisito	RF012
Descrizione	Partita - Abbandona partita
Input	Mouse
Processo	L'utente deve avere la possibilità di abbandonare la partita prima che questa termini. Se la partita termina in questo modo, il punteggio non viene salvato.
Output	La partita termina e si ritorna al menu principale.
Importanza	4
Stabilità	4

Numero del requisito	RF013
Descrizione	Partita – Riavvia partita
Input	Mouse
Processo	Il giocatore deve avere la possibilità di riavviare la partita. Se la partita termina in questo modo, il punteggio non viene salvato.
Output	La partita corrente viene abbandonata, si genera un nuovo scenario di gioco e si avvia una nuova partita dove punteggio, timer e difficoltà vengono azzerati.
Importanza	4
Stabilità	4

Numero del requisito	RF014
Descrizione	Partita - Punteggio
Input	Uccisione gorilla.
Processo	Ogni volta che Frank elimina un gorilla nemico, il giocatore guadagna 100 punti per la partita corrente.
Output	Aggiornamento punteggio.
Importanza	3
Stabilità	2

3.3. Vincoli di progettazione

Non sono richiesti particolari vincoli di progetto.

3.4. Requisiti di prestazione

È necessario che il software sia costruito in modo tale da mantenere tempi di risposta ragionevolmente contenuti, entro i limiti di praticità: l'esperienza di gioco deve essere il più possibile reattiva, sia nella risposta ai comandi in input che nell'elaborazione dei dati durante la creazione della partita.

3.5. Altri requisiti

3.5.1 *Requisiti logici del database*

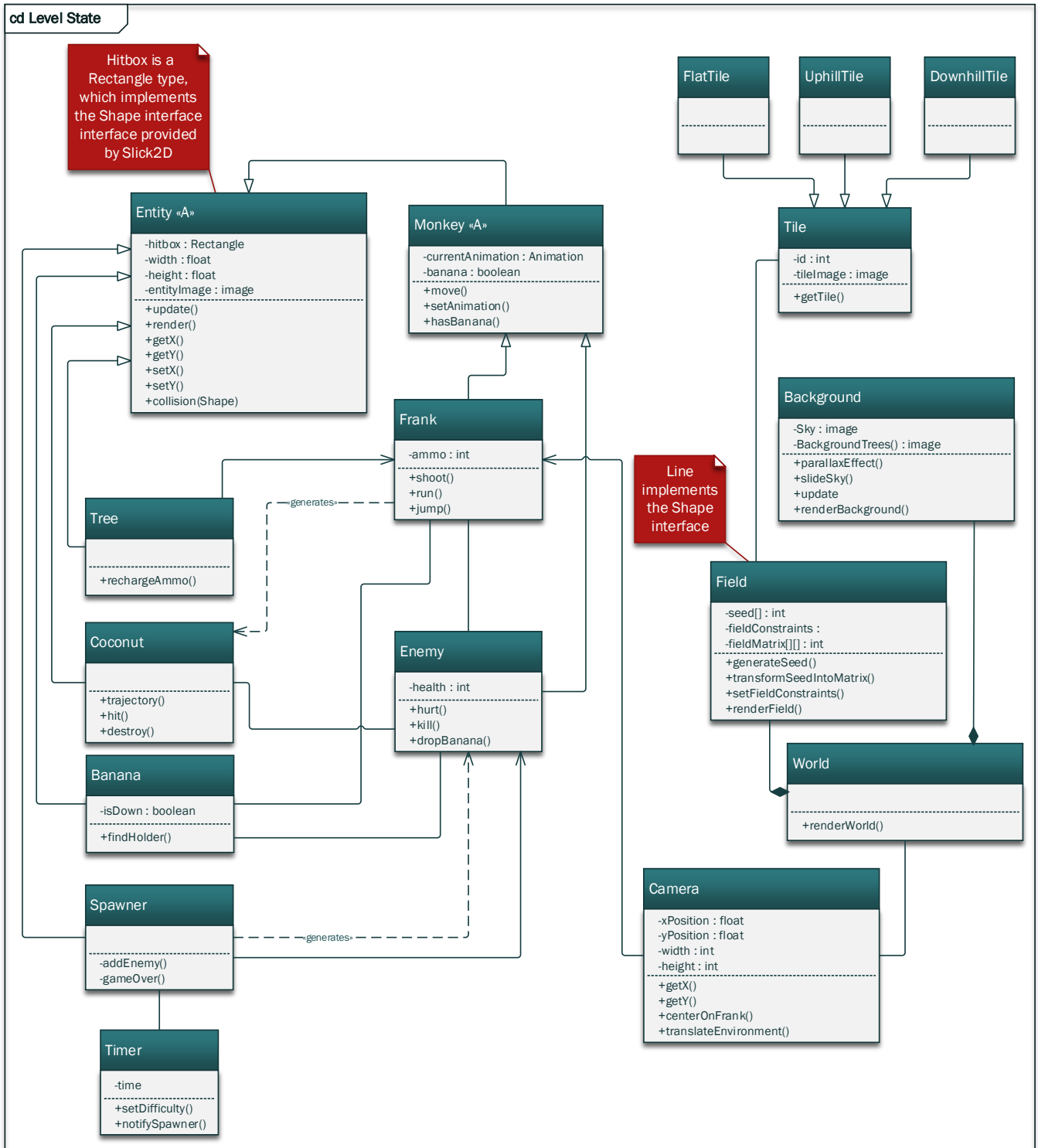
Il software non richiede l'uso di un database, poiché la leaderboard comprende solo una selezione dei migliori punteggi di gioco. Pertanto è sufficiente utilizzare un file salvato localmente che li memorizza.

