SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS) FOR

CS383 Spring 2014 Project: Swords and Sorcery

Version 1.0
5/9/14

Prepared for:
CS 383 Software Engineering Spring 2014

Prepared by:
Software Engineering Class Spring 2014
University of Idaho
Moscow, ID 83844-1010

SWORDS AND SORCERY SSRS
TABLE OF CONTENTS

Section Page

1 Introduction

1.1 IDENTIFICATION

The system which this document applies is titled Swords and Sorcery, or S & S for short. It is currently in version 1.0.

1.2 PURPOSE

The purpose of the system under development is to create a computerized version of the board game Swords and Sorcery. While the system will be used by anyone who wants to play the game , this document is intended to be read and understood by University of Idaho computer science software designers and coders.

1.3 SCOPE

This project began development at the beginning of Spring Semester 2014. As a class we began by identifying requirements, we then moved onto a brief phase of design before jumping right into implementation. We have done some testing, but most of this project is still underdevelopment. This project is sponsored by Dr. Jeffery, and is developed by the Software Engineering class. Currently the project is able to be ran on any machine capable of running the Java Virtual Environment.

1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

Term or Acronym	Definition
Alpha test	Limited release(s) to selected, outside testers
Beta test	Limited release(s) to cooperating customers wanting early access to developing systems
Final test	aka, Acceptance test, release of full functionality to customer for approval
DFD	Data Flow Diagram
SDD	Software Design Document, aka SDS, Software Design Specification
SRS	Software Requirements Specification
SSRS	System and Software Requirements Specification
S & S	Swords and Sorcery
TCP/IP	Transmission Control Protocol and Internet Protocol

1.5 OVERVIEW AND RESTRICTIONS

This document is for limited release only to UI CS personnel working on the project.

Section 2 of this document describes the system under development from a holistic point of view. Functions, characteristics, constraints, assumptions, dependencies, and overall requirements are defined from the system-level perspective.

Section 3 of this document describes the specific requirements of the system being developed. Interfaces, features, and specific requirements are enumerated and described to a degree sufficient for a knowledgeable designer or coder to begin crafting an architectural solution to the proposed system.

Section 4 provides the requirements traceability information for the project. Each feature of the system is indexed by the SSRS requirement number.

Sections 5 and up are appendices including original information and communications used to create this document.

2 OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

Swords and Sworcery is an independent product and all of it's functions are completely self-contained.

2.2 PRODUCT FUNCTIONS

This project's main function is to create a computerized version of the board game Swords and Sorcery, specifically the army game. This includes the game being a multi-player game to be playable over a network. The game functions are as follows

- Create/Join a game
- Play through the different phases:
  - Movement: move units and characters on the board
  - Combat: complete combat between two different stacks of units
  - Spell: cast a spell with a character
- Communicate the results of these phases over the network to other instances of the game
- Communicate with other players through chat over the network
- End the game

2.3 USER CHARACTERISTICS

The intended users of this product are anyone who wants to play the game. The user should have an education level of at least middle school so they can understand the rules of the game. They should have some experience of the rules of Swords and Sorcery. The intended user needs no technical expertise to run this application.

2.4 CONSTRAINTS

Swords and Sorcery is constrained by the rules set by the physical board game. Also due to the networking it is constrained to following the protocols over the network. Hardware requirements are just having enough memory to hold and run Swords and Sorcery, and have the hardware necessary to run the Java Run-Time Environment.

2.5 ASSUMPTIONS AND DEPENDENCIES

We assume that the Java Run-Time Environment will be able to run Java 8 code, but if future versions of the Java Run-Time Environment do not support Java 8 code then the Swords and Sorcery code base will need to be updated accordingly.

2.6 SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS

2.6.1 Site dependencies

Swords and Sorcery has very few site dependencies. It requires a mouse and keyboard for input and a monitor for output. To run Swords and Sorcery requires the Java Run Time Environment that can run Java 8 applications. Also the computer needs to have a network connection.

2.6.2 Safety, security and privacy requirements

Sword and Sorcery had no safety, security, or privacy requirements.

2.6.3 Performance requirements

Swords and Sorcery software application should be able to support up to seven users playing the same game over the network.

