**Scenari Applicazione**

|  |  |
| --- | --- |
| **Login App** | |
| **Descrizione** | Permette all’attore di accedere al sistema e alle funzionalità dell’applicazione mediante le proprie credenziali |
| **Attori** | Giocatore |
| **Input** | Username e password |
| **Pre-condizione** | L’attore è registrato nel sistema |
| **Output** |  |
| **Post-condizione** | Attore loggato al sistema |
| **Scenario principale** | 1. L’attore avvia l’applicazione 2. L’attore inserisce username e password 3. L’applicazione comunica al server centrale le credenziali immesse 4. Il sistema verifica l’autenticità dei dati 5. Il sistema verifica se l’attore non è presente nella lista di una partita “in corso” |
| **Scenario alternativo** | 1. Il sistema riscontra una inconsistenza nei dai inseriti 2. Ritorna al punto 2 3. Include “riconnessione alla partita” |

|  |  |
| --- | --- |
| **Riconnessione alla partita** | |
| **Descrizione** | Permette all’attore di riconnettersi alla partita alla quale era iscritto se ancora “in corso” |
| **Attori** | Giocatore |
| **Input** | ID\_Utente, ID\_Partita |
| **Pre-condizione** | Attore loggato al sistema |
| **Output** |  |
| **Post-condizione** | Stato dell’attore aggiornato all’interno della partita |
| **Scenario principale** | 1. Il sistema invia all’applicazione una richiesta di “riconnessione alla partita” 2. L’applicazione visualizza all’attore la richiesta di riconnessione ricevuta dal server 3. L’attore accetta la richiesta di riconnessione alla partita 4. L’applicazione invia la risposta dell’attore al server 5. Il sistema aggiorna lo stato dell’attore all’interno della partita 6. L’applicazione riconnette l’attore alla partita |
| **Scenario alternativo** | 1. L’attore rifiuta la richiesta di riconnessione alla partita 2. L’applicazione invia la risposta dell’attore al server 3. Il sistema rimuove l’attore dalla partita |

|  |  |
| --- | --- |
| **Creazione partita** | |
| **Descrizione** | Permette all’attore di creare una nuova partita |
| **Attori** | Giocatore |
| **Input** | Caratteristiche della partita da creare |
| **Pre-condizione** | Attore loggato |
| **Output** | 1. Messaggio di conferma: “Partita crea con successo” 2. Messaggio di errore:”Errore nella creazione della partita” |
| **Post-condizione** | Partita creata |
| **Scenario principale** | 1. L’attore accede alla sezione per la creazione di una nuova partita 2. L’attore inserisce le impostazioni della partita che si vuole creare 3. L’applicazione comunica al server i dati relativi alla partita che si vuole creare 4. Il sistema verifica la correttezza dei dati ricevuti 5. Il sistema crea la partita 6. Il sistema aggiorna l’elenco delle partite 7. Il sistema comunica all’applicazione l’avvenuta creazione della partita 8. L’applicazione visualizza all’utente il messaggio 1. |
| **Scenario alternativo** | 1. Il sistema comunica all’applicazione che la partita non può essere creata 2. L’applicazione visualizza all’utente il messaggio 2. 3. Ritorna al punto 2. |

|  |  |
| --- | --- |
| **Iscrizione a una partita** | |
| **Descrizione** | Permette all’attore di iscriversi ad una partita esistente |
| **Attori** | Giocatore |
| **Input** | IdPartita, IdUtente |
| **Pre-condizione** | Attore loggato |
| **Output** | 1. Messaggio di conferma: “Iscrizione effettuata con successo” 2. Messaggio di errore:”impossibile partecipare alla partita” |
| **Post-condizione** | Attore iscritto alla partita selezionata |
| **Scenario principale** | 1. L’attore visualizza l’elenco delle partite create 2. L’attore seleziona la partita alla quale vuole partecipare 3. L’applicazione comunica al server i dati relativi alla partita e all’attore che vuole parteciparvi 4. Il sistema verifica se l’attore può essere iscritto alla partita selezionata 5. Il sistema inserisce l’attore tra i partecipanti alla partita 6. Il sistema comunica all’applicazione l’avvenuta iscrizione alla partita 7. L’applicazione visualizza all’utente il messaggio 1. |
| **Scenario alternativo** | 1. Il sistema comunica all’applicazione che l’utente non può essere iscritto alla partita 2. L’applicazione visualizza all’utente il messaggio 2. 3. Ritorna al punto 1. |