Unified Modelling Language (UML)

4. Class Diagram

4.1. Scenario di gioco

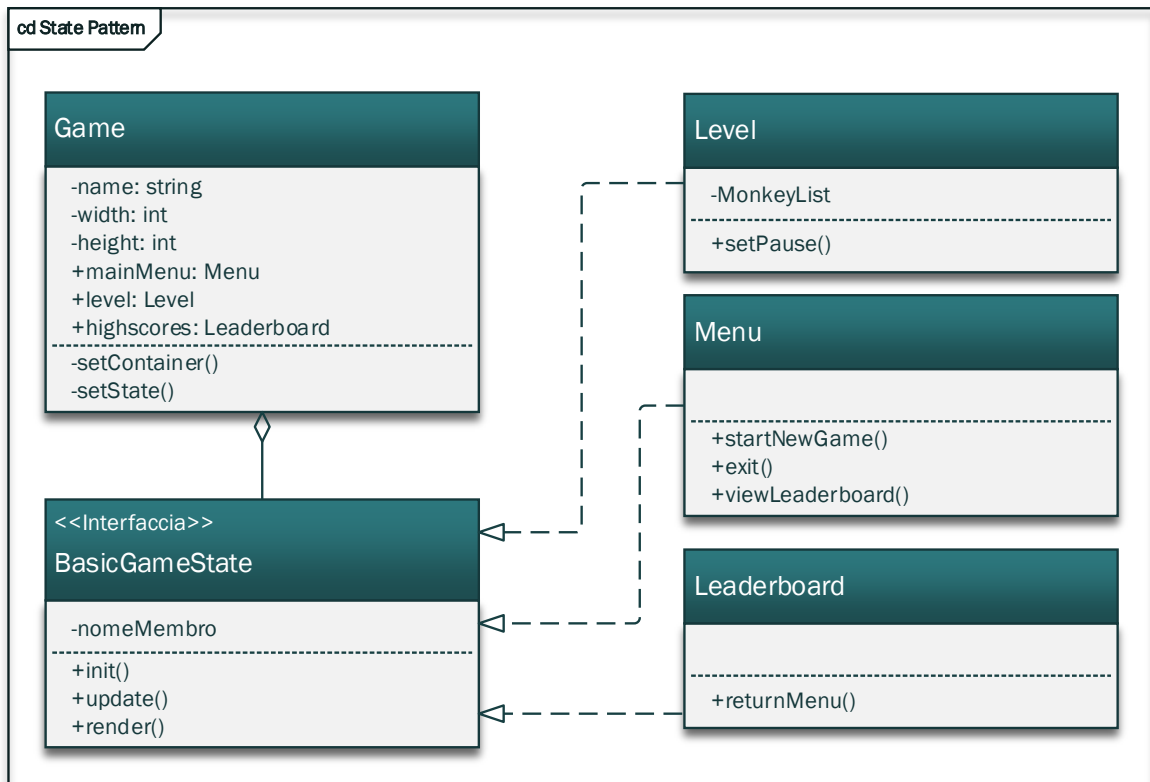
Questo diagramma mostra la suddivisione e l'interazione delle classi principali di una partita.