2.6.4 System and software quality

This project shall have the ability to perform all of the required functions with consistent results. Swords and Sorcery shall also be able to be playable on any machine capable of running Java as that is what Swords and Sorcery will run on. It shall also be able to be used by anyone capable of understanding the rules to the board game.

2.6.5 Packaging and delivery requirements

The game and all of its documentation is available at https://github.com/cjeffery/sworsorc

2.6.6 Personnel-related requirements

The system under development has no special personnel-related characteristics.

2.6.7 Training-related requirements

No training materials or expectations are tied to this project other than the limited help screens built into the software and the rule book for the board game.

2.6.8 Logistics-related requirements

The system requirements for Swords and Sorcery are any machine capable of running Java 8 run time environment.

2.6.9 Precedence and criticality of requirements

All requirements stated above have equal weight, since there are no requirements for safety, security, or privacy all of the requirements were given an equal weight.

3 SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 Hardware Interfaces

Swords and Sorcery did not require any hardware interfaces.

3.1.2 Software Interfaces

Swords and Sorcery only required TCP/IP as the only software interface due to needing a network connection.

3.1.3 User Interfaces

The user interface required for Swords and Sorcery is the Java Run-Time Environment. This is a requirement resulting from the implementation not the design of the project.

3.2 SYSTEM FEATURES

3.2.1 Use Case Diagrams

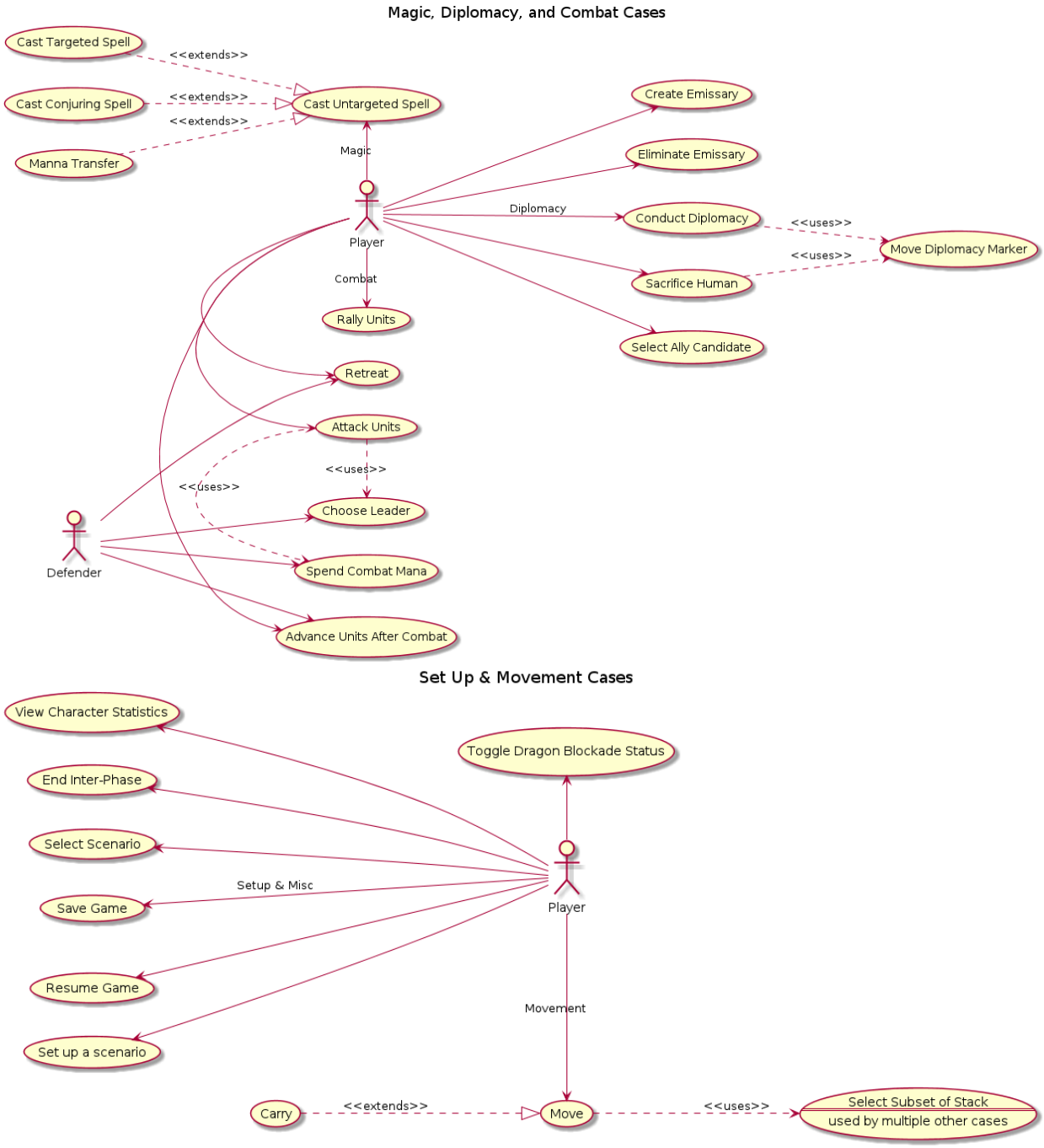
A pdf of our use case diagrams is inserted starting on the following page.

