



Basi di Dati  
Progetto A.A. 2021/2022

Risiko

0253048

Abdelrahman Abdelrahman

## Indice

<b>1. Descrizione del Minimondo.....</b>	<b>2</b>
<b>2. Analisi dei Requisiti.....</b>	<b>4</b>
<b>3. Progettazione concettuale.....</b>	<b>11</b>
<b>4. Progettazione logica.....</b>	<b>23</b>
<b>5. Progettazione fisica.....</b>	<b>35</b>
<b>Appendice: Implementazione.....</b>	<b>43</b>

Tutto il testo su sfondo grigio, all'interno di questo template, deve essere eliminato prima della consegna. Viene utilizzato per fornire informazioni sulla corretta compilazione del report di progetto.

Non modificare il formato del documento:

- Carattere: Times New Roman, 12pt
- Dimensione pagina: A4
- Margini: superiore/inferiore 2,5cm, sinistro/destro: 1,9cm

L'assegnazione della tesina può essere effettuata online, visitando il sito <https://www.pellegrini.tk/progetti/> ed inserendo i propri dati. Per qualsiasi problema, contattare il docente via email all'indirizzo [a.pellegrini@ing.uniroma2.it](mailto:a.pellegrini@ing.uniroma2.it)

## 1. Descrizione del Minimondo

Si vuole realizzare un servizio online che consenta di giocare ad un clone del famoso gioco Risiko, in modalità "conquista del mondo".

Al sistema hanno accesso due tipologie di utenti: i giocatori e i moderatori. I moderatori hanno la possibilità di creare stanze di gioco, in funzione della quantità di giocatori che utilizzano attualmente il sistema.

In particolare, i moderatori hanno la possibilità di visualizzare, tramite un report, quante stanze hanno attualmente partite in corso e quanti giocatori stanno partecipando alla partita.

Inoltre, in questo report, gli amministratori possono visualizzare il numero totale di giocatori che hanno effettuato almeno un'operazione negli ultimi 15 minuti che non sono all'interno di alcuna stanza di gioco.

Una stanza permette ad un numero massimo di sei giocatori di entrare e partecipare alla partita.

Una partita coinvolge almeno tre giocatori.

Quando il terzo giocatore entra in una stanza, viene attivato un countdown tale da avviare la partita dopo due minuti. In questi due minuti altri giocatori possono entrare, fino al massimo concesso.

All'avvio della partita, gli stati del tabellone vengono assegnati casualmente ai giocatori. I turni "girano" in funzione del tempo di ingresso dei giocatori nella stanza (chi è entrato prima gioca prima).

Un turno prevede che il giocatore possa compiere una delle seguenti azioni:

- \* Posizionare un numero arbitrario di carri armati in uno stato

- \* Scegliere uno stato da cui fare partire un attacco verso uno stato adiacente

- \* Spostare carri armati da uno stato ad un altro adiacente (almeno un carro armato deve restare nello stato di partenza)

L'attacco viene svolto nel seguente modo.

La fase di attacco si svolge tra il giocatore che attacca e quello che difende attraverso il lancio dei dadi.

Il numero dei dadi da lanciare è stabilito dal numero di armate che si decide di schierare in guerra meno una, fino ad un massimo di tre dadi per volta.

Ognuno dei giocatori lancia il numero di dadi corrispondenti, e poi si confrontano i valori ottenuti, il più alto dell'attaccante con il più alto del difensore, il secondo con il secondo e così via.

Per ogni punteggio più alto, il perdente deve togliere un'armata dal tabellone.

In caso di parità il punto va al difensore. Il lancio dei dadi viene simulato mediante la generazione di numeri pseudocasuali.

Se lo stato attaccato perde tutte le armate, questo viene conquistato e vengono spostate automaticamente in esso un numero di armate pari a quelle sopravvissute all'attacco.

Un apposito timer determina quando il tempo per svolgere un'azione da parte di un giocatore scade e il turno passa quindi al giocatore successivo.

Al termine del turno, se è stata svolta almeno un'azione, il giocatore riceve un numero di carri armati da posizionare pari al numero di stati posseduti diviso tre, arrotondato per eccesso.

Un'apposita procedura consente al client di sapere, quando è il turno del giocatore, tutto lo stato di gioco e quindi far scegliere quale azione effettuare.

Si ricorda, comunque, che tutta la logica applicativa è implementata nel DBMS.

Un giocatore può sempre visualizzare lo storico di tutte le partite giocate.

## 2. Analisi dei Requisiti

### Identificazione dei termini ambigui e correzioni possibili

Linea	Termine	Nuovo termine	Motivo correzione
5,12	Giocatori	Giocatori Attivi	Come vengono menzionati in un contesto che implica che I giocatori coinvolti sono Giocatori attualmente attivi
11	amministratore	Moderatore	Come è stato dichiarato prima come “moderatore”.
13	operazione	Azione Esterna	Come riferisce alle Azioni Esterne (diverse dalle Azioni di Gioco)
17 <sup>(1)</sup>	partita	Lobby (Partita Inattiva)	Come non viene definito quando viene creata la partita, ma vengono fatti i riferimenti alla stessa partita che impone che sia la stessa partita riferita che va in “Countdown” e poi viene “Avviata”. Perciò rimane in modalità “Lobby” ospitata nella stanza di gioco relativa, prima di soddisfare i requisiti per diventare “Partita In Countdown”.
17 <sup>(2)</sup>	partita	Partita In Countdown	Come viene definito che deve coinvolgere almeno 3 giocatori per entrare in “CountDown” per poi essere “Avviata”
9,10	partita	Partita Avviata	Come vengono chiamate “Partite In Corso”, che implica che sono state avviate.
22	Stati del tabellone	Territori	Come rappresentano le istanze delle nazioni assegnate ai giocatori durante una “Partita Avviata” che variano al secondo del contesto della partita.
23	giocatori	Giocatori Partecipanti	Come vengono definiti prima come “giocatori partecipanti stanno partecipando alla partita”.
27,28 <sup>(1)</sup> , 29	stato	Territorio Alleato	Come ogni giocatore può solamente far partire un’azione da un Territorio Alleato (Sotto il suo dominio).
28 <sup>(2)</sup>	stato	Territorio Nemico	Come ogni giocatore può solamente far partire attacchi contro Territorio Nemici (Sotto il dominio di un altro giocatore).
37	numero di armate	Numero Di Carri Armati	Come viene chiamato prima come “Carri Armati”
38	Guerra	Attacco	Come viene chiamata prima “Attacco”
34	Giocatore che attacca	Giocatore Attaccante	Come viene chiamato “Attaccante” dopo <sup>(41)</sup> .
34	Quello che Difende	Difensore	Come viene chiamato “Difensore” dopo <sup>(42)</sup> .
44,49	Armata	Carro Armato	Come viene riferita prima come “Carro Armato” <sup>(30)</sup>
26	Azioni	Azioni di Gioco	Come sono diverse dalle Azioni Esterne e riferisce alle azioni che si possono effettuare dal giocatore durante il suo turno

## Specifica disambiguata

Si vuole realizzare un servizio online che consenta di giocare ad un clone del famoso gioco Risiko, in modalità "conquista del mondo".

Al sistema hanno accesso due tipologie di utenti: i giocatori e i moderatori.

I moderatori hanno la possibilità di creare stanze di gioco, in funzione della quantità di giocatori attivi che utilizzano attualmente il sistema.

In particolare, i moderatori hanno la possibilità di visualizzare, tramite un report, quante stanze hanno attualmente partite avviate e quanti giocatori stanno partecipando alla partita.

Inoltre, in questo report, i moderatori possono visualizzare il numero totale di giocatori attivi che hanno effettuato almeno un'azione esterna negli ultimi 15 minuti che non sono all'interno di alcuna stanza di gioco.

Una stanza permette ad un numero massimo di sei giocatori di entrare e partecipare al Lobby.

Una partita in Countdown coinvolge almeno tre giocatori. Quando il terzo giocatore entra in una stanza, viene attivato un countdown tale da avviare la partita in countdown dopo due minuti.

In questi due minuti altri giocatori possono entrare, fino al massimo concesso.

All'avvio della partita, I territori vengono assegnati casualmente ai giocatori. I turni "girano" in funzione del tempo di ingresso dei giocatori nella stanza (chi è entrato prima gioca prima).

Un turno prevede che il giocatore possa compiere una delle seguenti azioni di gioco:

- \* Posizionare un numero arbitrario di carri armati in un Territorio Alleato
- \* Scegliere uno Territorio Alleato da cui fare partire un attacco verso uno Territorio Nemico adiacente.
- \* Spostare carri armati da uno Territorio Alleato ad un altro adiacente (almeno un carro armato deve restare nello stato di partenza)

L'attacco viene svolto nel seguente modo.

La fase di attacco si svolge tra il giocatore attaccante e quello difensore attraverso il lancio dei dadi.

Il numero dei dadi da lanciare è stabilito dal numero di carri armati che si decide di schierare in attacco meno una, fino ad un massimo di tre dadi per volta.

Ognuno dei giocatori lancia il numero di dadi corrispondenti, e poi si confrontano i valori ottenuti, il più alto dell'attaccante con il più alto del difensore, il secondo con il secondo e così via.

Per ogni punteggio più alto, il perdente deve togliere un carro armato dal tabellone.

In caso di parità il punto va al difensore. Il lancio dei dadi viene simulato mediante la generazione di numeri pseudocasuali.

Se lo stato attaccato perde tutte le armate, questo viene conquistato e vengono spostate automaticamente in esso un numero di carri armati pari a quelli sopravvissuti all'attacco.

Un apposito timer determina quando il tempo per svolgere un'azione da parte di un giocatore scade e il turno passa quindi al giocatore successivo.

Al termine del turno, se è stata svolta almeno un'azione, il giocatore riceve un numero di carri armati da posizionare pari al numero di stati posseduti diviso tre, arrotondato per eccesso.

Un'apposita procedura consente al client di sapere, quando è il turno del giocatore, tutto lo stato di gioco e quindi far scegliere quale azione effettuare. Si ricorda, comunque, che tutta la logica applicativa è implementata nel DBMS.

Un giocatore può sempre visualizzare lo storico di tutte le partite giocate.

