CS 246—Assignment 5, Group Project (Spring 2019)

Due Date 1: Tuesday, July 23, 5pm
Due Date 2: Tuesday, July 30th, 11:59pm

DO NOT EVER SUBMIT TO MARMOSET WITHOUT COMPILING AND TESTING FIRST. If your final submission doesn't compile, or otherwise doesn't work, you will have nothing to show during your demo. Resist the temptation to make last-minute changes. They probably aren't worth it.

This project is intended to be doable by three people in two weeks. Because the breadth of students' abilities in this course is quite wide, exactly what constitutes two weeks' worth of work for three students is difficult to nail down. Some groups will finish quickly; others won't finish at all. We will attempt to grade this assignment in a way that addresses both ends of the spectrum. You should be able to pass the assignment with only a modest portion of your program working. Then, if you want a higher mark, it will take more than a proportionally higher effort to achieve, the higher you go. A perfect score will require a complete implementation. If you finish the entire program early, you can add extra features for a few extra marks.

Above all, MAKE SURE YOUR SUBMITTED PROGRAM RUNS. The markers do not have time to examine source code and give partial correctness marks for non-working programs. So, no matter what, even if your program doesn't work properly, make sure it at least does something.

1 The Basics

Watch out: This project involves a number of details, and going about it in the wrong order may cause you to spend lots of time while making relatively little progress in terms of marks. You should read the advice section at the end before starting to implement your project to get some guidelines on how to approach it.

In this project, you will produce the game Sorcery, a card game based on collectible card games such as "Hearthstone: Heroes of Warcraft" and "Magic: the Gathering." If you have never played either of these games before, Hearthstone is free and might help you familiarize yourself with the genre.

Sorcery is played on the terminal, with commands issued via standard input or supplied from a file and output printed to standard output. Complete implementations of Sorcery will also include a graphical interface which may be optionally enabled by the user as a command-line argument.

1.1 Basic Definitions

The following words have special meanings in Sorcery:

- **Card:** *Cards* are the basic objects in Sorcery, making up players' decks, hands, and graveyards.
- **Deck:** The players' *deck* is a collection of cards which they may *draw* from.
- **Board:** A player's *board* is a collection of cards which they have played and which have not been moved to another zone.

- **Graveyard:** A player's *graveyard* is a collection of minions which have died.
- **Hand:** The players' *hand* is a collection of cards (to a maximum of 5) which they may play.
- **Draw:** To *draw*, a player takes a card from their deck and puts it into their hand. A player may only draw if their hand is not full.
- **Owner:** The *owner* of a card is the player whose hand, deck, graveyard, or board it is in.
- **Type:** A card's *type* is one of "minion," "enchantment," "ritual," or "spell."
- **Minion:** A *minion* is a card representing a character or creature which will help you achieve victory.
- **Die:** When a minion *dies*, it is moved from its owner's board to their graveyard.
- **Magic:** The player's *magic* is their main resource used to play cards and use special abilities.
- **Trigger:** *Triggers* are effects which occur when certain conditions are met.
- **Active Player:** The *active player* is the player whose turn it is. The other player is the *inactive player* (or non-active player).
- **Shuffle:** *Shuffling* a deck randomizes the order of the cards in the deck.
- **APNAP Order:** *APNAP order*, or "active player, non-active player order" is the default order in which simultaneous effects occur (for example, the order in which Blizzard damages minions or "a minion enters play" triggers activate). The order is:

  1. First, the minions owned by the active player, in left-to-right (oldest-played to newest-played) order.
  2. Next, the ritual owned by the active player.
  3. Finally, repeat the above two steps in the same order for the non-active player.

1.2 Basic Gameplay

The game's objective is to reduce the opposing player's life to 0, at which point the game ends. The game begins by first asking both players for their names. It then shuffles both player's decks. Once the decks are shuffled, the game begins with player 1. Both players start with 20 life, 5 cards in their hand, and 3 magic.

For the rest of the game, players alternate turns. A player's turn consists of the following:

- The player gains 1 magic.
- The player draws a card if their deck is non-empty.
- Any "At the start of the turn" effects occur.
- The player is allowed to take actions until they pass.
- Any "At the end of the turn" effects occur.

2 Cards

Every card has a name and a cost. Other than that, card effects are determined by their type.

