|  |  |
| --- | --- |
|  |  |

Basi di Dati

Progetto A.A. 2021/2022

TITOLO DEL PROGETTO

0253048

Abdelrahman Abdelrahman

**Indice**

[1. Descrizione del Minimondo 2](#_Toc606296459)

[2. Analisi dei Requisiti 4](#_Toc1289394997)

[3. Progettazione concettuale 5](#_Toc2081466291)

[4. Progettazione logica 6](#_Toc2147004904)

[5. Progettazione fisica 8](#_Toc518560220)

[Appendice: Implementazione 10](#_Toc403811585)

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)

# Descrizione del Minimondo

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65 | 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 adiacent (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. |

# 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(1) | 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(2) | 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 rappresentono le istanze delle nazioni assegnate ai giocatori durante una “Partita Avviata” che variano al secondo del contesto della partita. |
| 23 | giocatori | Giocatori Participanti | Come vengono definiti prima come “giocatori participanti stanno partecipando alla partita”. |
| 27,28(1),29 | stato | Territorio  Alleato | Come ogni giocatore può solamente far partire un’azione da un Territorio Alleato (Sotto il suo dominio). |
| 28(2) | 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 “Attaco” |
| 34 | Giocatore che attacca | Giocatore Attaccante | Come viene chiamato “Attaccante” dopo(41) . |
| 34 | Quello che Difende | Difensore | Come viene chiamato “Difensore” dopo(42). |
| 44,49 | Armata | Carro Armato | Come viene riferita prima come “Carro Armato”(30) |
| 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 attacante 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 attacante 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 attacante 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 attacante 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. |

# Progettazione concettuale

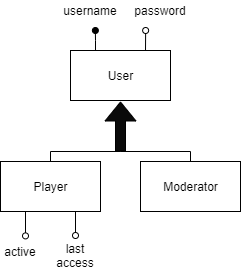
## Costruzione dello schema E-R

In questa sezione è necessario riportare tutti 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)

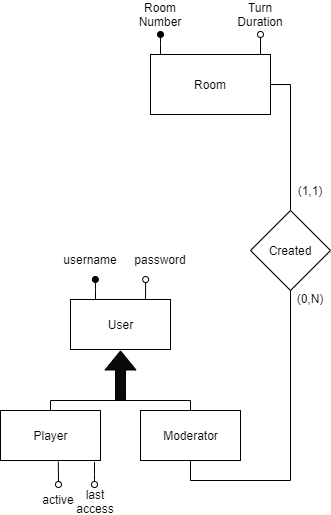
****

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.

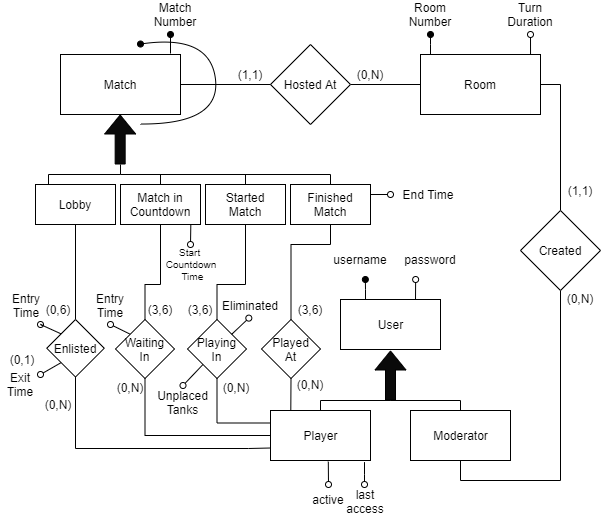
**Room** (Stanza)



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 supponere 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 ERe anche il pattern di Evoluzioneè 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*).

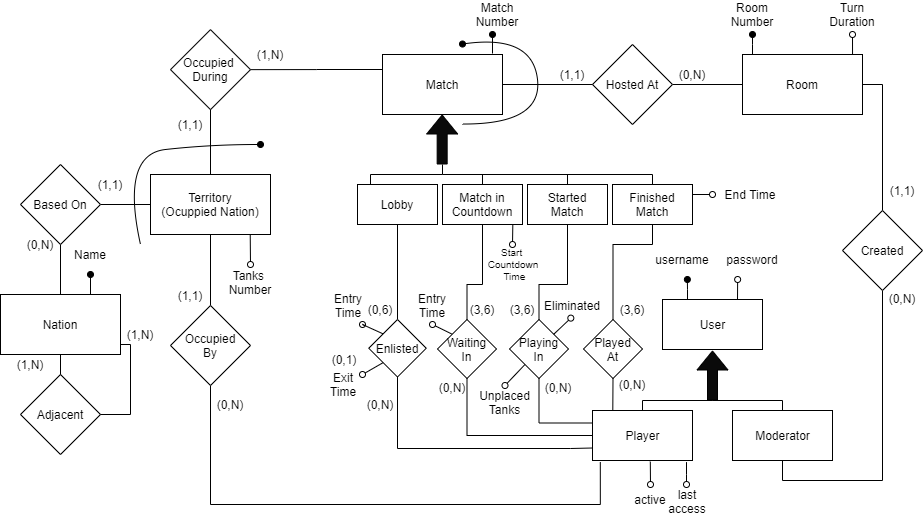
2-**Match in Countdown: Match** che ha almeno 3 **Player** e può andare in Countdown per avviarsi dopo 2 Minuti del *Start Countdown Time.*