CS 383 Swords & Sorcery
Master Use Case Descriptions & Diagram

February 18, 2014

Contents

Use Case Diagrams



1 New Game

Actor: Player

Goal: To start a new game

Summary: Player selects options from a menu, choosing what game to initiate.

Steps:

1. Player selects "new game".
2. System displays a list of scenarios.
3. Player selects a scenario.
4. Player selects what nation they will play.
5. System adds the game to the list available in the Lobby.
6. Player proceeds to Set Up.

2 Join Game

Actor: Player

Goal: To join an available game

Summary: Player selects game from the Lobby.

Steps:

- 1. System displays a list of available games/scenario/opponent in the Lobby from main menu.
2. Player selects a game.
3. Player selects what nation they will play.
4. System displays faction information such as victory conditions and starting units.
5. Player confirms selection.
6. System informs opponent(s) of the new player.
7. Player proceeds to Set Up.

3 Set Up

Actor: Players starting a game

Goal: To do scenario set-up

Summary: Player places the allocated units in valid hexes.

Steps:

-	1. System displays units to be placed and valid hexes.
2. Player places units in appropriate hexes until all available units from all players have been placed.	
3. Player indicates when they are 'satisfied' with placement.	4. System toggles "satisfied" off whenever enemy units are placed.
-	5. Scenario begins whenever all players are "satisfied".

4 Resume Game

Actor: Player

Goal: To resume a saved game

Precondition: A saved game exists. The player wishes to reconnect to the existing game rather than a new one.

Summary: The player restores a previously saved state

Related:

Steps:

1. The player selects the "Resume Game" option.	2. The system displays the available saved games.
3. The player selects the desired save file.	4. The system displays a prompt.
-	5. The system loads the saved state.

5 Save Game

Actor: Player

Goal: To save the current game state for later play

Precondition: A game is currently being played

Summary: The player selects a save option and the system saves the current state.

Related:

Steps:

1. The player selects the "Save Game" option.	2. The system displays a list of available save slots.
3. The player selects the desired slot.	4. The system displays the relevant confirmation prompt.
-	5. The system saves the current state of the game.

6 Select Scenario

Actor: Hosting Player

Goal: To select a pre-made game scenario.

Precondition: A new game has been initiated

Summary: The hosting player selects a pre-made game scenario, for a new game.

Related:

Steps:

-	1. The system displays the list of available pre-made scenarios for the hosting player to select.
2. The player selects the desired scenario from the list.	3. The system displays the relevant confirmation prompt.
-	4. The system loads the selected scenario.

7 End Inter-Phase

Actor: Player

Goal: To end the current player's phase.

Precondition: The game is in a phase the player has control to end.

Summary: The player wishes to end their current phase early. If the rules are not violated the turn is ended.

Related:

Steps:

1. The player selects an "End Phase" option	2. The system displays the relevant confirmation prompt.
-	3. The system ends the current phase.

8 Select Ally Candidate

Actors: All Players

Goal: To determine player alliances for the game-turn.