2.1 Spells

Spells are the simplest type of card. A spell simply changes the game in some way (such as by increasing its caster's life by 5 or killing a chosen minion) and is then removed from the game.

2.2 Minions

Minions are the main card type in the game and the primary means of achieving victory. When a minion is played, it is moved from the player's hand to one of the player's 5 minion slots on the board. If all 5 slots are occupied, the minion cannot be played. Minions occupy the leftmost spots on the board, and new minions are always added to the right of older minions.

In addition to the attributes of every card, minions have *attack* and *defence* values, as well as *actions*. Attack and defence represent a minion's combat strength, while the minion's number of actions is how many times it is allowed to attack or use an ability in one turn. If a minion's defence is ever 0 or less it immediately dies and is moved from the board to the top of the graveyard.

Definition: An x/y *minion* is a minion with x attack and y defence.

Minions start with 0 actions. At the start of every turn, every minion owned by the player whose turn it is is restored to 1 action. Note that a minion which already had 1 action remains at 1 action. A minion may spend an action in one of the following ways:

- To attack the opposing player. The opposing player loses life equal to the attack value of the minion.
- To attack an opposing minion. Both minions damage one another: minion A reduces minion B's defence by minion A's attack, and then minion B damages A in the same way.
- To use an *activated ability*.

Note: *Minions retain their attack and defence values even when they change zones.*

Example: *Brad plays Air Elemental (a 1/1) and Nomair then attacks it with his Earth Elemental (a 4/4), killing the Air Elemental and reducing the Earth Elemental's defence by 1. Brad then plays Unsummon to return Nomair's Earth Elemental to Nomair's hand. The Air Elemental in Brad's graveyard is a 1/-3 and the Earth Elemental in Nomair's hand is a 4/3*

In addition to attack, defence, and actions, some minions also have abilities. Abilities can be divided into two types:

- *Activated abilities* cost magic and an action point to use, and work similar to playing a spell card.
- *Triggered abilities* are activated for free whenever a certain condition is met.

A minion can only have one ability of each type.

Question: How could you design activated abilities in your code to maximize code reuse?

2.3 Enchantments

Enchantments are modifications that can be played on minions. An enchantment can modify any aspect of a minion: some possibilities include modifying attack and defence values or granting new abilities.

If a minion is enchanted by multiple enchantments, they are applied in oldest-to-newest order. For example, a 2/2 minion enchanted first with a $+1/+1$ enchantment and then a $*2/*2$ enchantment has 6 attack and defence, while if it was first enchanted with the $*2/*2$ enchantment it would have 5 attack and defence.

If an enchantment grants a minion a new activated ability, the minion's old activated ability may not be used. If multiple enchantments grant a minion activated abilities, only the newest enchantment's ability may be used.

If at any time a minion leaves the board (for example, it dies or is returned to its owner's hand), all enchantments on that minion are removed from the game.

Example: *Sean plays Apprentice Summoner (a 1/1) and then plays Giant Strength (an enchantment which gives the Apprentice Summoner +2/+2) on it. Ten plays Unsummon on Sean's Apprentice Summoner, returning it to Sean's hand. The Giant Strength enchantment is removed from the game, and next time Sean plays his Apprentice Summoner it will be a 1/1.*

Question: What design pattern would be ideal for implementing enchantments? Why?

2.4 Rituals

Rituals are special cards with a *triggered ability*, an *activation cost* and a number of *charges*. Every time the ritual's triggered ability activates, it expends a number of charges equal to its activation cost to do the effect. If it does not have enough charges left to activate, the ability's effect simply does not occur.

A player may only have one ritual on the board at any one time. If they play a second ritual while one is already active, the old ritual is removed from the game.

Example: *Brad has a Dark Ritual in play (whose triggered ability grants him a magic at the start of his turn) with an activation cost of 1 and 2 charges left, while Nomair has a Dark Ritual in play with an activation cost of 1 and 0 charges left. On Brad's next turn he gains an extra magic from his Dark Ritual, reducing its charges to 1. On Nomair's next turn, his Dark Ritual does not grant him an extra magic and remains at 0 charges.*

3 Triggers

There are four triggers which can be used by triggered abilities or other game rules. Triggers can only be activated on cards that are currently on the board. If multiple triggers are activated, they activate in APNAP order.