## Glossario dei Termini

Realizzare un dizionario dei termini, compilando la tabella qui sotto, a partire dalle specifiche precedentemente disambiguate

Termine	Descrizione	Sinonimi	Collegamenti
Utenti	Indica l'utente generico del sistema		Moderatore, Giocatore
Moderatore	Indica l'utente moderatore del sistema		Utente, Stanza, Giocatore
Giocatore	Indica il giocatore autenticato che usa il sistema		Utente, Stanza, Lobby, Partita in Countdown, Partita Avviata, Turno
Stanza	Indica la Stanza Ospitante dove si svolgono le Partite		Giocatore, Moderatore, Lobby, Partita in Countdown, Partita Avviata,

Partita	Indica la Partita Generica che si svolge in una Stanza Ospitante		Giocatore, Stanza, Lobby, Partita in Countdown, Partita Avviata, Partita Giocata
Lobby	Indica la Partita Inattiva che si svolge in una Stanza che poi va in Countdown	Partita Inattiva	Giocatore, Stanza, Partita, Partita in Countdown
Partita in Countdown	Indica il Lobby che ha (Min 3, Max 6) Giocatori iscritti, che va in Countdown per 2 Minuti prima di avviarsi		Giocatore, Stanza, Partita, Lobby, Partita Avviata
Partita Avviata	Indica la Partita Avviata dopo un Countdown di 2 Minuti		Moderatore, Giocatore, Stanza, Partita, Partita in Countdown, Partita Giocata, Territori, Turni
Partita Giocata	Indica le Partite terminate a cui il Giocatore ha partecipato	Partita Terminata	Giocatore, Partita, Partita Avviata, Territori
Turno	Indica il Turno attuale durante la Partita		Giocatore, Partita Avviata, Partita Giocata, Azione Di Gioco
Azione di Gioco	Indica l'Azione di Gioco che il Giocatore può svolgere durante il suo Turno (Posizionamento, Attacco e Spostamento)		Giocatore, Partita Avviata, Turno
Nazione	Indica i stati definiti dalle Regole del Gioco	Stati	Partita Avviata, Territori
Territori	Indica lo stato attuale del Tabellone durante la Partita con la relazione tra le Nazioni, Giocatori e I Carri Armati corrispondenti	Tabellone	Giocatore, Partita Avviata, Azione Di Gioco, Nazione
Lancio di Dadi	Indica un Lancio di Dadi che decide il risultato di un'Attacco durante un Turno		Giocatore, Partita Avviata, Turno, Azione Di Gioco

## Raggruppamento dei requisiti in insiemi omogenei

Per ciascun elemento “più importante” della specifica (riportata anche nel glossario precedente), estrapolare dalla specifica disambiguata le frasi ad esso associate. Compilare una tabella separata per ciascun elemento individuato.

### Utente

Al sistema hanno accesso due tipologie di utenti: i giocatori e i moderatori.

### Moderatore

I moderatori hanno la possibilità di creare stanze di gioco, in funzione della quantità di giocatori autenticati che utilizzano attualmente il sistema.

In particolare, i moderatori hanno la possibilità di visualizzare, tramite un report, quante stanze hanno attualmente partite avviate e quanti giocatori stanno partecipando alla partita.

Inoltre, in questo report, i moderatori possono visualizzare il numero totale di giocatori che hanno effettuato almeno un'azione esterna negli ultimi 15 minuti che non sono all'interno di alcuna stanza di gioco.

### Giocatore

Una stanza permette ad un numero massimo di sei giocatori di entrare e partecipare al Lobby.

Una partita in Countdown coinvolge almeno tre giocatori.

I turni "girano" in funzione del tempo di ingresso dei giocatori nella stanza (chi è entrato prima gioca prima).

Un turno prevede che il giocatore possa compiere una delle seguenti azioni di gioco:

- \* Posizionare un numero arbitrario di carri armati in un Territorio Alleato
- \* Scegliere uno Territorio Alleato da cui fare partire un attacco verso uno Territorio Nemico adiacente.
- \* Spostare carri armati da uno Territorio Alleato ad un altro adiacente (almeno un carro armato deve restare nello stato di partenza)

La fase di attacco si svolge tra il giocatore attaccante e quello difensore attraverso il lancio dei dadi.

In caso di parità il punto va al difensore.

Al termine del turno, se è stata svolta almeno un'azione, il giocatore riceve un numero di carri armati da posizionare pari al numero di stati posseduti diviso tre, arrotondato per eccesso.

Un'apposita procedura consente al client di sapere, quando è il turno del giocatore, tutto lo stato di gioco e quindi far scegliere quale azione effettuare.

Un giocatore può sempre visualizzare lo storico di tutte le partite giocate.



**Stanza**

Una stanza permette ad un numero massimo di sei giocatori di entrare e partecipare al Lobby.

Una partita in Countdown coinvolge almeno tre giocatori. Quando il terzo giocatore entra in una stanza, viene attivato un countdown tale da avviare la partita in countdown dopo due minuti.

All'avvio della partita, I territori vengono assegnati casualmente ai giocatori. I turni "girano" in funzione del tempo di ingresso dei giocatori nella stanza (chi è entrato prima gioca prima).

**Lobby**

Una stanza permette ad un numero massimo di sei giocatori di entrare e partecipare al Lobby.

**Partita In Countdown**

Una partita in Countdown coinvolge almeno tre giocatori. Quando il terzo giocatore entra in una stanza, viene attivato un countdown tale da avviare la partita in countdown dopo due minuti.

In questi due minuti altri giocatori possono entrare, fino al massimo concesso.

**Partita Avviata**

All'avvio della partita, I territori vengono assegnati casualmente ai giocatori. I turni "girano" in funzione del tempo di ingresso dei giocatori nella stanza (chi è entrato prima gioca prima).

**Partita Giocata**

Un giocatore può sempre visualizzare lo storico di tutte le partite giocate.

**Turno**

I turni "girano" in funzione del tempo di ingresso dei giocatori nella stanza (chi è entrato prima gioca prima).

Un turno prevede che il giocatore possa compiere una delle seguenti azioni di gioco:

- \* Posizionare un numero arbitrario di carri armati in un Territorio Alleato
- \* Scegliere uno Territorio Alleato da cui fare partire un attacco verso uno Territorio Nemico adiacente.
- \* Spostare carri armati da uno Territorio Alleato ad un altro adiacente (almeno un carro armato deve restare nello stato di partenza)

Un apposito timer determina quando il tempo per svolgere un'azione da parte di un giocatore scade e il turno passa quindi al giocatore successivo.

Al termine del turno, se è stata svolta almeno un'azione, il giocatore riceve un numero di carri armati da posizionare pari al numero di stati posseduti diviso tre, arrotondato per eccesso.

**Azione Di Gioco**

Un turno prevede che il giocatore possa compiere una delle seguenti azioni di gioco:

- \* Posizionare un numero arbitrario di carri armati in un Territorio Alleato
- \* Scegliere uno Territorio Alleato da cui fare partire un attacco verso uno Territorio Nemico adiacente.
- \* Spostare carri armati da uno Territorio Alleato ad un altro adiacente (almeno un carro armato deve restare nello stato di partenza)

L'attacco viene svolto nel seguente modo. La fase di attacco si svolge tra il giocatore attaccante e quello difensore attraverso il lancio dei dadi.

Se lo stato attaccato perde tutte le armate, questo viene conquistato e vengono spostate automaticamente in esso un numero di carri armati pari a quelli sopravvissuti all'attacco.

**Territori**

All'avvio della partita, I territori vengono assegnati casualmente ai giocatori.

Se lo stato attaccato perde tutte le armate, questo viene conquistato e vengono spostate automaticamente in esso un numero di carri armati pari a quelli sopravvissuti all'attacco.

Ognuno dei giocatori lancia il numero di dadi corrispondenti, e poi si confrontano i valori ottenuti, il più alto dell'attaccante con il più alto del difensore, il secondo con il secondo e così via.

Per ogni punteggio più alto, il perdente deve togliere un carro armato dal tabellone.

**Lancio Di Dadi**

La fase di attacco si svolge tra il giocatore attaccante e quello difensore attraverso il lancio dei dadi.

Il numero dei dadi da lanciare è stabilito dal numero di carri armati che si decide di schierare in attacco meno una, fino ad un massimo di tre dadi per volta.

Ognuno dei giocatori lancia il numero di dadi corrispondenti, e poi si confrontano i valori ottenuti, il più alto dell'attaccante con il più alto del difensore, il secondo con il secondo e così via.

In caso di parità il punto va al difensore.

Il lancio dei dadi viene simulato mediante la generazione di numeri pseudocasuali.

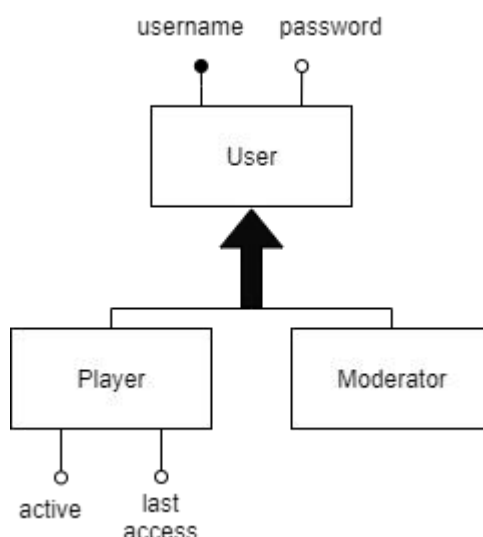
### 3. Progettazione concettuale

#### Costruzione dello schema E-R

In questa sezione è necessario riportare tutti i passi seguiti per la costruzione dello schema E-R finale, a partire dalle specifiche raccolte ed organizzate nel capitolo precedente. Non è richiesto un procedimento specifico: si può adottare una strategia top-down, bottom-up, a macchia d'olio o mista. L'importante è descrivere e commentare tutti i passi della costruzione, andando anche ad inserire "schemi parziali" utilizzati nel processo.

Ho usato un approccio misto che varia al secondo del concetto da rappresentare, e per mostrare concetti e possibili Design Pattern modo più semplice possibile:

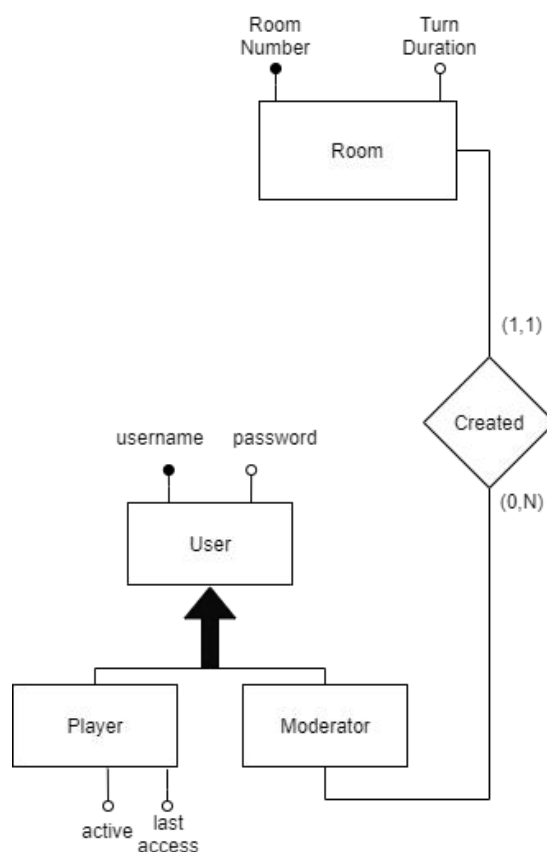
#### USER (Utente)



Nel testo si parte con i **User** e le due tipologie definite dal testo che sono **Player** (Giocatore) e **Moderator** (Moderatore).