4.2. Design Patterns

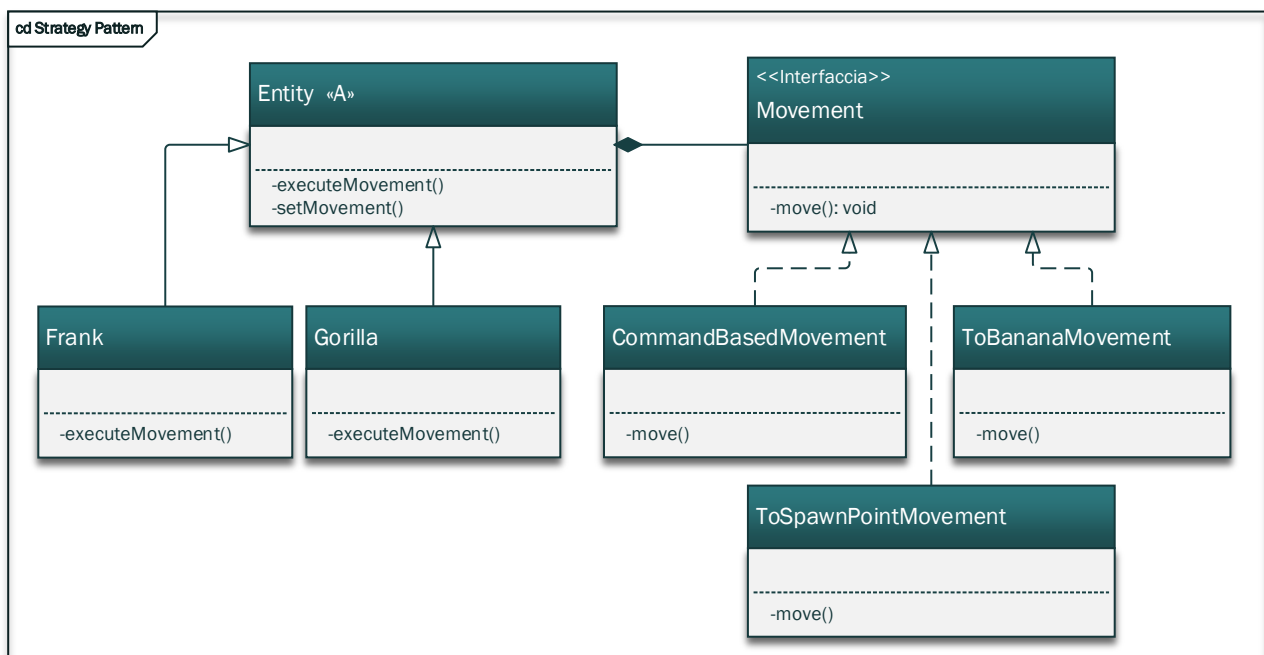
4.2.1 State Pattern

Si utilizza un design pattern di tipo State per facilitare la navigazione all'interno del videogioco. In base ai tasti selezionati nei diversi menù, si otterranno scenari diversi.



