### UC *01* – *Load a game from a file*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Version | Date | Author | | Changes |
| 1 | 18.01.2015 | Bartsch & Niederhumer | | None |
| Scope & Level | Loading, primary | | | | |
| Goal in context | Game is loaded and waiting for user input. | | | | |
| Preconditions | * Have a valid JSON file containing a JTextadventure game. * Have Java RE installed on the system. * Know how to run a jar file with an argument. | | | | |
| Successful outcome | Game starts and loads the provided game file. | | | | |
| Failure outcomes | Failure | | | Outcome | |
| 1. File not found. 2. File invalid. | | | 1. Program exits. 2. Program exits. | |
| Primary actor | User | | | | |
| Secondary actors | JTextadventure (System) | | | | |
| Main scenario | * User: Navigate to jar folder and run the jar with the path to a valid game file. * System: Check if the file exists, and parse it. * System: Display the start message and wait for user input. * USECASE ENDED | | | | |
| Alternatives | 1. Program outputs error message, indicating that the file was not found. 2. Program outputs error message, indicating that the file was invalid. | | | | |

### UC *02 – Enter commands and change game state*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Version | Date | Author | | Changes |
| 1 | 18.01.2015 | Bartsch & Niederhumer | | None |
| Scope & Level | Gameplay, primary | | | | |
| Goal in context | Change current gamestate. | | | | |
| Preconditions | * Game is successfully loaded. * Game is not yet finished. | | | | |
| Successful outcome | Gamestate has changed via user commands. | | | | |
| Failure outcomes | Failure | | | Outcome | |
| 1. Command is invalid. 2. Command parameters are invalid. 3. Requested interaction is not possible. | | | 1. State is not changed. 2. State is not changed. 3. The requested action is not performed. | |
| Primary actor | User | | | | |
| Secondary actors | JTextadventure (System) | | | | |
| Main scenario | * System waits for user input. * User inputs valid command. * System parses command and performs interaction. * System displays new state to the user. * USECASE ENDED | | | | |
| Alternatives | 1. Display error message and possible similar commands. 2. Display error message and possible similar parameters (objects). 3. Display message why the interaction is not possible. | | | | |
| Related information | * Parameters depend on the interaction * Objects are targets for interaction defined in the game file. | | | | |