Play Splendor

Use Case: Play Splendor
Scope: Splendor
Level: User goal
Intention in Context: The intention of the *Player* is to play a game of Splendor against other *Players*
Multiplicity: Multiple *Players* can play Splendor concurrently. A given *player* is not allowed to play multiple games simultaneously.
Primary Actor: *Player*
Secondary Actors: *Players* (who play the roles of opponents)
Main Success Scenario:

1. *Player* *logs into Lobby Service*.
2. *System* shows *Player* *the available game sessions*.
3. *Player* chooses to join a game session.

Step 4 is executed once enough players have joined the game

4. *Players* take turns.

Step 5 *is executed after the last person in the round during which someone got 15 prestige points plays their turn*.

5. *System* informs all *Players* about who won the game.

Extensions:

1a. *Player* was not able to login. Use case continues at step 1.
3a. *Player* was not able to join a game session; use case continues at 3b.
3b. *Player* has the ability to manage game sessions.
3c. *Player* may request to perform Universal Actions.

PerformUniversalActions

Use Case: PerformUniversalActions
Scope: Splendor
Level: Subfunction
Intention in Context: Intention of *Player* is to perform some actions they can do at any point in time
Multiplicity: Multiple *Players* can perform universal actions concurrently.
Primary Actor: *Player*
Secondary Actors: *Players* (who play the roles of opponents)
Main Success Scenario:

1. *Player* requests the *System* to perform an action at any given time. The possible actions are :
    - Pause game for everyone,
    - Leave game session
    - Tweak the game settings,
    - View *Players*' assets:
        i. Every Player's reserved nobles
        ii. Other Player's reserved bonuses
        iii. The Player's own reserved bonuses
        iv. Other Players' reserved bonuses (hides bonuses reserved from deck)
        v. Everyone Player's gems and bonuses
    - Inspect a card on the game board

TakeTurn

Use Case: TakeTurn
Scope: Splendor
Level: Subfunction
Intention in Context: The intention of the *Players* is to play his turn.
Multiplicity: Only one *Player* can take gems in each turn.
Primary Actor: *Player*
Secondary Actors: *Players* (who play the roles of opponents)
Main Success Scenario:

1. *System* informs *Player* that it is his turn.
2. Current *Player* performs one of the following:
    - buy a Card,
    - take Gems,
    - reserve a card.
3. *System* informs *Player* if they are visited by a noble.
4. *System* informs *Player* that it’s the end of their turn.

Extensions:

3a. If *Player* is visited by multiple nobles, they inform the system which one they chose. Use case continues at step 4.

ReserveACard

Use Case: Reserve a Card
Scope: Splendor
Level: Subfunction
Intention in Context: Intention of *Player* is to reserve a development card from the table.
Multiplicity: Only one *Player* can reserve a card in each turn.
Primary Actor: *Player*
Secondary Actors: *Players* (who play the roles of opponents)
Main Success Scenario:

1. *Player* informs *System* about which card they want to reserve.
2. *System* informs *Player* that they received a gold token.
3. *System* informs *Players* of new game state.

Extensions:

1a. *System* ascertains that *Player* already have three reserved cards in hand

1a.1 *System* informs *Player* that they are not allowed to reserve more than three cards. Use case continues at step 3.

2a. System ascertains that there is no gold left. Use case continues at step 3.

TakeGems

Use Case: TakeGems
Scope: Splendor
Level: Subfunction
Intention in Context: The intention of the *Players* is to take gems and to add them to their bank.
Multiplicity: Only one *Player* can take gems in each turn.
Primary Actor: *Player*
Secondary Actors: *Players* (who play the roles of opponents)
Main Success Scenario:

1. Current *Player* informs *System* about how many gems and of which colors they want to add to their bank. They can perform one of the following :
   - take 2 gems of the same color,
   - take 3 gems of different colors.
2. *System* updates current *Player*'s bank.
3. *System* informs *Players* of new game state.