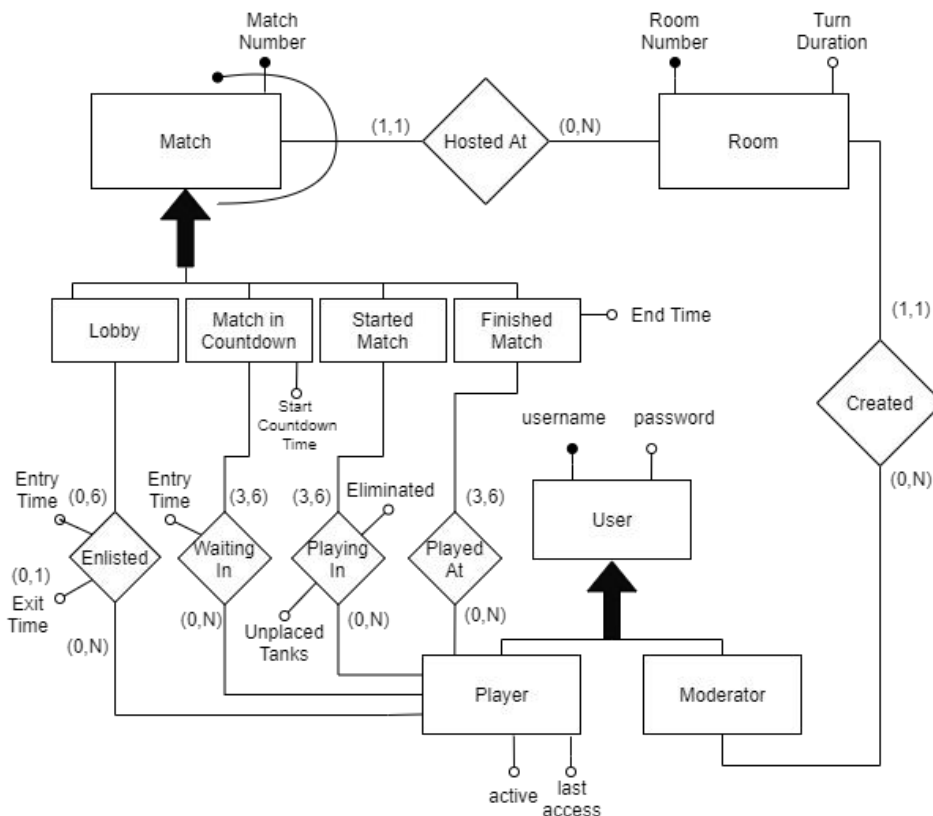
Entrambi i tipi di **User** hanno in comune (*Username*) e (*Password*) che vengono usati per l'autenticazione al sistema.

Un **Player** ha in più gli attributi (*active*) e (*last access*), come viene chiesto nel testo che il **Moderator** abbia la possibilità di visualizzare dei report che sono basati su il numero di **Player** attivi ed anche quando quelli **Player** hanno fatto l'accesso, come l'accesso viene considerato un'azione esterna importante per uno dei report.

**Room** (Stanza)

Nel testo viene definito che i **Moderator** possono creare dei **Room** che ospita le partite, e su cui si basa la lunghezza massima di un turno al secondo del *Turn Duration* di quella **Room**, poi le **Room** vengono identificate tramite il numero della stanza *Room Number*.

Nota: Non c'è una relazione che collega direttamente **Player** e **Room** come la **Room** viene trattata come l'arena dove si svolgono le partite perciò è più chiaro trattarla come un'entità statica su cui si basano e vengono ospitate le partite che sono dinamiche.

**Match** (Partita)

È logico che il **Player** sia collegato ad un **Match** una volta che è entrato in una **Room** anche se quella **Match** non è ancora avviata come non viene definito quando un **Match** viene creato perciò possiamo supporre che una volta una nuova **Room** viene creata o un **Match** si conclude, un altro **Match** di tipo **Lobby** viene ospitato in quella **Room** per accogliere i nuovi **Player**.

Come ogni **Match** deve essere ospitato in una **Room**, allora tutti i **Match** sono entità deboli verso **Room**.

Come nel testo viene descritto il cambiamento dello stato di un **Match** al secondo dei **Player** partecipanti e l'evoluzione del **Match** da uno stato all'altro; Il modo più chiaro per rappresentare i requisiti per ogni stato di **Match** tramite il diagramma ER e anche il pattern di Evoluzione è tramite la generalizzazione del **Match** e la cardinalità che varia al secondo dei requisiti per quello stato e la relazione che contiene i dati relativi a quello stato.

Tutti i stati del **Match** sono identificati tramite il *Match Number* (Candidato Per Chiave Primaria) ed anche il *Room Number* della **Room** dove si svolge la partita.

L'evoluzione di un **Match** è il seguente:

**1-Lobby: Match** che non è ancora avviata, e non ha ancora la possibilità di andare in Countdown (Ha meno di 3 **Player**) come viene descritto nella specifica.

Nella relazione tra **Lobby** e **Player** (*Enlisted*), si tiene traccia di quando un giocatore è entrato (*Entry Time*) e quando è uscito se il **Player** decide di uscire dalla **Room** prima che il **Match** va in Countdown (*Exit Time*).

**2-Match in Countdown: Match** che ha almeno 3 **Player** e può andare in Countdown per avviarsi dopo 2 Minuti del *Start Countdown Time*.

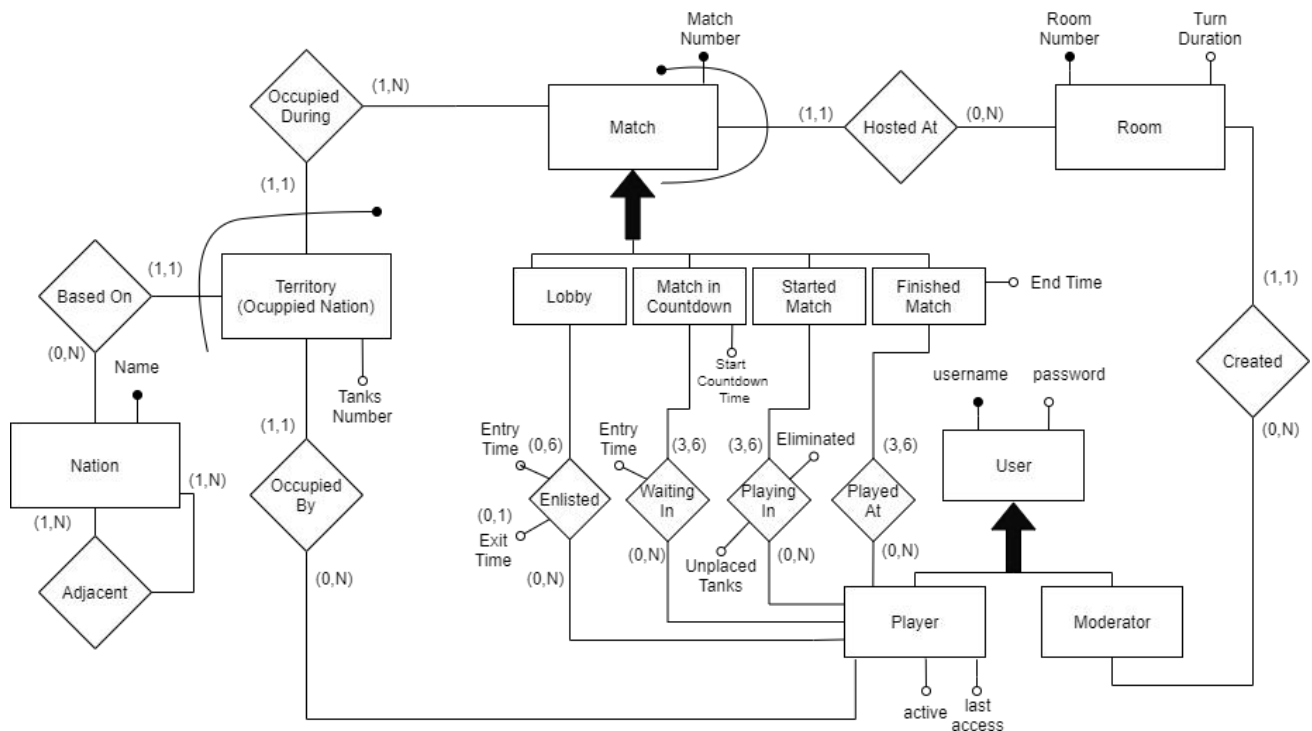
Nella relazione tra **Match in Countdown** e **Player** (*Waiting in*), si tiene traccia di quando un giocatore è entrato (*Entry Time*) prima che il **Match** viene avviato.

**2-Started Match: Match** avviato che ha 3-6 **Player**.

In questo stato l'importante è tenere traccia dei dati di gioco non i dati della partecipazione come nei stati precedenti perciò; Nella relazione tra **Started Match** e **Player** (*Playing In*), si tiene traccia dei carri armati non posizionati su nessun Territorio (*Unplaced Tanks*) e se il **Player** è stato eliminato o meno (*Eliminated*).

**3-Finished Match: Match** che è stato concluso e ha in più l'attributo (*End Time*) che tiene traccia di quando una partita è stata conclusa.

## Territory (Stati del tabellone)



Come il gioco è basato sul gioco *Risiko*, allora ci sono 42 stati in ogni **Match**, ed ogni tabellone deve tenere traccia di tutti i quarantadue.

Il concetto dei stati presenti nel regole del gioco viene incapsulato nell'entità **Nation**, come tutte le **Nation** sono unici basta avere il nome come Identificatore (*Name*).

