CS383: Software Engineering

HW1
Spring 2014

Sean Shepherd (4, 5, 6)
shep1244@vandals.uidaho.edu
Chi-Hsiang Wang (7, 8, 9, LaTeX Layout)
wang0162@vandals.uidaho.edu
Johnathan Flake (10, 11, 12)
flak4202@vandals.uidaho.edu
Wayne Fuhrman (13, 14, 15)
fuhr0438@vandals.uidaho.edu
Keith Drew (16, 17, 18, 19)
keithd@vandals.uidaho.edu
Joe Higley (20)
higley@vandals.uidaho.edu

Contents

1 Use Case Description

1.1 Pre-Game (Shepherd Sean)

1.1.1 Playing a scenario

Goal	Complete a single game-turn
Summary	Player is directed through the 7 steps of a game-turn by the computer.
Actors	• Human player(s)
Related Use Cases	• Inform Players of Random Event • Determine Player Order • Form Alliances

Steps	• Players are informed of any random events. • Players attempt to form any alliances they wish for. • Players are informed of the order in which they will take their turns. • The first player begins his turn: 1. Movement Phase 2. Magic Phase a) Spell Segment b) Counterspell Segment 3. Combat Phase a) Unit Rallying Segment b) Magic Allocation Segment c) Combat Resolution Segment • Other players go through their turns in order in the same manner as the first player. • Players observe the movements of randomly-moving units. • Players attempt diplomacy with neutrals. • Players are informed of their mana regeneration and the position of the three suns.

1.1.2 Inform Players of Random Event

Preconditions	Game has just started or the end of a game-turn has just occurred.
Summary	The computer will calculate if there is to be a random event and inform the players of the consequences.

1.1.3 Determine Player Order

Preconditions	Players have just attempted to form alliances.
Summary	Computer randomly-generates the player order for the current game-turn and informs the players.
Alternatives	The player order on the first turn is given by the scenario and is not randomly-generated.

1.1.4 Form Alliances

Actors	• Human Players
Preconditions	The Random Event Determination Inter-Phase has just ended.
Steps	• Players will figure out if they want any alliances. • Players will acknowledge who they want to ally with secretly. • Two players are allied if they both acknowledged eachother. • A group of more than two players are allied if all the players in the group acknowledge that they wish to be allied with all the others in the group.

1.1.5 Play as Allies

Actors	• Human Players
Preconditions	Players have formed an alliance during the Player-Order Determination Inter-Phase.
Summary	Players will play during the same player-turn, will ignore zones of control from allied units, and cannot attack allied units.

1.2 Movement (Chihsiang Wang)

1.2.1 Army Units Movement

Movement	
Summary	Current player moves their units, they can do as many number of units as they want.
Actors	• Current Player • Friendly Player • Enemy Player
Preconditions	• In the current player's movement phase
Primary Scenario: General Movement	• Current player picks one unit and decide which adjacent hex they want to move(any directions) • Before move to adjacent hex, check conditions: 1. Movement Points: If and only if certain unit has enough MP to move 2. Hex occupied by other players: Certain unit can't move to same hex which has other players' unit 3. Hex in Zone of control: Certain unit must stop when get into this hex • Players need to follow movement restriction • When a game turn is finished, all Movement Allowance will be removed, and get full at next game turn

Alternative Scenario:	• Flying units always expend one Movement Point per hex • If the current unit got special effects(magic, spells, random events), it may changes Movement Allowance(include flying units) • If units move into the hex with enemy's character, the character will be captured • Units can move to certain hex by Teleportation

1.2.2 Character Movement

Movement	
Summary	Current player moves their characters, they can do as many number of characters as they want.
Actors	• Current Player
Preconditions	• In the current player's movement phase