4.2.2 Strategy Pattern

All'interno del gioco sono presenti molte entità in grado di muoversi, facendolo in maniera diversa tra di loro: l'uso di uno strategy pattern, oltre che ad evitare la riscrittura del codice, renderebbe possibile l'implementazione dell'OCP in maniera particolarmente efficace aprendo all'estensione l'insieme dei movimenti a disposizione del gruppo delle entità in grado di muoversi.

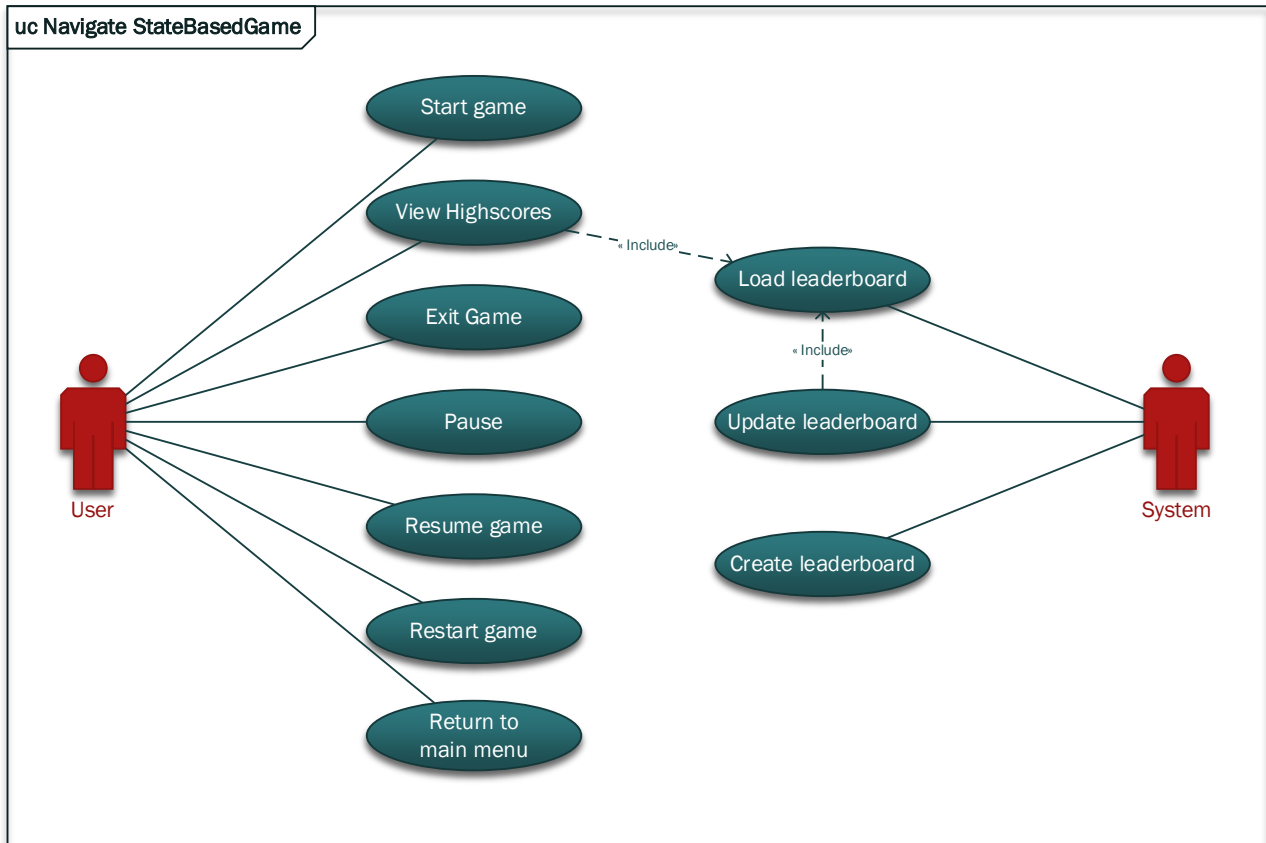


5. Use Case Diagram

Questo diagramma ha lo scopo di fornire una visione d'insieme semplice delle azioni che sono di responsabilità dei diversi attori.

