Bilkent University
CS 319 Spring 2020
Iteration 1 - Project Analysis Report
Group 1B-SS

Can Cebeci	21703376
Cem Cebeci	21703377
Gizem Karal	21703094
Gökberk Boz	21602558
Sena Sultan Karataş	21604078
Mehmet Ali Altunsoy	21702531

1 Introduction

Slay the Spire is a single player roguelike deck building game where the goal is to follow a map through three levels while fighting various monsters and collecting cards and upgrading the deck. This report describes and analyses the game in terms of functional requirements, non-functional requirements, system models and user interface. In the functional requirements section, there are detailed descriptions of what the game should offer to its users. In the non-functional requirements section, there is a detailed description of how the game works. In the system models section, there are several models that describe how the game works and how users can play the game. In the state machine diagram that shows the different states of the game and the transitions between them, a use case diagram for each state shows what users can do in that state. Also, there is a class diagram that shows the application domain objects and the associations between them. At the end of the report there is a user interface section that demonstrates several frames of how the game will look like in different states of the game.

2 Functional Requirements

The functional requirements of our implementation of the game are described in this section.

2.1 The state of a run at any instance

The state of a run is described by the following:

- Contents of the deck
- The amount of gold the player has
- The relics the player has
- The amounts of maximum HP and current HP the player has
- The potions the player has and the maximum number of potions they can carry
- The map and the player's current position
- The current cost of using the card removal service
- If a combat is ongoing, the state of the combat (explained in detail later)

2.2 Starting a Run

The application boots to the main menu. The player can either start a new run or if an unfinished run exists, keep playing it. The application only keeps record of a single run and starting a new run will require abandoning the current run. The process of starting a fresh run will be described here.

2.2.1 Choosing a character class

After choosing to start a new run from the main menu, the user will be presented with the choice of the character class they will be playing for the run. Each run's character class is determined before it starts and can not be changed once chosen. The character class determines the starting state and the card pool of a run. There are four classes in the game:

- The Ironclad
- The Silent

- The Defect
- The Watcher

The starting states of the four character classes are described in detail in Section 1.2.2.

2.2.2 The starting state of the run

After a character class is chosen, the starting state of the run is determined. The starting state has no randomness involved and it is the same for each run with the same character class.

2.2.2.1 Starting state of The Ironclad

HP: 80 / 80
Gold: 99
Relics:

- Burning blood: “At the end of combat, heal 6 HP”

Potions: None (3 potion slots, all empty)
Cost of Card Removal: 75
Deck:

- 5 Strike
- 4 Defend
- 1 Bash

(10 cards total)

2.2.2.2 Starting state of The Silent

HP: 70 / 70
Gold: 99
Relics:

- Ring of the Snake: “At the start of each combat, draw 2 additional cards”

Potions: None (3 potion slots, all empty)
Cost of Card Removal: 75
Deck:

- 5 Strike
- 5 Defend
- 1 Survivor
- 1 Neutralize

(12 cards total)

2.2.2.3 Starting state of The Defect

HP: 75 / 75
Gold: 99
Relics:

- Cracked Core: “At the start of each combat, *Channel* 1 *Lightning*”

Potions: None (3 potion slots, all empty)
Cost of Card Removal: 75
Deck:

- 4 Strike
- 4 Defend
- 1 Zap
- 1 Dualcast

(10 cards total)

2.2.2.4 Starting state of The Watcher

HP: 72 / 72
Gold: 99
Relics:

- Pure Water: “At the start of each combat,add a *Miracle* into your hand”

Potions: None (3 potion slots, all empty)
Cost of Card Removal: 75
Deck:

- 4 Strike
- 4 Defend
- 1 Eruption
- 1 Vigilance

(10 cards total)

2.2.3 Initial bonus whale

After the starting state is determined, the player is presented with four choices that change the run’s state by a mysterious whale. The first two choices are without a cost and they provide minor bonuses to the run’s state. The possible choices in the first two slots are:

- Max HP +8/6/7/7 (Ironclad/Silent/Defect/Watcher)
- Enemies in the next three combat will have one health.
- Remove a card.
- Transform a card.
- Upgrade a card.
- Choose one of three random cards to obtain.
- A random rare card.
- A random uncommon colorless card.
- A random common relic.
- Receive 100 gold.