Primary Scenario: General Movement

- Current player picks one character and decide which adjacent hex they want to move(any directions)
- Before move to adjacent hex, check conditions:
  1. Movement Points:
     If and only if certain unit has enough MP to move(9 Movement Allowance for every characters)
  2. Hex occupied by other players:
     Certain unit can't move to same hex which has other players' unit
  3. Hex in Zone of control:
     Different as units, Character can move into those hexes controlled by enemy, but need to follow:
     a) The hex is occupied by friendly units
     b) The hex is both zone of enemy's and friendly player's control
- Players need to follow movement restriction
- When a game turn is finished, all Movement Allowance will be removed, and get full at next game turn

---

Alternative Scenario:

- If a flying unit moves through or begins the movement phase in the same hex as Character that has not moved. The fly unit can carry character and the character will not expend any movement point
- If the certain character got special effects(magic, spells, random events), it may changes Movement Allowance(include flying units)
- Units can move to certain hex by Teleportation

1.3 Stacking (Chihsiang Wang)

Stacking	
Summary	Maximum of 2 friendly units and any numbers of characters can stack in the same hex
Actors	• Current Player • Friendly Player
Preconditions	• In the current player's movement phase • In the friendly player's movement phase • Maximum of 2 units in the same hex(unlimited numbers of characters)
Primary Scenario:	• Current player pick a unit or character and move it to adjacent hex • If the unit stop on the hex contained with no more than one friendly unit, it stack together • If the character stop one the hex contained with units or characters, they stack together
Alternative Scenario:	• Units will never be stacked with enemy's units • Unit may move through the hex already contained maximum numbers of units, but can't stop on that hex • flying units can move through enemy's units, but not which has Archers or flying units • At the end of movement phase, if there are more than 2 units in the same hex, the units owner may decide which unit should be eliminated

1.4 Zones of Control (Chihsiang Wang)

Zones of control	
Summary	The six hexagons surrounding a unit constitute the Zone of Control of that unit
Actors	• Current Player's • Enemy Player's • Monsters(considered Characters)
Preconditions	• In the current player's movement phase
Primary Scenario:	• When a unit moves into the enemy-controlled zone, it must be stop moving • Units may leave zone of control at the beginning of its movement phase • Units will never move from one Enemy-Controlled hex to another
Alternative Scenario:	• Flying units are not affect by zone of control, except Archer and other enemy flying units zone of control • Characters can move into enemy-controlled zone if the hex is controlled by friendly units or contained with friendly units

1.5 Magic (Johnathan Flake)

1.5.1 Casting a Spell

Casting a Spell

Summary	Player selects and casts a spell
Actors	• Current Player
Preconditions	• It is the player's movement or spell casting phase. • The player has a character able to cast spells.
Primary Scenario:	• Player selects a casting character • Caster selects a spell equal to or less than selected character's power level • Player selects a target for the spell • Character allocates appropriate manna • Spell effects resolve
Related Cases	• Movement

1.5.2 Casting a Spell of a higher power level

Casting a Spell of a higher power level	
Summary	Player selects and casts a spell of a higher power level
Actors	• Current Player
Preconditions	• It is the player's movement or spell casting phase. • The player has a character able to cast spells.

Primary Scenario:	• Player selects a casting character • Player selects a spell one level greater than selected character's power level • Player selects a target for the spell • Character allocates appropriate manna • System computes a spell success • Spell effects resolve
Alternate	• System computes a spell failure • Spell fails
Alternate Extension	• System computes a critical failure • Casting character dies
Related Cases	• Movement

1.5.3 Counter Spells

Counter Spells	
Summary	Player selects and casts a counter spell.
Actors	• Spell casting Player • Counter-spell casting Player
Preconditions	• It is the player's movement or spell casting phase. • The player has a character able to cast spells.

Primary Scenario:	• Spell casting Player casts a spell • Counter-spell casting Player selects a casting character • Counter-spell casting Player selects a spell that counters Spell casting Player's spell • Counter-spell casting Player allocates appropriate manna • Spell casting Player's spell is negated
Related Cases	• Casting a spell • Casting a spell of a higher power level

1.6 Combat Phase Use Cases (Wayne Fuhrman)

1.6.1 Initiate Battle

Initiate Battle	
Summary	The phasing player initiates a battle.
Actors	• The Phasing Player
Preconditions	• The game-turn is in the combat phase.
Scenario	• The player selects the target hex. • The player selects the participating units. • The defending player is notified.
Postconditions	• The game state proceeds to the Magic Allocation Phase.

