

Progetto Ingegneria del Software

Realizzazione Risiko con Unity

Documento dei requisiti

Si vuole realizzare il gioco del Risiko per 2/3/6 giocatori.

La mappa base del gioco, modellata sulla mappa reale del mondo, è costituita da 6 continenti a loro volta costituiti da stati, in totale 42.

Gli stati possono essere considerati confinanti anche se non fisicamente adiacenti sulla mappa (esempio passaggi oceanici).

Il gioco è diviso in momenti:

Documento dei requisiti

Momento iniziale:

A ogni giocatore si assegna un colore identificativo e un nome univoco.

Ogni territorio (sinonimo di stato) è assegnato a un giocatore in modo tale che la spartizione risulti equa.

Ogni giocatore ha diritto a 63/42/21 armate iniziali (a seconda del numero di giocatori). Un'armata è posizionata obbligatoriamente su ogni stato assegnato al giocatore; le altre saranno assegnabili a piacere.

Ad ogni giocatore è assegnato un obiettivo segreto.

Documento dei requisiti

Momento di gioco, diviso in fasi:

Preturno:

Il giocatore di turno ha diritto a tante armate di rinforzo quanti sono i territori diviso 3 (divisione intera).

Se il giocatore possiede un continente (cioè ogni stato del continente stesso) ha diritto ad un numero di armate supplementari (variabile a seconda del continente).

Tutte le armate di rinforzo sono posizionabili a piacere sui territori posseduti.

Documento dei requisiti

Attacco:

Il giocatore può decidere di attaccare uno stato confinante non già in suo possesso.

L'attacco può essere effettuato con un massimo di 3 armate purché ne resti almeno un'altra nel territorio stesso.

L'attacco può essere ripetuto a piacere finché il giocatore non specifica esplicitamente il “fine fase” oppure non ha più mosse a disposizione.

Ad ogni armata usata per attaccare corrisponde il lancio di un dado di attacco.

Il giocatore proprietario dello stato attaccato si difende con un numero di armate pari al minimo tra le armate in proprio possesso e quelle usate dall'attaccante.

Ad ogni armata usata per la difesa corrisponde un lancio di un dado di difesa.

Documento dei requisiti

L'esito dell'attacco è determinato dal confronto tra i risultati dei lanci presi ordinatamente (il maggiore d'attacco è confrontato con il maggiore della difesa). In caso di parità, vince la difesa.

Le armate perse (quelle impegnate nei confronti persi) sono rimosse dal gioco.

Post-turno:

Opzionalmente, il giocatore può spostare un numero arbitrario di armate da un solo territorio in suo possesso ad un altro confinante (sempre in suo possesso).

Documento dei requisiti

Momento finale:

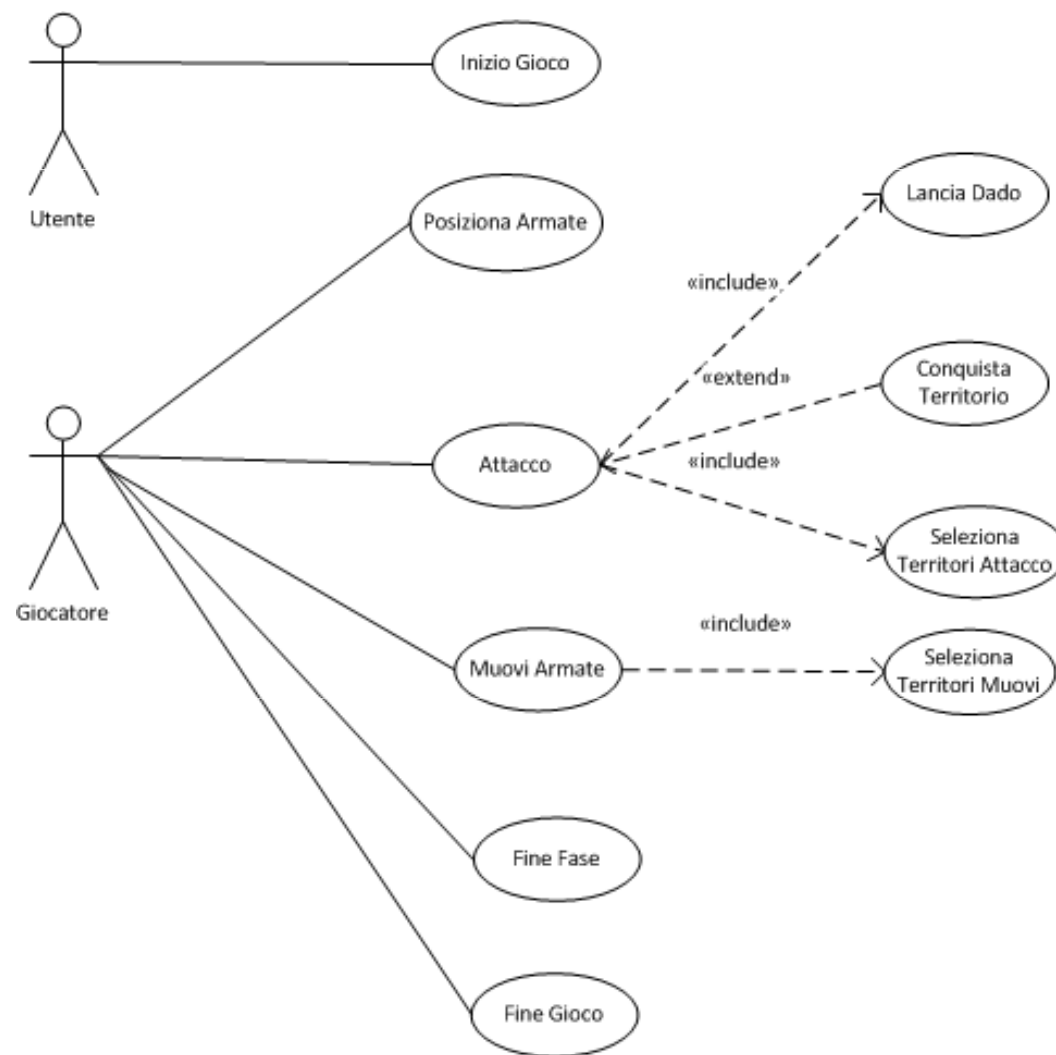
Il gioco termina quando un giocatore raggiunge il proprio obiettivo segreto.

Dettagli aggiuntivi:

Gli obiettivi possono essere di diverso tipo (es.: conquistare un certo numero di territori, conquistare certi continenti, sconfiggere un certo giocatore etc).

Il momento di gioco viene effettuato in sequenza dai giocatori, in maniera ciclica fino al raggiungimento della fase finale.

Casi d'uso



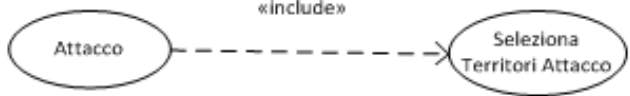
Casi d'uso



Scenari

Titolo	Inizio gioco
Descrizione	Decisione dell'utente di iniziare una nuova partita
Relazioni	<pre> graph LR IG([Inizio Gioco]) -.-> «include» AC([Assegna Colore]) IG -.-> «include» AT([Assegna Territori]) IG -.-> «include» AO([Assegna Obiettivo]) IG -.-> «include» AA([Assegna Armate]) IG -.-> «include» SNG([Sceita numero giocatori]) SNG -.-> «include» SN([Sceita Nome]) </pre>
Attori	Utente
Precondizioni	Non deve esistere una partita già in corso
Postcondizioni	E' iniziata una partita
Scenario Principale	<ol style="list-style-type: none"> 1. L'utente stabilisce il numero di giocatori (CU "Sceita numero giocatori") 2. L'utente stabilisce i nomi dei giocatori (CU "Sceita nome") 3. Il sistema assegna a ciascun giocatore un colore 4. Il sistema assegna a ciascun giocatore un ugual numero di territori 5. Il sistema assegna l'obiettivo segreto 6. Il sistema assegna delle armate iniziali
Scenari Alternativi	
Requisiti non funzionali	
Punti aperti	


Scenari

Titolo	Selezione Territori Attacco
Descrizione	Viene effettuata la selezione di due stati: la sorgente e la destinazione dell'attacco
Relazioni	 <pre> graph LR Attacco((Attacco)) -.-> «include» SelezionaTerritoriAttacco((Seleziona Territori Attacco)) </pre>
Attori	Giocatore
Precondizioni	<ul style="list-style-type: none"> - Esiste una partita in corso - E' il turno del giocatore - E' la fase di attacco del giocatore - Il giocatore possiede un territorio da cui attaccare (2+ armate e confinante con stato nemico)
Postcondizioni	
Scenario Principale	<ol style="list-style-type: none"> 1. Il giocatore seleziona uno stato in suo possesso 2. Il giocatore seleziona uno stato avversario confinante con quello scelto al punto 1 o uno suo stato con più di 2 armate 3. La selezione diventa effettiva
Scenari Alternativi	<ol style="list-style-type: none"> 1.a Se il giocatore non ha almeno due armate nello stato selezionato <ol style="list-style-type: none"> 1.a.1 Si torna al passo 1 2.a Se lo stato selezionato è uno stato avversario ma non confinante <ol style="list-style-type: none"> 2.a.1 Si torna al passo 2 2.b Se lo stato selezionato è di proprietà del giocatore ma ha solo 1 armata <ol style="list-style-type: none"> 2.b.1 Si torna al passo 2
Requisiti non funzionali	
Punti aperti	