Il primo attore è l'utente, attraverso questa rappresentazione è possibile comprendere quali sono complessivamente le azioni che questo può compiere nel contesto di un gioco basato a stati, quindi fondamentalmente azioni di navigazione tra gli stati.

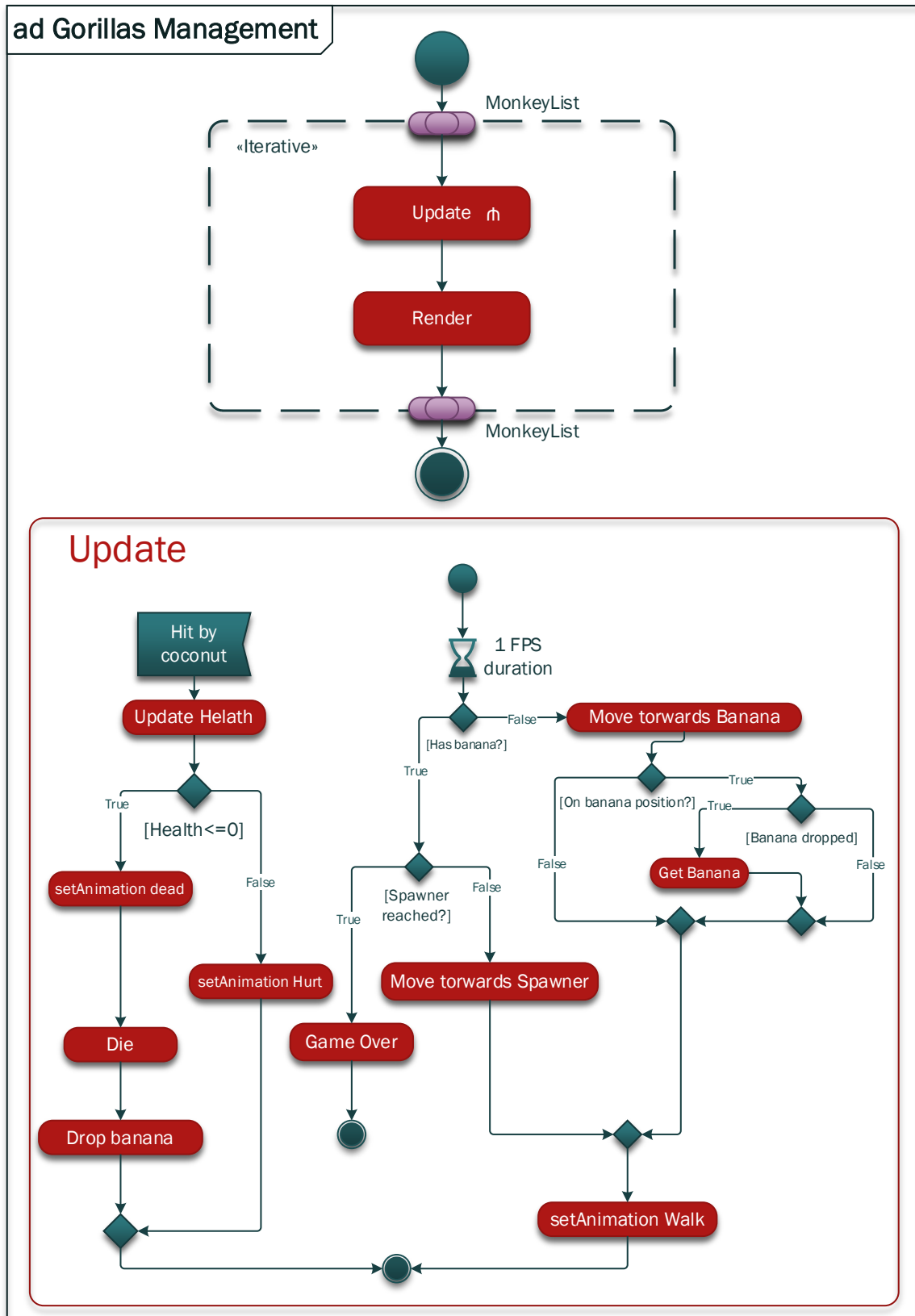
Il secondo attore è il sistema stesso che, in base alle richieste del giocatore, deve essere sostanzialmente in grado di fornire o elaborare le risorse opportune.