Extension:

1a. *Player* tries to take 2 gems and *System* ascertains color has less than 4 gems left to be taken.
   1a.1. *System* informs *Player*. Use case continues at step 1.

1b. *System* ascertains that one of the color requested by *Player* is out in the game's bank :
   1b.1. System informs Player. Use case continues at step 1.

2a. *System* ascertains that current *Player* has more than 10 gems :
   2a.1. *Player* tells *System* which gem from their bank they want to discard. Use case continues at step 2.

BuyACard

Use Case: BuyACard
Scope: Splendor
Level: Subfunction
Intention in Context: The intention of the *Player* is to buy a card as the move for their turn.
Multiplicity: Only one *Player* can buy a card in each turn.
Primary Actor: *Player*
Secondary Actors: *Players* (who play the roles of opponents)
Main Success Scenario:

1. Current *Player* informs *System* about which card to buy.
2. System informs *Player* of the *Card*'s price. It could be in :
   - Tokens,
   - Cards.
3. *Player* informs *System* about the payment they wish to use for the card :
   - using *Tokens* (data: BonusCard and Tokens to use),
   - using other Cards (data: Cards to discard).
4. Current *Player* informs *System* that they want to proceed with the payment.

Step 5 is skipped if the Card is not a Bonus.

5. Current *Player* informs *System* about which Color the Card should be.

Step 6 is skipped if the Card is not an OrientCard.

6. Current *Player* informs *System* about which item to get at no cost. The item depends on the acquired *Card*, if it was a :
   - level 2 OrientCard (data: face-up level 1 card to acquire),
   - level 2 OrientCard (data: Noble to reserve),
   - level 3 OrientCard (data: face-up level 2 card to acquire).
7. System informs *Player* of new game state.

Extensions:

4a. Current *Player* informs System that he wishes to cancel the proposed trade.

4.a.1 System brings back *Player* to the GameBoard. Use case ends in abandon (success).

Lobby-related use cases

User actions

LogintoLobbyService

Use case: LoginToLobbyService
Scope: System, Lobby Service
Level: Subfunction
Intention in Context: *Player* intends to login to Lobby Service.
Multiplicity: Only one *Player* is allowed to login at a time.
Primary Actor: *Player*
Secondary Actors: *Lobby Service*
Precondition: The *Player* isn't already logged in.
Main Success Scenario:
The login screen/popup is shown by System when Player tries to enter the Lobby screen.

1. *Player* enters credentials into *System*.
2. *Player* submits credentials to *System*.
3. *System* sends login request to the *Lobby Service* with the provided credentials.
4. *Lobby Service* validates the credentials and returns a pair of authentication tokens to the *System*.
5. *System* shows *Player* a "Login Successful" message.
6. *System* automatically redirects *Player* to the Lobby screen.

Extensions:
1a. *Player* informs *System* that they want to cancel the login.

> 1a.1 *System* closes the login popup/screen and redirects *Player* to main menu; use case ends in failure.

2a. *Player* hasn't entered name and/or password; *System* displays the message "Please provide both your name and password"; use case continues at step 1.
2b. *Player's* entered name doesn't adhere to corresponding requirements; *System* displays the message "Your name is invalid" under Login Form; use case continues at step 1.
2c. *Player's* entered password doesn't adhere to corresponding requirements; *System* displays the message "Your password is invalid" under Login Form; use case continues at step 1.
2d. *Player* has entered a name/password combination that doesn't exist in *Lobby Service's* database.

> 4 II. *Lobby Service* returns an error 403. Use case ends in failure.
> 5 II. *System* displays a message "The user with provided credentials doesn't exist, please check and try again" under Login Form; use case continues at step 1.

4a. The *Lobby Service* doesn't respond to the request within 15 seconds or returns an Internal Server Error. Use case ends in failure.

> 5 III. *System* displays a message "Login failed, please try again later" under Login Form; use case continues at step 1.

