**TakeSameColouredGems**

**Use Case**: TakeSameColouredGems

**Scope**: Splendor

**Level**: Sub Function

**Intention in Context**: Intention of the Players is to take 2 gem tokens of the same colour from the gem piles and add them to their inventory.

**Multiplicity**: Only one Player can take gem tokens simultaneously.

**Primary Actor**: Player

**Secondary Actors**: Player (who play the role of opponent)

**Main Success Scenario**:

1. Current *Player* informs *System* that they would like to take two gems of the same colour.

2. *System* informs current *Player* which piles they can choose from (the pile cannot have less than 4 tokens in it).

3. Current *Player* informs *System* which pile they would like to choose 2 gems from.

4. *System* informs *Players* of new game state.

**Extensions**:

2a. *System* determines that there are no possible piles for the current *Player* to choose; the use case ends in failure

3a. *System* determines that the amount of gem tokens in the current *Player*’s inventory exceeds 10

3a.1. *System* informs *Player* to return tokens until they only have 10 tokens left in their inventory.

3a.2. *Player* informs *System* which tokens they would like to return; the use case continues at step 4.

**TakeDifferentColouredGems**

**Use Case**: TakeDifferentColouredGems

**Scope**: Splendor

**Level**: Sub Function

**Intention in Context**: Intention of the Players is to take 3 gem tokens of different colours from the gem piles and add them to their inventory.

**Multiplicity**: Only one Player can take gem tokens simultaneously.

**Primary Actor**: Player

**Secondary Actors**: Player (who can view the current game state)

**Main Success Scenario**:

1. Current *Player* informs *System* that they would like to take three gems of the same colour.

2. *System* informs current *Player* which piles they can choose from (the pile cannot be empty).

3. Current *Player* informs *System* which gem token they would like to take.

*Steps 2 and 3 are repeated for each token.*

4. *System* informs *Players* of new game state.

**Extensions**:

2a. *System* determines that there are no possible piles for the *Player* to choose; the use case ends in failure

3a. *System* determines that the amount of gem tokens in the current *Player*’s inventory exceeds 10.

3a.1. *System* informs current *Player* to return tokens until they only have 10 tokens left in their inventory.

3a.2.Current *Player* informs *System* which tokens they would like to return; the use case continues at step 4.