6. Activity Diagram

L'activity diagram raffigura lo svolgimento di una particolare attività dell'applicazione.

Con la seguente rappresentazione, si mette in luce il funzionamento di una partita, la gestione delle condizioni in cui si trovano i nemici e dei loro movimenti durante lo svolgimento della partita.

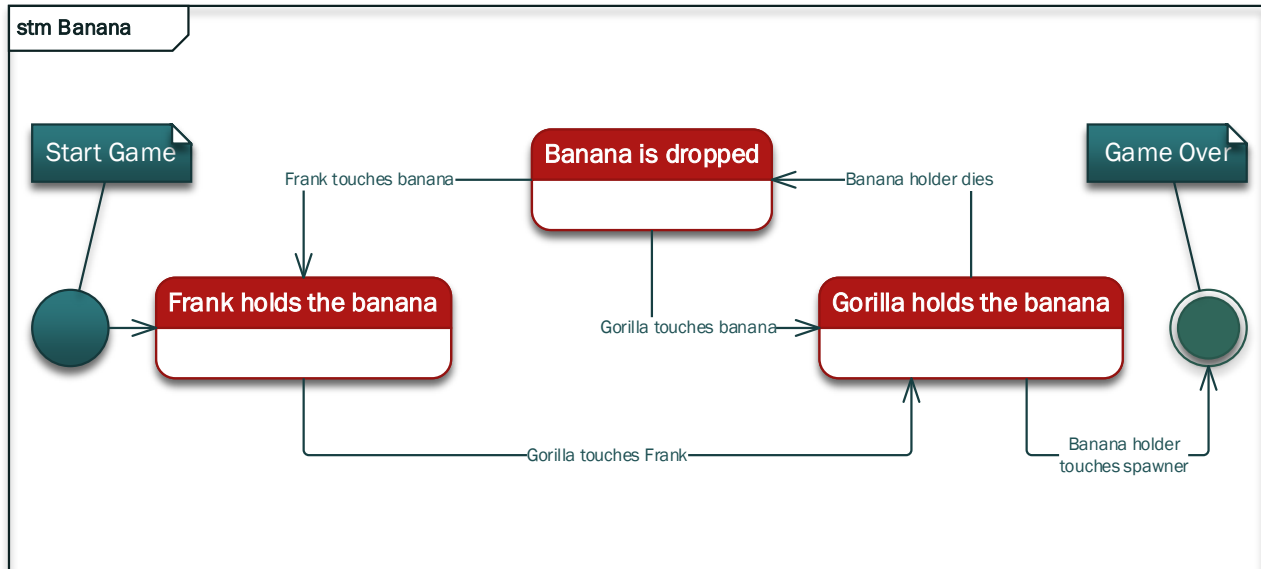


7. State Diagrams

Lo state diagram raffigura i vari stati in cui può trovarsi un oggetto nel corso del tempo.

