|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Player moves character | ID: 110 | | Importance: High |
| Primary Actor: Dealer | | Use Case Type: Business | |
|  | | | |
| Brief Description: The player clicks the directional key, and the character moves in the specified direction. | | | |
| Trigger: After the game starts, the screen displays the character, and the player clicks the directional keys. | | | |
| Relationships:  Association: Player  Include: null  Extend: Player attacks entity  Generalization: null | | | |
| Normal Flow of Events:  1. Entering the game screen displays the character  2. Player clicks the direction keys  3. The game checks whether it is within the legal beat window and jumps to sub-process 1.  4. On the beat, the game performs collision detection and jumps to subroutine 2.  5. Allow movement, execute movement state action strategy | | | |
| SubFlows：  Sub1: Game checks the beat window   1. The game checks whether the player's keystroke timing falls within the beat window.   If not within the beat window  The game ignores player keystrokes and displays the prompt "Beat Skipped".  If the beat is illegal (the player enters in a non-ready state)  The game ignores player keystrokes and displays the prompt "Beat Skipped".  Sub2: Game collision detection   1. The game checks whether the player's target position is unoccupied.   If the target is a boundary/occupied  Enter subprocess 3  Other, the game setting status is mobile  Sub3 State Machine Detection   1. Game detects whether the current state allows movement.   If the target position is already occupied  Game settings status set to attack  If the target position is the boundary  Game settings status is ready  Other, the game settings status is ready. | | | |
|  | | | |
|  | | | |