1.6.2 Rally Units

Rally Units	
Summary	The attacking player attempts to rally a demoralized unit.
Actors	• The attacking player.
Preconditions	• The game-turn is in the combat phase. A battle has just been initiated.
Scenario	• The user selects a character • The user selects a demoralized unit. • The system computes the result and changes the state of any re-moralized character.
Alternatives	• In steps 1 and 2, the system may allow a group of demoralized units and a group of characters to be selected.

1.6.3 Allocate Magic Support

Allocate Magic Support	
Summary	Both players commit to the amount of manna each character will use during battle. Both players must commit before knowing the other's commitment.
Actors	• The Attacking Player • The Defending Player

Scenario	• The attacker commits to a manna amount. • The defender commits to a manna amount. • The system reveals each player's commitment to the other player.
Postconditions	• The combat strength is adjusted based on magic support. • The game state proceeds to the Combat Resolution Phase

1.6.4 Resolve Combat

Resolve Combat	
Summary	The battle results are computed. Both players eliminate and retreat troops as required. The attacking player may pursue retreating troops.
Actors	• The Attacking Player • The Defending Player
Scenario	• The system computes and displays the battle result. • The defending player selects units to eliminate. • The defending player selects units to retreat and selects each retreat path. • The attacking player selects units to eliminate. • The attacking player selects units to retreat. • If the attacking player has units remaining in the original hex, the player may pursue any retreating enemy troops.

Alternatives	• For each user, the order of elimination and retreating may be interleaved. However, the defending player must always resolve first. • Each of the steps 2, 3, 4, and 5 may be optional or required, depending on the magnitude of the damage suffered by the player.
Postconditions	• Any retreating troops are demoralized.

1.7 Sections 16-19 (Keith Drew)

Movement Phase	**Teleport**
Summary	Teleport unit(s) from teleport hex to teleport hex.
Actors	Player/User
Preconditions	Unit(s) on teleport hex; in current movement phase.

Steps:

1. User selects teleport option
2. System determines random teleport hex destination, displays results, updates screen, and goes to "Combat" use case if the destination contains enemy unit(s).
3. User continues their turn.

Alternative 1: User has cast spell of "Teleport Control". Here the user must select their destination hex.

Alternative 2: User has cast spell of "Teleport Protection". The system may not eliminate any of the user's units as a result of teleporting.

Related Use Cases:	Combat

Magic Phase: Spell Segment	**Summon Demon**
Summary	Player summons a demon monster.
Actors	Player/User
Preconditions	Player is in Magic Phase, can cast spell.

Steps:

1. Player selects "Summon Demon" spell to cast.
2. System randomly determines Demon Characteristics.
3. Player selects and casts "Bind Demon" spell - see "Bind Demon"
   OR
   Player does not select and cast "Bind Demon", meaning he does not control the demon.
4. System evaluates "Bind Demon" spell, if applicable, and applies results.

Related Use Cases:	Spell "Bind Demon"

Movement Phase	**Blockade Dragon Tunnel Complex (DTC)**
Summary	A dragon monster unit blockades a DTC.
Actors	Player/User
Preconditions	Dragon unit has not moved this game-turn. Player controls dragon unit.

Steps:

1. User selects dragon unit and indicates they wish to blockade.
2. System indicates which tunnel complex entrances can be blocked.
3. User selects any/all of the available entrances.
4. System records blocked entrances and disallows movement through them.

Movement Phase	**Create Emissary**
Summary	A player controlled character unit spawns an emissary.
Actors	Player/User

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

Steps:

1. User selects character which will spawn emissary.
2. System determines the characteristics of the emissary and updates the game board with the new piece.