|  |  |
| --- | --- |
| **Recupero armate con carte** | |
| **Descrizione** | L’attore sceglie le carte da giocare e l’applicazione gli assegna il numero di armate corrispondenti |
| **Attori** | Giocatore |
| **Input** | Carte per il recupero delle armate |
| **Pre-condizione** | Turno di attacco dell’attore, Fase di disposizione armate non avvenuta |
| **Output** | Numero delle armate di rinforzo |
| **Post-condizione** | Fase di recupero delle armate terminata, il numero delle armate in possesso dell’attore viene incrementato del quantità corrispondete alle carte selezionate |
| **Scenario principale** | 1. L’applicazione visualizza le carte in possesso dell’attore 2. L’attore sceglie le tre carte da utilizzare per il recupero delle armate 3. L’applicazione visualizza il numero di armate corrispondenti al tris selezionato 4. L’attore conferma il recupero delle armate 5. L’applicazione assegna all’attore il numero di armate che gli spetta 6. Le carte utilizzate vengono reinserite nel mazzo 7. Include “aggiornamento tavolo” |
| **Scenario alternativo** | 1. L’attore annulla l’operazione di recupero della armate 2. Il numero di armate visualizzate è uguale a zero 3. Ritorna al punto 1 |

|  |  |
| --- | --- |
| **Disposizione armate** | |
| **Descrizione** | Permette di disporre le armate sulla mappa |
| **Attori** | Giocatore |
| **Input** | Territorio, numero armate |
| **Pre-condizione** | Turno di gioco dell’attore, fase di attacco e fase di spostamento non effettuate |
| **Output** |  |
| **Post-condizione** | Fase di disposizione delle armate terminata |
| **Scenario principale** | 1. L’applicazione controlla che l’attore abbia armate da posizionare 2. L’attore sceglie il territorio sul quale posizionare le armate 3. L’applicazione controlla che il territorio scelto sia utilizzabile dall’attore 4. L’applicazione posiziona un’armata nel territorio selezionato 5. Ritorna al punto 1 |
| **Scenario alternativo** | 1. L’attore non ha ulteriori armate da posizionare 2. L’attore conferma la disposizione delle armate 3. Include “aggiornamento tavolo” 4. Il territorio non viene selezionato 5. Ritorna al punto 2 6. L’attore annulla la disposizione delle armate 7. Il numero di armate da posizionare ritorna al valore di partenza 8. Ritorna al punto 2 |

|  |  |
| --- | --- |
| **Attacco** | |
| **Descrizione** | Permette all’attore di attaccare un territorio |
| **Attori** | Giocatore |
| **Input** | Territorio di partenza, territorio di destinazione, numero truppe |
| **Pre-condizione** | Turno di gioco dell’attore, fase di disposizione delle armate terminata, fase di spostamento non eseguita |
| **Output** | Resoconto attacco |
| **Post-condizione** | Fase di attacco terminata |
| **Scenario principale** | 1. L’attore sceglie il territorio dal quale lanciare l’attacco 2. L’applicazione controlla che il territorio scelto sia utilizzabile dall’attore per l’attacco 3. Il controllo risulta valido e l’applicazione seleziona il territorio 4. L’attore sceglie un territorio non in suo possesso sul quale effettuare l’attacco 5. L’applicazione controlla che il territorio scelto sia attaccabile 6. L’attore sceglie il numero di truppe per l’attacco 7. L’attore conferma l’attacco 8. L’applicazione genera i dati di attacco 9. L’applicazione genera i dati di difesa 10. L’applicazione visualizza all’attore l’esito dell’attacco 11. Include “aggiornamento tavolo” |
| **Scenario alternativo** | 1. Il controllo non risulta valido e Ritorna al punto 1 2. L’attore seleziona un proprio territorio 3. Ritorna al punto 2 4. Il controllo non risulta valido e Ritorna al punto 1 5. L’attore annulla l’operazione di attacco |

|  |  |
| --- | --- |
| **Spostamento truppe** | |
| **Descrizione** | Permette all’attore di spostare le proprie truppe |
| **Attori** | Giocatore |
| **Input** | Territorio di partenza, territorio di destinazione, numero truppe |
| **Pre-condizione** | Turno di gioco dell’attore, fase di attacco terminata |
| **Output** |  |
| **Post-condizione** | Fase di spostamento truppe terminata, turno giocatore terminato |
| **Scenario principale** | 1. L’attore sceglie il territorio di partenza per lo spostamento 2. L’applicazione controlla che il territorio scelto sia utilizzabile dall’attore come partenza dello spostamento 3. Il controllo risulta valido e l’applicazione seleziona il territorio 4. L’attore sceglie un territorio come destinazione dello spostamento 5. L’applicazione controlla che il territorio scelto sia utilizzabile come destinazione dello spostamento 6. L’attore sceglie il numero di truppe per lo spostamento 7. L’applicazione effettua lo spostamento 8. Include “aggiornamento tavolo” 9. Include “passa turno” |
| **Scenario alternativo** | 1. Il controllo non risulta valido e Ritorna al punto 1 2. L’attore annulla l’operazione di spostamento 3. Il controllo non risulta valido e Ritorna al punto 5 |

