

Analisi dei Casi d'Uso - Orbit

Attori del Sistema

Gli attori rappresentano le entità che interagiscono con il sistema. Nel contesto di Orbit logicamente distinguiamo:

- **Utente Registrato (Player):** L'attore principale che usufruisce dei servizi di acquisto e gioco.
- **Publisher:** Un utente con permessi speciali per caricare e vendere i propri giochi sulla piattaforma.
- **Amministratore (Admin):** Un super-utente responsabile della moderazione della piattaforma.
- **Database (Sistema Esterno):** Entità passiva che salva le informazioni (transazioni, utenti, giochi, recensioni).

Casi d'Uso Primari

1 - Registrazione e Autenticazione (Login/Signup)

- **Attore:** Utente o Nuovo Utente.
- **Descrizione:** L'utente crea un nuovo account o accede a uno esistente per sbloccare le funzionalità della piattaforma.
- **Flusso:**
 1. L'utente inserisce nickname e password.
 2. Il sistema verifica nel Database se le credenziali sono corrette (Login) o se il nickname è disponibile (Signup).
 3. In caso di successo, il sistema carica il profilo utente con i relativi ruoli (Admin/Publisher) e il saldo.
- **Eccezioni:** UserNotFoundException, WrongPasswordException, PlayerAlreadyExistException.

2 - Acquisto di un Gioco

- **Attore:** Utente Registrato.
- **Descrizione:** L'utente compra un gioco presente nello store utilizzando il proprio saldo virtuale.
- **Prerequisiti:** L'utente è loggato e non possiede già il gioco.
- **Flusso:**
 1. L'utente seleziona un gioco dal negozio.
 2. L'utente conferma l'intenzione di acquisto.
 3. Il sistema verifica se il saldo è sufficiente (AmountNotValidException).
 4. Il sistema sottrae l'importo dal saldo dell'utente.
 5. Il sistema aggiunge il gioco alla libreria dell'utente.
 6. Il sistema registra la transazione nello storico.
- **Conseguenze:** Il saldo è aggiornato e il gioco è accessibile nella libreria.

3 - Pubblicazione di un Gioco

- **Attore:** Publisher.
- **Descrizione:** Un publisher inserisce un nuovo titolo nel catalogo di Orbit per la vendita.
- **Flusso:**
 1. Il publisher accede alla sezione di pubblicazione.
 2. Inserisce i metadati del gioco: Titolo, Prezzo Base, Genere/Tag e percorso dell'immagine di copertina (cover_path).
 3. Il sistema salva il nuovo gioco nel Database.
- **Postcondizioni:** Il gioco è immediatamente visibile nello Shop per tutti gli utenti.

4 - Gestione del Portafoglio (Riscatto Gift Card)

- **Attore:** Utente Registrato.
- **Descrizione:** L'utente incrementa il proprio saldo riscattando un codice univoco.
- **Flusso:**
 1. L'utente accede allo shop e clicca per comprare un gioco.
 2. Inserisce il codice della Gift Card.
 3. Il sistema verifica la validità del codice nel Database (checkGiftCard).
 4. Se valido, il sistema aggiunge il valore al saldo utente e rimuove/invalida la carta dal Database.
- **Eccezioni:** CodeNotFoundException.

5 - Acquisto Licenza di Vendita

- **Attore:** Publisher.
- **Descrizione:** Un publisher attiva la sua licenza direttamente dalla home per sbloccare la possibilità di pubblicare un suo gioco.
- **Prerequisiti:** Lo user ha fatto il login con un ruolo PUBLISHER e non ha già acquistato una licenza in precedenza.
- **Flusso:**
 1. Il publisher accede alla home page.
 2. Il sistema mostra le informazioni dell'account, identificando lo user come publisher.
 3. Il sistema mostra un bottone "BUY LICENSE" cliccabile.
 4. Il publisher clicca il bottone "BUY LICENSE".
 5. Il sistema concede immediatamente la licenza all'account.
 6. Il sistema disabilita il bottone "BUY LICENSE" a indicare che la licenza è attiva.
- **Postconditions:** Il publisher è in possesso di una licenza attiva che gli garantisce l'accesso alla funzione "PUBLISH GAME".

Casi d'Uso Secondari

Consultazione Libreria

- **Attore:** Utente Registrato.
- **Descrizione:** L'utente visualizza l'elenco dei giochi acquistati.

- **Flusso:** Il sistema interroga il Database recuperando tutti gli ID gioco associati all'ID utente e ne mostra i dettagli (Copertina, Titolo).

Scrittura Recensione

- **Attore:** Utente Registrato.
- **Descrizione:** L'utente lascia un feedback su un gioco che possiede.
- **Precondizioni:** L'utente deve possedere il gioco in libreria.
- **Flusso:**
 1. L'utente seleziona un gioco dalla propria libreria.
 2. L'utente assegna un voto (1-5) e scrive un commento testuale.
 3. Il sistema salva la recensione e ricalcola la media voto del gioco (score).

Moderazione Utenti (Ban)

- **Attore:** Amministratore.
- **Descrizione:** L'amministratore blocca l'accesso a un utente o a un publisher che ha violato le regole.
- **Flusso:** L'admin ricerca l'utente e imposta il flag di ban. L'utente non potrà più effettuare il login.

Moderazione dei Giochi (Ban Gioco)

- **Attore:** Amministratore.
- **Descrizione:** L'amministratore blocca la vendita di un gioco pubblicato che viola le linee guida della piattaforma.
- **Flusso:** 1. L'amministratore ricerca il gioco nel catalogo. 2. Il sistema imposta il flag `is_banned` a 1 nel Database per quel gioco. 3. Il gioco non è più visibile nello Store per i nuovi acquisti (ma rimane nella libreria di chi lo ha già comprato).

Acquisto tramite Carta di Credito

- **Attore:** Utente Registrato.
- **Descrizione:** L'utente compra un gioco pagando direttamente con un metodo di pagamento esterno (es. Carta di Credito) anziché usare il portafoglio di Orbit.
- **Flusso:**
 1. L'utente seleziona un gioco dal negozio.
 2. Sceglie di pagare con Carta di Credito al Checkout.
 3. Il sistema processa il pagamento tramite l'interfaccia esterna (simulata).
 4. Il sistema aggiunge il gioco alla libreria e registra la transazione senza intaccare il saldo dell'utente.
- **Eccezioni:** PaymentFailedException (se la carta viene rifiutata).

Gestione degli Sconti (Discount)

- **Attore:** Publisher.

- **Descrizione:** Un publisher decide di applicare uno sconto temporaneo a un proprio gioco per incentivare le vendite.
- **Flusso:**
 1. Il publisher seleziona un proprio gioco pubblicato.
 2. Inserisce una percentuale di sconto (es. 20%).
 3. Il sistema calcola il nuovo currentPrice tramite il DiscountManager.
 4. Il sistema aggiorna il prezzo nel Database.
- **Eccezioni:** AmountNotValidException (se la percentuale o il prezzo non sono validi).