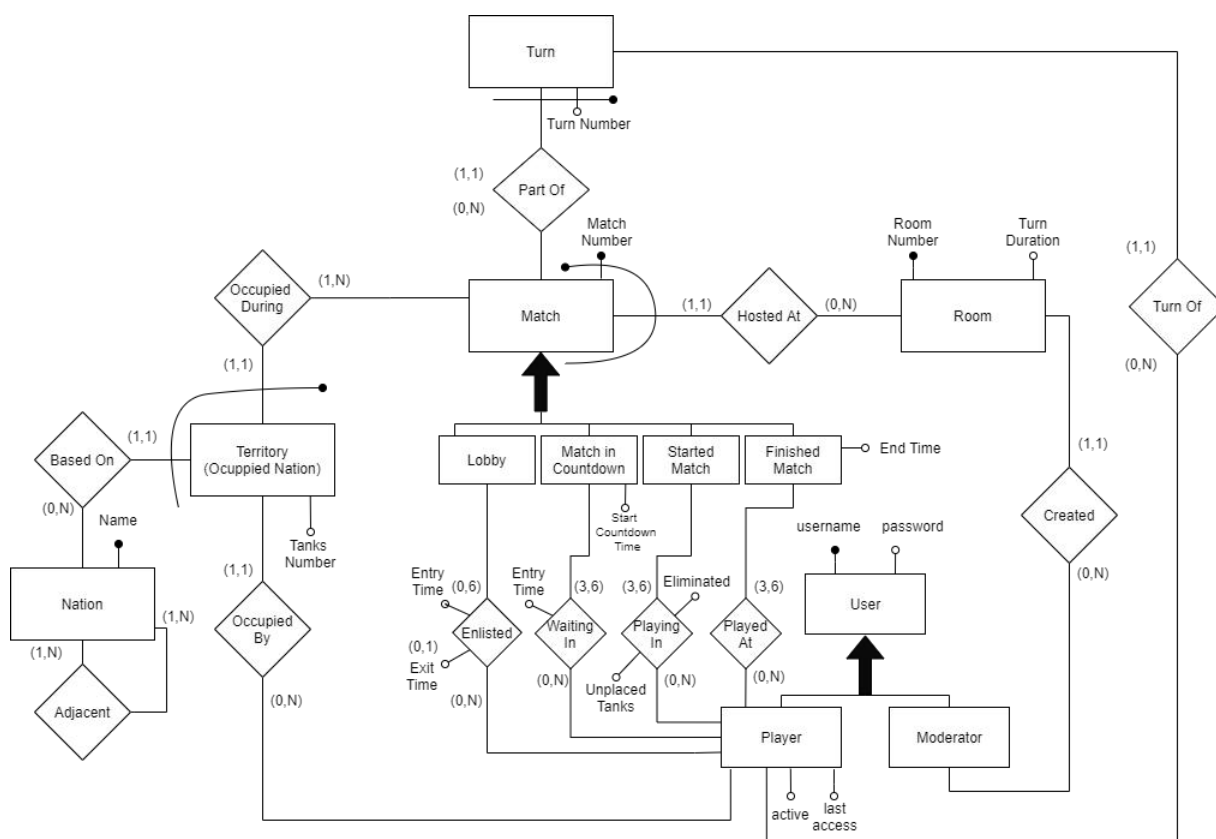
Secondo le regole del gioco ogni **Nation** è adiacente ad almeno un'altra **Nation**, perciò si realizza la relazione ricorsiva (*Adjacent*) tra **Nation** e se stessa.

Nel testo viene specificato che ad ogni **Player** che partecipa ad un **Started Match** ha almeno un **Territory** assegnato all'inizio del **Match**, e il contrario implica che il **Player** è stato eliminato.

Come **Territory** rappresenta il tabellone per una partita con ogni **Nation** presente una ed una sola volta per partita e cambia solamente il **Player** che la occupa e il numero di carri armati attualmente su quel **Territory** (*Tanks Number*).

L'entità **Territory** viene rappresentata come entità debole che dipende dalla **Nation** e **Match**, come un **Territory** può esistere una sola volta per **Match**.

## Turn (Turno)



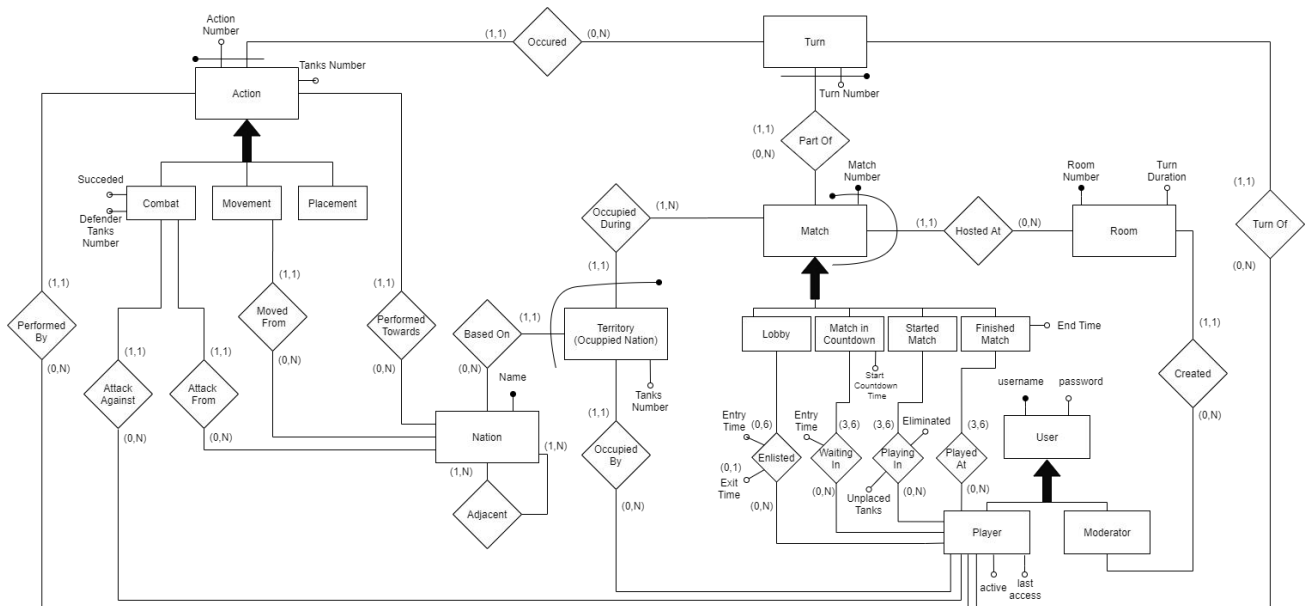
Nella specifica viene definito che ogni **Turn** ha un solo **Player** che può fare un'azione durante quel **Turn**.

Ogni **Turn** è unico per partita, perciò i **Turn** vengono rappresentati come entità deboli che vengono identificati tramite il **Match** relativo e il numero del turno (*Turn Number*).

Ogni **Turn** ha il tempo dell'inizio del turno (*Turn Start Time*) per riconoscere i **Turn** che vanno in Timeout.



## Action (Azione di Gioco)



Nella specifica viene definito che ogni **Action** è unica per **Turn**, viene effettuata da un **Player** verso una **Nation** e coinvolge almeno un carro armato; Queste specifiche sono in comune tra tutti i possibili **Action** perciò vengono rappresentati come:

- *Tanks Number*: Il numero di carri armati coinvolti in quel **Action**.
- *Action Number*: Il numero del **Action** durante quel **Turn**, e viene usato in combinazione con gli identificatori di **Turn** e **Match** per identificare un **Action** come è un'entità debole.
- *Performed Towards*: La relazione rappresenta un **Action** e la **Nation** verso cui è stato fatto quel **Action**.
- *Performed By*: La relazione rappresenta un **Action** e il **Player** che ha effettuato quel **Action**.

Nella specifica vengono definiti diversi tipi di **Action** che si possono fare durante un **Turn**:

- **Placement** (Posizionamento) dei carri armati su una **Nation**, e viene reindirizzato tramite *Tank Number* in **Territory** che rappresenta lo stato attuale.

> **Placement** non richiede altri dati aggiuntivi per rappresentare quel **Action**.

- **Movement** (Spostamento) dei carri armati da una **Nation** ad un'altra **Nation** adiacente.

> **Movement** richiede in più la relazione *Moved From* tra **Movement** e **Nation** per rappresentare da quale **Nation** parte quel **Movement**.

- **Combat** (Attaccare) verso da una **Nation** verso un'altra adiacente.

> **Combat** è l'unico **Action** che richiede multipli attributi e relazioni aggiutivi:

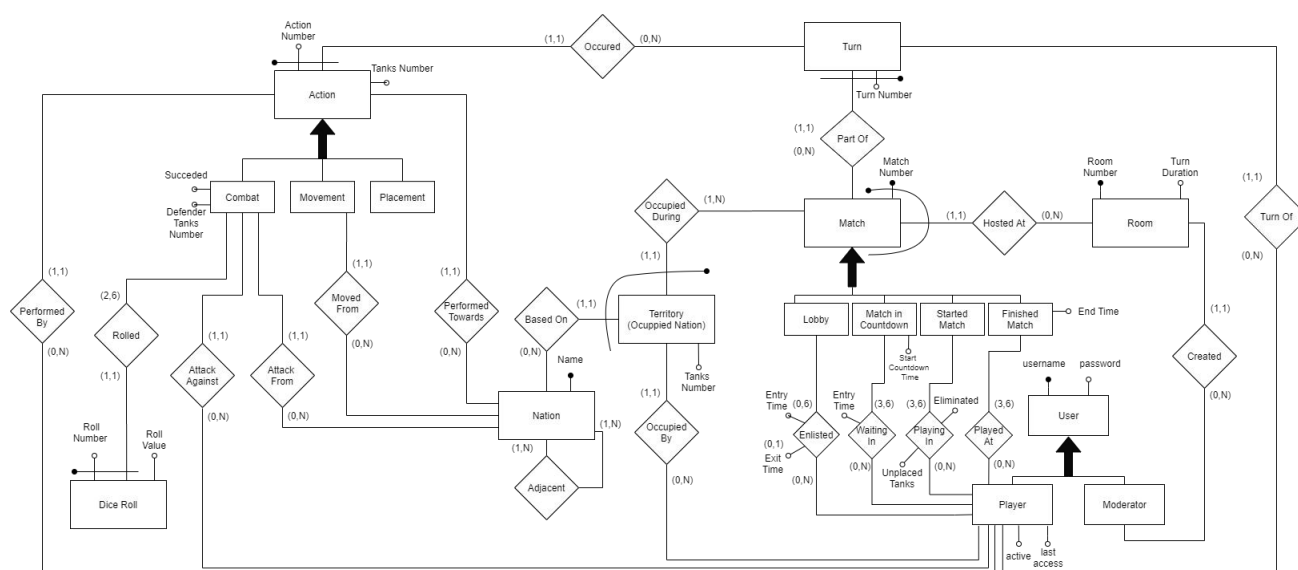
>> *Succeded*: Rappresenta se quel **Combat** è riuscito o meno.

>> *Defender Tanks Number*: Rappresenta il numero di carri armati in difesa contro quel **Combat**.

>> *Attack From*: La relazione rappresenta la **Nation** da parte il **Combat**.

