ECGR 2104 Spring 2018

Final Project:

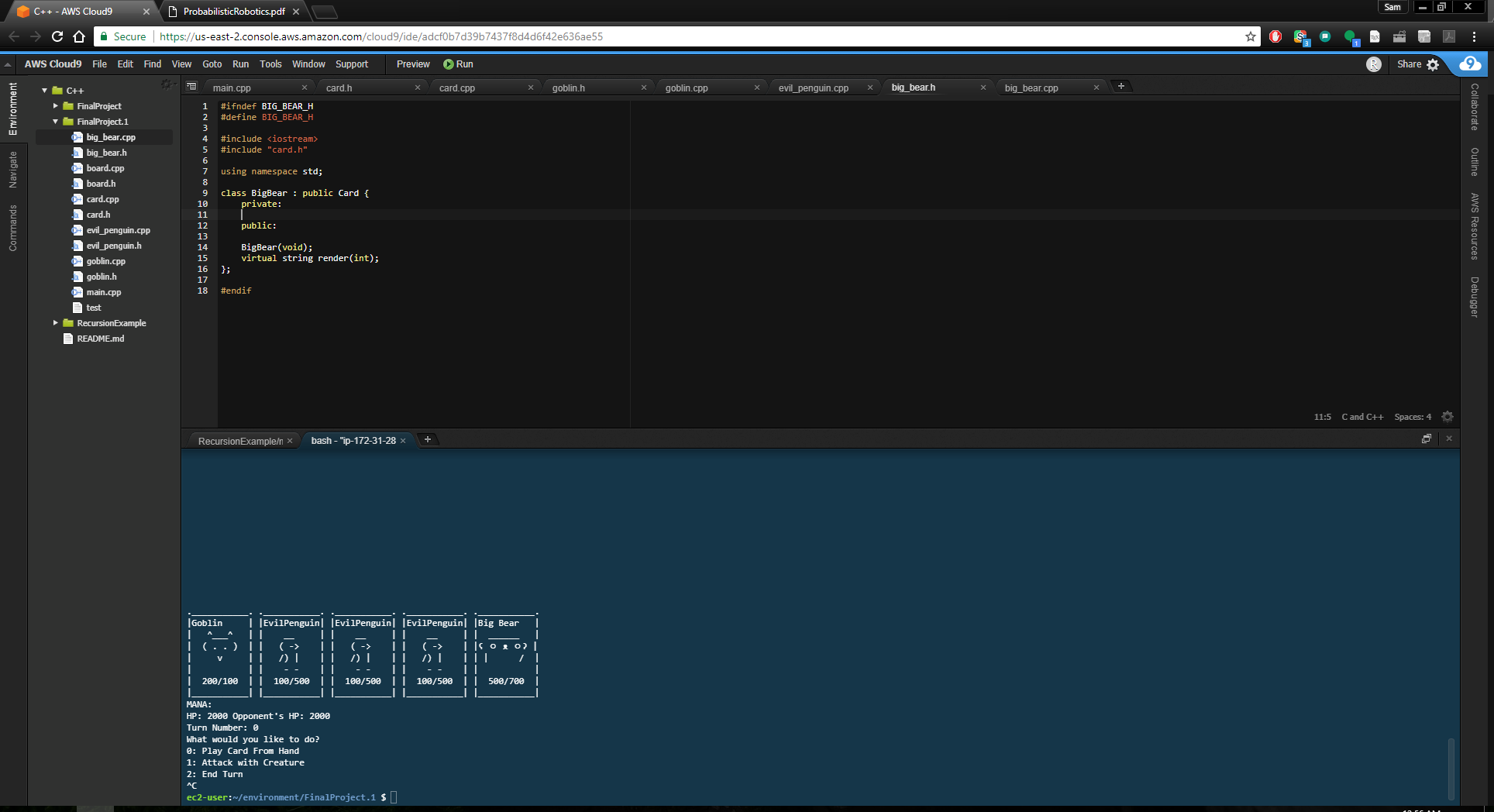
Hearthstone--

**Introduction:**

This project will involve building a program to play a game called “Hearthstone--“. Hearthstone-- is a derivative of the popular online card game Hearthstone, and functions similarly to other collectible card games. The game of Hearthstone--is a game where two players summon creatures to attack each other until their health points are reduced to zero.