ManageGameSessions

Use case: ManageGameSessions
Scope: System, Lobby Service
Level: User Goal
Intention in Context: *Player* wants to perform operations on game sessions.
Multiplicity: *Player* can only perform one operation at a time.
Primary Actor: Player
Secondary Actors: Lobby Service
Precondition: The *Player* is already logged in.
Main Success Scenario:

1. *Player* can perform any of the following operations:

- Create a new game session
- Join a game session
- Leave a game session
- Delete a game session
- Launch a game session

CheckAvailableGameSessions

Use case: CheckAvailableGameSessions
Scope: System, Lobby Service
Level: Subfunction
Intention in Context: *Player* intends to see which game sessions are active at a given moment.
Multiplicity: Only one request for game session list can be performed at a time per *Player*.
Primary Actor: *Player*
Secondary Actors: *Lobby Screen*
Precondition: The *Player* is logged in.
Main Success Scenario:
The Player is at the Lobby screen.

1. *System* sends a request to *Lobby Service* to fetch a list of game sessions.
2. *Lobby Service* returns to *System* a list of currently active game sessions.
3. *System* parses the list and displays the sessions to *Player*.

Extensions:
2a. *Lobby Service* returns an Internal Server Error; *System* displays a message "We're having trouble fetching game sessions, please try again later."; use case ends in failure.
2b. *Lobby Service* doesn't respond; *System* waits until a response is returned; use case continues at step 2.

> 3 II. *System* displays a message "There are no active sessions at the moment. Press the Create button to start a game!"; use case ends in success.

3a. *System* cannot parse the list due to invalid/missing attributes returned from *Lobby Service*.

3 III. *System* displays a message "We're having trouble fetching game sessions, please try again later."; use case ends in failure.

CreateGameSession

Use case: CreateGameSession
Scope: System, Lobby Service
Level: Subfunction
Intention in Context: *Player* intends to create a game session.
Multiplicity: *Player* can create multiple game sessions but not simultaneously.
Primary Actor: *Player*
Secondary Actors: *Lobby Service*
Precondition: *Player* is logged in.
Main Success Scenario:
The Player is at the Lobby screen.

1. *Player* instructs *System* that they want to create a new game session.
2. *System* displays Lobby creation client to *Player*.
3. *Player* submits Lobby information into *System*.
4. *System* sends lobby information to *Lobby Service*.
5. *Lobby Service* instantiates game session and updates the list of game sessions for *System*.
6. *System* returns *Player* to the lobby main screen.

Extensions:
It is possible to choose a savegame as a template for lobby settings.
2a. *System* displays previously created savegames; *Player* chooses a savegame as a template for lobby settings; *System* fills in the settings from the savegame for the *Player*; use case continues at step 3.
3a. *Player* doesn't submit anything and exits the lobby creation menu; use case ends in abandon.
3b. *System* is down and can't send the request; *System* UI displays the message "We're having trouble creating this game session, please try again later."; use case ends in failure.
4a. *Lobby Service* returns an Internal Server Error; use case continues at 3b.
4b. *Lobby Service* doesn't respond within 15 seconds; use case continues at 3b.

LaunchGameSession

Use case: LaunchGameSession
Scope: System, Lobby Service
Level: Subfunction
Intention in Context: *Player* intends to launch a game session.
Multiplicity: Each *Player* can only launch 1 game session at a time
Primary Actor: *Player*
Secondary Actors: *Lobby Service, Game Service*
Precondition: *Player* is logged in and is the owner of a session
Main Success Scenario:
The Player is at the Lobby screen.

1. *Player* instructs *System* that they want to launch a new game session.
2. *System* informs *Lobby Service* to start game.
3. *Lobby Service* informs *System* that the game is started.
4. *System* transfers *Player* to the game screen in the *Game Service*.

Extensions:
2a. *System* is down and can't send the request; *System* UI displays the message "We're having trouble launching this game session, please try again later."; use case ends in failure.
3a. *Lobby Service* returns an Internal Server Error; use case continues at step 2a.
3b. *Lobby Service* doesn't respond within 15 seconds; use case continues at step 2a.