>> *Attack Against*: La relazione rappresenta il **Player** contro cui pare il **Combat**.  
(**Player** Difensore)

## Dice Roll (Lancio dei dadi)



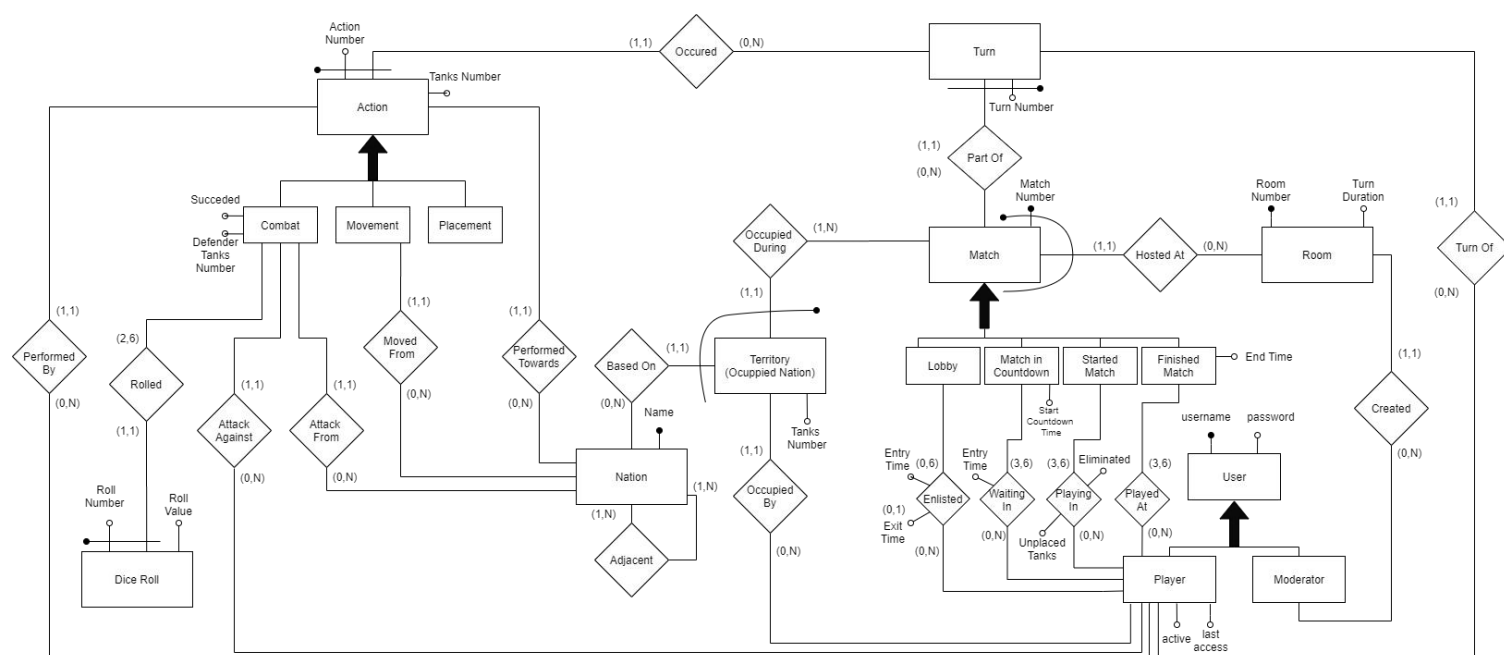
L'ultimo concetto mancante per completare la rappresentazione del **Action** di tipo **Combat** è il **Dice Roll**.

Tutti i **Combat** vengono decisi tramite i **Dice Roll**, ogni **Combat** ha *almeno due Dice Roll* (Un **Dice Roll** per **Player** [1 **Player** Attaccante, 1 **Player** Difensore] ) e *al massimo sei Dice Roll* associati (Tre **Dice Roll** per **Player** [3 **Player** Attaccante, 3 **Player** Difensore] ) e questo concetto viene rappresentato tramite la cardinalità della relazione *Rolled*.

Ogni **Dice Roll**, è un'entità debole verso il **Combat** relativo e viene identificato tramite gli identificatori del **Combat** e il numero del dado lanciato (*Roll Number*).

Ogni **Dice Roll**, ha un valore (*Roll Value*) che rappresenta il numero uscito dal lancio di quel dado.

## Integrazione finale



## Regole aziendali

*Regole di vincolo:-*

- Un Player non può partecipare ad un Match che ha già 6 partecipanti
- Ogni Match in Countdown deve avviarsi dopo 2 minuti dall'inizio del tempo di Countdown
- I turn devono iniziare da 1 per ogni Match distinto
- Si può entrare in una Room solo quando il Match sta in stato di Lobby o Countdown
- Si può uscire da una Room solo quando il Match è in stato di Lobby o dopo

l'eliminazione

- Il sistema deve far partire un timer all'inizio di ogni turn per saltarlo se il turn va in timeout
- Un Lobby viene ospitato subito in una Room alla creazione della Room o subito dopo la conclusione dell'ultimo Match precedentemente ospitato in quella Room
- Ogni Player deve aggiornare lo stato di attività e l'ultimo accesso ad ogni accesso al sistema
- Un Match non può avere nessun Turn affinché non è arrivato allo stato di Started Match.
- Ogni Match ha 42 Territory che riflettono lo stato attuale del tabellone

*Regole di derivazione:-*

- Le informazioni di un Player che sta in difesa si ottiene dal Combat contro esso
- Il vincitore di un Match si ottiene cercando il Player che occupa 42 Territory o sottraendo tutti i Player eliminati da quelli partecipanti
- La fine di un Match si ottiene quando tutti i Player sono stati eliminati tranne uno
- Il timeout di un Turn si rileva controllando se  $(Turn\ Start\ Time + Turn\ Duration)$  è nel passato
- Il numero di una Room si ottiene sommando il numero di Player che partecipano al Match ospitato in quella Room

## Dizionario dei dati

Completare la progettazione concettuale riportando nella tabella seguente il dizionario dei dati

Entità	Descrizione	Attributi	Identificatori
<b>User</b>	Rappresenta un utente del sistema	Username, Password	Username
<b>Player</b>	Rappresenta un giocatore	Active, Last Access	Username
<b>Moderator</b>	Rappresenta un moderatore del sistema		Username
<b>Room</b>	Rappresenta una stanza creata dal moderatore e che ospita le partite	Room Number, Turn Duration	Room Number
<b>Match</b>	Rappresenta una partita generica	Match Number	Match Number, Room Number
<b>Lobby</b>	Rappresenta la partita in stato di Lobby che permette ai utenti di entrare ed uscire		Match Number, Room Number
<b>Match in Countdown</b>	Rappresenta la partita in stato di Countdown quando ha almeno 3 giocatori e ha fatto partire il Countdown per l'avvio della partita	Start Countdown Time	Match Number, Room Number
<b>Started Match</b>	Rappresenta la partita avviata		Match Number, Room Number
<b>Finished Match</b>	Rappresenta la partita terminata	End Time	Match Number, Room Number
<b>Nation</b>	I stati del gioco definiti dalle regole del gioco	Name	Name
<b>Territory</b>	Rappresenta lo stato attuale del tabellone	Tanks Number	Name, Match Number, Room Number
<b>Turn</b>	Rappresenta un turno durante una partita	Turn Number, Turn Start Time	Turn Number, Match Number, Room Number
<b>Action</b>	Rappresenta un'azione accaduta durante un turno	Action Number, Tanks Number	Action Number, Turn Number, Match Number, Room Number
<b>Combat</b>	Rappresenta un'azione di combattimento da giocatore contro un'altro	Succeded, Defender Tanks Number	Action Number, Turn Number, Match Number, Room Number
<b>Movement</b>	Rappresenta un'azione di movimento di carri armati da uno stato ad un'altro		Action Number, Turn Number, Match Number, Room Number
<b>Placement</b>	Rappresenta un'azione di posizionamento di carri armato su uno stato		Action Number, Turn Number, Match Number, Room Number

<b>Dice Roll</b>	Rappresenta un lancio di dado durante un combattimento	Roll Number, Roll Value	Roll Number, Action Number, Turn Number, Match Number, Room Number
------------------	--	----------------------------	--

## 4. Progettazione logica

### Volume dei dati

Questa sezione serve ad illustrare qual è il carico che la base di dati dovrà sopportare. A tal fine, è necessario prevedere un volume di dati attesi. Compilare la tabella sottostante, per ciascun concetto identificato nello schema E-R. I volumi devono essere stimati dallo studente in maniera ragionevole rispetto all'operatività presunta dell'applicativo.

Concetto nello schema	Tipo <sup>1</sup>	Volume atteso
User	E	40.500
Player	E	40.000
Moderator	E	500
Room	E	2000 (Ogni Moderator ha creato 4 Room)
Match	E	15.000 (La maggioranza dei Match è composta da 4 giocatori)
Lobby	E	15.000
Match in Countdown	E	14.000
Started Match	E	14.000
Finished Match	E	13.500
Nation	E	42
Territory	E	588.000
Turn	E	750.000 (Ogni Match in media è composto da 50 turni)
Action	E	740.000 (10.000 Turni sono stati fatti senza nessuna azione)
Placement	E	300.000
Movement	E	240.000
Combat	E	200.000
Dice Roll	E	480.000 10% 1 dado lanciato (20.000) 40% 2 dadi lanciati (160.000) 50% 3 dadi lanciati (300.000)
Created	R	2000
Hosted At	R	15.000
Enlisted	R	64.000 (Tra cui 4000 che sono entrati ed usciti prima del Countdown)
Waiting In	R	62.000 (Tra cui 2000 sono entrati durante il Countdown)

<sup>1</sup> Indicare con E le entità, con R le relazioni

Playing In	R	62.000
Played At	R	62.000
Occupied During	R	588.000
Based On	R	588.000
Occupied By	R	588.000
Adjacent	R	163
Turn Of	R	750.000
Occurred	R	740.000
Performed Towards	R	740.000
Performed By	R	740.000
Moved From	R	240.000
Attack From	R	200.000
Attack Against	R	200.000
Rolled	R	480.000

## Tavola delle operazioni

Rappresentare nella tabella sottostante tutte le operazioni sulla base di dati che devono essere supportate dall'applicazione, con la frequenza attesa. Le operazioni da supportare devono essere desunte dalle specifiche raccolte.

Cod.	Descrizione	Frequenza attesa
01C	Effettuare Login	5000 al giorno (Supponendo che ci sono 5000 User attivi )
02M	Creare una Room e il suo Lobby	100 al mese (Può variare drasticamente in caso di picco nella quantità dei nuovi Player)
03M	Visualizzare i Match avviati e quanti Player stanno partecipando ad essi	10 al giorno
04M	Visualizzare il numero totale di Player che hanno fatto un'azione esterna negli ultimi 15 minuti	5 al giorno
05P	Entrare in una Room ospitante ad un Lobby	10.000 al giorno
06P	Entrare in una Room ospitante ad un Match in Countdown	2000 al giorno
07P	Uscire da una Room	1000 al giorno (Supponendo che il 10% dei Player che entrano in Room escono prima del Countdown)
08P	Visualizzare lo storico dei Match giocati	1000 a settimana
09P	Visualizzare le Room a cui il Player può entrare	12.000 al giorno
10P	Posizionare carri armati su un Territory	25.000 al giorno
11P	Spostare carri armati da un Territory ad un'altro	15.000 al giorno
12P	Attaccare un altro Territory	20.000 al giorno
13P	Lanciare un dado	40.000 al giorno (Supponendo che la maggioranza delle volte servono solo 2 Dadi)
14P	Effettuare Logout	4500 al giorno



15P	Aggiornare lo stato di attività	9000 al giorno (Ogni volta che l'operazione 01C o 13P viene effettuata da un Player)
16S	Saltare un Turn a causa di Timeout	5000 al giorno (Supponendo che 5000 turni vanno in timeout)
17S	Assegnare i Territory casualmente ai Player partecipanti	2500 al giorno (Supponendo che la maggioranza dei Match sia fatta da 4 Player)
18S	Avviare un Match dopo Countdown	2500 al giorno
19S	Concludere un Match	2500 al giorno
20S	Creare un Lobby in una Room	2500 al giorno (Ogni volta che l'operazione 17S eseguita)
21S	Concludere un Turn e iniziarne un altro	60.000 al giorno (Ogni volta che un Turn non va in Timeout eseguendo una delle operazioni 10P, 11P o 12P)

## Costo delle operazioni

In riferimento a tutte le operazioni precedentemente indicate, calcolarne il costo supponendo, per questa fase del progetto, che il costo in scrittura di un dato sia doppio rispetto a quello in lettura.