Variable	**Eliminate Emissary**
Summary	Enemy units in the same stack eliminate emissary.
Actors	Player A, Player B
Preconditions	Emissary occupies same hex as enemy unit(s).

Steps:

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

Diplomacy Inter-phase	**Conduct Diplomacy**
Summary	Player conducts diplomacy to gain favor with neutral camp.
Actors	Player/User
Preconditions	Player controls an emissary or character that occupies a neutral Capital hex.

Steps:

1. User indicates they with to conduct diplomacy.
2. System indicates which emissary/character pieces are eligible.
3. User selects unit to conduct diplomacy with.
4. System evaluates results of diplomacy indicates success/failure, along with appropriate game board change updates and influence updates.

Alternative: If multiple players have characters or emissaries in the same capital, they conduct diplomacy in the same order as the player-turn order this game-turn. When one player succeeds all other emissaries present are removed from play.

Alternative: If an enemy player's character occupies the same capital hex, the player may choose to skip conducting diplomacy to subtract their diplomacy rating from the "die roll" of any enemy - of course this will be automated.

Alliance Determination Phase	**Sacrifice**
Summary	A player sacrifices a non-monster unit to gain favor with neutral camp.
Actors	Player/User
Preconditions	Non-monster unit(s) on boarder of neutral camp who favor is desired with.

Steps:

1. User selects sacrifice option.
2. System determines and displays valid units for sacrifice.
3. User selects the unit desired for sacrifice (see alternative).
4. System displays sacrifice animation (if any) and adjusts diplomacy as necessary.

Alternative: In step 3, if the sacrificial unit is bordering two neutral territories that are both eligible for sacrifice, the player must indicate the territory s/he desires to sacrifice the unit to.

Movement Phase	**Invade Neutral Province**

Summary A unit is moved into a neutral province with which they are not allied, as invasion.

Actors Player/User

Preconditions Player is neutral with province; unit is able to move past border of province; unit is not a character.

Steps:

1. User selects move option (see movement use case) and attempts to enter a neutral province.
2. System indicates the move is an invasion and prompts user confirmation of invasion.
3. User indicates if they would like to invade.
4. System updates diplomacy "charts" as necessary.
5. User continues movement phase.

Related Use Cases: Move Unit

Movement Phase	**Release Character**
Summary	A unit stack detaining a character moves away from the character, leaving it alone in any hex.
Actors	Player/User
Preconditions	Character is "captured"; units with character can move.

Steps:

1. User calls use case Move Unit(s).
2. User chooses to move units without captured character.
3. System prompt asks if user would like to release the character.
4. User confirms their decision.
5. System updates display and moves character unit instantly as far away as allowed.

Related Use Cases:	Move Unit(s)

Manna Regeneration Phase	**Attempt Escape (character)**
Summary	A character attempts to escape capture.
Actors	Player/User
Preconditions	Manna regeneration phase; character is captured.

Steps:

1. User selects "Attempt Escape" option.
2. System indicates which characters may attempt escape.
3. User indicates character to attempt escape.
4. System determines results of attempt, indicates results to user, and updates the game board as necessary.

Alternative: Character trying to escape is a magic-user. In this case the system must ask if the player would like to have his character use magic to assist their escape.

Diplomacy Interphase	**Eliminate Prisoner**
Summary	A player has brought a captured character to his capital and wishes to make an example out of them to the masses.
Actors	Player/User
Preconditions	Diplomacy Inter-phase; captured character is in capital.

Steps:

1. User selects "Public Execution" option.
2. System confirms user selection and plays execution animation.
3. System updates display accordingly.

1.8 Selecting Scenario(Joe Higley)

Selecting Scenario	
Summary	The players are lead through the scenario selection process.
Actors	• Human player(s)
Preconditions	• Game client has been started

Use Case	• Players select the number of players for the game • Game will display a list of scenarios for the given number of players. • Players select the scenario they wish to play from the available list. • Game will display the background story and a list of roles for this scenario • Players select their role from the given list. • Game initiallizes the board then gives players the option to move pieces in order of first game turn. • Player selects a piece then selects a legal position to move it to. • Player selects finished when they have completed repositioning. • Once all players have completed repositioning, game will start. • Victory Conditions will be displayed.