DeleteGameSession

Use case: DeleteGameSession
Scope: System, Lobby Service
Level: Subfunction
Intention in Context: *Player* intends to delete a game session.
Multiplicity: *Players* can delete any game sessions they've created.
Primary Actor: *Player*
Secondary Actors: *Lobby Service*
Precondition: *Player* is logged in
Main Success Scenario:
The Player is at the Lobby screen.

1. *Player* instructs *System* that they want to delete a game session.
2. *System* displays deletion confirmation screen to *Player*.
3. *Player* confirms to *System* that they wish to delete the game session.
4. *System* sends request to delete session to *Lobby Service*.
5. *Lobby Service* deletes game session and updates the list of game sessions for *System*.
6. *System* returns *Player* to the lobby main screen.

Extensions:
4a. *Player* cancels deletion request; use case ends in abandon.
4b. *System* is down and can't send the request; *System* UI displays the message "We're having trouble deleting this session, please try again later."; use case ends in failure.
5a. *Lobby Service* returns an Internal Server Error; use case continues in 4b.
5b. *Lobby Service* doesn't respond within 15 seconds; use case continues in 4b.

LeaveGameSession

Use case: LeaveGameSession
Scope: System, Lobby Service
Level: Subfunction
Intention in Context: *Player* intends to leave a game session.
Multiplicity: *Player* can leave as many game sessions as they are in.
Primary Actor: *Player*
Secondary Actors: *Lobby Service*
Precondition: *Player* is logged in and the game session hasn't already been launched.
Main Success Scenario:
The Player is at the Lobby screen.

1. *Player* instructs *System* that they want to leave a game session that they are a part of.
2. *Player* submits Lobby information into *System*.
3. *System* sends lobby information to *Lobby Service*.
4. *Lobby Service* removes the *Player* for the list of players in the game session and updates the list of game sessions for *System*.

Extensions:
3a. *System* is down and can't send the request; *System* UI displays the message "We're having trouble letting you out of the game session, please try again later."; use case ends in failure.
4a. *Lobby Service* returns an Internal Server Error; use case continues at 3a.
4b. *Lobby Service* doesn't respond within 15 seconds; use case continues at 3a.

JoinRunningGameSession

Use case: JoinRunningGameSession
Scope: System, Lobby Service
Level: Subfunction
Intention in Context: *Player* intends to join a running game session.
Multiplicity: Each *Player* can only join 1 game session at a time
Primary Actor: *Player*
Secondary Actors: *Lobby Service, Game Service*
Precondition: *Player* is logged in and there is at least one available game lobby to join.
Main Success Scenario:
The Player is at the Lobby screen.

1. *Player* instructs *System* that they want to join a running game session.
2. *System* informs *Lobby Service* to start game.
3. *Lobby Service* informs *System* that the game is started.
4. *System* transfers *Player* to the game screen in the *Game Service*.

Extensions:
2a. *System* is down and can't send the request; *System* UI displays the message "We're having trouble launching this game session, please try again later."; use case ends in failure.
3a. *Lobby Service* returns an Internal Server Error; use case continues at 2a.
3b. *Lobby Service* doesn't respond within 15 seconds; use case continues at 2a.

JoinGameSession

Use case: JoinGameSession

Scope: System, Lobby Service

Level: Subfunction

Intention in Context: *Player* intends to join a game session.

Multiplicity: *Player* can join multiple game sessions but not simultaneously.

Primary Actor: *Player*

Secondary Actors: *Lobby Service*, *Game Service*

Precondition: *Player* is logged in and there is at least one available game lobby to join.

Main Success Scenario:

The Player is at the Lobby screen.

1. *Player* instructs the *System* that they want to join a game session.
2. *System* informs the *Lobby Service* that the *Player* wants to join the game session.
3. *Lobby Service* informs *System* that *Player* has joined.
4. *System* transfers *Player* to the Lobby Info Screen.