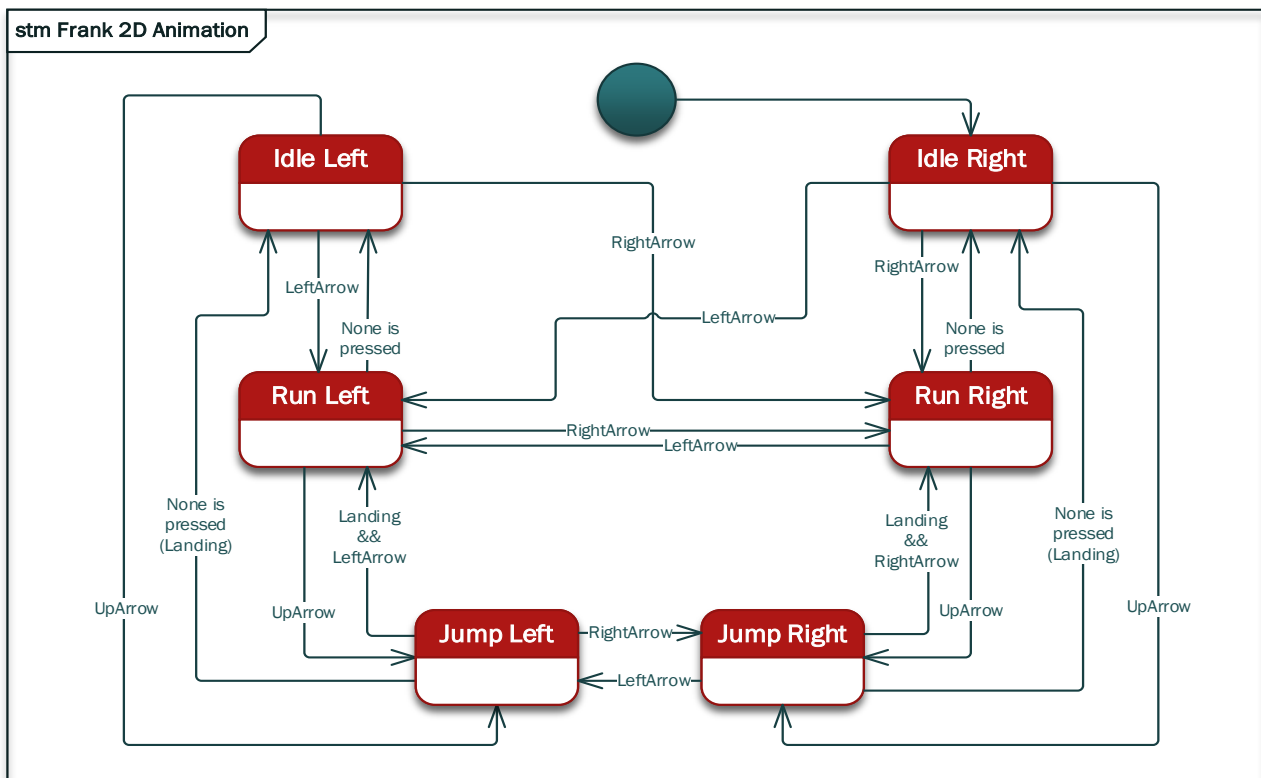
7.1. Scenario di gioco

Questo diagramma ha l'obiettivo di mostrare cosa può accadere alla banana in base a quali entità interagiscono con questa durante lo svolgimento di una partita.



7.2. Gestione delle animazioni

Nel seguente diagramma sono illustrati gli stati che può assumere Frank in conformità ai comandi che gli vengono imposti dal giocatore. Da questi stati dipendono le animazioni che dovranno essere mostrate a schermo.



8. Sequence Diagram

In questa rappresentazione si evidenzia cosa comporta premere i tasti a cui è associato Frank, in particolare, i comandi relativi ai movimenti e al lancio di una noce di cocco, oltre a descrivere le conseguenze di un colpo andato a segno con un proiettile.