2 Tasks by Wayne Fuhrman

2.1 Analysis of Combat Tasks (Fuhrman)

Combat is broken into an orderly series of steps. Each step is performed, and the user continues to the next step. A user decision at a previous step may influence a later step.

The division of the combat phase into discrete tasks is a little complicated. I've used a tree, with leaf nodes containing reasonably atomic tasks:

1. The user selects which troops will engage in combat.
   - The user selects which hex will be attacked.
   - The user selects which troops will attack.
2. Combat is executed.
   a) Unit rallying (attacker only)
      - The user selects a character.
      - The user selects an eligible unit to rally.
      - State change: The rallying attempt is performed (a dice roll and comparison).

   b) Magic allocation (attacker and defender)
      - The user selects a character.
      - The user selects an amount of manna to spend.
      - State change: The effective combat strength of the attacking troops is updated.

   c) Combat Resolution (attacker and defender)
      - State Change: Battle results are calculated.
      - Based on the results, the participants either retreat or eliminate units.
      - The attacking user may advance troops into newly vacated territory.

Some of these steps would require user action when playing manually, but might be handled by a computer without user input.

2.2 Assessment of Combat Phase (Fuhrman)

A rule-conforming adaptation looks easy. An adaptation that is what-the-user-wants-conforming may be harder.

Some manually elements of combat are simple to digitize:

- Random numbers should be generated by the software.
- Table results should be fetched by the computer.
- Various small calculations (like summing combat strength and calculating terrain effects) should be handled by the program.

In these cases, the user may still wish to see information about the impending calculation: For example, the user might be presented with the effective combat ratio before committing to an attack.

Other elements may require more thought to digitize:

- The selection process

  The game rules allow a large number of individual choices, and the decision process could easily become tedious. For example, not all units that can attack must attack. But requiring the user to select each unit would seem tedious. Two ideas that might streamline selection are group selection (click and drag to select), and the use of defaults (e.g. we might default to always rallying units when possible).

- Combat Resolution

  The retreat and elimination phase is governed by a large number of specific rules. For example, there is a sequence of enforced retreat preferences, where the user is allowed to use certain retreat paths, but only if other paths are not available. The logic isn't too tricky, but it may be hard to present the selection in a non-aggravating way to the user. We might consider simplifying the retreat path rules.

- Rallying

  The troop rallying might be combined, so that the user selects all troops they will attempt to rally at once, and then the program informs the user of the results.

3 Tasks by Chihsiang Wang

3.1 Analysis of Game Rules

3.1.1 Movement

- Phasing Player can move any, all or none unites and Characters in any directions. Units and Character are moved one at a time.

3.1.2 The Movement Allowance

- Movement Allowance is printed on the face of the card. MA of all characters is 9, but can be altered by Events or Spall. MA for units also can be altered.
- MA only can be use in a single game-turn, this means when a game turn is finished, all unused MA will be removed, and will get full MA in next game turn.

3.1.3 How to Move

- Movement expend Movement Points Clear Hex: one point Others: See the chart
- Units and Characters can only move from hex to adjacent hexes. Exception: Teleportation.
- Road or Trail may negate the effect of other terrain.
- Flying units: w symbol to represent their racial Only need to expend one Movement Point to enter any kinds of hexes. And flying units may enter some impassable hexes.
- When two players are allied, will move together, more details in Alliance (Section 6.0)

3.1.4 Racial Type

- Each units and characters has racial type
- certain race and characters has certain cost of movement points.

3.1.5 Interrelationship of Units

- There are four different types of units (in relationship):
  1. Friendly:
     is any units that comes from same nation. Usually same color. Conjured units is in this category.
  2. Friendly Allied:
     under control of the same player OR a player that is allied with the owning player.
  3. Enemy:
     Enemy units are any units controlled by another player who is not allied.
  4. Neutral:
     Not yet controlled by any players, listed in the scenario as Neutral units.