3.1 At the start of your turn

When a player's turn starts, all triggered abilities titled "at the start of your turn" (or similar) on cards on that player's board activate. These occur immediately after the player gains magic and draws a card (if the deck is not empty), and before the board is displayed for that turn.

3.2 At the end of your turn

When a player's turn ends, all triggered abilities titled "at the end of your turn" (or similar) on cards on that player's board activate.

3.3 Whenever a minion enters play

When a minion is placed on the board by any means (for example, played from a player's hand or created by a spell) all cards on the board (including the minion entering play) with this trigger activate in APNAP order.

3.4 Whenever a minion leaves play

When a minion leaves the board by any means (for example, returned to a player's hand or killed) all cards on the board (including the minion leaving play) with this trigger activate in APNAP order.

Watch out: The minion activates its trigger before enchantments are removed from it.

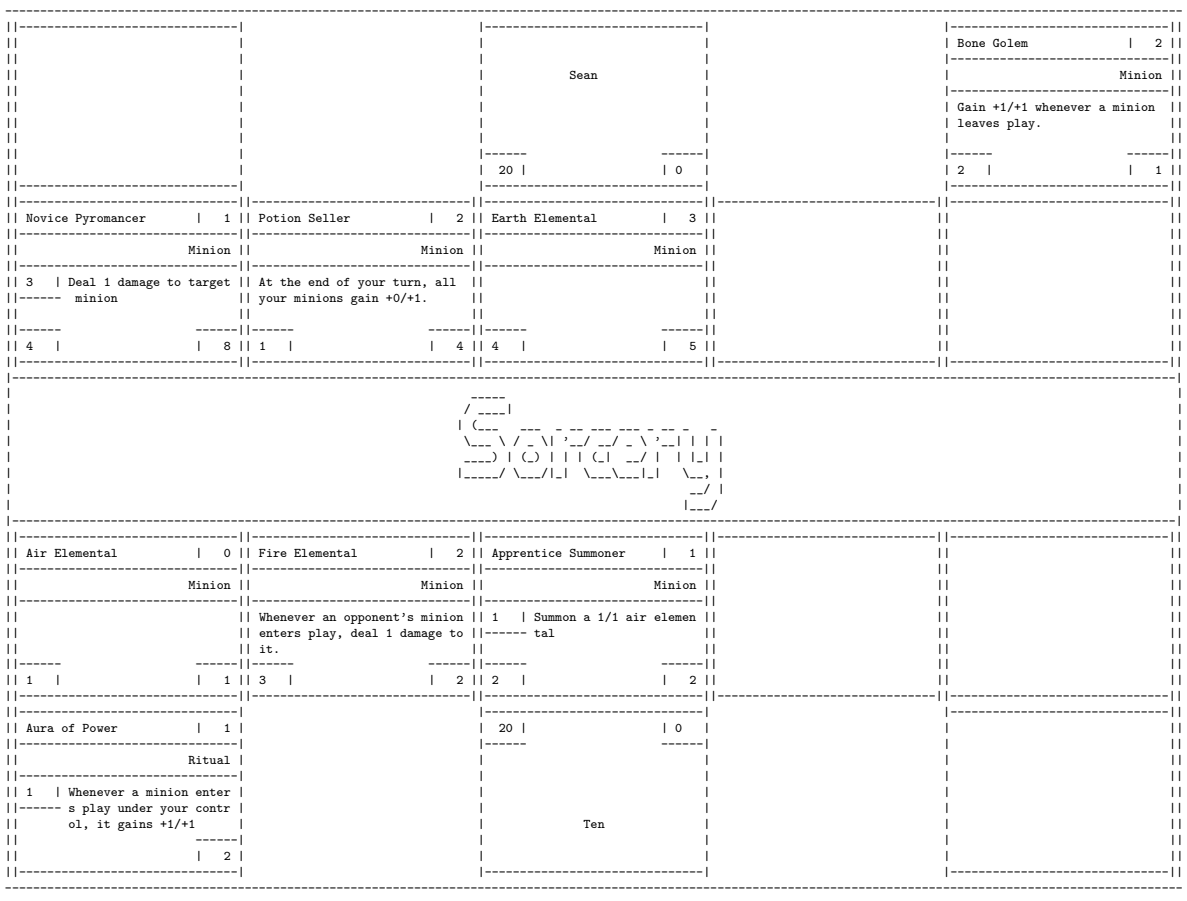
Example: *Sean and Nomair both have a Bone Golem in play, a 1/3 which gains +1/+1 whenever a minion dies. Sean's Bone Golem attacks Nomair's Bone Golem, leaving them both as 1/2s. Next, Sean casts Blizzard, which deals 2 damage to both Bone Golems. Sean's Bone Golem is killed first, but Nomair's survives since it gains +1/+1 when Sean's dies. Note also that Sean?s bone golem is now a 2/1, but is still dead.*