**How to play:**

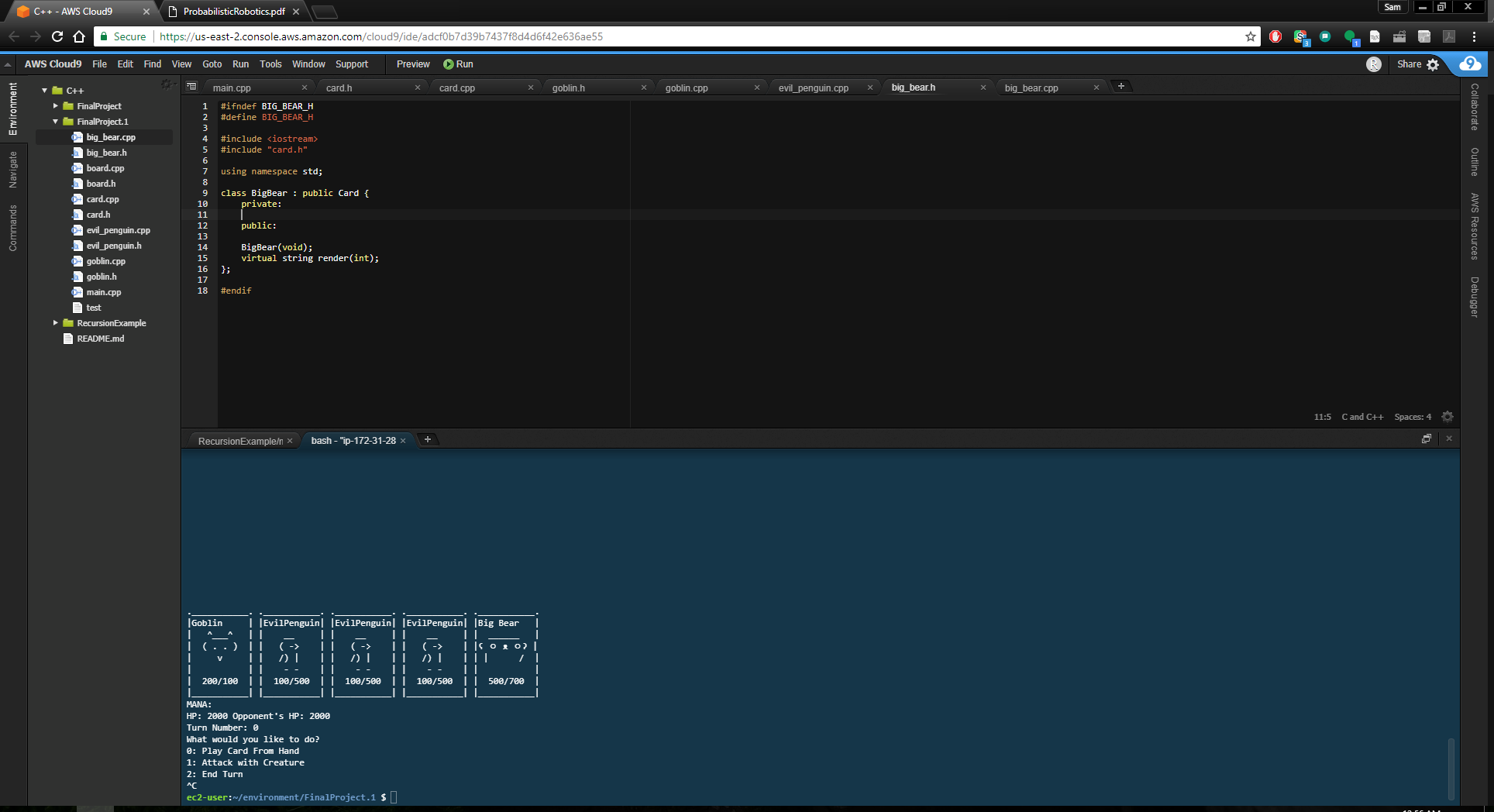
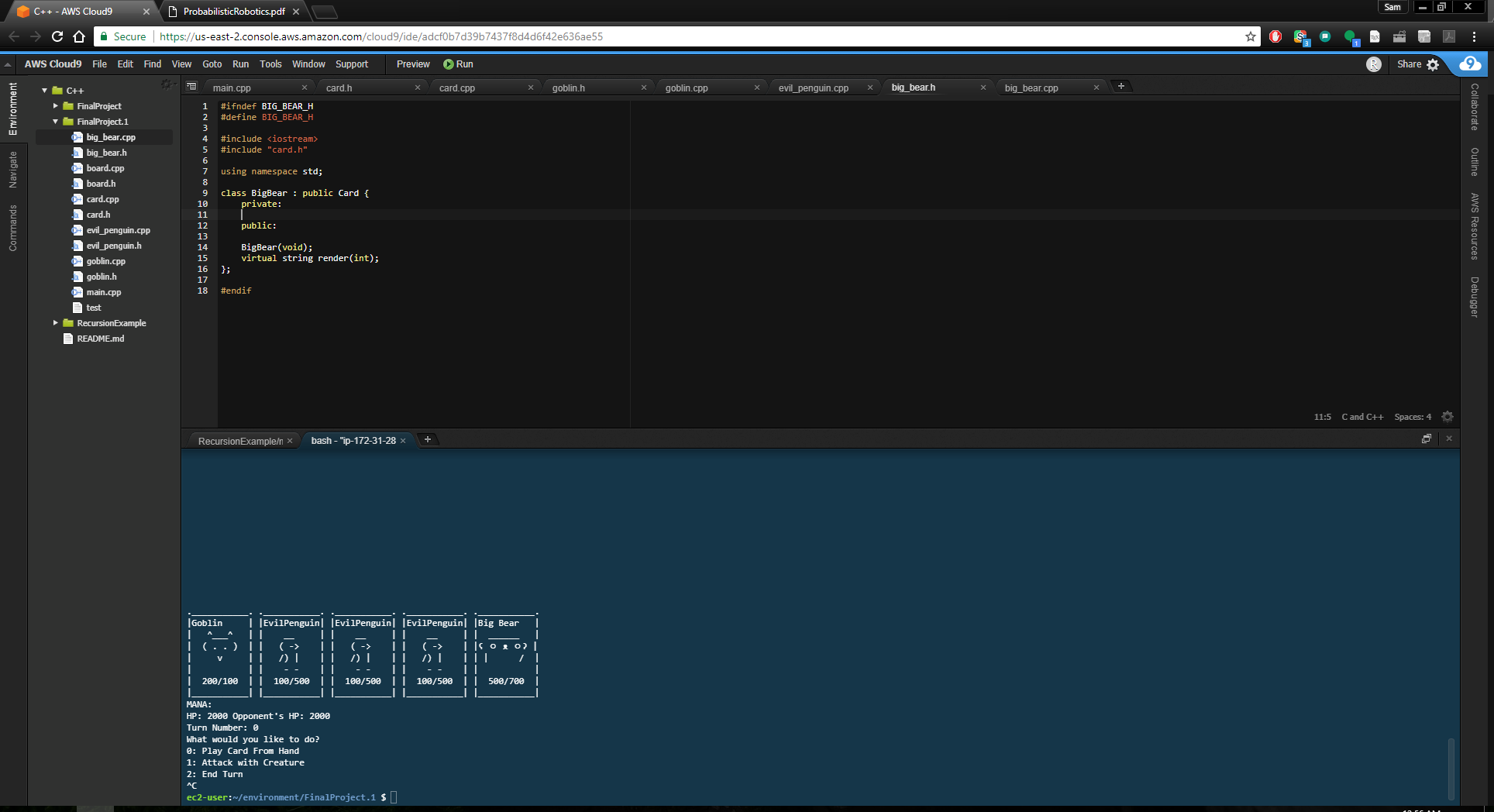
Each player has a customized 20-card deck. This deck is filled with various “creature” cards, each with different properties. Let’s look at an example below:



Here, we have an ASCII representation of a “Goblin” card. Each card has five different properties: a name, attack, defense, mana cost, and exhaustion. This ASCII card displays the name at the top of the card, an “image” of the creature in the center, and two numbers separated by a slash at the bottom. These numbers represent the attack and defense of the creature, respectively. The attack represents how hard the creature “hits” other creatures and opponents, and the defense represents how “hard of a hit” the creature can take before dying. The creature also has a mana cost and exhaustion, but we’ll define that later.

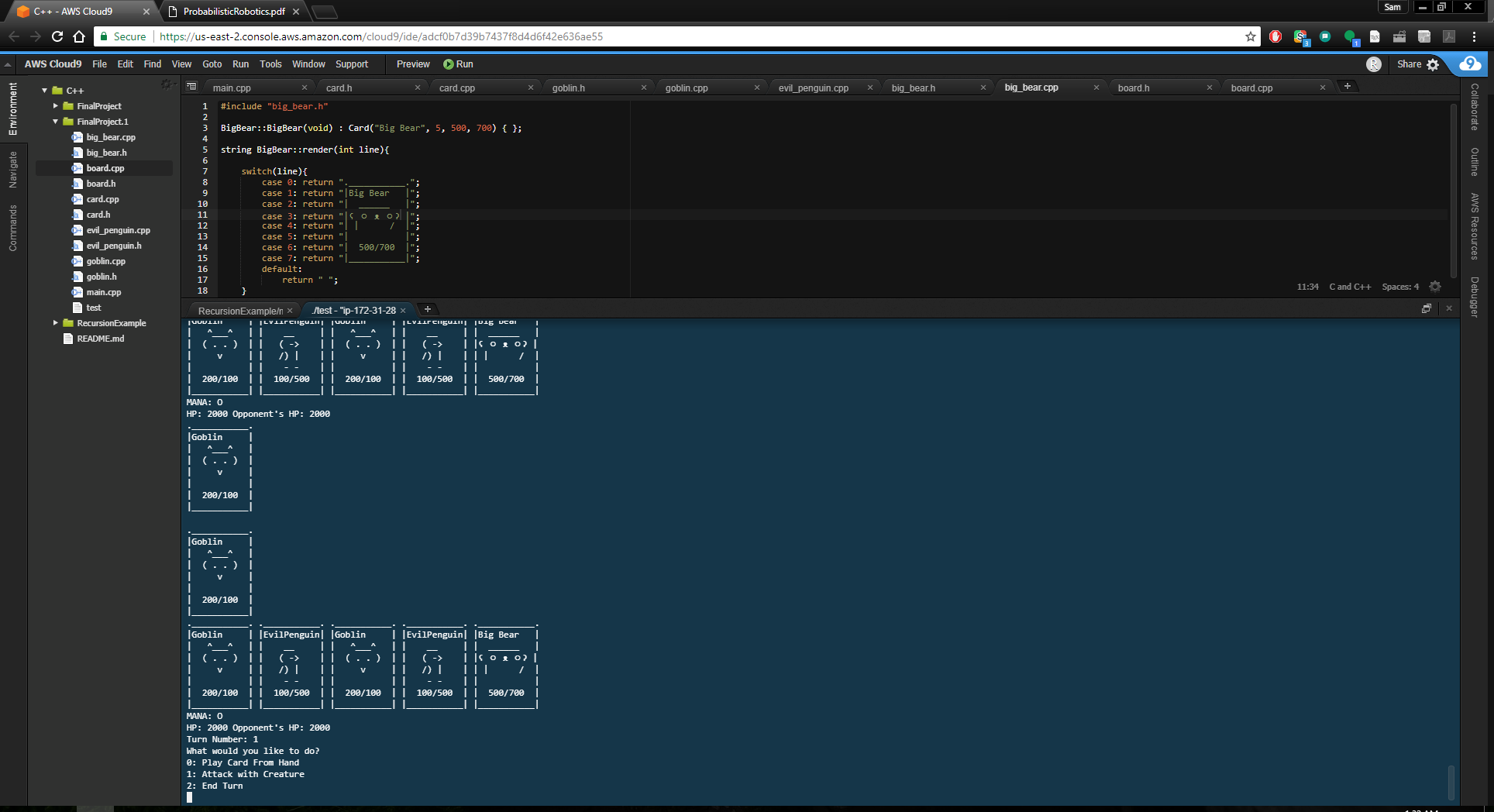
Each player begins the game with 2000 health points, shuffles their decks, and draws 5 cards for an initial hand. A coin toss can be used to determine which player goes first. At the beginning of each turn (except for the first turn of the player going first), each player draws 1 card. To play a card, the player must spend mana according to the card’s mana cost. Mana is a resource generated each round, and is equal to the number of turns that have past. (e.g., On turn 1, you have 1 mana available. On turn 2, you have 2 mana available. On turn 7, you have 7 mana available.) Each card has a mana cost required to play it, and you must have that much mana available to spend on that card. For example, a Goblin requires 1 mana to summon, so if it’s the first card I play on turn 3, I’ll have 2 mana left over after summoning. Mana is refreshed each round, and unspent mana on your turn doesn’t roll over.