3.1.6 Movement Inhibitions and Prohibitions

- a friendly unit cant move into any hex which has enemy units. In addition, if the hex already contained with units, other players unit cant move in even its Friendly allied units. Same as Neutral units.
- Units of different armies or nations may not enter hexes containing units or characters of another Army or Nation.
- A unit must cease(stop) movement when it enters an Enemy-Controlled hex. And units cant move anymore till end of game turn.
- A unit or character never expends more MP than its MA in any army movement segment.
- Characters are not forced to cease movement upon entering an enemy-controlled hex.
- If any enemy units enter a hex which occupied by a character, the character will be captured by enemy player.
- A character moves as freely as units do. If a flying unit in the same hex(should be beginning of movement phase and in same hex and character is not moved), flying units may pick up character. Flying units can pick up any numbers of friendly characters. Flying monster only can pick one character each.

3.1.7 Flying Units

- Always expend one MP per hex, unless effected by magic or random events.

3.1.8 Stacking

- Maximum of 2 friendly units and any number of characters can be stacked in one hex at the end of any Friendly Movement Phase.

3.1.9 Which Units may Stack

- Units of Allies and units of 2 different armies or nations controlled by one player may never stack each other.

3.1.10 Character Stacking

- No limit of numbers of characters that may occupy same hex. Monster (though tread as units) are considered Characters for stacking purpose (can be stacked in any numbers of monster in same hex). But Monster may never stack with armies units.

3.1.11 Character and Army Stacking

- Character owned by different forces: can be stacked
  Units: can not

3.1.12 Effects of Flying Units On Stacking

- Flying units may move through hexes containing with enemy units(but cant stop at that hex). Flying units may not move through the hex containing with enemy archer units or any enemy flying units. And flying units are not effected by zones of control of enemy units(exception: if the units mentioned above)

3.1.13 Monster and Army Stacking

- No Monster may ever stack with an Army unit.

3.1.14 Zones of Control

- 6 hex surrounded the units is zone of controls. Monster considered as characters.
- When a unit moves into the enemy- controlled zone, it must be stop moving
- Units may leave zone of control at the be- ginning of its movement phase
- Units will never move from one Enemy- Controlled hex to another
- Flying units are not affect by zone of control, except Archer and other enemy flying units zone of control

- Characters can move into enemy-controlled zone if the hex is controlled by friendly units or contained with friendly units

3.2 Play Test

- Our team played the Swords & Sorcery twice, this helped us to understand this game easier. While we played Scenario 3, a 7 players game, we spent almost an hour to set up everything, and spent another house to finish one game turn. I realized that this game has extensive rules but it's good to convert to be a computer game since it has very well domain.

3.3 Assessment of Requirements

- Most of Movement elements are easy to to be computerize. Every characters or units will have movement information:
  1. Movement Allowance
  2. Movement Points expend
  3. Units Racial
  4. Stacking condition
  5. Magic effect
  6. Zone of controlled
  7. Fly unit
- And also need to consider Terrain types, so that we can simply use some statements(simplest using if/else statement). With enough information from the class units or characters, we can determine which hex can be moved in, which are not.

4 Tasks by Sean Shepherd

4.1 Assessment of Requirements for Sections 4, 5 and 6

Section 4.0	• Players will be guided through the steps of a game-turn and will always know whose turn it is and what they need to be doing at the time.
Section 5.0	• Random Events can be decided by the computer instead of a player having to roll the dice. • The effects of a random event can also be instant, which means the human players won't have to figure out where to move the suns anymore or where killer penguins will spawn.
Section 6.0	• Alliances will require private conversations between players so there will have to be a chat thing of some kind for players to communicate during this phase and also once players are allied so they can coordinate their attacks.