Precondition: It is currently the player-Order Determination Inter-Phase.

Summary: Each player selects any other player(s) they would like to be allied with for the turn.

Steps:

-	1. The system displays to each player a list of potential allies (the other, eligible, players).
2. The players select any other player(s) they wish to ally with for the game-turn.	
3. The player indicates they are finished selecting hopeful allies.	4. The system moves to the next phase.

9 Move

Actor: Player

Goal: To move unit(s) or character(s) from one tile to another

Summary: Move a unit, character, monster, stack, or subset of a stack from a starting tile to a destination tile, depending on terrain or movement points.

Precondition: It is the player's movement phase

Related: Carry, **Select Subset of Stack**

Steps:

1. The player selects a unit, character, monster, stack, or stack subset.	2. The system displays what hexes are reachable by those unit(s) based on their remaining Movement Points.
3. The player selects a destination hex to move the unit(s) to.	4. The unit(s) are moved and their movement points are subtracted accordingly.

Alternatives:

1. If unit is moved into an enemy zone of control, it's movement might immediately end.
2. If a unit movement would trigger invasion, the player is prompted to confirm the action.
3. If a unit move ends on an enemy character, the character is captured.
4. The user can cancel movement after being shown the reachable hexes.
5. Some unit(s) will not be able to make a valid movement for various reasons, in this case the user must cancel the movement.

10 Teleport

Actor: Player

Goal: To carry a character using a flying unit

Summary: Flying units or monsters can carry other characters if said other characters haven't moved yet.

Precondition: It is the movement phase and the player has a flying unit that has room on it's back, in range of a character that hasn't moved yet

Related: Movement

Steps:

1. Player moves on top of a portal hexagon.
2. Player is provided a dialog giving them the option to use the portal.
3. the player chooses to teleport his units individually or as a group.
4. Perform appropriate teleportation.

Alternatives:

1. Should an enemy unit occupy an output portal, the teleported units should be retreated by one tile.
2. A player can choose not to teleport in step 2.
3. A player may have a spell (Teleport Protection, Teleport Control) affect the outcome of step 4.

11 Carry

Actor: Player

Goal: To carry a character using a flying unit

Summary: Flying units or monsters can carry other characters if said other characters haven't moved yet.

Precondition: It is the movement phase and the player has a flying unit that has room on it's back, in range of a character that hasn't moved yet

Related: Movement

Steps:

1. The player moves a flying unit over unmoved character(s) as per **Movement**
2. The system asks which, if any, characters should be carried.
3. The player indicates which, if any, characters they want carried.
4. Those characters are stacked with the flying unit for the remainder of the movement phase.

12 Select Subset of Stack

Actor: Player

Goal: To select one or more units out of a stack

Summary: There's lots of unit stacking in Swords & Sorcery. The player needs an easy way to both use a stack as a whole or to use specific units from a stack.

Steps:

1. The user selects a stack.
2. The user is given the choices of using the stack as a whole, selecting one or more units out of the stack, or cancelling the selection.
3. The user either cancels or selects the desired unit(s).
4. Those units become selected.
5. The user continues on with whatever action they were going to do using the selected units.

Alternatives:
1. In some states it might only make sense to select a single unit out of a stack. In this case the user is not given the choice of selecting multiple units or the whole stack
2. In some states such as **Eliminate Unit** the player may be forced to select some number of units. In this case there is no option to cancel

13 Attack Units

Actor: Player

Goal: Attack opposing player through combat

Precondition: Player is in the Combat Resolution Segment

Summary: Player selects units to attack and unit to be attacked. Combat is resolved by the system.

Related: Choose Leader, **Spend Combat Manna**

Steps:
1. Attacker selects units to attack.
2. Attacker selects unit to be attacked.
3. System notifies defending player.
4. If multiple leaders (Choose Leader).
5. If magic capable leaders (Spend Combat Manna).
6. System displays effects of combat.

Alternatives:
1. User deselects units in step 1.
2. Combat is not valid in step 2.

14 Advance Units After Combat

Actor: Player

Goal: To move units forward into the enemies retreating path

Precondition: Player has won combat and opposing player has retreated units.