When you spend mana to summon a creature, that creature goes onto your side of the “field”. Each turn players can attack with creatures on their side of the field. A creature can be used to attack either the opponent or other creatures. If the creature attacks the opponent, the opponent loses health points equal to the attack of the attacking creature. If the opponent’s health points are reduced to 0, then they lose. If the creature attacks another creature, the defending creature dies if its defense is lower than the attacking creature’s attack. If a creature dies, it is sent to the discard pile.

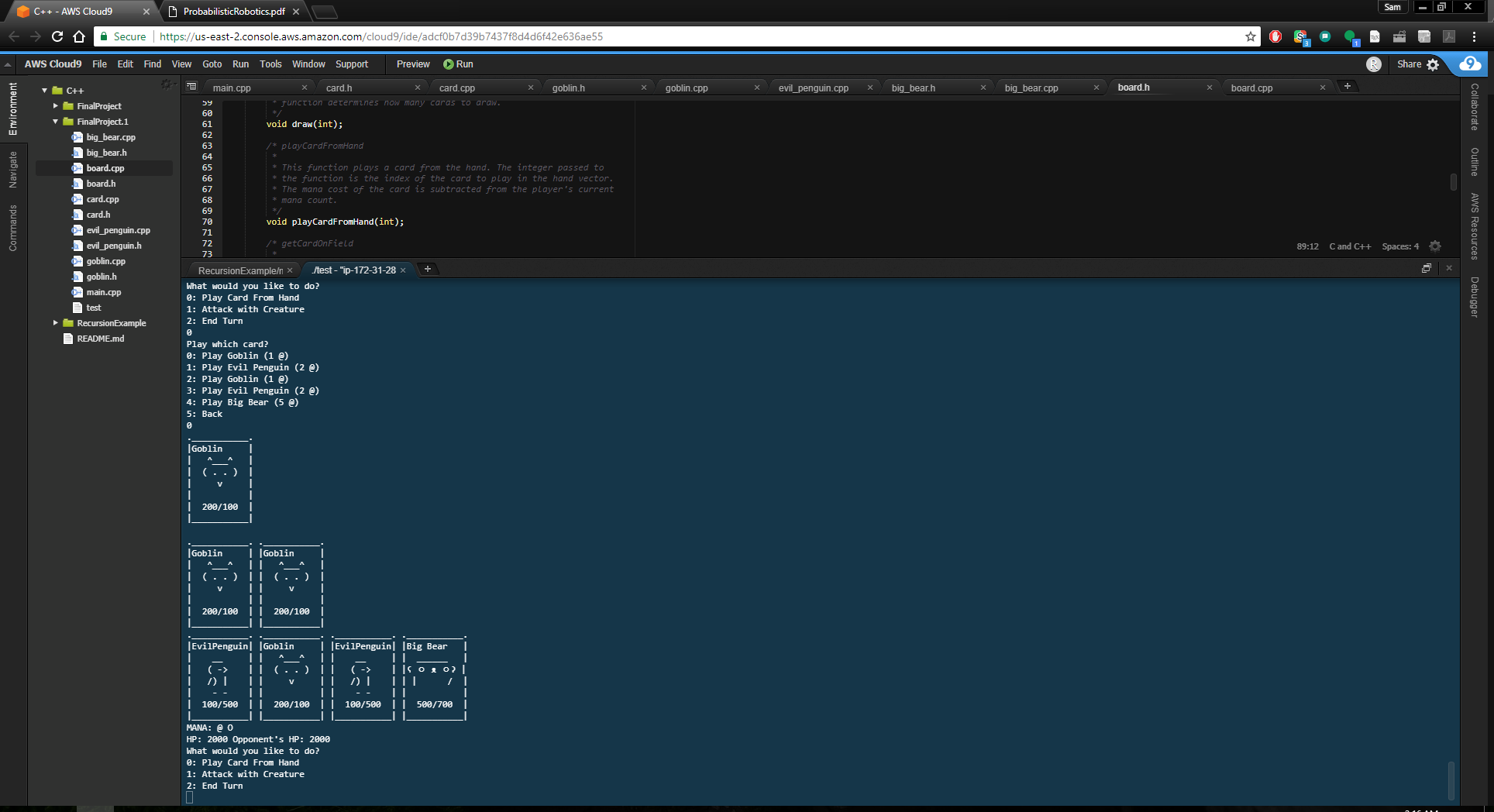


If “Evil Penguin” attacks “Goblin”, “Goblin” will die and be discarded from the field to the discard pile.