4 The Display

All information in Sorcery is displayed in a text-based manner. If the `-graphics` command-line argument is supplied, information should also be displayed in a graphical window.

4.1 The Board

An example Sorcery board follows:



Note: *You are given two options for drawing the board: the simple style in this document or a fancier style using unicode characters. Both styles are available in the files* `ascii_graphics.cc/h` *provided to you.*

The top and bottom left cards are player 1 and 2's rituals respectively, while the top and bottom right cards are their graveyards. The middle-top row of 5 cards are player 1's minions, while the middle-bottom row are player 2's minions. Any empty slots are filled with a blank rectangle. The top and bottom centre boxes represent the players themselves: the left value is the player's life, the right value is their magic, and the centre value is their name.

4.2 The Hand

To display a hand, simply display the cards within it in a row. For example:

```
|------------------------------||------------------------------||------------------------------||------------------------------||------------------------------|
| Silence                |   1 || Potion Seller          |   2 || Bone Golem             |   2 || Blizzard               |   3 || Aura of Power          |   1 |
|------------------------------||------------------------------||------------------------------||------------------------------||------------------------------|
|                   Enchantment ||                       Minion ||                       Minion ||                       Spell ||                       Ritual |
|------------------------------||------------------------------||------------------------------||------------------------------||------------------------------|
| Enchanted minion cannot use a || At the end of your turn, all  || Gain +1/+1 whenever a minion  || Deal 2 damage to all minions  || 1   | Whenever a minion enter |
| bilities                     || your minions gain +0/+1.      || leaves play.                  ||                              ||------ s play under your contr |
|                              ||                               ||                               ||                              ||       ol, it gains +1/+1      |
|                              ||------                  ------||------                  ------||                              ||                        ------|
|                              || 1   |                 |   3 || 1   |                 |   3 ||                              ||                        |   4 |
|------------------------------||------------------------------||------------------------------||------------------------------||------------------------------|
```

4.3 Inspecting a minion

To inspect a minion, display the minion (in the exact same way it appears on the board), and then on a new line display its enchantments, oldest to newest, five per line.

For Example, if Enrage was the newest enchantment played:

```
|------------------------------|
| Air Elemental          |   0 |
|------------------------------|
|                       Minion |
|------------------------------|
|                              |
|                              |
|                              |
|------                  ------|
| 26  |                  | 26 |
|------------------------------|
|------------------------------||------------------------------||------------------------------||------------------------------||------------------------------|
| Giant Strength         |   1 || Giant Strength         |   1 || Giant Strength         |   1 || Giant Strength         |   1 || Giant Strength         |   1 |
|------------------------------||------------------------------||------------------------------||------------------------------||------------------------------|
|                   Enchantment ||                   Enchantment ||                   Enchantment ||                   Enchantment ||                   Enchantment |
|------------------------------||------------------------------||------------------------------||------------------------------||------------------------------|
|                              ||                              ||                              ||                              ||                              |
|                              ||                              ||                              ||                              ||                              |
|                              ||                              ||                              ||                              ||                              |
|------                  ------||------                  ------||------                  ------||------                  ------||------                  ------|
| +2  |                  |  +2 || +2  |                  |  +2 || +2  |                  |  +2 || +2  |                  |  +2 || +2  |                  |  +2 |
|------------------------------||------------------------------||------------------------------||------------------------------||------------------------------|
|------------------------------||------------------------------|
| Giant Strength         |   1 || Enrage                 |   2 |
|------------------------------||------------------------------|
|                   Enchantment ||                   Enchantment |
|------------------------------||------------------------------|
|                              ||                              |
|                              ||                              |
|                              ||                              |
|------                  ------||------                  ------|
| +2  |                  |  +2 || *2  |                  |  *2 |
|------------------------------||------------------------------|
```

4.4 Individual cards

Whenever a card needs to be displayed in any of the previous areas, it should be displayed as follows:

4.4.1 Spells

```
|------------------------------|
| Banish                 |   2 |
|------------------------------|
|                        Spell |
|------------------------------|
| Destroy target minion or ritu |
| al                           |
|                              |
|                              |
|                              |
|------------------------------|
```

The card's name is "Banish", indicated in the top left, and its cost is 2, indicated in the top right. Its type is listed in the middle box and a description of its effect is in the lower box

4.4.2 Minions

Minions can be displayed in one of two ways, depending on whether they have an activated ability or not.