Extensions:

2a. *System* is down and can't send the request; *System* UI displays the message "We're having trouble joining this game session, please try again later."; use case continues at step 2.

3a. *Lobby Service* returns an Internal Server Error; see extension 2a.

3b. *Lobby Service* doesn't respond within 15 seconds; see extension 2a.

3c. *Lobby Service* returns a "Game is full" error; *System* UI displays the message "This game is full, please try again later."; use case ends in failure.

If the game session is already running, only the players that joined the game session when it was launched can re-join it.

3d. *Lobby Service* returns a "Game is already running" error; *System* UI displays the message "This game is already running, please choose another one."; use case ends in failure.

4a. The game session is already running; *System* transfers *Player* to the screen in the *Game Service*.

TweakSettings

Use case: TweakSettings
Scope: System, Lobby Service
Level: User Goal
Intention in Context: Player intends to change their personal settings.
Multiplicity: Players can change their settings as many times as they want, but not simultaneously.
Primary Actor: Player
Secondary Actors: Lobby Service
Precondition: Player is logged in
Main Success Scenario:

The Player is at the Settings screen.

1. *Player* can change any of the following settings:
   - Preferred color
   - Screen Resolution
   - Sound level
2. *System* sends a request to *Lobby Service* to update the *Player*'s chosen settings.
3. *Lobby Service* notifies *System* of successful settings change.

Extensions:

If Player is an Admin, they can modify another Player's preferred color via the Edit button in the Player List screen. All other aspects of this use case (including extensions) are the same.

1a. *Player* doesn't click/press anything; *System* is blocked from sending requests until at least one setting has changed; use case ends in failure.
1b. *Player* clicks/presses the Close Button; *System* redirects *Player* to the previous screen; use case ends in failure.
4a. *Lobby Service* returns an Internal Server Error; *System* displays a message "We're having trouble updating settings, please try again later."; use case continues at step 1.
4b. *Lobby Service* doesn't respond within 15 seconds; see extension 4a.

Admin-exclusive actions

AddNewPlayerToDatabase

Use case: AddNewPlayerToDatabase
Scope: System, Lobby Service
Level: Subfunction
Intention in Context: *Player* intends to create an account for a new *Player*.
Multiplicity: Only one request for account creation can be performed at a time per *Player* with unique payload.
Primary Actor: *Player*
Secondary Actors: *Database, Lobby Service*
Precondition: The *Player* is logged in and is an admin.
Main Success Scenario
The admin Player clicked on "Create User" button on Player List screen and is redirected to the Register Player screen.

1. *Player* informs *System* of name and password and selects preferred color of the new account in corresponding form fields.