The third choice provides a greater bonus than the first two choices but it always comes with a disadvantage. The possible disadvantages are:

- Lose X max health 8/6/7/7 (Ironclad/Silent/Defect/Watcher).
- Take X damage 18/15/18/21 (Ironclad/Silent/Defect/Watcher).
- Obtain a Curse.
- Lose all gold.

And the possible bonuses in the third choices are:

- Remove 2 cards.
- Transform 2 cards.
- Gain 250 gold.
- Choose a rare card to obtain.
- Obtain 2 random colorless cards.
- Obtain a random rare relic.
- Max HP +16/12/14/14.

The fourth choice is always to lose your starting relic and obtain a random boss relic.

2.3 Playing Through a Run

While playing through a run, the player repeatedly chooses a vertex on the map and performs the actions associated with the vertex until either the game is won or the player's HP reaches zero and the game ends. The game is over when the final boss is defeated. The vertices change different components of the run state. In the following sections; the map, the vertices, the actions associated with them and how they affect the run state will be explained.

2.3.1 The map

The map is a directed acyclic graph with a boss vertex that can be reached from any other vertex. All edges are directed vertically upwards. Consequently, the boss vertex is always at the very top. All vertices immediately preceding the boss vertex are rest vertices. The game consists of three acts and a different map is generated for each act. There are multiple vertices from which the player can start the act. The complete map of the current act is visible to the player at all times. There are seven types of vertices differing in the actions they associate with:

- Combat Vertices
- Elite Combat Vertices
- Boss Vertices
- Merchant Vertices
- Treasure Vertices
- Rest Vertices
- Mystery Vertices

The types of vertices and the actions associated with them are explained in the following sections.

2.3.2 Combat vertices

The player engages in combat and can only keep moving once combat is over. After each combat victory, the player earns loot. Loot may include some amount of gold, potions and a choice between three cards to add into the deck. The player may choose not to claim the loot. Combat is explained in detail in section 1.4.

2.3.3 Elite combat vertices

Elite combat vertices are combat vertices that are significantly harder and they reward the player with a relic if they manage to defeat the elite foes.

2.3.4 Boss vertices

In boss vertices, the player engages in combat with the toughest foes. If they can defeat the foes, they will be rewarded with some gold, one of three random rare cards to choose and a boss relic, which are the most powerful of relics. Also, the player is healed to max HP after defeating a boss vertex.

2.3.5 Merchant vertices

In the merchant vertices, the player can exchange gold for relics, potions or cards. The items a merchant has are randomized but merchants always have seven cards, 3 relics and 3 potions. Alternatively, the player can pay an amount of gold to remove a card from the deck permanently, once per merchant vertex. The price of removing a card from the deck starts as 75g and increases by 25g every time the service is used. The player may make any number of purchases (including zero) until they run out of gold.

2.3.6 Treasure vertices

In a treasure vertex, a relic and some gold are offered to the player. The player may choose to take or leave the loot.

2.3.7 Rest vertices

After selecting a rest vertex, the player should choose between recovering %20 of their HP and upgrading a card. Upgrading a card increases its strength. This may be in the form of reducing the cards energy cost, increasing its damage or block or adding a new effect to the card. Each card can be upgraded once unless the card states otherwise.

2.3.8 Mystery vertices

A mystery vertex may behave like a combat vertex, a merchant vertex, a treasure vertex, a rest vertex or it may display some text describing the player's situation and ask them to choose between a number of options, resulting in different trade-offs. The player does not know what a mystery vertex will do until they select the vertex.

2.4 Combat

2.4.1 The state of a combat at any instance

The state of a combat at any instance is described by the following:

- A set of enemies, each having
    - An amount of current HP and maximum HP.
    - A set of status effects (described in detail in section 1.4.4)
    - An intent (described in detail in section 1.4.5)

- The amount of energy and maximum energy the player has.
- Three piles of cards, which are
    - The draw pile, which is the pile the player draws cards from.
    - The hand, which are the cards that are currently playable.
    - The discard pile, which is the pile that contains cards recently played or discarded.
- A set of status effects the player is afflicted by.