|  |  |
| --- | --- |
| **Passaggio del turno** | |
| **Descrizione** | Permette all’attore di terminare il proprio turno di gioco |
| **Attori** | Giocatore |
| **Input** | IDUtente |
| **Pre-condizione** | Turno di gioco dell’attore, fase di disposizione armate terminata |
| **Output** |  |
| **Post-condizione** | Turno giocatore terminato, passaggio del turno al giocatore successivo |
| **Scenario principale** | 1. L’attore sceglie di terminare il proprio turno di gioco 2. Include “recupero carta” 3. Include “aggiornamento tavolo” |
| **Scenario alternativo** |  |

|  |  |
| --- | --- |
| **Recupero carta** | |
| **Descrizione** | L’applicazione assegna all’attore una carte se almeno uno degli attacchi effettuati durante il turno in corso è stato vincente |
| **Attori** | Giocatore |
| **Input** |  |
| **Pre-condizione** | Turno di gioco dell’attore, fase di attacco effettuata, fase di passaggio del turno in esecuzione. |
| **Output** | Carta recuperata |
| **Post-condizione** | Carta recuperata rimossa dal mazzo, carta recuperata inserita nel mazzo dell’attore |
| **Scenario principale** | 1. L’applicazione controlla che l’attore abbia effettuato nel suo turno di gioco almeno un attacco vincente 2. L’applicazione preleva una carta dal mazzo 3. L’applicazione inserisce la carta recuperata tra le carte dell’attore |
| **Scenario alternativo** |  |

|  |  |
| --- | --- |
| **Aggiornamento tavolo** | |
| **Descrizione** | L’applicazione aggiorna il tavolo di gioco degli attori |
| **Attori** | Giocatore |
| **Input** | ID\_Utente, numero di armate recuperate, lista di oggetti contenenti {Territorio, numero armate}, tripla {partenza attacco, destinazione attacco, truppe di attacco}, tripla {partenza spostamento, destinazione spostamento, numero di armate spostate} |
| **Pre-condizione** |  |
| **Output** |  |
| **Post-condizione** |  |
| **Scenario principale** | 1. L’applicazione dell’attore notifica agli altri giocatori che è necessario aggiornare il proprio tavolo 2. Le applicazioni dei singoli giocatori determinano il tipo di azione da replicare 3. Le applicazioni dei singoli giocatori eseguono l’azione richiesta 4. L’applicazione che effettua l’aggiornamento del tavolo rileva che l’operazione da replicare è di tipo “attacco” 5. L’applicazione riceve dall’attore la tripla {partenza attacco, destinazione attacco, truppe di attacco } 6. L’applicazione riproduce l’attacco 7. L’applicazione controlla che l’attore non abbia raggiunto il suo obiettivo |
| **Scenario alternativo** | 1. L’applicazione che effettua l’aggiornamento del tavolo rileva che l’operazione da replicare è di tipo “recupero armate con carte” 2. L’applicazione riceve dall’attore il numero di armate recuperate 3. L’applicazione incrementa il numero delle armate in possesso dell’attore 4. L’applicazione che effettua l’aggiornamento del tavolo rileva che l’operazione da replicare è di tipo “disposizione armate” 5. L’applicazione riceve dall’attore una lista di oggetti contenenti {Territorio, numero armate} 6. L’applicazione aggiorna la disposizione delle armate dell’attore 7. L’applicazione che effettua l’aggiornamento del tavolo rileva che l’operazione da replicare è di tipo “passaggio del turno” 8. L’applicazione aggiorna il turno di gioco 9. L’applicazione che effettua l’aggiornamento del tavolo rileva che l’operazione da replicare è di tipo “spostamento” 10. L’applicazione riceve dall’attore la tripla {partenza spostamento, destinazione spostamento, numero di armate spostate} 11. L’applicazione riproduce lo spostamento 12. Include “comunicazione vittoria” |

|  |  |
| --- | --- |
| **Comunicazione vittoria** | |
| **Descrizione** | L’applicazione comunica al server la conclusione della partita |
| **Attori** | Giocatore |
| **Input** | ID\_Utente, ID\_Partita |
| **Pre-condizione** | Obiettivo raggiunto dall’attore |
| **Output** |  |
| **Post-condizione** | Partita terminata |
| **Scenario principale** | 1. L’applicazione comunica al server l’ID dell’attore e l’ID della partita 2. Il server memorizza l’esito della partita |
| **Scenario alternativo** |  |