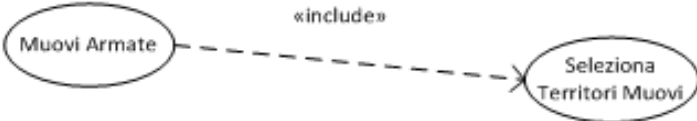
Scenari

Titolo	Posiziona Armate
Descrizione	Viene effettuato l'assegnamento delle armate non ancora posizionate
Relazioni	
Attori	Giocatore
Precondizioni	<ul style="list-style-type: none">- Esiste una partita in corso- E' il turno del giocatore- E' la fase di preturno
Postcondizioni	
Scenario Principale	<ol style="list-style-type: none">1. Il giocatore può selezionare uno stato in suo possesso a cui aggiungere un'armata o decidere di annullare tutte le aggiunte già fatte2. Il giocatore può terminare la fase (CU "Fine Fase"), altrimenti<ol style="list-style-type: none">2.a Si torna al passo 1
Scenari Alternativi	<ol style="list-style-type: none">1.a Se lo stato non è di suo possesso o non ha armate da posizionare<ol style="list-style-type: none">1.a.1 Si torna al passo 1
Requisiti non funzionali	
Punti aperti	

Scenari

Titolo	Selezione Territori Muovi
Descrizione	Viene effettuata la selezione di due stati: la sorgente e la destinazione dello spostamento
Relazioni	 <pre> graph LR A((Muovi Armate)) -.-> «include» B((Seleziona Territori Muovi)) </pre>
Attori	Giocatore
Precondizioni	<ul style="list-style-type: none"> - Esiste una partita in corso - E' il turno del giocatore - E' la fase di spostamento del giocatore
Postcondizioni	
Scenario Principale	<ol style="list-style-type: none"> 1. Il giocatore seleziona uno stato in suo possesso 2. Il giocatore seleziona un altro stato in suo possesso 3. La selezione diventa effettiva
Scenari Alternativi	<ol style="list-style-type: none"> 1.a Se il giocatore non ha almeno due armate nello stato selezionato oppure lo stato non è in suo possesso <ol style="list-style-type: none"> 1.a.1 Si torna al passo 1 2.a Se lo stato non è in suo possesso o è in suo possesso ma non è confinante con quello scelto al punto 1 e ha una armata <ol style="list-style-type: none"> 2.a.1 Si torna al passo 2 2.b Se lo stato è in suo possesso ma non è confinante con quello scelto al punto 1 e ha almeno due armate <ol style="list-style-type: none"> 2.b.1 Si sovrascrive con questo stato la scelta effettuata al punto 1 2.b.2 Si torna al punto 2
Requisiti non funzionali	
Punti aperti	

Scenari

Titolo	Muovi Armate
Descrizione	Il giocatore ridistribuisce le armate nella sua fase di post-turno
Relazioni	 <pre> graph LR A((Muovi Armate)) -.-> «include» B((Seleziona Territori Muovi)) </pre>
Attori	Giocatore
Precondizioni	<ul style="list-style-type: none"> - Esiste una partita in corso - E' il turno del giocatore - E' la fase di post-turno
Postcondizioni	- Le armate nel primo territorio sono calate nella stessa misura in cui sono aumentate le armate nel secondo territorio
Scenario Principale	<ol style="list-style-type: none"> 1. Il giocatore usa CU "Seleziona Territori Muovi" 2. Il giocatore seleziona quante armate (max N-1 dove N sono quelle presenti nel primo territorio selezionato) spostare dal 1° al 2° territorio
Scenari Alternativi	
Requisiti non funzionali	
Punti aperti	

Scenari

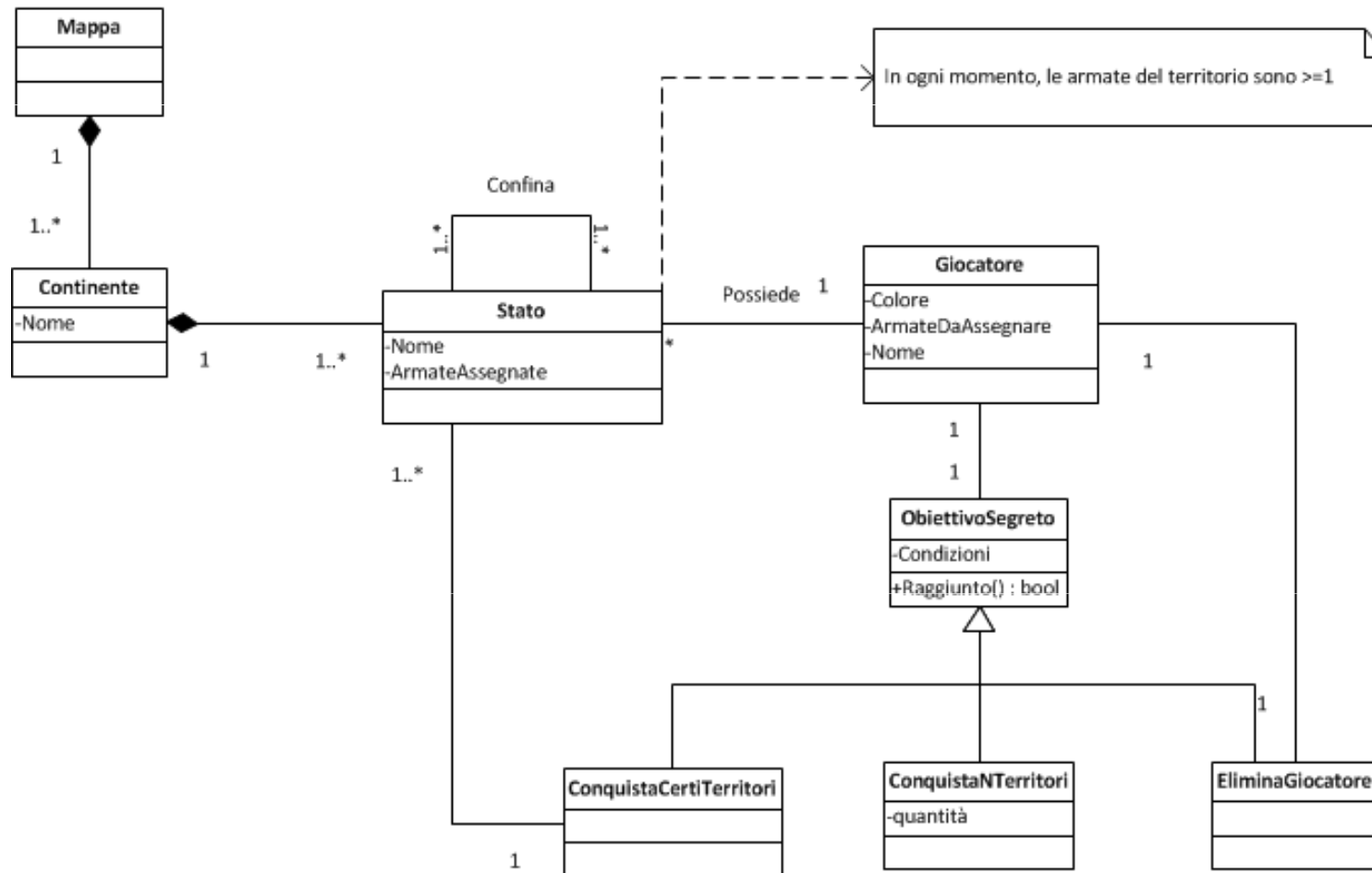
Titolo	Attacco
Descrizione	Viene effettuato un attacco
Relazioni	<pre> graph TD Attacco((Attacco)) -.-> «includes» LanciaDado((Lancia Dado)) Attacco -.-> «extend» ConquistaTerritorio((Conquista Territorio)) Attacco -.-> «includes» SelezionaTerritoriAttacco((Seleziona Territori Attacco)) </pre>
Attori	Giocatore
Precondizioni	<ul style="list-style-type: none"> - Esiste una partita in corso - E' il turno del giocatore - E' la fase di attacco - Il giocatore possiede almeno uno stato, confinante con un avversario, con più di 1 armata
Postcondizioni	<p>Il numero di armate negli stati selezionati è rimasto consistente con l'esito del lancio dei dadi.</p> <p>Con più precisione:</p> <ol style="list-style-type: none"> 1) Il dado con il punteggio più alto ottenuto dall'attaccante si confronta con il punteggio più alto del difensore. 2) Se è più alto il punteggio dell'attaccante, il difensore dovrà togliere dal territorio attaccato una delle sue armate (riponendola fra quelle in dotazione). 3) Se è più alto il punteggio del difensore, sarà l'attaccante a dover ritirare una delle sue armate dal territorio dal quale ha sferrato l'attacco. 4) In caso di pareggio vince sempre il difensore. 5) Se ciascun giocatore ha lanciato un solo dado, il confronto avviene immediatamente e l'attacco si conclude con il ritiro di una delle armate dal territorio del perdente. 6) Se, invece, l'attaccante o il difensore hanno lanciato due o più dadi, il confronto avviene per eliminazione progressiva del punteggio più basso.
Scenario Principale	<ol style="list-style-type: none"> 1. Il giocatore usa CU "Seleziona Territori Attacco" 2. Il giocatore decide con quante armate attaccare, minimo 1, massimo 3 3. In base all'esito del lancio dei dadi corrispondenti: <ol style="list-style-type: none"> 3.a Se Il giocatore possiede almeno uno stato, confinante con un avversario, avente più di 1 armata, si torna al passo 1 3.b Altrimenti, il giocatore può passare alla fase successiva (CU "Fine Fase")
Scenari Alternativi	
Requisiti non funzionali	
Punti aperti	