2.4.2 Initial state of a combat

- Combat starts with the set of enemies defined randomly. This random definition depends on the type of vertex that initiated combat as well as the progress of the run. Enemies get tougher increasingly as the run proceeds.
- The amount of maximum energy is 3, unless it is modified by relics. The amount of current energy is insignificant since, as stated in section X.X, it is set to the amount of maximum energy at the start of the first turn.
- The draw pile is initially an exact copy of the player's deck. The hand and the discard pile are empty.
- There are no status effects that afflict the player at the start of a combat, unless stated otherwise by a relic.

2.4.3 How combat is played

The player enters combat by

- Choosing a combat vertex on the map
- Choosing an elite vertex on the map
- Choosing a boss vertex on the map
- Choosing a mystery vertex on the map (and the vertex resolving to be a combat)

Combat is played in turns and lasts until either all of the enemies have been defeated or the player has lost all of their HP and lost the game.

Each turn proceeds as follows:

1) The player draws five cards from their draw pile.
2) The energy of the player is set to its maximum value, which is usually three and is subject to modification by relics.
3) The intent of each enemy is declared.
4) The player plays a sequence of cards from their hand. The effects of each card are resolved and then they are put into the discard pile when played.

5) The player decides to end their turn. This usually happens when the player has run out of cards or energy.
6) All cards in the player's hand are discarded, unless an effect states otherwise.
7) Each enemy realizes their declared intent.
8) Another turn follows.

A turn can end half-way at any point if combat is finished.

The main mechanics of combat are to deal damage to enemies using *attack* cards and to protect yourself from damage by gaining *block* using defensive cards.

2.4.4 Status effects

During combat, the player and the enemies may be afflicted by a number of status effects, which can have both positive and negative impact on the afflicted party. Status effects may be applied to a party by relics, potions, cards or enemy actions. In addition, some enemies can start combat with a status effect.

Some status effects have a specified number of turns before they decay, which is referred to as their duration, others last until the end of combat. Some status effects applied by relics may be effective across all combat as long as the player has the source relic
.
When applied to a party multiple times, status effects may stack by duration or intensity or they may not stack at all.

Here are some examples of status effects:

 Strength: Increases attack damage by X.

 Thorns: When attacked, deals X damage back.

 Dexterity: Increases  Block gained from cards by X.

2.4.5 Enemy intents

In each turn, all enemies declare an intent. An intent tells the player what action the enemy is going to take once the player decides to end the turn.
The intent of an enemy may be:

- Aggressive: The enemy will attack the player. The amount of damage the attack will deal is declared as well.
- Strategic: The enemy will inflict a negative status effect on the player. The actual effect is unknown until the intent is realized.
- Defend: The enemy will gain block. The amount of block is not declared.
- Buff: The enemy will apply a positive status effetc to itself.
- Unknown: The intent of the enemy is unknown
- Any two-intent combination of the upper four types.

3 Non-Functional Requirements

3.1 User Interface Requirements

The comprehensibility of the User Interface is a very crucial component of the game for especially new players. Even for players who are experienced in deck-building games, the easy and reliable User Interface of our game will reduce learning time. Any user must be able to navigate around the game menus in less than 10 minutes. This will also increases the focus on gameplay. Besides the interface, having clear and understandable attributes is very important especially for games based on strategic gameplay. In Slay the Spire, our card, relic and potion explanations should be very clear and understandable for at least %90 of the players who speak English.

3.2 Hardware Requirements

Since the program will be written in Java, the game should run on any machine that has Java Virtual Machine.