Operazione 01C → costo 1			
Concetto	Costrutto	Accessi	Tipo
User	E	1	L
Operazione 02M → costo 4			
Concetto	Costrutto	Accessi	Tipo
Room	E	1	S
Lobby	E	1	S
Operazione 03M → costo 28.000			
Concetto	Costrutto	Accessi	Tipo
Started Match	E	14.000	L
Playing In	R	14.000	L
Operazione 04M → costo 120.000			
Concetto	Costrutto	Accessi	Tipo
Player	E	40.000	L
Enlisted	R	40.000	L
Waiting In	R	40.000	L
Operazione 05P → costo 4			
Concetto	Costrutto	Accessi	Tipo
Room	E	1	L
Lobby	E	1	L
Enlisted	R	1	S
Operazione 06P → costo 4			
Concetto	Costrutto	Accessi	Tipo
Room	E	1	L
Match in Countdown	E	1	L

Waiting In	R	1	S
<b>Operazione 07P→ costo 4</b>			
<b>Concetto</b>	<b>Costrutto</b>	<b>Accessi</b>	<b>Tipo</b>
Room	E	1	L
Lobby	E	1	L
Enlisted	E	1	S
<b>Operazione 08P→ costo 64.000</b>			
<b>Concetto</b>	<b>Costrutto</b>	<b>Accessi</b>	<b>Tipo</b>
Played At	R	64.000	L
<b>Operazione 09P→ costo 10.000</b>			
<b>Concetto</b>	<b>Costrutto</b>	<b>Accessi</b>	<b>Tipo</b>
Room	E	2000	L
Lobby	E	2000	L
Enlisted	R	2000	L
Match in Countdown	E	2000	L
Waiting In	R	2000	L
<b>Operazione 10P→ costo 11</b>			
<b>Concetto</b>	<b>Costrutto</b>	<b>Accessi</b>	<b>Tipo</b>
Playing In	R	1	L
Territory	E	1	S
Placement	E	1	S
Performed Towards	E	1	S
Performed By	E	1	S
Playing In	R	1	S
<b>Operazione 11P→ costo 5</b>			
<b>Concetto</b>	<b>Costrutto</b>	<b>Accessi</b>	<b>Tipo</b>
Territory	E	1	L
Movement	E	1	S
Moved From	R	1	S
<b>Operazione 12P→ costo 11</b>			
<b>Concetto</b>	<b>Costrutto</b>	<b>Accessi</b>	<b>Tipo</b>
Territory	E	1	L
Attack From	E	1	S
Attack Against	E	1	S
Performed Towards	R	1	S
Performed By	R	1	S
Combat	E	1	S
<b>Operazione 13P→ costo 4</b>			
<b>Concetto</b>	<b>Costrutto</b>	<b>Accessi</b>	<b>Tipo</b>
Dice Roll	E	1	S
Rolled	R	1	S
<b>Operazione 14P→ costo 2</b>			
<b>Concetto</b>	<b>Costrutto</b>	<b>Accessi</b>	<b>Tipo</b>
Player	E	1	S
<b>Operazione 15P→ costo 2</b>			
<b>Concetto</b>	<b>Costrutto</b>	<b>Accessi</b>	<b>Tipo</b>
Player	E	1	S
<b>Operazione 16S→ costo 4</b>			

Concetto	Costrutto	Accessi	Tipo
Started Match	E	1	L
Turn	E	1	L
Turn	E	1	S
<b>Operazione 17S→ costo 300</b>			
Concetto	Costrutto	Accessi	Tipo
Nation	E	42	L
Playing In	R	6	L
Territory	E	42	S
Occupied During	R	42	S
Occupied By	R	42	S
<b>Operazione 18S→ costo 2</b>			
Concetto	Costrutto	Accessi	Tipo
Started Match	E	1	S
Playing In	R	6	S
<b>Operazione 19S→ costo 2</b>			
Concetto	Costrutto	Accessi	Tipo
Finished Match	E	1	S
Lobby	E	1	S
<b>Operazione 20S→ costo 3</b>			
Concetto	Costrutto	Accessi	Tipo
Room	E	1	L
Lobby	E	1	S
<b>Operazione 21S→ costo 5</b>			
Concetto	Costrutto	Accessi	Tipo
Turn	E	1	L
Turn	E	1	S
Turn Of	R	1	S

- |                                    |                             |
|------------------------------------|-----------------------------|
| 1. Costo 1, Al giorno 5.000        | Costi al giorno 5.000       |
| 2. Costo 4, Al mese 100            | Costi al mese 400           |
| 3. Costo 28.000, Al giorno 10      | Costi al giorno 280.000     |
| 4. Costo 120.000, Al giorno 5      | Costi al giorno 600.000     |
| 5. Costo 4, Al giorno 10.000       | Costi al giorno 40.000      |
| 6. Costo 4, Al giorno 2.000        | Costi al mese 8.000         |
| 7. Costo 4, Al giorno 1.000        | Costi al giorno 70          |
| 8. Costo 64.000, A Settimana 1.000 | Costi al giorno 460         |
| 9. Costo 10.000, Al giorno 12.000. | Costi al giorno 120.000.000 |
| 10. Costo 11, Al giorno 25.000     | Costi al giorno 275.000     |
| 11. Costo 5, Al giorno 15.000      | Costi al giorno 75.000      |
| 12. Costo 11, Al giorno 20.000     | Costi al giorno 220.000     |
| 13. Costo 4, Al giorno 40.000      | Costi al giorno 160.000     |
| 14. Costo 2, Al giorno 4.500       | Costi al giotno 9.000       |
| 15. Costo 2, Al giorno 9.000       | Costi al giorno 18.000      |
| 16. Costo 4, Al giorno 5.000       | Costi al giorno 20.000      |
| 17. Costo 300, Al giorno 2.500     | Costi al giorno 750.000     |
| 18. Costo 2, Al giorno 2.500       | Costi al giorno 5.000       |
| 19. Costo 2, Al giorno 2.500       | Costi al giorno 5.000       |
| 20. Costo 3, Al giorno 2.500       | Costi al giorno 7.500       |

21. Costo 5, Al giorno 60.000

Costi al giorno 300.000

## Ristrutturazione dello schema E-R

Descrivere (laddove necessario fornendo anche degli schemi) quali passi vengono adottati per ristrutturare lo schema E-R, ad esempio in termini di:

- 1) Analisi delle ridondanze
- 2) Eliminazione delle generalizzazioni
- 3) Scelta degli identificatori primari

Si noti che in questa fase è possibile fare riferimento al costo delle operazioni precedentemente realizzato per guidare le scelte. Ad esempio, un leggero spreco di memoria legato alla non rimozione di ridondanze può essere facilmente giustificato da un guadagno in termini di prestazioni.

### Analisi delle ridondanze:

- [09P] Visualizzare le Room a cui il Player può entrare ; Accessi:12.000

[Senza dato ridondante: Numero Di Partecipanti] Costo: 10.000

[Con dato ridondante: Numero Di Partecipanti] Costo: 6.000

Operazione 09P→ costo 6.000			
Concetto	Costrutto	Accessi	Tipo
Room	E	2000	L
Lobby	E	2000	L
Match in Countdown	E	2000	L

- [03M] Visualizzare i Match avviati e quanti Player stanno partecipando ad essi; Accessi: 10

[Senza dato ridondante: Numero Di Partecipanti] Costo: 28.000

[Con dato ridondante: Numero Di Partecipanti] Costo: 14.000

\*Riduzione del costo del 50%

Operazione 03M→ costo 14.000			
Concetto	Costrutto	Accessi	Tipo
Started Match	E	14.000	L

## Eliminazione delle generalizzazioni:

### 1- Generalizzazione dell'entità User e le sue entità figlie

Come l'entità **Player** ha in più gli attributi (*active*, *last access*) che sono utili per il reintracciamento dell'attività del **Player**. E Come il numero di **Moderator** è trascurabile confrontato con il numero di **Player**, e come i due attributi presentano una ridondanza nel caso dei **Moderator** ma hanno un valore costruttivo più che distruttivo (Possono essere utili in modo costruttivo se nell'entità **Moderator** serve reintracciare l'attività con un minimo o poca logica aggiuntiva ma senza modificare lo schema).

Si è deciso di:

- 1- Incorporare le entità figlie **Player** e **Moderator** nell'entità padre **User**.
- 2- Differenziare l'identità incorporata tramite un attributo aggiuntivo (*Role*) che riferisce alla tipologia di **User** a cui si riferisce.

### 2- Generalizzazione dell'entità Match e le sue entità figlie (Stati del match)

Come l'entità figlie (**Lobby**, **Match in Countdown**, **Started Match** e **Finished Match**) rappresentano stati del **Match** che evolve nel tempo al secondo dei requisiti discussi precedentemente e anche per ridurre il costo delle operazioni che accedono all'entità figlie in un modo distinto.

Si è deciso di:

- 1- Incorporare le entità figlie **Lobby**, **Match in Countdown**, **Started Match** e **Finished Match** nell'entità padre e differenziare lo stato del **Match**.
- 2- Differenziare l'identità incorporata tramite un attributo aggiuntivo (*State*) che riferisce allo stato attuale del **Match**.
- 3- Unire le relazioni tra **Player** e i stati del **Match** prima dell'incorpamento in un'unica relazione
- 4- Gli attributi *End Time* e *Start Countdown Time* diventano opzionali nell'entità incorporata come sono opzionali nello stato di **Lobby** e **Started Match**.