A minion without an activated ability is displayed as follows:

```
|-------------------------------|
| Bone Golem              |   2 |
|-------------------------------|
|                       Minion  |
|-------------------------------|
| Gain +1/+1 whenever a minion  |
| leaves play.                  |
|                               |
|------                  ------|
| 1   |                  |   3 |
|-------------------------------|
```

In this case, the Bone Golem has a triggered ability, displayed in its effect box. The only difference from the layout of the spell card above is the presence of an attack box in the lower left and defence box in the lower right. If the minion has no triggered ability, the effect box will simply be empty.

If the minion has an activated ability, there is an additional box to describe its cost:

```
|-------------------------------|
| Apprentice Summoner     |   1 |
|-------------------------------|
|                       Minion  |
|-------------------------------|
| 1   | Summon a 1/1 air elemen |
|------ tal                     |
|                               |
|------                  ------|
| 1   |                  |   1 |
|-------------------------------|
```

Question: Suppose we found a solution to the space limitations of the current user interface and wanted to allow minions to have any number and combination of activated and triggered abilities. What design patterns might help us achieve this while maximizing code reuse?

4.4.3 Enchantments

Enchantments are laid out similar to minions: the pieces they modify are highlighted by their layout. For example, an enchantment which doubles a minion's attack and defence may look like:

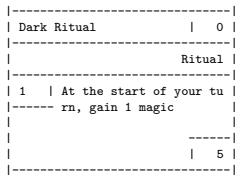
```
|-------------------------------|
| Enrage                  |   2 |
|-------------------------------|
|                   Enchantment |
|-------------------------------|
|                               |
|                               |
|                               |
|------                  ------|
| *2  |                  | *2  |
|-------------------------------|
```

While an enchantment which prevents a minion from using abilities looks like:

```
|-------------------------------|
| Silence                 |   1 |
|-------------------------------|
|                   Enchantment |
|-------------------------------|
| Enchanted minion cannot use a |
| bilities                      |
|                               |
|                               |
|                               |
|-------------------------------|
```

4.4.4 Rituals

Rituals are laid out similarly to minions with abilities, except without an attack box. The ritual's activation cost is located where the ability cost would be located for a minion, while its number of remaining charges is located where the minion's defence would be.



5 Commands and Command Line Arguments

5.1 Commands

During a player's turn, they may issue the following commands. Each command must be issued on its own line. A command may have extra whitespace before and after every word in it.

5.1.1 help

The `help` command displays a message describing the commands and their formats. You may use the following help message or create your own:

```
Commands: help -- Display this message.
          end  -- End the current player's turn.
          quit -- End the game.
          attack minion other-minion -- Orders minion to attack other-minion.
          attack minion -- Orders minion to attack the opponent.
          play card [target-player target-card] -- Play card, optionally targeting target-card owned by target-player.
          use minion [target-player target-card] -- Use minion's special ability, optionally targeting target-card owned by target-player.
          inspect minion -- View a minion's card and all enchantments on that minion.
          hand -- Describe all cards in your hand.
          board -- Describe all cards on the board.
```

5.1.2 end

The `end` command ends the current player's turn. A player may end their turn at any time.

5.1.3 quit

The `quit` command ends the game immediately with no winner.

5.1.4 draw

The `draw` command draws a card, similar to the effect if the player just started their turn. This command is only available in `-testing` mode.

5.1.5 discard

The `discard i` command discards the `i`th card in the player's hand, simply removing it from their hand (the card does not go to the graveyard, trigger leave play effects or anything else). This command is only available in `-testing` mode.

5.1.6 attack

The `attack` command follows one of two formats:

- `attack i` orders minion `i` to attack the opposing player, where 1 is the leftmost minion and 5 is the rightmost minion.
- `attack i j` orders the active player's minion `i` to attack the inactive player's minion `j`, where both `i` and `j` are as above.