3.3 Responsivity Requirements

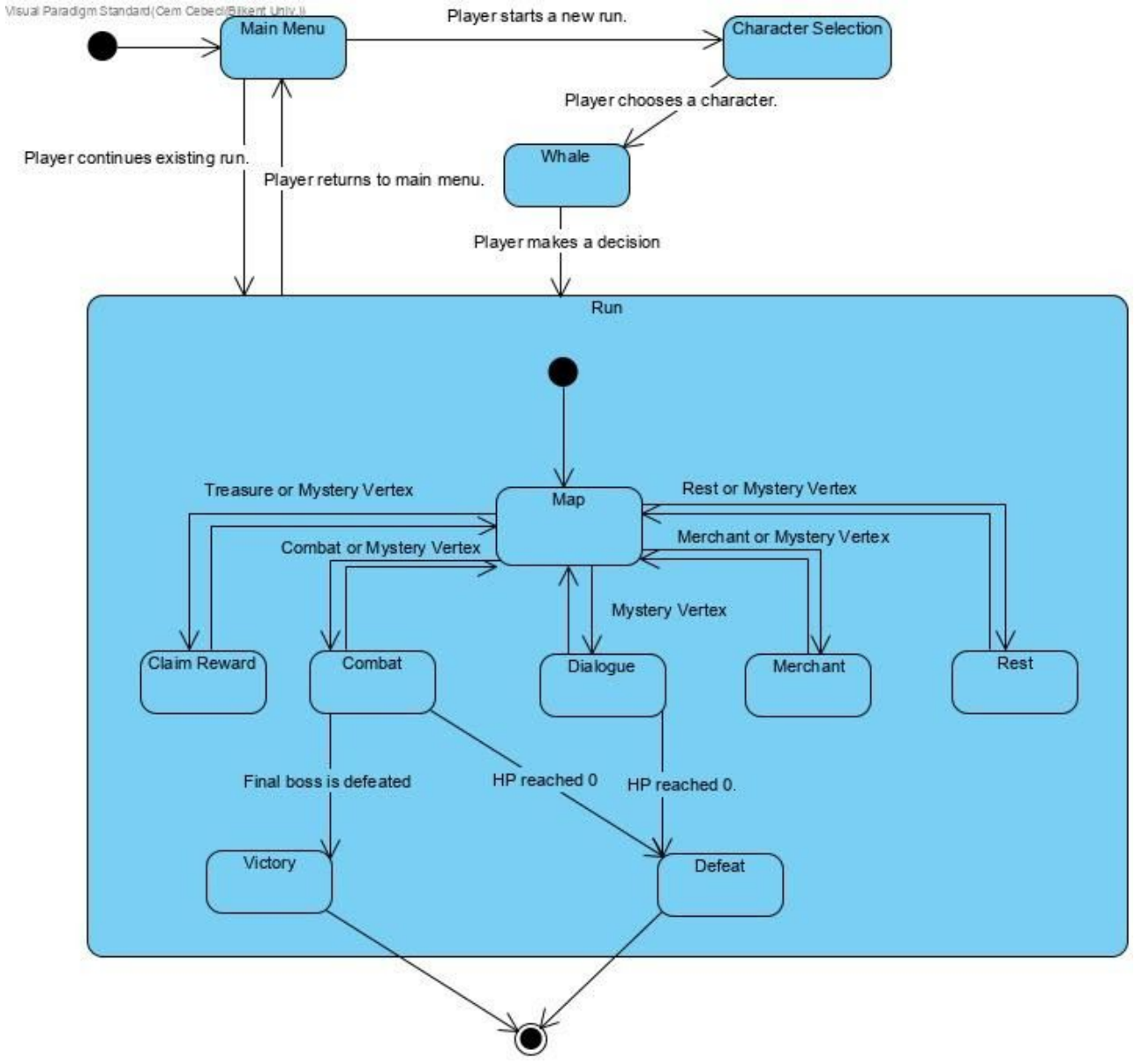
In order to allow a smooth gameplay experience, the game must respond to any action in less than one second.

4 System Models

In this section, the game is described with the help of different UML models.

4.1 State Machine Diagram of the Game

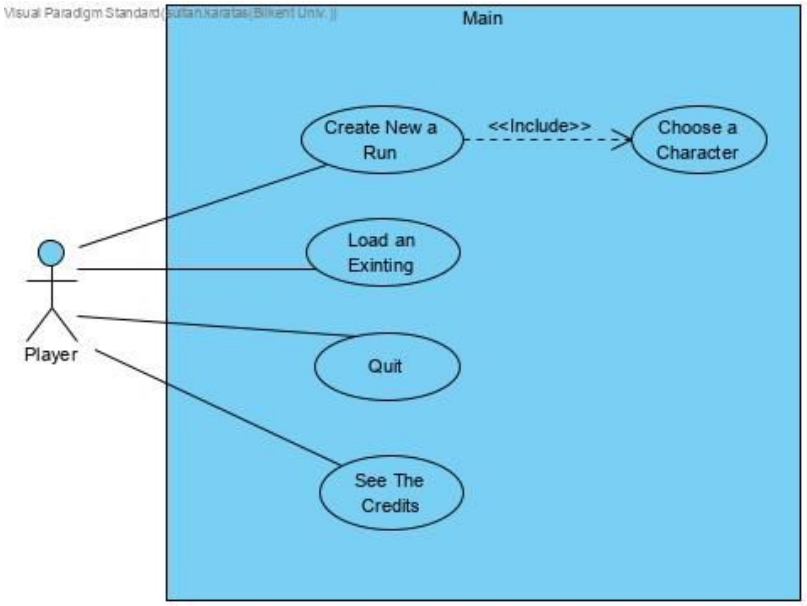
The various states of the game and how transitions happen between them are described by the following state machine diagram.