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.

1. **Started Match: Match** avviatoche 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*).

1. **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)  


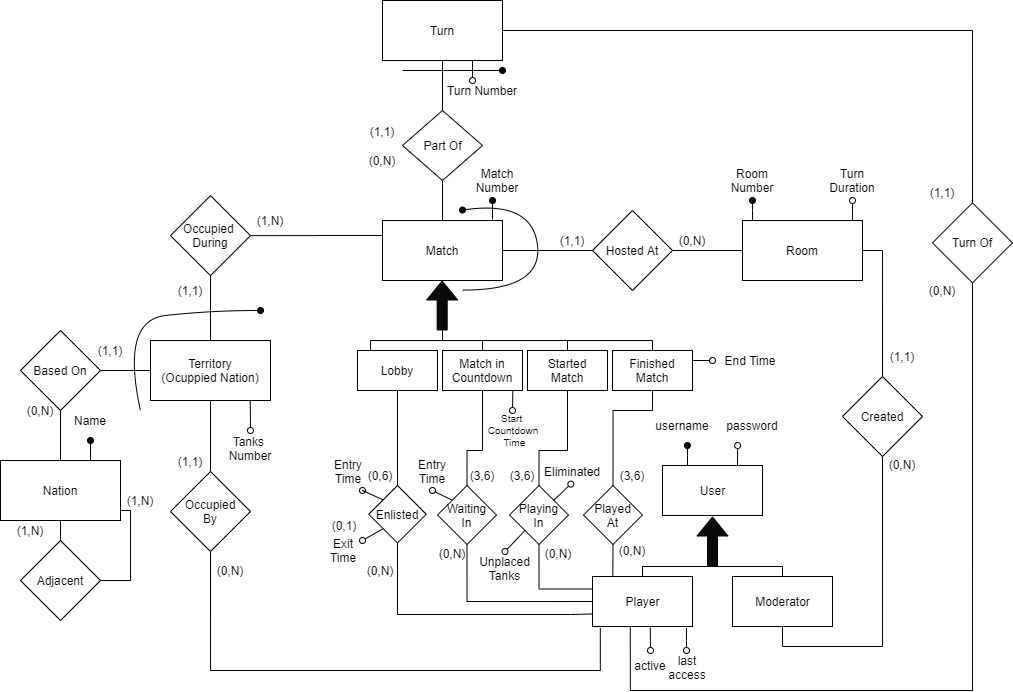
Come il gioco è basato sul gioco *Risiko*, allora ci sono *42 stati* in ogni **Match**, ed ogni tabellone deve tenere traccia di tutti i quarantadue.

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*).