Scenari

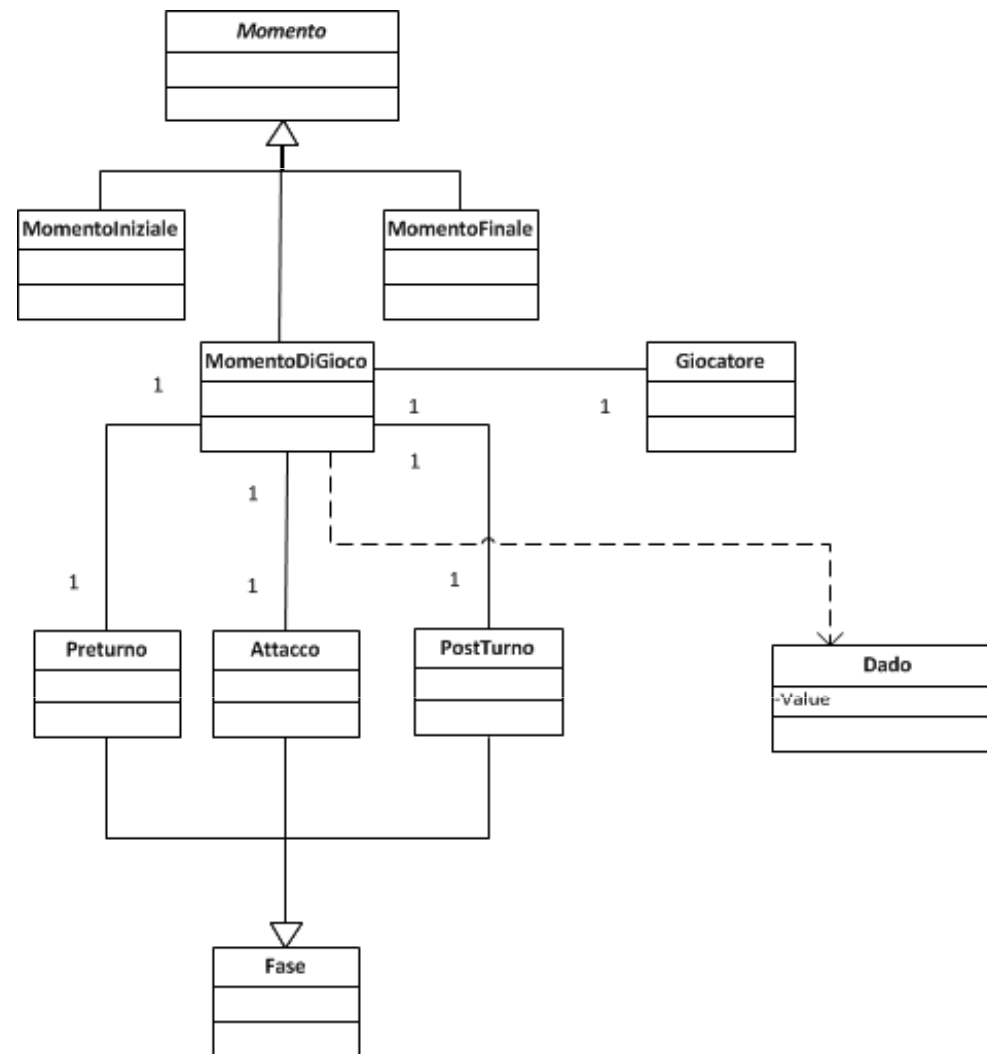
Titolo	Fine Fase
Descrizione	Si dichiara conclusa la propria fase di gioco
Relazioni	
Attori	Giocatore
Precondizioni	- Esiste una partita in corso - E' il turno del giocatore
Postcondizioni	- Si è passati alla fase successiva del turno del giocatore o si è passati alla prima fase del turno del giocatore successivo
Scenario Principale	1. Il giocatore dichiara la fine della fase corrente, eventualmente del proprio turno
Scenari Alternativi	
Requisiti non funzionali	
Punti aperti	

Titolo	Fine Gioco
Descrizione	Il gioco finisce per il raggiungimento dell'obiettivo segreto da parte di un giocatore
Relazioni	
Attori	Giocatore
Precondizioni	- E' in corso una partita - E' stato appena raggiunto un obiettivo segreto
Postcondizioni	- Non esiste una partita in corso
Scenario Principale	1. Il giocatore ha vinto e termina la partita
Scenari Alternativi	
Requisiti non funzionali	
Punti aperti	