Summary: Player can choose to move units into retreating path of enemy after combat.

Steps:
1. Player selects units to advance.
2. Player selects path of advance.
3. System checks if path is valid and displays result.

Alternatives:
1. Player chooses not to move.
2. Player deselects units in step 1.
3. Path is not valid so unit does not move.
4. If a unit advance ends on an enemy character, the character is captured.

15 Retreat

Actor: Player

Goal: To retreat units after losing combat

Precondition: Player has just lost combat

Summary: The player can choose to retreat units after combat or kill off units.

Steps:
1. System shows player units to be retreated.
2. Player selects units to kill.
3. Player chooses path for remaining units.
4. System checks if path is valid and displays results.

Alternatives:
1. Units are all killed in combat.
2. Player kills all units in step 2.
3. Path not valid in step 3

16 Choose Leader

Actor: Player

Goal: To select one leader if multiple are present

Precondition: Player has more than one leader.

Summary: Player chooses which leader to use.

Steps:
1. Player selects leader from stack.
2. System shows player which leader is to be used.

Alternative: Player deselects unit in step 1. Leader is not allowed to be used.

17 Spend Combat Manna

Actor: Player

Goal: Cast magic from leaders in combat

Precondition: Player is in combat and has magical leader with manna points

Summary: Player chooses which leader to use.

Steps:
1. Player selects character.
2. Player chooses amount of manna to spend.
3. System shows player combat strength of spell.

Alternatives:
1. Player deselects unit in step 1.
2. Player does not have enough manna in step 2.

18 Rally Units

Actor: Player

Goal: Attempt to rally units that have been demoralized

Precondition: Player has demoralized units and leader is present in stack. player in Unit Rallying Segment of Army Combat Phase

Summary: Player chooses leader to use if multiple and is shown if rally was passed.

Steps:

1. Player selects demoralized units to be rallied.
2. Player selects leader to rally with, if there are more than one.
3. System shows result of rally.

Alternatives:

1. Player deselects units in step 1.
2. Leader is not allowed to be used.
3. Units cannot be rallied.

19 View Character/Unit Statistics

Actors: Player

Goal: view the statistics and current state of a certain character, unit, or monster in play.

Precondition: a character, unit, or monster is in play.

Summary: The user performs this action when he wishes to view the full statistics relating to a character, unit, or monster in play. It is similar to consulting the characters card in the manual system.

Steps:

1. Player selects the character, unit, or monster
2. System displays a menu asking what the user wants to do with the selected object (may include Move, Attack, Cast spell, View, etc. . . depending on game phase).
3. Player chooses View.
4. System displays the character, unit or monsters relevant statistics (those on the game card plus manna level and movement points remaining if applicable) in a new window
5. User closes the window.

Alternative: User can choose Cancel instead in Step 3 to close the menu.

20 Cast Untargeted Spell

Actors: Player

Goal: To cast a particular spell on a particular target, which may be a group of hexes, hex sides, characters, or units.

Precondition: Player fulfills a set of preconditions for at least one spell. These preconditions depend on the characters available, their manna levels, and the game phase.

Summary: The actor will choose one of his characters, and select a spell.

Steps:

1. Player selects one of his or her characters with a mouse click	2. System displays a menu asking what the user wants to do with the character (may include Move, Attack, Cast spell, View, etc. . . depending on game phase).
3. Player chooses Cast spell.	4. System displays a window listing all spells that the character is capable of attempting given its Magic Power Level, current Manna points, the game phase, which types of spells the character has cast that turn, and how many game-turns are left in the game play.
5. User selects one of the untargetted spells.	6. System calculates the result if the character was attempting a higher order spell, then makes internal adjustments to create the force walls.
-	7. System reports the spells results in a dialog box.
8. User accepts results, closing dialog box.	

Alternative: User may choose to cancel in steps 3, or 5.

21 Cast Targeted Spell

Actors: Player

Goal: To cast a particular spell on a particular target, which may be a group of hexes, hex sides, characters, or units.

Precondition: Player fulfills a set of preconditions for at least one spell. These preconditions depend on the characters available, their manna levels, the targets available, and the game phase.