4.2 Use Case Diagrams of the States

How the player can interact with the game in the game's different states are described by use case diagrams in this section. Only the states with nontrivial use case diagrams are described.

4.2.1 Use cases of the main menu state



Use Case: Create a New Run

Participating Actor: Initiated by Player

Flow of Events: 1. Player clicks on the Start Game button.

2. Player name inputs are filled**.**

Entry Conditions: 1. Game has not started.

2.Player clicks on the Start Game button.

Exit Conditions: Player goes to the Choose a Character use case.

Use Case: Choose a Character

Participating Actor: Initiated by Player

Flow of Events: Player chooses one of the character options.

Entry Conditions: Game has not started.

Exit Conditions: Game has started.

Use Case: Load an Existing Run

Participating Actor: Initiated by Player

Flow of Events: Player resumes the game from where they left off.

Entry Conditions: 1. New game has not started.
2.Character has not selected.

Exit Conditions: The game is continued.

Use Case: Quit

Participating Actor: Initiated by Player

Flow of Events: Player clicks on the Quit button.

Entry Conditions: Game has started.

Exit Conditions: Player should click on Quit button.

Use Case: See the Credits

Participating Actor: Initiated by Player

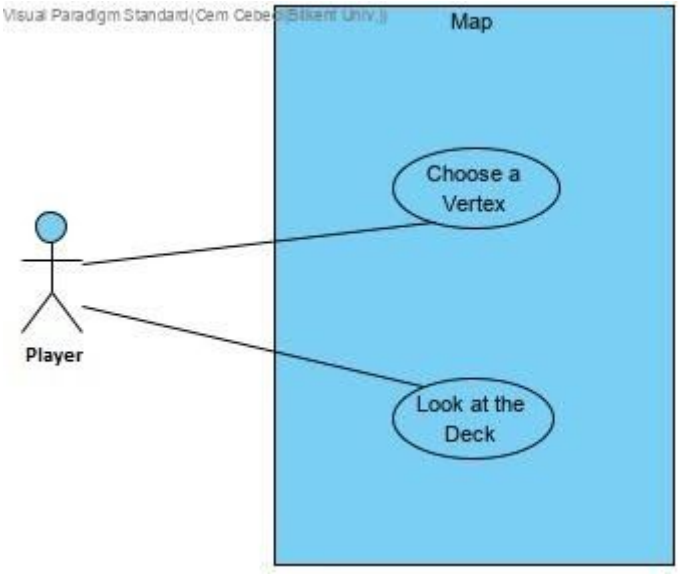
Flow of Events: 1. Player clicks on the settings button.

2.Player sees all people who contributed to the development of the game.

Entry Conditions: No condition.

Exit Conditions: Player click on the credits screen.

4.2.2 Use cases of the map state



Use Case Name: Choose a Vertex

Participant Actor: Player

Flow of events: 1. Player gives the corresponding gold amount to purchase a card.

2. New card is added to the deck

Entry Condition:1. Player should have enough HP.

2. If the new vertex is opened after the previous vertex, then the player can choose it.

Exit Condition: New combat is started.