5 Tasks by Keith Drew

5.1 Play-test Results

Our group actually play-tested Swords & Sorcery twice, both times being an incomplete play-through. However, we managed to make it through a full game turn each time. This helped the group to collectively understand the extensive rules and sequence of play for this complicated board game. To the end of understanding the sequence of play of Swords & Sorcery, we found success. It is also worth noting that we tried different scenarios for each play-test, first Scenario 1, then Scenario 3. Scenario 1 is a two-player scenario, while Scenario 3 is a seven-player scenario. Also, Scenario 1 includes neutral camps, while Scenario 3 does not.

5.2 Army Game

5.2.1 Distribution of Sections

- Sean Shepherd - 4, 5, 6
- Chi-Hsiang Wang - 7, 8, 9
- Johnathan Flake - 10, 11, 12
- Wayne Fuhrman - 13, 14, 15
- Keith Drew - Sections 16, 17, 18, 19
- Joe Higley - Section 20

5.2.2 User Tasks - Sections 16-19

For my given sections there are distinct tasks, 11, that I have identified:

1. Teleport
2. Summon Demon
3. Blockade Dragon Tunnel Complex
4. Create Emissary
5. Eliminate Emissary

6. Conduct Diplomacy
7. Sacrifice Unit/Character
8. Invade Neutral Territory
9. Release Character
10. Escape Capture (Character only)
11. Eliminate Prisoner

5.2.3 Assessment of Requirements for Computer Adaptation

The class of game details (parts) that will be automated and invisible to the user is extensive. Some of these details include:

- Random number generation
- Literal placement of game pieces
- Random event determination
- Scenario set up
- etc.

These details will mostly involve program logic and minor user action.

The class of game details (parts) that will be visible and interactive to/with the user includes:

- The game board
- Necessary charts or links to them (those needed for interaction or strategy)
- etc.

More specifically, the solar chart needs to be accessible to the players. To me this means adding a "live" display on the game screen (the "HUD," if you will). My idea for this specific chart would be a pair of tracks (one for yellow sun, one for blue/red suns) that showed the relational positions of the suns as necessary for strategy. However, actually implementations of the sun chart's effects would all be under the hood, possible in a completely unrelated are of code.

6 Tasks by Wayne Fuhrman

6.1 Task

It is notable that a considerable amount of the tasks behind the Magic section would likely be automated, and thus the amount of tasks for the board game version varies drastically when compared to a computerized version. As such I have broken tasks into two sections: "User tasks" or tasks that a user would complete regardless, and "automated tasks" that the player would note in the board game, but would most likely be computerized, and therefore not necessarily be seen as a user task when automated.

6.1.1 User tasks for Magic

- Casting
    1. Select eligible character
        a) Select spell within character's power level
        b) Select spell one above character's power level
    2. Select target
    3. Resolve spell effects

6.1.2 Automated tasks for Magic

1. Sun Phases
    a) Check current sun phase
        - Set all characters' power levels to their appropriate level for the sun phase
    b) Set suns to next phase
2. Casting
    a) Check eligible character's available manna
        i. Expend required amount of manna
        ii. Spell fails if manna is insufficient
    b) Compare selected spell to caster's spell level

- For spells one level above character's power level, check for spell failure and critical failure

c) Check range of spell for casting character

d) Resolve spell effects

3. Manna Regeneration
   - Check current sun phase
     - Check for characters with expended manna
       a) Check character's color
       b) Replenish the manna of the character by an amount relevent to their color

6.2 Assessment of Magic

- As previously noted, many aspects of Magic can be translated to an automated setting and would actually increase the playability of the game by streamlining the spell casting process. Many checks, such as available manna, manna regernation, and spell caster's level could be done without much attention from the user. Automation would also allow the user to select spells much easier, as inappropriate or ineligable entries would be removed from the list of available spells. Other aspects, such as phases of the sun, and their respective ascensions/descensions would be translated into a simple element of UI.

- The majority of the elements of Magic would likely be the same, or very similar to the board game version of Swords and Scorcery. However, much of what is largely a time consuming process would be simplified as many of its elements would be calculated by the computer out of sight of the user and the same actions would be much more user-friendly.