\*Miglioramenti a causa delle decisioni 1 e 3:

- Riduzione dei costi:

> [09P] Costo Originale: 10.000; Costo con la Ridondanza: 6.000;

Costo con l'incorpamento: 4.000

Riduzione in totale del 60%

> [04p] Costo Originale: 120.000; Costo con l'incorpamento: 80.000

Riduzione del: 33.3%

### 3- Generalizzazione dell'entità **Action** e le sue entità figlie (Tipi del **Action**)

Come le tre entità figlie **Placement**, **Movement** e **Combat** condividono degli attributi che sono essenziali per ogni **Action** (*Action Number*, *Tanks Number*) ma alcuni di loro (**Movement** e **Combat**) hanno dettagli in più che sono essenziali mentre non servono ulteriori dettagli per **Placement**.

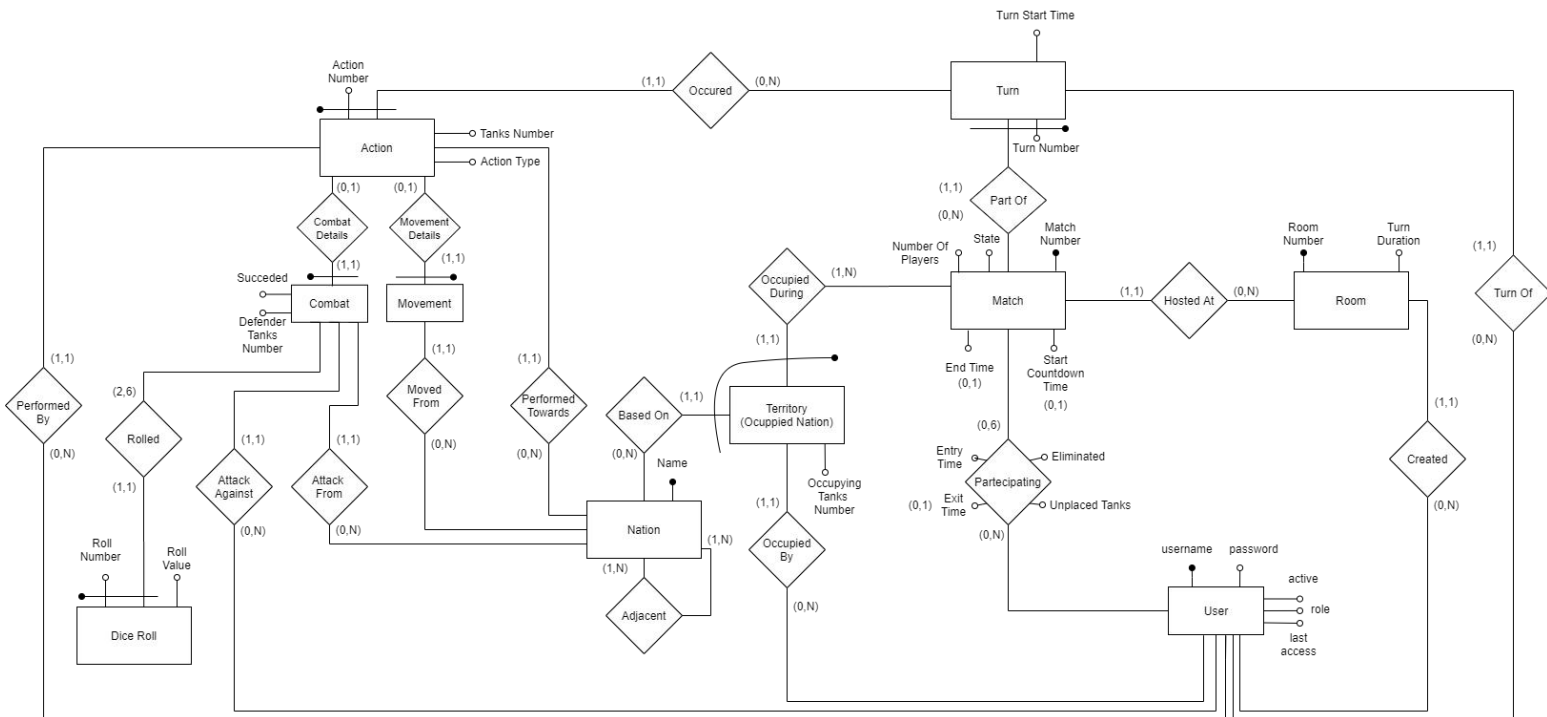
- 1- Aggiungere un attributo di tipo (*Action Type*) per identificare il tipo di **Action** a cui si riferisce.
- 2- Eliminare l'entità figlia **Placement** come verrà rappresentata tramite l'entità padre **Action** e verrà identificata tramite l'attributo (*Action Type*).
- 3- Sostituire la generalizzazione di **Movement** e **Combat** con relazioni di dipendenza verso **Action** per salvare i dati relativi nei loro contesti in un modo separato e verranno richiesti in un modo Lazy.

### Scelta degli identificatori primari:

Entità	Identificatore Primario
User	Username
Room	Room Number
Match	Match Number
Turn	Match Number, Turn Number
Action	Match Number, Turn Number, Action Number
Movement	Match Number, Turn Number, Action Number
Combat	Match Number, Turn Number, Action Number
Dice Roll	Match Number, Turn Number, Action Number, Roll Number
Nation	Name
Territory	Match Number, Name

\*Nota: Nello schema ER **Match** era un'entità debole verso **Room** ma con *Match Number* che era un candidato per essere chiave primaria; Alla fine si sceglie *Match Number* come identificatore primario senza la dipendenza su **Room** (*Room Number*) per eliminare una ridondanza enorme che risulterà nella catena di entità deboli dipendenti su **Match** (Turn, Action, Movement, Combat e Dice Roll) come alcune di quelle entità deboli hanno una frequenza elevatissima.

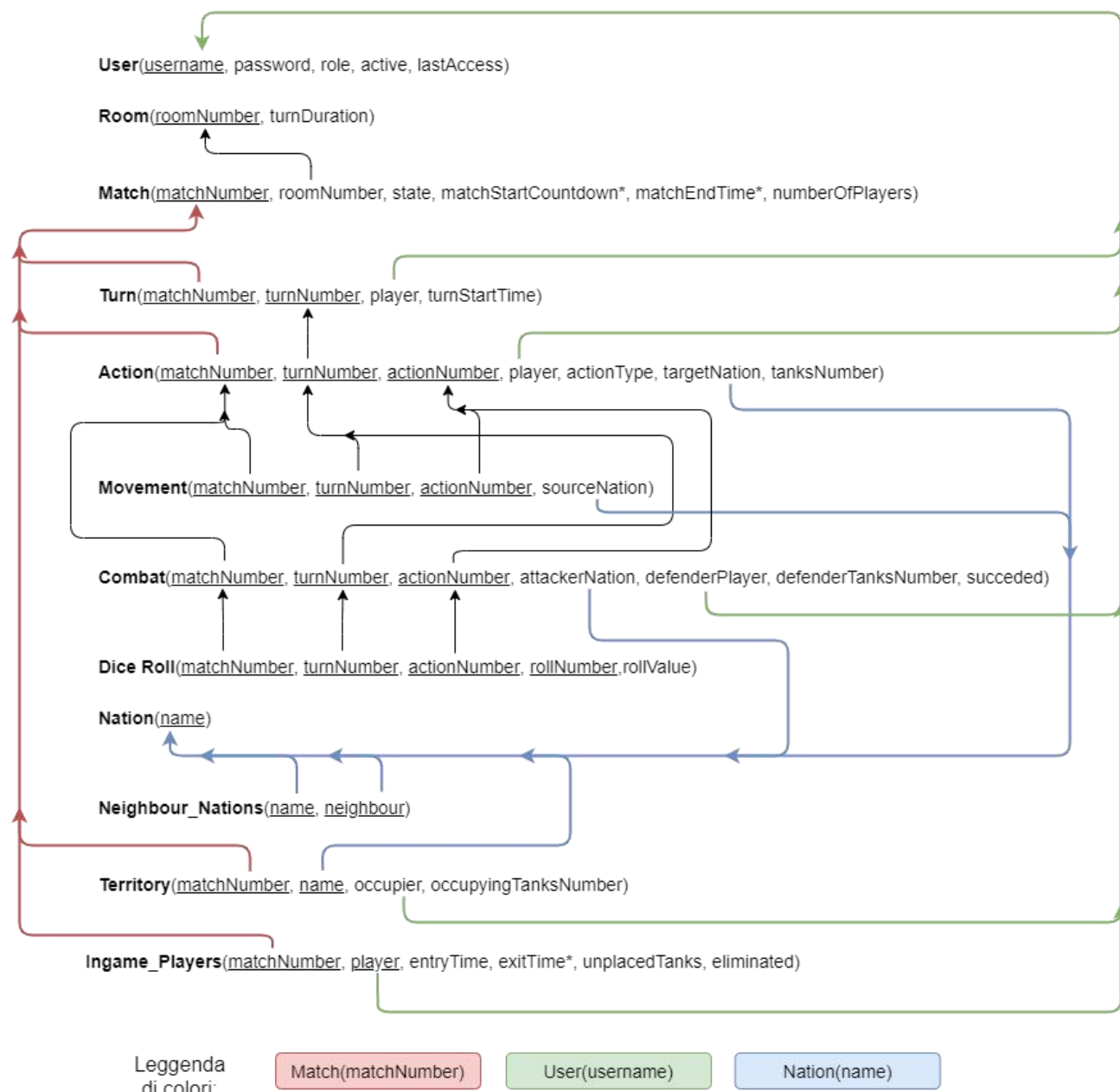
## Modello Ristrutturato



## Trasformazione di attributi e identificatori

- Nell'entità **Territory** l'attributo *Tanks Number* viene rinominato a *Occupying Tanks Number* per differenziare il suo contesto da *Tanks Number* nell'entità **Action**.