Use Case Name: Look at the Deck

Participant Actor: Player

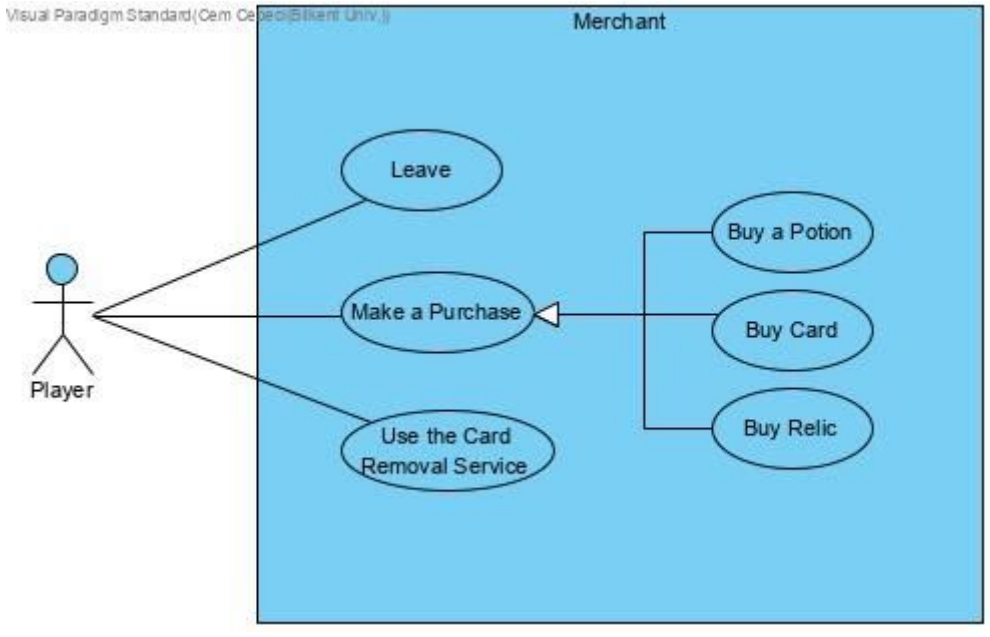
Flow of events: 1. Player sees all of the cards.

2. Player can click on a card and view an upgrade of that card.

Entry Condition: No condition.

Exit Condition: No condition.

4.2.3 Use cases of the merchant state



Use Case Name: Leave

Participant Actor: Player

Flow of events: No flow of events

Entry Condition: Player should have chosen a merchant vertex.

Exit Condition: Player leaves with or without purchases.

Use Case Name: Make a Purchase

Participant Actor: Player

Flow of events: Player decides on the type of the item that will be purchased

Entry Condition: Player must have enough gold to purchase

Exit Condition: **1.** Player does not want to purchase

2. Player does not have enough gold to continue purchasing

Use Case Name: Buy a Potion

Participant Actor: Inherited from **Make a purchase** case

Flow of events: 1. Player gives the corresponding gold amount to purchase a potion.

2. New potion is added to the inventory

Entry Condition: Inherited from **Make a purchase** case

Exit Condition: Inherited from **Make a purchase** case

Use Case Name: Buy Card

Participant Actor: Inherited from **Make a purchase** case

Flow of events: 1. Player gives the corresponding gold amount to purchase a card.

2. New card is added to the deck

Entry Condition: Inherited from **Make a purchase** case

Exit Condition: Inherited from **Make a purchase** case

Use Case Name: Buy Relic

Participant Actor: Inherited from **Make a purchase** case

Flow of events: 1. Player gives the corresponding gold amount to purchase a card.

2. New relic is added to the character

Entry Condition: Inherited from **Make a purchase** case

Exit Condition: Inherited from **Make a purchase** case

Use Case Name: Use the Card Removal Service

Participant Actor: Player

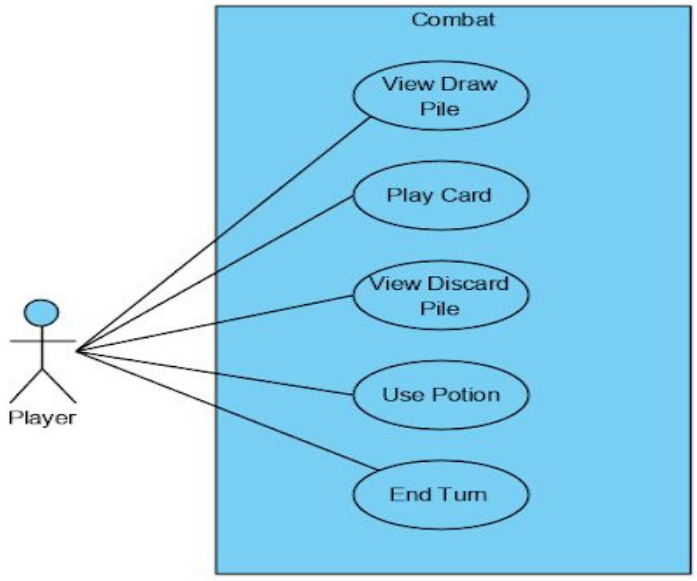
Flow of events: 1. Player pays the needed amount of gold to remove a card form the deck.