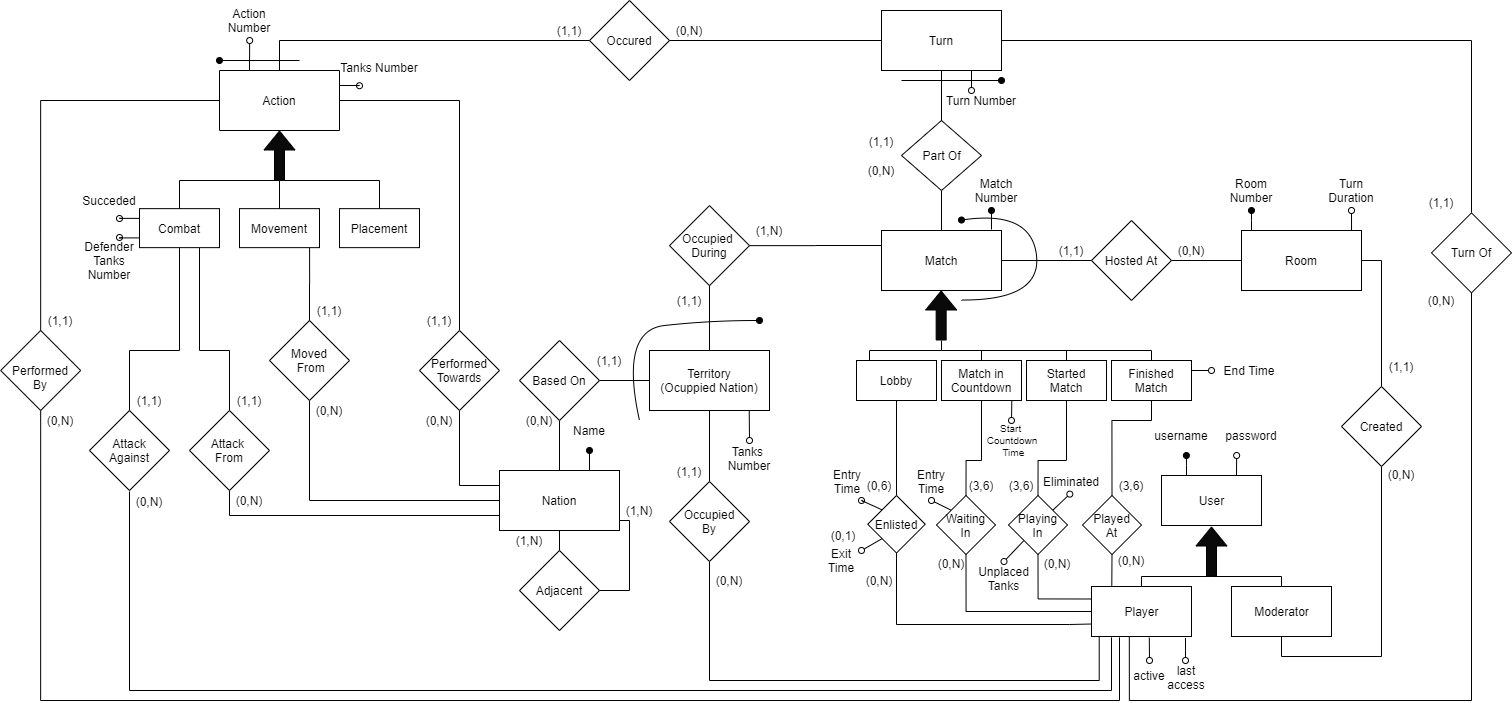
Secondo le regole del gioco ogni **Nation** è adiacente ad almeno un’altra **Nation**, perciò si realizza la relazione ricorsiva (*Adjacent*) tra **Nation** e se stessa.

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’entita **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*).

Un **Match** può non avere nessun **Turn** affinchè non è arrivato allo stato di **Started Match**.  
  
  
  
  
  
  
  
  
  
  
**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** comeè un’entita debole.

- *Performed Towards*:La relazione rappresenta un **Action** ela **Nation** verso cui è stato fatto quel **Action**.

- *Performed By*:La relazione rappresenta un **Action** e il **Player** che haeffettuato 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 reintraciato 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** versoun’altra adiacente.