## Traduzione di entità e associazioni





**User**(username, password, role, active, lastAccess)

**Room**(roomNumber, turnDuration)

**Match**(matchNumber, roomNumber, state, matchStartCountdown\*, matchEndTime\*, numberOfPlayers)

**Turn**(matchNumber, turnNumber, player, turnStartTime)

**Action**(matchNumber, turnNumber, actionNumber, player, actionType, targetNation, tanksNumber)

**Movement**(matchNumber, turnNumber, actionNumber, sourceNation)

**Combat**(matchNumber, turnNumber, actionNumber, attackerNation, defenderPlayer, defenderTanksNumber, succeeded)

**Dice Roll**(matchNumber, turnNumber, actionNumber, rollNumber, rollValue)

**Nation**(name)

**Territory**(matchNumber, name, occupier, occupyingTanksNumber)

**Ingame\_Players**(matchNumber, player, entryTime, exitTime\*, unplacedTanks, eliminated)

**Neighbour\_Nations**(name, neighbour)

Vincoli referenziali:

Match(roomNumber)  $\subseteq$  Room(roomNumber)

Turn(matchNumber)  $\subseteq$  Match(matchNumber)

Turn(player)  $\subseteq$  User(username)

Action(matchNumber)  $\subseteq$  Match(matchNumber)

Action(turnNumber)  $\subseteq$  Turn(turnNumber)

Action(player)  $\subseteq$  User(username)

Action(targetNation)  $\subseteq$  Nation(name)

Movement(matchNumber)  $\subseteq$  Action(matchNumber)

Movement(turnNumber)  $\subseteq$  Action(turnNumber)

Movement(actionNumber)  $\subseteq$  Action(actionNumber)

Movement(sourceNation)  $\subseteq$  Nation(name)

Combat(matchNumber)  $\subseteq$  Action(matchNumber)

Combat(turnNumber)  $\subseteq$  Action(turnNumber)

Combat(actionNumber)  $\subseteq$  Action(actionNumber)

Combat(attackerNation)  $\subseteq$  Nation(name)

Combat(defenderPlayer)  $\subseteq$  User(username)

Dice\_Roll(matchNumber)  $\subseteq$  Combat(matchNumber)

Dice\_Roll(turnNumber)  $\subseteq$  Combat(turnNumber)

Dice\_Roll(actionNumber)  $\subseteq$  Combat(actionNumber)

Territory(matchNumber)  $\subseteq$  Match(matchNumber)

$\text{Territory}(\text{name}) \subseteq \text{Nation}(\text{name})$

$\text{Territory}(\text{occupier}) \subseteq \text{User}(\text{username})$

$\text{Ingame\_Players}(\text{matchNumber}) \subseteq \text{Match}(\text{matchNumber})$

$\text{Ingame\_Players}(\text{player}) \subseteq \text{User}(\text{username})$

## **Normalizzazione del modello relazionale**

Tutte le tabelle sono in 3NF.

## 5. Progettazione fisica

### Utenti e privilegi

Descrivere, all'interno dell'applicazione, quali utenti sono stati previsti con quali privilegi di accesso su quali tabelle, giustificando le scelte progettuali.

#### Utente (Ruolo):

- Login (guest)
- Moderator (moderator)
- Player (player)

#### Privilegi:

### Strutture di memorizzazione

Compilare la tabella seguente indicando quali tipi di dato vengono utilizzati per memorizzare le informazioni di interesse nelle tabelle, per ciascuna tabella.

Tabella User		
Colonna	Tipo di dato	Attributi <sup>2</sup>
username	VARCHAR(45)	PK, NN
password	VARCHAR(45)	NN
role	ENUM('player', 'moderator')	NN
lastAccess	TIMESTAMP	NN
active	BIT	0

Tabella Room		
Colonna	Tipo di dato	Attributi <sup>3</sup>
roomNumber	INT(6)	PK, NN, UN
turnDuration	INT(3)	NN, UN
createdBy	VARCHAR(45)	NN

<sup>2</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

<sup>3</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

Tabella Match		
Colonna	Tipo di dato	Attributi <sup>4</sup>
matchNumber	INT	PK,NN,UN,AI
roomNumber	INT(6)	NN, UN
state	ENUM('lobby', 'countdown', 'started', 'finished')	NN
matchStartCountdown	TIMESTAMP	
matchEndTime	TIMESTAMP	
numberOfPlayers	SMALLINT	NN

Tabella Turn		
Colonna	Tipo di dato	Attributi <sup>5</sup>
matchNumber	INT	PK,NN,UN
turnNumber	INT(3)	PK,NN,UN
player	VARCHAR(45)	NN
turnStartTime	TIMESTAMP	NN

Tabella Action		
Colonna	Tipo di dato	Attributi <sup>6</sup>
matchNumber	INT	PK,NN,UN
turnNumber	INT(3)	PK,NN,UN
actionNumber	INT(2)	PK,NN,UN
player	VARCHAR(45)	NN
actionType	ENUM('placement', 'movement', 'combat')	NN
targetNation	VARCHAR(32)	NN
tanksNumber	INT(3)	NN

---

<sup>4</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

<sup>5</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

<sup>6</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

Tabella Movement		
Colonna	Tipo di dato	Attributi <sup>7</sup>
matchNumber	INT	PK,NN,UN
turnNumber	INT(3)	PK,NN,UN
actionNumber	INT(2)	PK,NN,UN
sourceNation	VARCHAR(32)	NN

Tabella Combat		
Colonna	Tipo di dato	Attributi <sup>8</sup>
matchNumber	INT	PK,NN,UN
turnNumber	INT(3)	PK,NN,UN
actionNumber	INT(2)	PK,NN,UN
attackerNation	VARCHAR(32)	NN
defenderPlayer	VARCHAR(45)	NN
defenderTanksNumber	INT(3)	NN
succeded	BIT	NN

Tabella Dice_Roll		
Colonna	Tipo di dato	Attributi <sup>9</sup>
matchNumber	INT	PK,NN,UN
turnNumber	INT(3)	PK,NN,UN
actionNumber	INT(2)	PK,NN,UN
rollNumber	SMALLINT	PK,NN,UN
rollValue	SMALLINT	NN

Tabella Nation		
Colonna	Tipo di dato	Attributi <sup>10</sup>
name	VARCHAR(32)	PK,NN

<sup>7</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

<sup>8</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

<sup>9</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

<sup>10</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

Tabella Neighbour_Nations		
Colonna	Tipo di dato	Attributi <sup>11</sup>
name	VARCHAR(32)	PK,NN
neighbour	VARCHAR(32)	PK,NN

Tabella Territory		
Colonna	Tipo di dato	Attributi <sup>12</sup>
matchNumber	INT	PK,NN,UN
nation	VARCHAR(32)	PK,NN
occupier	VARCHAR(45)	NN
occupyingTanksNumber	INT(3)	NN

Tabella Ingame_Players		
Colonna	Tipo di dato	Attributi <sup>13</sup>
matchNumber	INT	PK,NN,UN
player	VARCHAR(45)	PK,NN
entryTime	TIMESTAMP	NN
exitTime	TIMESTAMP	
unplacedTanks	INT	NN
eliminated	BIT	NN

---

<sup>11</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

<sup>12</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

<sup>13</sup> PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

## Indici

Tabella User	
Indice PRIMARY	Tipo <sup>14</sup> :
username	PR

Tabella Room	
Indice PRIMARY	Tipo <sup>15</sup> :
roomNumber	PR
Indice IDX_CREATED_BY	Tipo <sup>16</sup> :
createdBy	IDX

Tabella Match	
Indice PRIMARY	Tipo <sup>17</sup> :
matchNumber	PR
Indice FK_MATCH_ROOM	Tipo <sup>18</sup> :
roomNumber	IDX

Tabella Turn	
Indice PRIMARY	Tipo <sup>19</sup> :
matchNumber	PR
turnNumber	PR
Indice IDX_FK_PLAYER	Tipo <sup>20</sup> :
player	IDX

---

<sup>14</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>15</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>16</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>17</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>18</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>19</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>20</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<b>Tabella Action</b>	
<b>Indice PRIMARY</b>	<b>Tipo<sup>21</sup>:</b>
matchNumber	PR
turnNumber	PR
actionNumber	PR
<b>Indice IDX_FK_TARGET_NATION</b>	<b>Tipo<sup>22</sup>:</b>
targetNation	IDX

<b>Tabella Movement</b>	
<b>Indice PRIMARY</b>	<b>Tipo<sup>23</sup>:</b>
matchNumber	PR
turnNumber	PR
actionNumber	PR
<b>Indice IDX_FK_SOURCE_NATION</b>	<b>Tipo<sup>24</sup>:</b>
sourceNation	IDX

<b>Tabella Combat</b>	
<b>Indice PRIMARY</b>	<b>Tipo<sup>25</sup>:</b>
matchNumber	PR
turnNumber	PR
actionNumber	PR
<b>Indice IDX_FK_DEFENDER</b>	<b>Tipo<sup>26</sup>:</b>
defenderPlayer	IDX
<b>Indice IDX_FK_ATTACKER_NATION</b>	<b>Tipo<sup>27</sup>:</b>
attackerNation	IDX

---

<sup>21</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>22</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>23</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>24</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>25</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>26</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>27</sup> IDX = index, UQ = unique, FT = full text, PR = primary.



Tabella Dice_Roll	
Indice PRIMARY	Tipo <sup>28</sup> :
matchNumber	PR
turnNumber	PR
actionNumber	PR
rollNumber	PR

Tabella Nation	
Indice PRIMARY	Tipo <sup>29</sup> :
name	PR

Tabella Neighbour_Nations	
Indice PRIMARY	Tipo <sup>30</sup> :
nation	PR
neighbour	PR
Indice IDX_FK_NATION	Tipo <sup>31</sup> :
nation	IDX
Indice IDX_FK_NEIGHBOUR	Tipo <sup>32</sup> :
neighbour	IDX

Tabella Territory	
Indice PRIMARY	Tipo <sup>33</sup> :
matchNumber	PR
nation	PR
Indice IDX_FK_NATION	Tipo <sup>34</sup> :
nation	IDX
Indice IDX_FK_OCCUPIER	Tipo <sup>35</sup> :
occupier	IDX

<sup>28</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>29</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>30</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>31</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>32</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>33</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>34</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>35</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

Tabella Ingame Players	
Indice PRIMARY	Tipo <sup>36</sup> :
matchNumber	PR
player	PR
Indice IDX_FK_USER	Tipo <sup>37</sup> :
player	IDX

## Trigger

Descrivere quali trigger sono stati implementati, mostrando il codice SQL per la loro istanziazione. Si faccia riferimento al fatto che il DBMS di riferimento richiede di utilizzare trigger anche per realizzare vincoli di check ed asserzioni.

## Eventi

Descrivere quali eventi sono stati implementati, mostrando il codice SQL per la loro istanziazione. Si descriva anche se gli eventi sono istanziati soltanto in fase di configurazione del sistema, o se alcuni eventi specifici vengono istanziati in maniera effimera durante l'esecuzione di alcune procedure.

## Viste

Mostrare e commentare il codice SQL necessario a creare tutte le viste necessarie per l'implementazione dell'applicazione.

*Non ci viste previste*

## Stored Procedures e transazioni

Mostrare e commentare le stored procedure che sono state realizzate per implementare la logica applicativa delle operazioni sui dati, evidenziando quando (e perché) sono state realizzate operazioni transazionali complesse.

---

<sup>36</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

<sup>37</sup> IDX = index, UQ = unique, FT = full text, PR = primary.

## Appendice: Implementazione

### Codice SQL per istanziare il database

Riportare il codice SQL necessario ad istanziare lo schema del DB. Le stored procedure, le viste, i trigger, gli eventi e tutto quello che è stato già inserito all'interno della relazione di progetto nelle sezioni precedenti non deve essere inserito in questa appendice.

Sì, avete letto bene: **riportare il codice SQL**. Frasi del tipo “il codice è nel file allegato” non rispondono alla richiesta di riportare il codice SQL.

### Codice del Front-End

Riportare (correttamente formattato) il codice C del thin client realizzato per interagire con la base di dati.

Sì, avete letto bene: **riportare il codice C**. Frasi del tipo “il codice è nei file allegati” non rispondono alla richiesta di riportare il codice C.