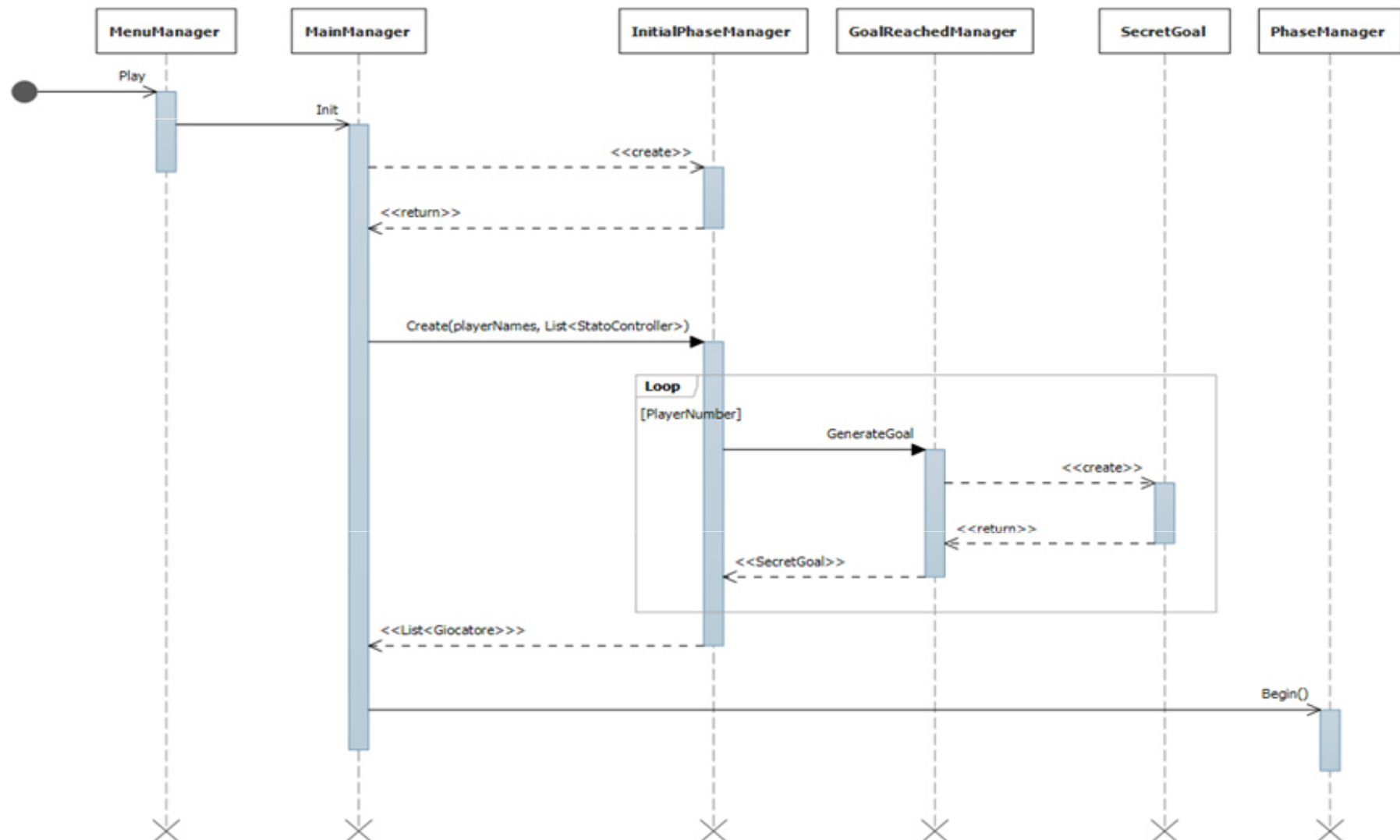
Modello statico: diagramma delle classi



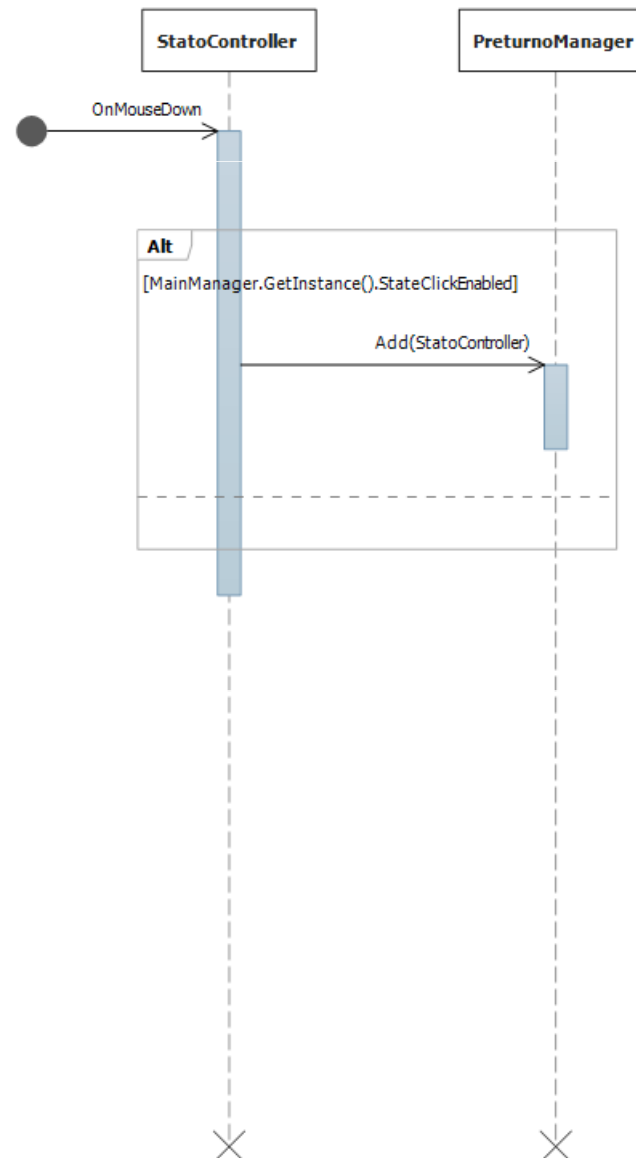
Modello statico: diagramma delle classi



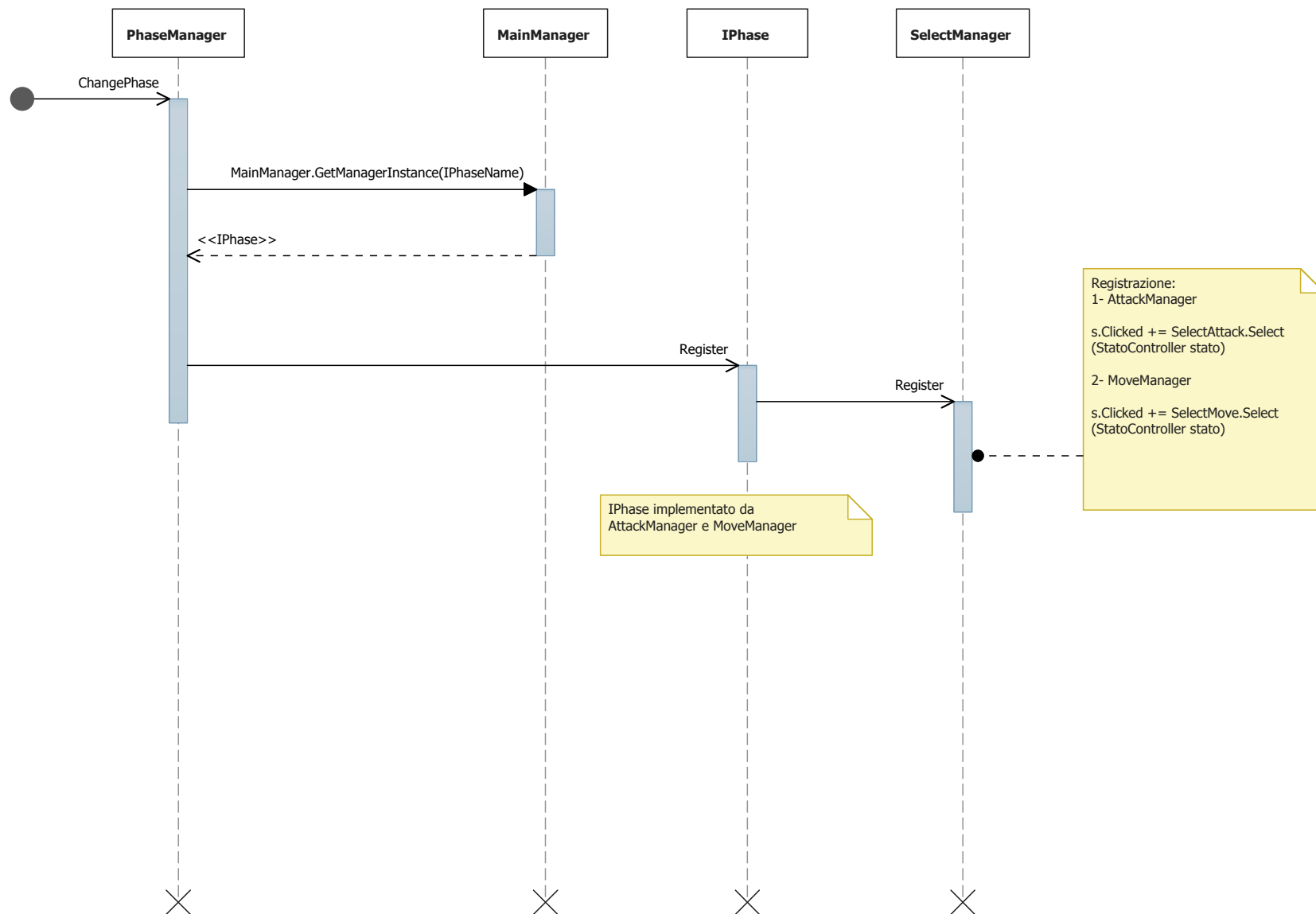
Modello dinamico: diagramma di sequenza – Inizio Gioco