Summary: The actor will choose one of his characters, select a spell, and then choose one or more targets based on the cost of the spell.

Related: Uses "Select Subset of Stack" and extends "Cast Untargeted Spell".

Steps:

1. Player selects one of his or her characters with a mouse click	2. System displays a menu asking what the user wants to do with the character (may include Move, Attack, Cast spell, View, etc. . . depending on game phase).
3. Player chooses Cast spell.	4. System displays a window listing all spells that the character is capable of attempting given its Magic Power Level, current Manna points, the game phase, which types of spells the character has cast that turn, and how many game-turns are left in the game play.
5. User selects one of the targetted spells.	6. System highlights possible target hex sides and asks the user to choose one.
7. User selects up to as many of the specific type of target as his character can afford to cast the spell on at a cost of 2 Manna points per hex side. When done, user can press continue.	8. System calculates the result if the character was attempting a higher order spell, then makes internal adjustments to create the force walls.
-	9. System reports the spells results in a dialog box.
10. User accepts results, closing dialog box.	

Alternative: User may choose to cancel in steps 3, 5, or 7.

22 Cast Conjuring Spell

Actors: Player

Goal: To conjure a particular type of unit to aide the player.

Precondition: It must be the player's Magic phase. The user must have a character of attempting a conjuring spell and have Manna points for it. No non-conjuring spells may have been cast by this character this game-turn.

Summary: The actor will choose one of his characters, select a spell, and then choose how long the unit is to remain in play.

Related: Extends "Cast Untargeted Spell".

Steps:

1. Player selects one of his or her characters with a mouse click	2. System displays a menu asking what the user wants to do with the character (may include Move, Attack, Cast spell, View, etc. . . depending on game phase).
3. Player chooses Cast spell.	4. System displays a window listing all spells that the character is capable of attempting given its Magic Power Level, current Manna points, the game phase, which types of spells the character has cast that turn, and how many game-turns are left in the game play.
5. User selects one of the targetted spells.	6. System highlights possible target hex sides and asks the user to choose one.
7. User selects up to as many game-turns as he can afford, given the cost of the spell.	8. System calculates the result if the character was attempting a higher order spell, then makes internal adjustments to create the force walls.
-	9. System reports the spells results in a dialog box.
10. User accepts results, closing dialog box.	

Alternative: User may choose to cancel in steps 3, 5, or 7.

23 Cast CounterSpell

Actors: Player

Goal: To negate the effects of certain spells

Precondition: It must be an opponent player's CounterSpell segment. The opponent must have just cast a counter-able spell: Fear, etc. The Player must have a magic character with a level high enough to cast the appropriate counter spell.

Summary: All Opponent players may choose to cast counterspells, without knowing whether each other are doing so [e.g. chat cut-off].

Steps:

-	1. System displays a dialog informing the user that so-and-so has cast the such-and-such spell, and asks if they want to counter it.
2. Player chooses Cast counter spell.	3. System displays a window with the outcome.

24 Manna Transfer

Actors: Player

Goal: To transfer Manna between a player's characters.

Precondition: It must be this players Magic phase, and the user must have a character capable of attempting a rst level spell, and this character must have 2 Manna points. Another of the players magic-capable characters must reside in the same hex as the casting character.

Summary: the user selects a character, chooses this spell, and then selects how much Manna to transfer and to what character.

Related: Extends "Cast Untargeted Spell".

Steps:

1. Player selects one of his or her characters
2. System displays a menu asking what the user wants to do with the character (may include Move, Attack, Cast spell, View, etc. . . depending on game phase).
3. Player chooses Cast spell.
4. System displays a window listing all spells that the character is capable of attempting given its Magic Power Level, current Manna points, the game phase, which types of spells the character has cast that turn, and how many game-turns are left in the game play.
5. User selects Manna Transfer.
6. System asks user to select a character to transfer Manna to.
7. User selects one of his characters in the same hex as the casting character.
8. System asks how much Manna is to be transfered.
9. User enters a number no greater than half of the casting character's current Manna level and no greater than the receiving character's maximum Manna level.
10. System calculates the result if the character was attempting a higher order spell, then makes internal adjustments to transfer the Manna.
-
11. System reports the spell's results in a dialog box.
12. User accepts results, closing dialog box.