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

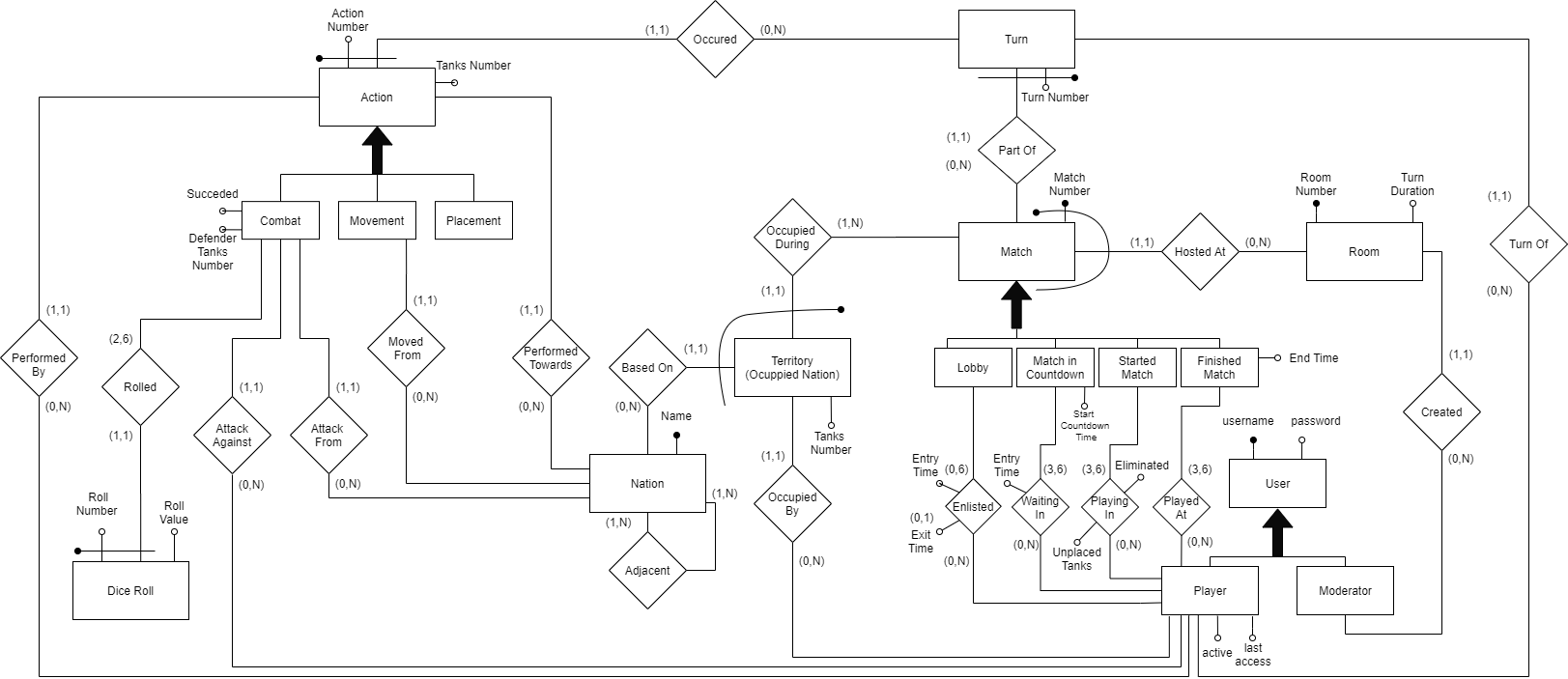
>> *Succeded*: Rappresenta se quel **Combat** è riuscito o meno.

>> *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** Attacante, 1 **Player** Difensore])e *al massimo sei* **Dice Roll** associati

(*Tre* **Dice Roll** per **Player** [3 **Player** Attacante, 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

Nell’integrazione finale delle varie parti dello schema E-R è possibile che si evidenzino dei conflitti sui nomi utilizzati e dei conflitti struttuali. Prima di riportare lo schema E-R finale, descrivere quali passi sono stati adottati per risolvere tali conflitti.

## Regole aziendali

Laddove la specifica non sia catturata in maniera completa dallo schema E-R, corredare lo stesso in questo paragrafo con l’insieme delle regole aziendali necessarie a completare la progettazione concettuale.

## Dizionario dei dati

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Entità** | **Descrizione** | **Attributi** | **Identificatori** |
|  |  |  |  |

# 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 ciasun 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[[1]](#footnote-0)** | **Volume atteso** |
|  |  |  |

## 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** |
|  |  |  |

## 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.

## 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:

Analisi delle ridondanze

Eliminazione delle generalizzazioni

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.

## Trasformazione di attributi e identificatori

Qualora siano presenti, in questa fase della progettazione, attributi ripetuti o identificatori esterni, descrivere quali trasformazioni vengono realizzate sul modello per facilitare la traduzione nello schema relazionale.

## Traduzione di entità e associazioni

Riportare in questa sezione la traduzione di entità ed associazioni nello schema relazionale.

Fornire una rappresentazione grafica del modello relazionale completo.

## Normalizzazione del modello relazionale

Effettuare la normalizzazione del modello relazionale precedentemente descritto (in forma grafica) andando a mostrare le forme 1NF, 2NF, 3NF.

# 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.

## 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 <nome>** | | |
| **Colonna** | **Tipo di dato** | **Attributi[[2]](#footnote-1)** |
|  |  |  |

## Indici

Compilare la seguente tabella, per ciascuna tabella del database in cui sono presenti degli indici. Descrivere le motivazioni che hanno portato alla creazione di un indice.

|  |  |
| --- | --- |
| **Tabella <nome>** | |
| **Indice <nome>** | **Tipo[[3]](#footnote-2):** | |
| Colonna 1 | <nome> | |

## Trigger

Descrivere quali trigger sono stati implementati, mostrando il codice SQL per la loro instanziazione. 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 instanziazione. 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.

## 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.

# Appendice: Implementazione

## Codice SQL per instanziare 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.

1. Indicare con E le entità, con R le relazioni [↑](#footnote-ref-0)
2. PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna. [↑](#footnote-ref-1)
3. IDX = index, UQ = unique, FT = full text, PR = primary. [↑](#footnote-ref-2)