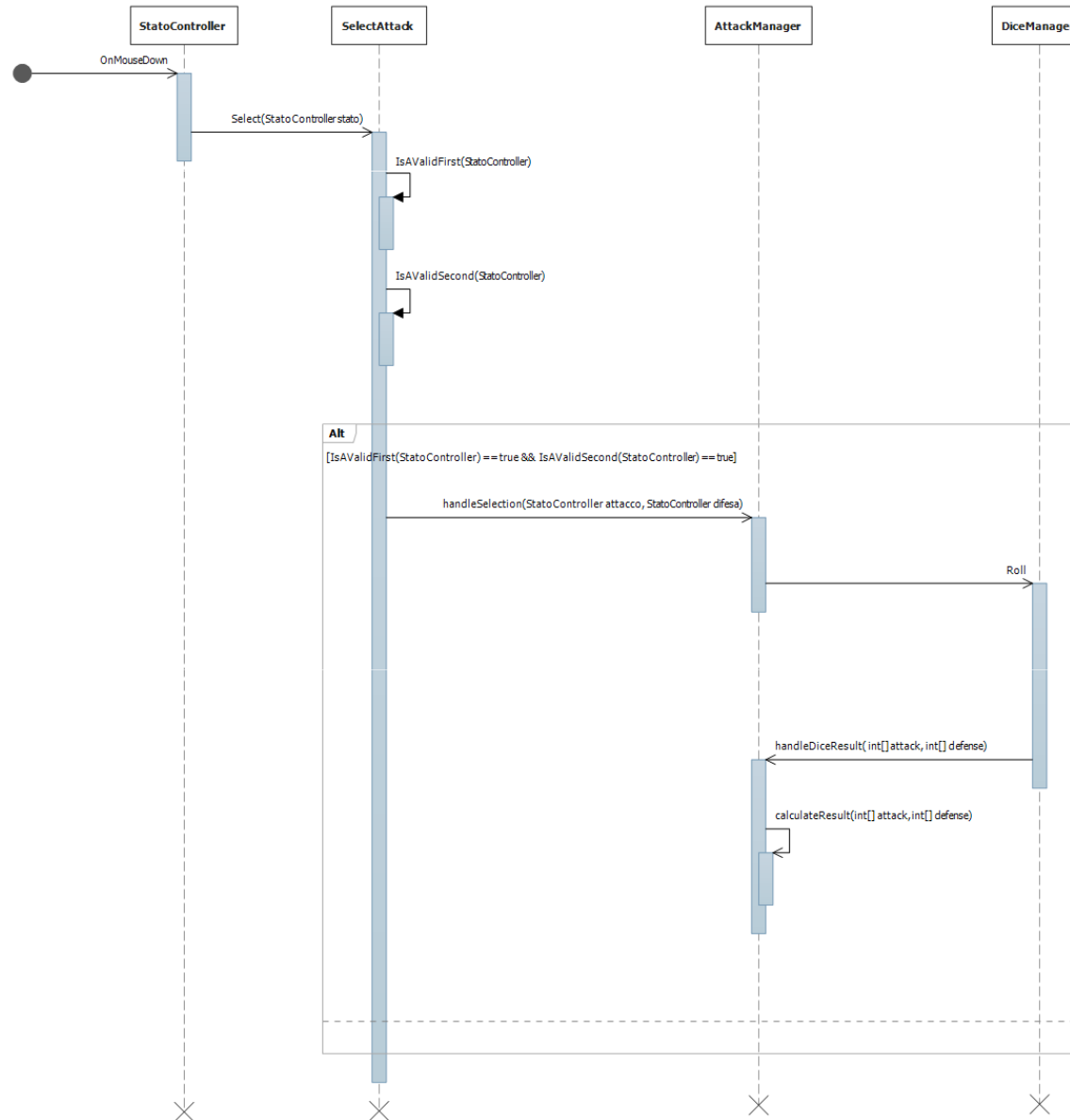
Modello dinamico: diagramma di sequenza – Aggiunta Armate



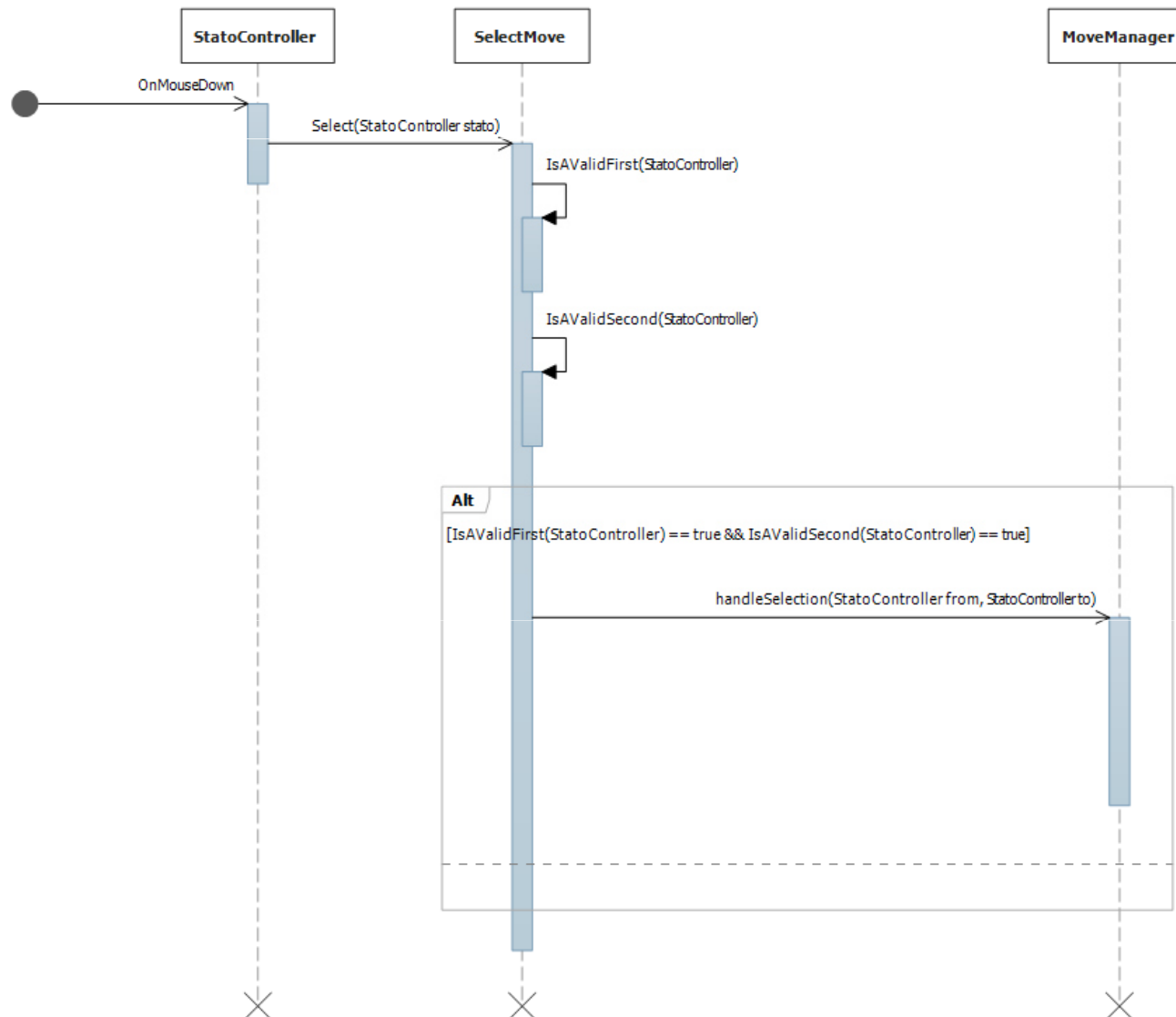
Modello dinamico: diagramma di sequenza – Cambio fase



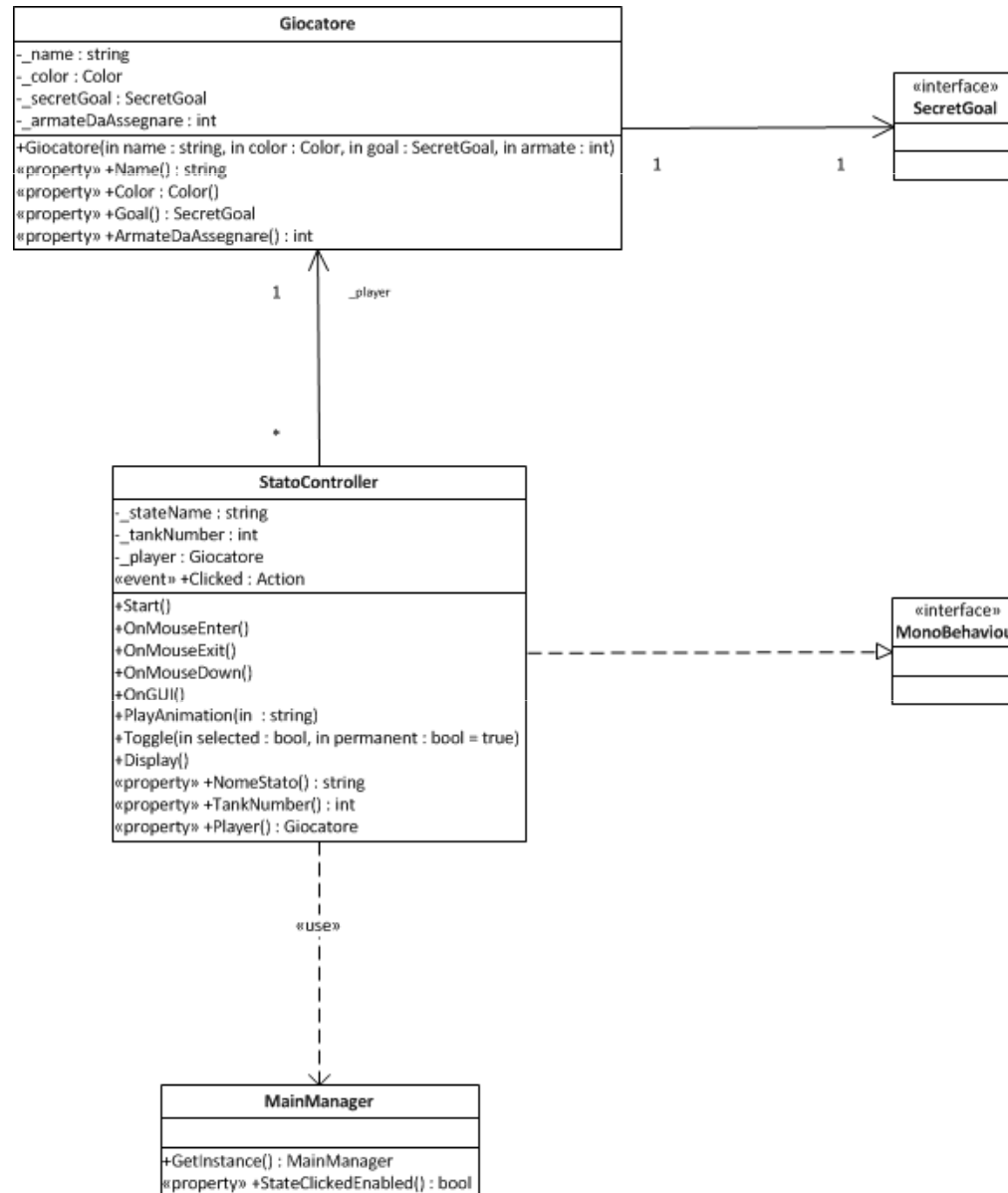
Modello dinamico: diagramma di sequenza – Attacco