Alternative: User may choose to cancel in steps 3, 5, 7, or 9.

25 Create Emissary

Actor: Player

Preconditions: Friendly movement phase; character with diplomatic rating greater than 0; character has 1 or 0 emissaries.

Summary: The player creates an emissary

Steps:

1. User selects character which will spawn emissary.
2. System creates the emissary and updates the board.

26 Eliminate Emissary

Actor: Player A, Player B

Precondition: Emissary occupies same hex as enemy unit(s).

Summary: One player eliminates another players emissary.

Steps:

1. Player A indicates they wish to terminate Player B's emissary.
2. System removes emissary and updates the game board.

27 Conduct Diplomacy

Actor: Player

Precondition: Player controls an emissary or character that occupies a neutral Capital hex.

Summary: Player conducts diplomacy using a character or emissary at a neutral Capital.

Related: Move Diplomacy Marker

1. User chooses to conduct diplomacy.
2. System indicates which emissary or character pieces are eligible.
3. User selects emissary or character.
4. System calculates results of diplomacy.
5. User moves diplomacy marker as in **Move Diplomacy Marker**

Alternative: System calculates a negative result for diplomacy and moves the diplomacy marker itself.

28 Move Diplomacy Marker

Actor: Player

Goal: Move a player's diplomacy marker.

Precondition: Positive results from **Conduct Diplomacy** or **Sacrificing a Unit**.

Summary: Player moves their diplomacy marker on the Diplomacy Track.

Steps:

- 1. System displays Diplomacy Track.
2. User moves Diplomacy marker and confirms movement.
3. System updates Diplomacy Track.

Alternative: User moves Diplomacy marker to a players hex. That neutral is now allied with that player and the diplomacy marker is removed from play.

29 Sacrifice

Actor: Player

Goal: Sacrifice a character or unit in order to sway a neutral's allegiance.

Precondition: Movement is done during movement phase. The sacrifice is done during the diplomacy phase.

Summary: A player sacrifices a unit or eligible character.

Related: Move Diplomacy Marker

Steps:

1. User moves character, unit, or captured character to a hex adjacent to a character or unit of the neutral.
2. During the diplomacy the system removes the "sacrifice"
3. User moves neutral diplomacy marker as in **Move Diplomacy Marker**

Alternative: User isn't allowed to sacrifice characters because of their religion.

30 Toggle Dragon Blockade Status

Actor: Player

Goal: Change the status of blockades in a dragon tunnel complex.

Precondition: The dragon must be inside or adjacent to an entrance of a Dragon Tunnel Complex.

Summary: The player controlling the dragon chooses to either blockade entrances to a tunnel complex, or to remove a blockade. Either of these replace the dragon's movement for that game-turn.

Steps:

1. User selects a dragon player and which type of blockade change will be happening
2. System asks which hex sides will be effected by the change
3. User selects one or more hex sides.

31 Attempt to Escape

Actor: Player

Precondition: Happens during manna regeneration phase. A Player must have a captured character.

Summary: Captured characters can attempt to escape their grisly fate.

Steps:

-

1. System asks during manna regeneration whether the user wants character to attempt escape.
2. User chooses to attempt escape.
3. System calculates results of escape attempt.

Alternatives:

1. User chooses not to escape
2. Magic-using characters are prompted prior to step 3 whether they wish to use manna to aid their escape.

32 Eliminate Prisoner

Actor: Player

Precondition: Diplomacy interphase. A Player must have a captured character in a capitol.

Summary: Captured characters can be executed.

Steps:

-	1. System asks during diplomacy interphase whether the user wishes to execute captured characters in capitols.
2. User chooses to execute the prisoner.	3. System updates state and maps and informs players of the grisly result.

Alternative: User chooses not to execute (for now).

33 Eliminate Unit from Full Stack

Actor: Player

Precondition: The player ends an inter-phase with an over-sized stack that has to be trimmed.

Summary: The player selects which units to eliminate from a stack that is too full

Related: Select from stack

Steps:

-	1. The game shows the player the units in an oversized stack.
2. The player selects units to eliminate until the stack is small enough.	