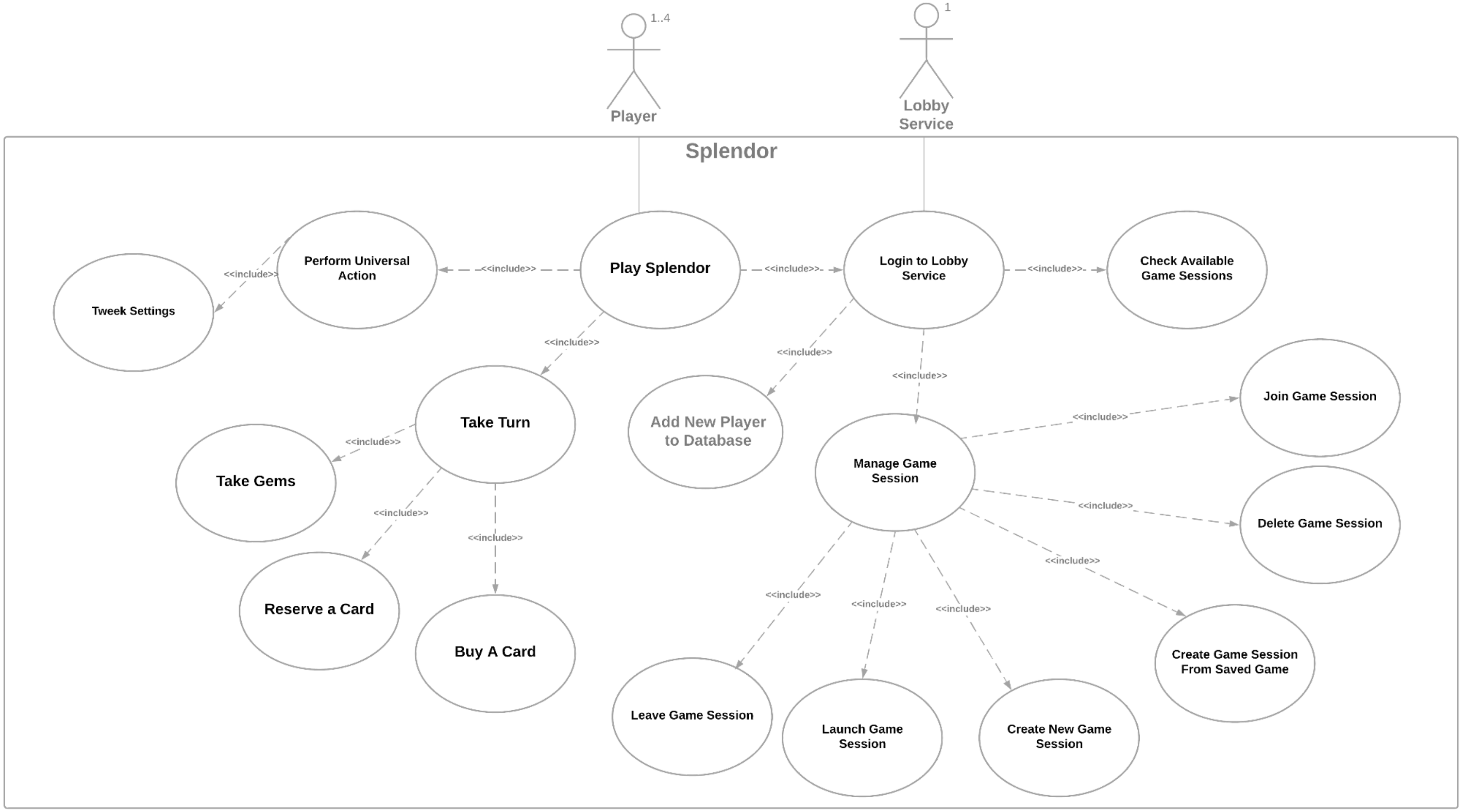
2. *System* sends a request to the *Lobby Service* to create a new player account.

3. *Lobby Service* informs *System* that the account was created successfully.

4. *System* redirects *Player* back to the User List screen and the newly created account is shown at the top of the list.

Extensions:
1a. *Player* cancels creation of player; *System* redirects *Player* back to the Player List screen; use case ends in abandon.
2a. *System* is down and can't send the request; *System* UI displays the message "We're having trouble setting your color, please try again later."; use case ends in failure.
3a. *Lobby Service* returns an Internal Server Error; use case continues at step 2a.
3b. *Lobby Service* doesn't respond within 15 seconds; use case continues at step 2a.
3c. *Lobby Service* returns an error saying that an account with this name exists; *System* UI displays the message "This username is already taken, please enter another one."; use case continues at step 1.

Use Case Diagram


1..4
Player
1
Lobby Service
Splendor
Tweek Settings
<<include>>
Perform Universal Action
<<include>>
Play Splendor
<<include>>
Login to Lobby Service
<<include>>
Check Available Game Sessions
<<include>>
<<include>>
Take Turn
<<include>>
Take Gems
Add New Player to Database
<<include>>
Manage Game Session
<<include>>
Join Game Session
<<include>>
Delete Game Session
<<include>>
<<include>>
Reserve a Card
<<include>>
Buy A Card
<<include>>
<<include>>
<<include>>
Create Game Session From Saved Game
Leave Game Session
Launch Game Session
Create New Game Session