Modello dinamico: diagramma di sequenza – Spostamento



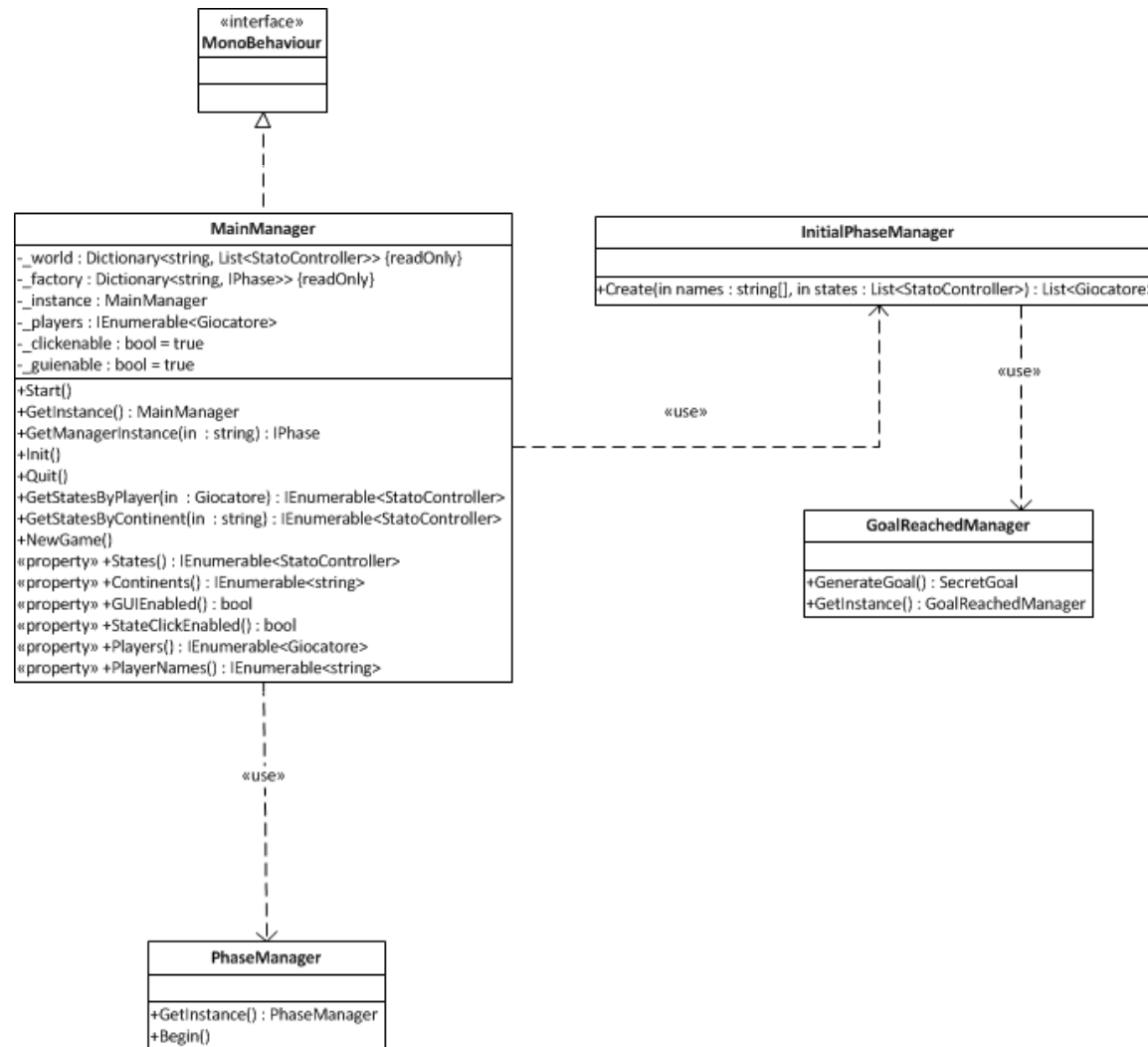
Progettazione – Stati e giocatori



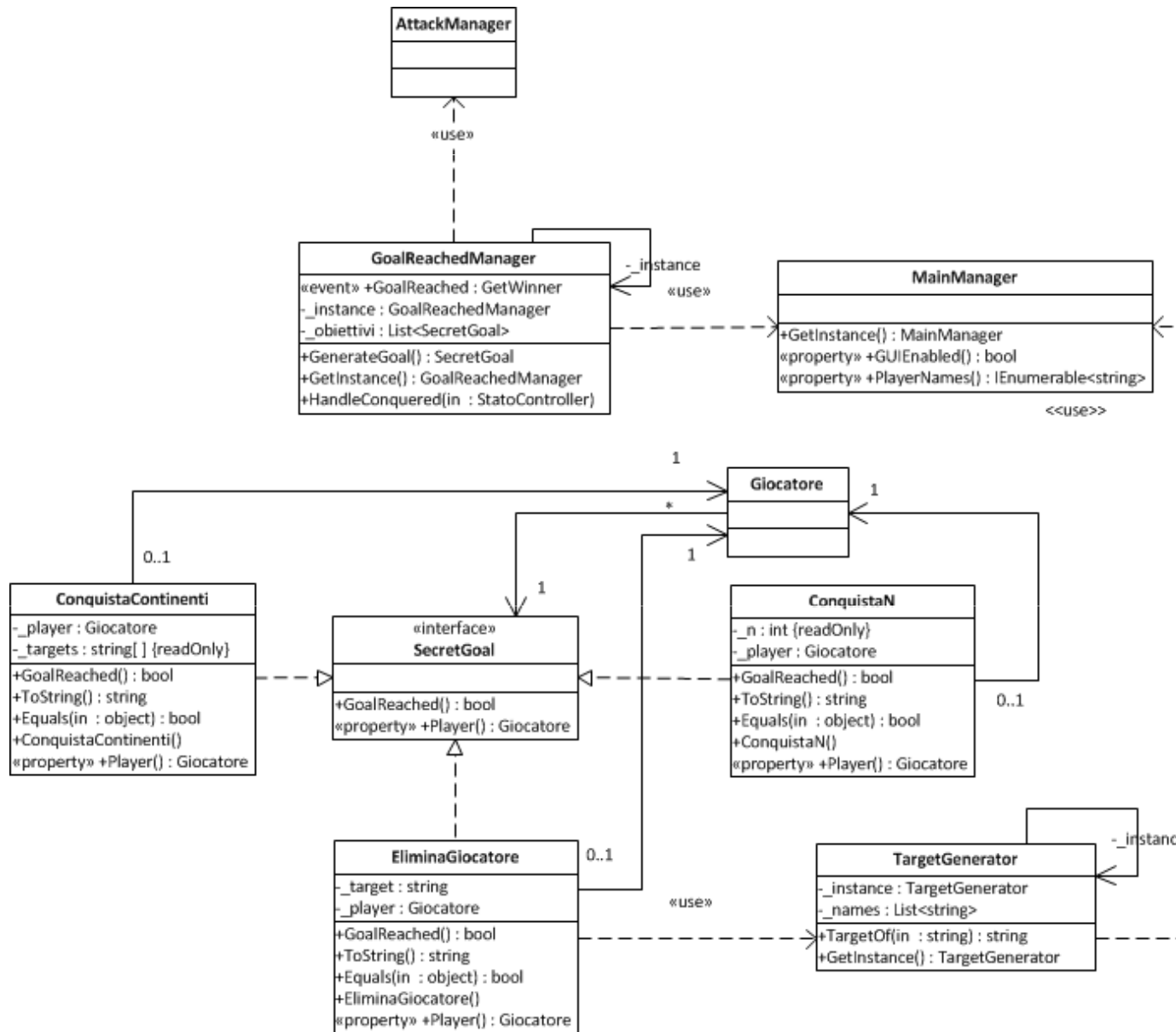
Progettazione – Unity



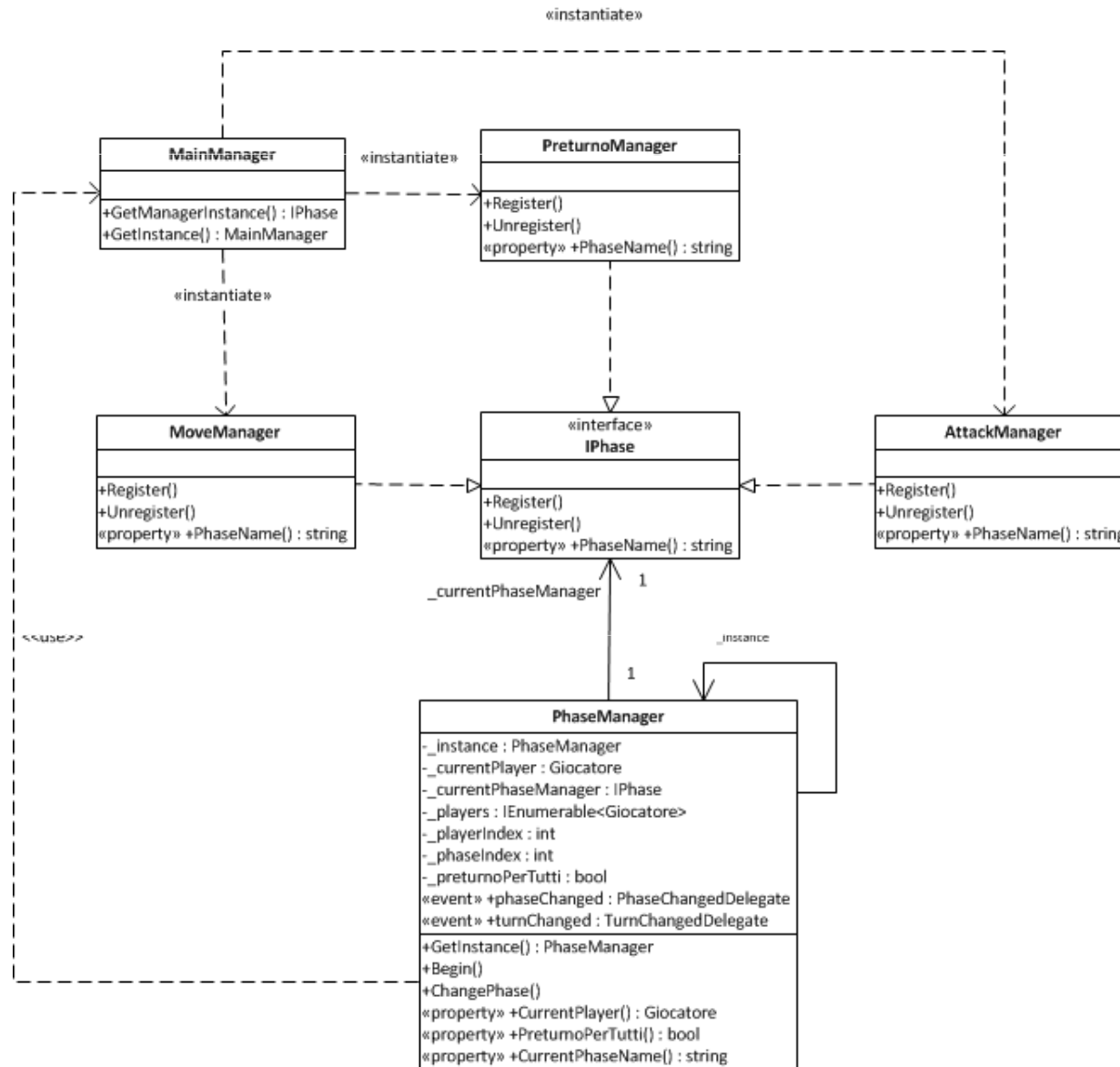
Progettazione – Main Manager



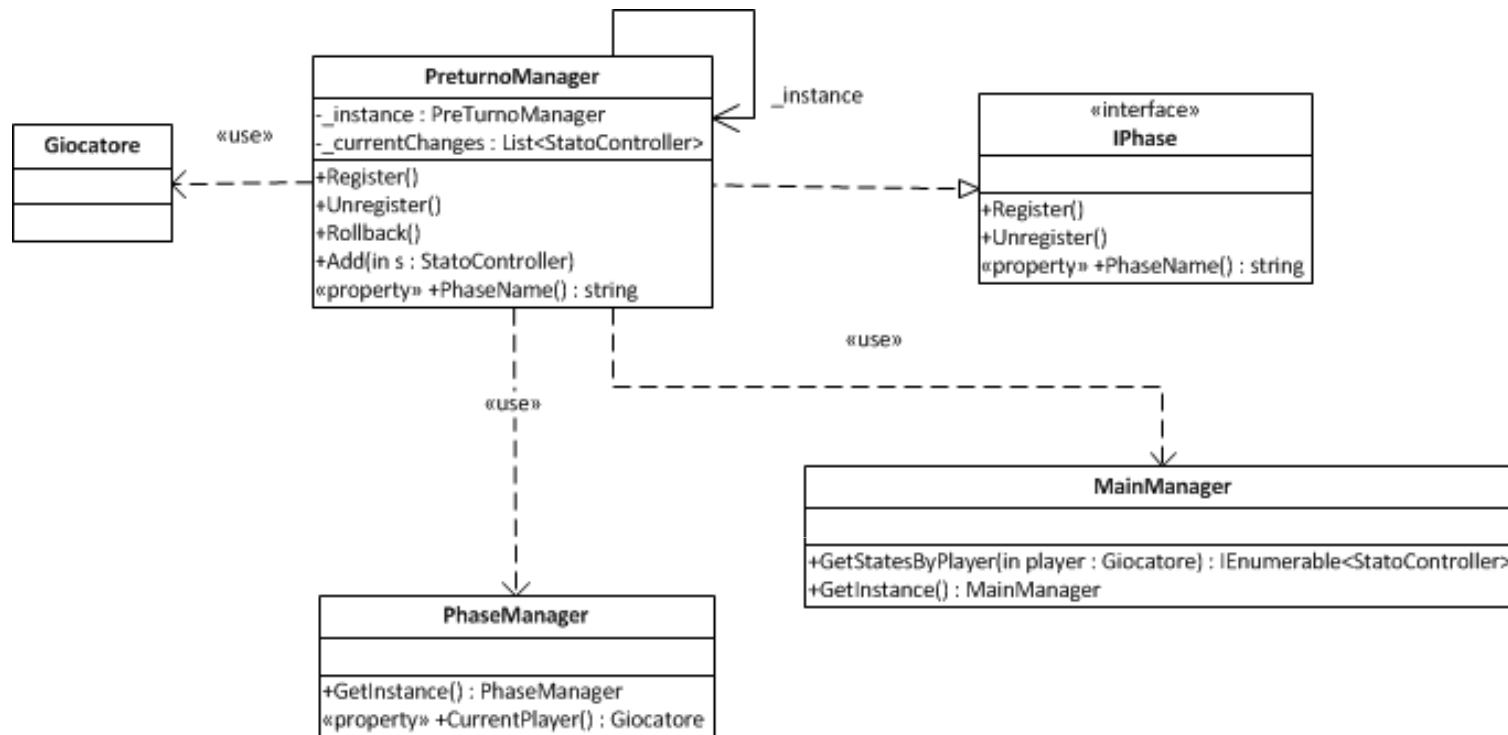
Progettazione – Obiettivi Segreti



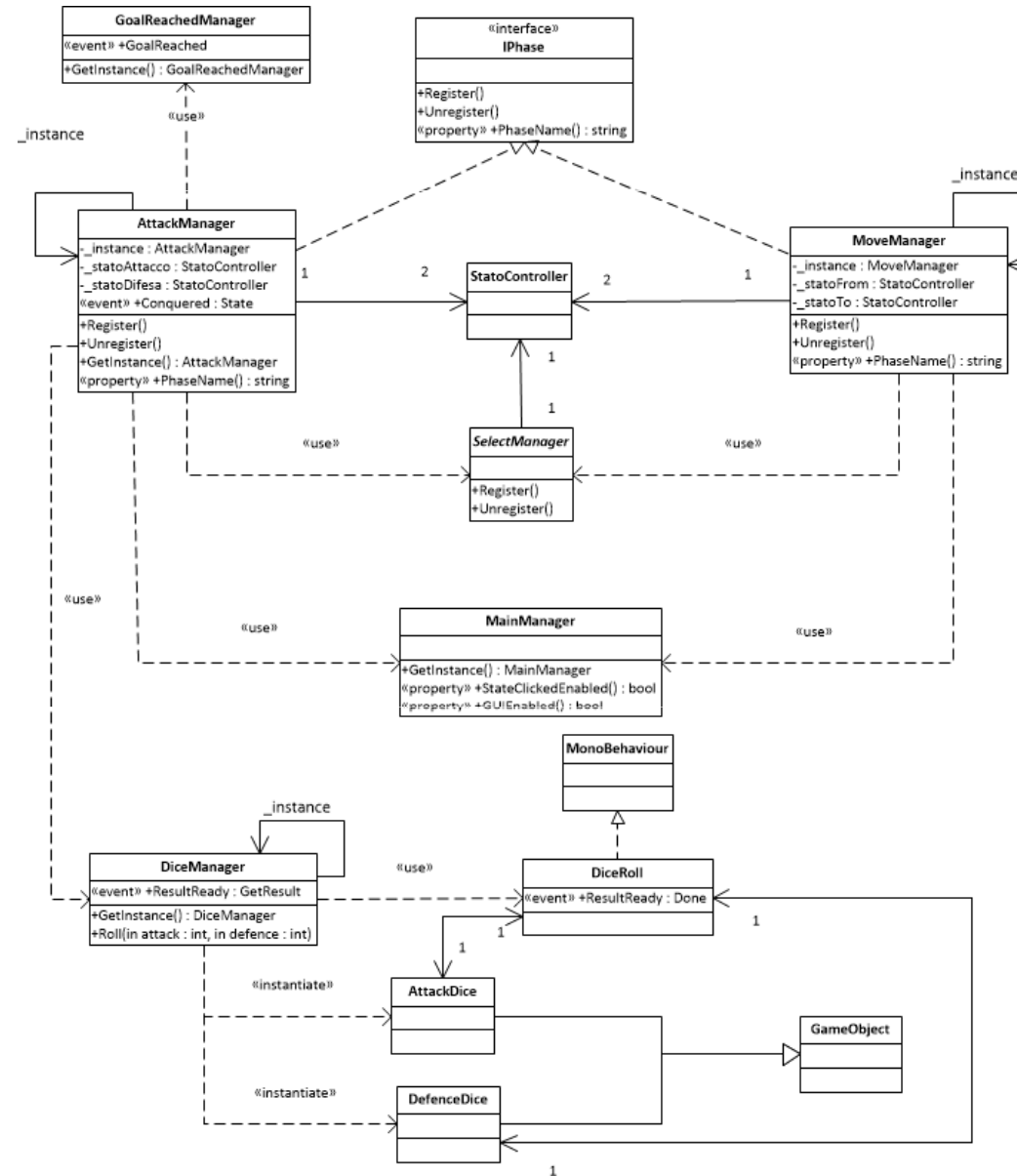
Progettazione – Phase Manager



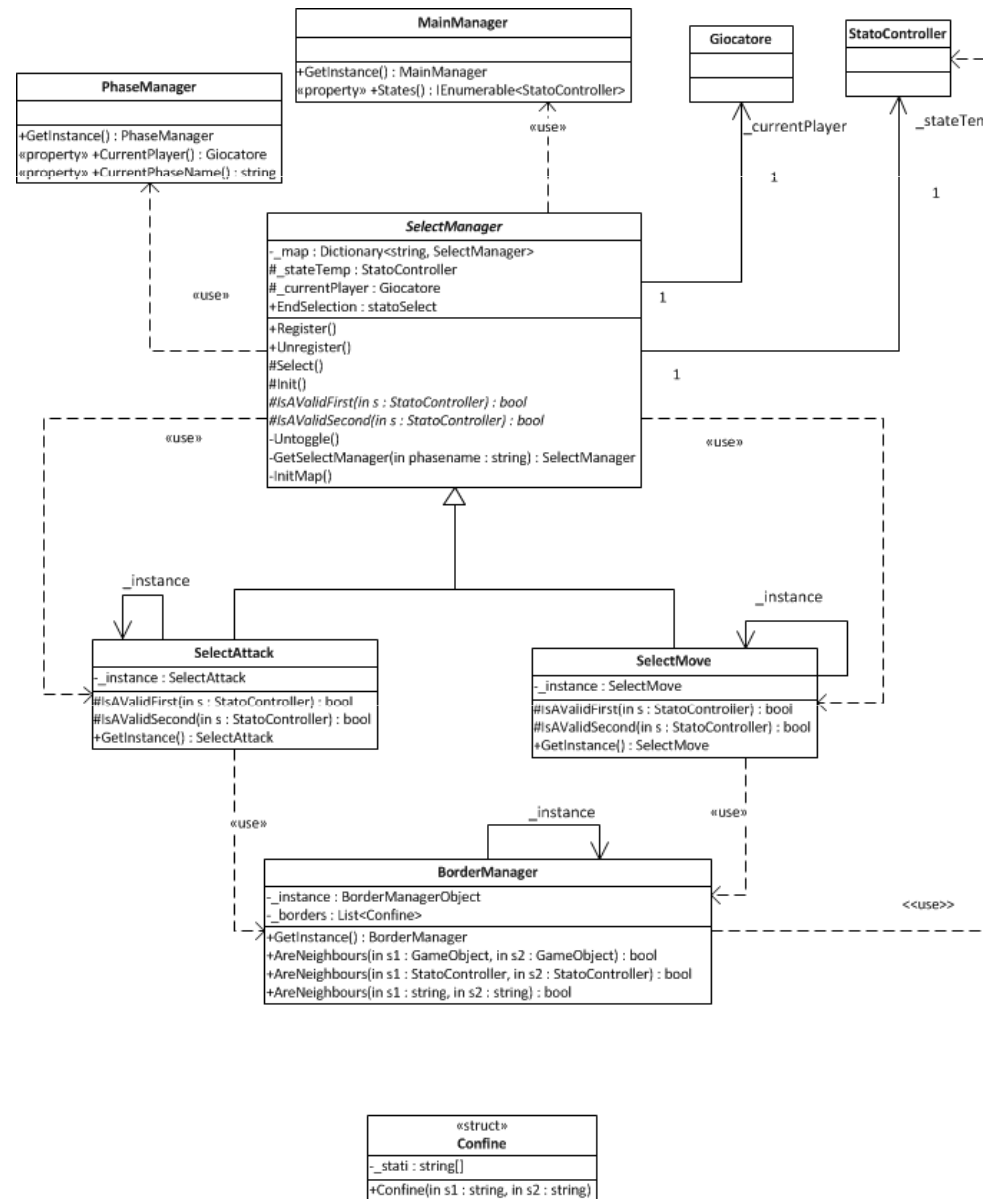
Progettazione – Preturmo Manager



Progettazione – Attack/Move Manager



Progettazione – Select Manager



Design Pattern e principi di progettazione

Al fine di garantire flessibilità nella realizzazione degli obiettivi segreti, è stato applicato il pattern **Strategy** realizzando un'interfaccia SecretGoal. Quest'ultima