5.1.7 play

The `play` command follows one of two formats:

- `play i` plays the `i`th card in the active player's hand with no target. For example, this can be used to play minions, rituals, and spells with no targets. Note that `i` ranges from `1` to `5`.
- `play i p t` plays the `i`th card in the active player's hand on card `t` owned by player `p`. `p` may be equal to `1` or `2` to represent player `1` or `2` respectively. `t` is either `1, 2, 3, 4, 5` (the `i`th minion owned by player `p`) or `r` (the ritual owned by player `p`). This can be used to play enchantments and spells with targets.

5.1.8 use

The `use` command follows the same format as the `play` command and has the same meaning, except that `i` refers to the `i`th minion owned by the current player, and the command orders that minion to use its activated ability on the provided target (or on no target).

5.1.9 describe

The `inspect i` command inspects the `i`th minion owned by the active player, as described in the "inspecting a minion" subsection of the Display section.

5.1.10 hand

The `hand` command displays the active player's hand, as described in the "hand" subsection of the Display section.

5.1.11 board

The `board` command displays the board, as described in the "board" subsection of the Display section.

5.2 Command Line Arguments

The following command line arguments may be specified to Sorcery in any order:

5.2.1 -deck1 and -deck2

The `-deck1` *filename* argument specifies that player 1's deck will be supplied in *filename*. `-deck2` works similarly but for player 2. If either player's deck is not specified using one of these commands, that player should use the file `default.deck` to specify their deck, which can be assumed to exist in the current directory. Deck files are simply a list of card names, one per line: see the provided `default.deck` for an example.

5.2.2 -init

The `-init` *filename* arguments specifies that the game will be initialized using *filename*. Filename consists of a sequence of commands to read from standard input before prompting the user for additional input (this includes player names). For example, if *filename* contains:

```
Sean
Ten
play 1
play 1
```

Then the game will begin with Player 1 named Sean, Player 2 named Ten, and Sean attempting to play the first two cards from his hand. After this, play continues using input from standard input as normal.

5.2.3 -testing

The `-testing` argument enables testing mode, changing gameplay in four ways:

- If a player attempts to play a spell or activate an ability and does not have enough magic to do so, their magic is simply set to `0` and they play the spell or activate the ability as if they had enough magic.
- Players may now use the `discard i` command to discard the `i`th card in their hand.
- Players may now use the `draw` command to draw a card.
- Decks are no longer randomized at the beginning of the game.

These two commands do not need to be described in the `help` command.

5.2.4 -graphics

The `-graphics` option enables a graphical interface. This does not disable the text interface. The graphical interface should have a similar layout to the text interface (with the current player's hand also displayed at all times), but other than that the details of the graphical interface are up to you, such as any extra flair to add, whether enchantments will be visible on the board at all times, and so forth.

Note that graphics should not be enabled by default.

Question: How could you make supporting two (or more) interfaces at once easy while requiring minimal changes to the rest of the code?

6 Individual Card Descriptions

By default, Sorcery includes the cards listed in this section. Once you are done the main game, you're welcome to add more cards out of interest or as a bonus.

6.1 Spells

Banish	2	Unsummon	1	Recharge	1
Spell		Spell		Spell	
Destroy target minion or ritual		Return target minion to its owner's hand		Your ritual gains 3 charges	
Disenchant	1	Raise Dead	1	Blizzard	3
Spell		Spell		Spell	
Destroy the top enchantment on target minion		Resurrect the top minion in your graveyard and set its defence to 1		Deal 2 damage to all minions	

Notes:

- Recharge and Raise Dead cannot be played if the ritual slot or graveyard respectively are empty.

6.2 Minions

Air Elemental	0	Earth Elemental	3	Bone Golem	2	Fire Elemental	2
Minion		Minion		Minion		Minion	
				Gain +1/+1 whenever a minion leaves play.		Whenever an opponent's minion enters play, deal 1 damage to it.	
1	1	4	4	1	3	2	2
Potion Seller	2	Novice Pyromancer	1	Apprentice Summoner	1	Master Summoner	3
Minion		Minion		Minion		Minion	
At the end of your turn, all your minions gain +0/+1.		1	Deal 1 damage to target minion	1	Summon a 1/1 air elemental	2	Summon up to three 1/1 air elementals
1	3	0	1	1	1	2	3

Notes:

- The Apprentice and Master Summoner abilities cannot be used if their owner already has 5 minions on the board.
- Master Summoner's ability may be used if there is room for at least one more minion but not all three. In that case, it simply summons enough to fill the board.
- Novice Pyromancer and Fire Elemental do not take damage from the minions they damage with their abilities.