Creatures also become exhausted after they attack and when they are summoned. If a creature is exhausted, it cannot attack. Therefore, creatures can only attack once per turn, and not in the same turn when they are summoned.



Above is an example of the board. Here, we have 5 cards in hand, a goblin on the player’s side of the field, and the opponent has a goblin on their side of the field. The current amount of mana is expressed as “@” and “O” symbols. Here, on turn 1 a goblin was summoned, depleting the 1 mana available. Spent mana slots are expressed as “O”. If there is still mana available, it is expressed with a “@” symbol.



Above, it is turn 2, and 1 mana has been spent. We still have 1 mana available, shown by the remaining “@”.

If the player runs out of cards, when the player draws from an empty deck, that player instead loses half their health each turn.

**Programming Hearthstone--:**

Hearthstone--will use two primary classes: Card and Board. The Card class will contain all generic card properties, such as name, mana cost, attack, defense, and exhaustion. Card will also be used as a base class for the other derived classes, such as Goblins and Penguins. The Board class will manage all the other aspects of the game for each player, such as health, deck, hand, and discard pile.

Before the start of the game, the player boards will need to be created. Create a Board object for each player. Decks should be populated for each player, shuffled, and an initial hand should be drawn.

The game should continue to alternate between the player and the opponent taking turns until one player’s health is depleted. The player should be prompted on their turn to either play a card from the hand, attack with a creature on the field, or end the turn. If the player decides to play a card from the hand, they should be prompted for which card to play and display the mana cost for each card. If the player decides to attack with a card on field, the player should be prompted for which card to attack with, and which target to attack, either a card on the opponent’s side of the field or the opponent directly.

**Grading Hearthstone--:**

40 Points Total + Bonus Objective Points

* Generating the Board (5 Points)

You are able to create a “board” object for both players. This means the deck, hand, field, and discard pile are all successfully populated with Card pointers and accessor functions to these vectors work. The Board and Card classes all compile and function as expected. You have the proper mechanics in place for the game.

* Setting up the Game (10 Points)

You are able to shuffle the deck, set player life points to 2000, draw 5 cards, and initialize mana. You are also able to render the hand and field correctly. The deck is also field with creature cards with derivative Card classes. Mana is properly initialized and rendered. A random number generator should be used to “flip a coin” to determine which player goes first.

* Playing the Game (15 Points)
  + Resolving Turns (5 Points)

Each turn a player should draw a card (Except for the first turn of the player going first). Mana totals should increase and mana should be refreshed each turn. The game should continue to alternate between turns for each player until one player’s health is reduced to 0. The program should then acknowledge the winner and stop taking turns/gathering user input for the game.

* + Playing cards from hand (5 Points)

You are able to collect user input and play cards from the hand. The code also checks for proper mana costs when playing a card, and consumes the mana when played or prevents playing the card if the mana cost is not met. The card is also correctly removed from the hand vector and added to the field vector.

* + Combat (5 Points)

You are able to request user input to select a card to attack and determine a target for that card. Applicable targets include opponent’s creatures and the opponent. Combat resolves correctly, causing creatures destroyed by combat to be removed from the field vector and added to the discard vector. If an opponent is attacked

* Adding New Creatures (5 Points)
  + A minimum of 10 different creature cards should be added to the game and made available to each player. At least 10 different creature cards should be made present in either players deck, however, quantities of these cards are up to the programmer’s discretion.
* Proper Rendering and Aesthetic Appeal (5 Points)
  + Various actions will change the state of the board. Each time a change occurs that may alter a player’s health, mana, hand, or field, the game should be re-rendered to reflect this. Proper error messages for invalid actions should be displayed.

Bonus Objectives!

* Deck Building Interface
  + Create an interface for allowing players to customize their decks before playing. (5 Points)
  + Allow users to save and load deck lists by writing them to a text file. (10 Points)
* MakeFiles and GitHub
  + Create a makefile to build your project. (2 Points)
  + Manage your project using GitHub. (2 Points)