è implementata da tutte le classi che rappresentano un tipo di obiettivo segreto, permettendo così una facile estensione a nuove tipologie in caso di necessità, pur rispettando il **design principle Open/Closed**.

Per ovviare a simili problemi incontrati nella concretizzazione del SelectManager, è stato nuovamente applicato il pattern **Strategy** al fine di implementare diverse logiche di selezione dei territori. SelectAttack e SelectMove sono quindi due classi concrete che implementano i metodi IsAValidFirst() e IsAValidSecond() della classe astratta SelectManager, permettendo una facile espansione in caso di bisogno. In entrambi i casi sopracitati si è garantito il **Dependency Inversion Principle** in modo da eliminare la dipendenza da classi concrete fornendo interfacce stabili, le quali rendono i moduli maggiormente riusabili e facilmente estendibili.

Design Pattern e principi di progettazione

Infine, essendo il progetto fortemente basato sugli eventi, è stato applicato numerose volte il pattern **Observer** per permettere agli oggetti Observer di essere aggiornati sullo stato di oggetti Observed ai quali si è registrato. Un esempio è il cambiamento di fase innescato dalla pressione del tasto Next da parte del giocatore, la quale scatena `OnNextClicked()` della classe `GUIController`, che a sua volta invoca l'evento `nextClicked()`. A quest'ultimo si è precedentemente registrato il `PhaseManager`, il quale decide con la funzione `ChangePhase()` quale manager di fase far registrare agli eventi generati dagli input del giocatore.