6.3 Enchantments

Giant Strength	1	Enrage	2	Haste	1
Enchantment		Enchantment		Enchantment	
				Enchanted minion gains +1 action each turn	
+2	+2	*2	*2		

Magic Fatigue	0	Silence	1
Enchantment		Enchantment	
Enchanted minion's activated ability costs 2 more		Enchanted minion cannot use abilities	

Notes:

- Silence and Magic Fatigue can be played on minions with no activated ability, in which case they do nothing but remain on the minion as an enchantment.
- Haste grants its action immediately, meaning a minion which has just been played may take 1 action if it is then enchanted with Haste.
- Extra actions granted by Haste do not last between turns, so a minion enchanted with Haste can take two actions per turn but cannot take one action one turn and then three the next.

6.4 Rituals

Dark Ritual	0	Aura of Power	1	Standstill	3
Ritual		Ritual		Ritual	
1	At the start of your turn, gain 1 magic	1	Whenever a minion enters play under your control, it gains +1/+1	2	Whenever a minion enters play, destroy it
	5		4		4

Notes:

- Standstill affects your own minions.
- Due to APNAP order, if you play a 1/1 (such as Air Elemental) with Aura of Power in play while your opponent has Fire Elemental in play, your 1/1 will gain the +1/+1 first, and therefore survive as a 2/1 once it takes damage from Fire Elemental.

7 Advice

Sorcery is a serious project which will take both time and some clever software engineering to complete. Since we can only assign marks to working components of your program, this section contains advice on how to go about tackling the project. Finally, this section has some advice for bonus marks.

7.1 Tackling the project

This section provides some advice on the order in which you might want to approach Sorcery to maximize the number of marks you get for the work you've completed:

Watch out: To test most of the functionality of Sorcery we will require the `-testing` and `-init` arguments to work properly. Make sure the features described by those arguments are available!

1. Decide on the basic classes you will use in your program, and their high-level relationships.
2. Implement players (only having names for now), the game loop, and the `-init` command line argument. Make each command simply echo itself for now, so that you can verify that all of these work correctly.
3. Implement skeleton functionality to load decks from a file called `default.deck`. If you cannot get deck loading from a file working, start by instead hardcoding the players' decks to start with the provided `default.deck` (make sure the cards are listed in the same order as they are in that file!).
4. Implement abstract cards and the ability for a player to have a hand of cards, including giving each player a deck and the functionality to draw from that deck. Implement the ability for players to start and end their turn, including drawing a card at the start of their turn if their hand isn't full.
5. Implement minions with no activated or triggered abilities, and allow them to attack players (with no limit on the number of actions per turn). Keep in mind that they will need to be enchantable later.
6. Implement spells which interact with minions.
7. Allow minions to attack other minions.
8. Implement rituals and triggered abilities.
9. Implement simple enchantments, such as enchantments that modify the attack of a minion.
10. Implement activated abilities.
11. Implement details that have been left out thus far (magic, actions, etc).
12. Implement the more complicated remaining cards.

If Things Go Well

If you complete the entire project, you can earn up to 10% extra credit for implementing extra features. These should be outlined in your design document, and markers will judge the value of your extra features.

To earn significant credit, enhancements must be algorithmically difficult, or solve an interesting problem in object-oriented design. Trivial enhancements will not earn a significant fraction of the available marks.

8 Submission Instructions

See **project_guidelines.pdf** for important instructions including what to submit when, bonus, plan of attack and final design document.

9 A Note on Random Generation

To complete this project, if you require random generation (or rather, pseudo-random) generation of numbers, you have two options available to you. In <cstdlib>, there are commands `rand` and `srand`, which generate a random number and seed the random generator respectively (typically, seeded with `time()` from <ctime>). Alternatively, you can use the `prng` class from `prng.h`, which provides an encapsulated random number generating algorithm. Either is fine for the project; it is not required to use one over the other.