2. Player chooses the card that will be removed.

Entry Condition: Player should have enough gold to use removal service

Exit Condition: The card is removed from the deck.

4.2.4 Use cases of the combat state



Use Case Name: View Draw Pile

Participant Actor: Player

Flow of events: Player looks at the cards in the draw pile.

Entry Condition: If there is a card in the draw pile, player can see the cards.

Exit Condition: No condition

Use Case Name: Play Card

Participant Actor: Player

Flow of events: Player uses the card he wants.

Entry Condition: Cards can be used if there is enough energy.

Exit Condition: No condition.

Use Case Name: View Discard Pile

Participant Actor: Player

Flow of events: Player looks at the cards in the discard pile.

Entry Condition: If there is a card in the discard pile, player can see the cards.

Exit Condition: No condition

Use Case Name: Use Potion

Participant Actor: Player

Flow of events: Player uses a potion he wants. The potion is consumed

Entry Condition: The player must have the potion

Exit Condition: No condition.

Use Case Name: End Turn

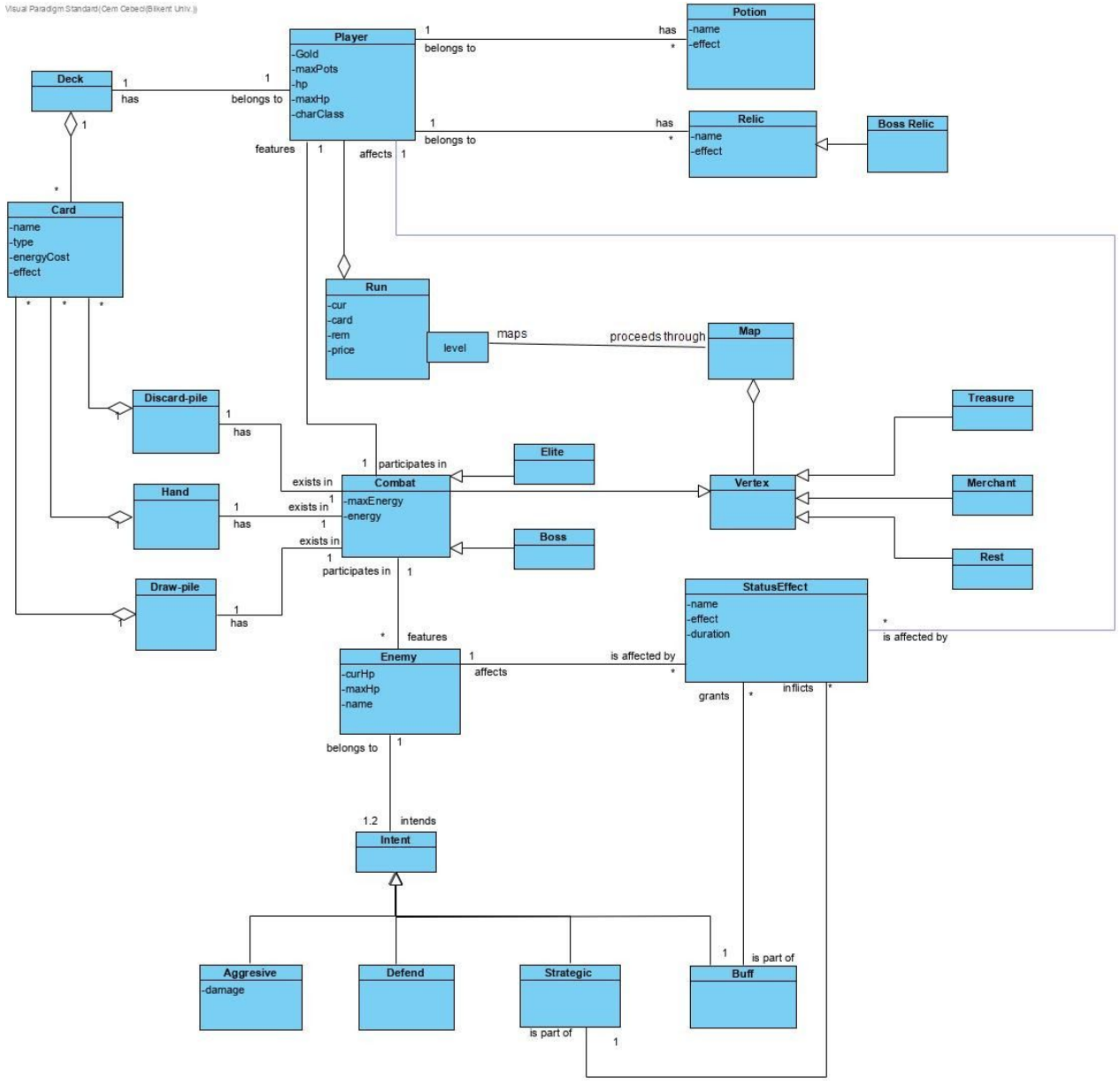
Participant Actor: Player

Flow of events: Player ends his turn and it is enemies' turn .

Entry Condition: If player does not want to choose card, he can end turn without choosing any card.

Exit Condition: No condition.

4.3 Class Diagram of the Game



5 User Interface

In this section, we provide mockups of our user interface design and explain the function of different screens and buttons.

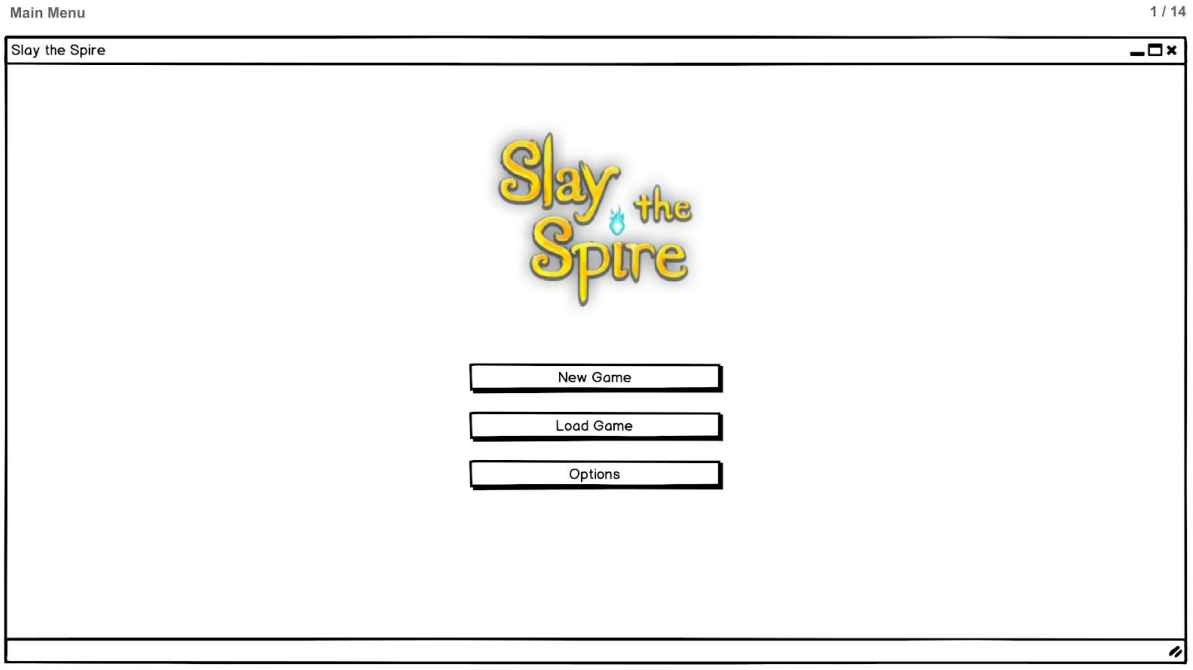


Figure 5.1

The player can choose to either start a new game or load a previously saved game (if any). Also options button leads to technical or gameplay settings as in Figure 5.1

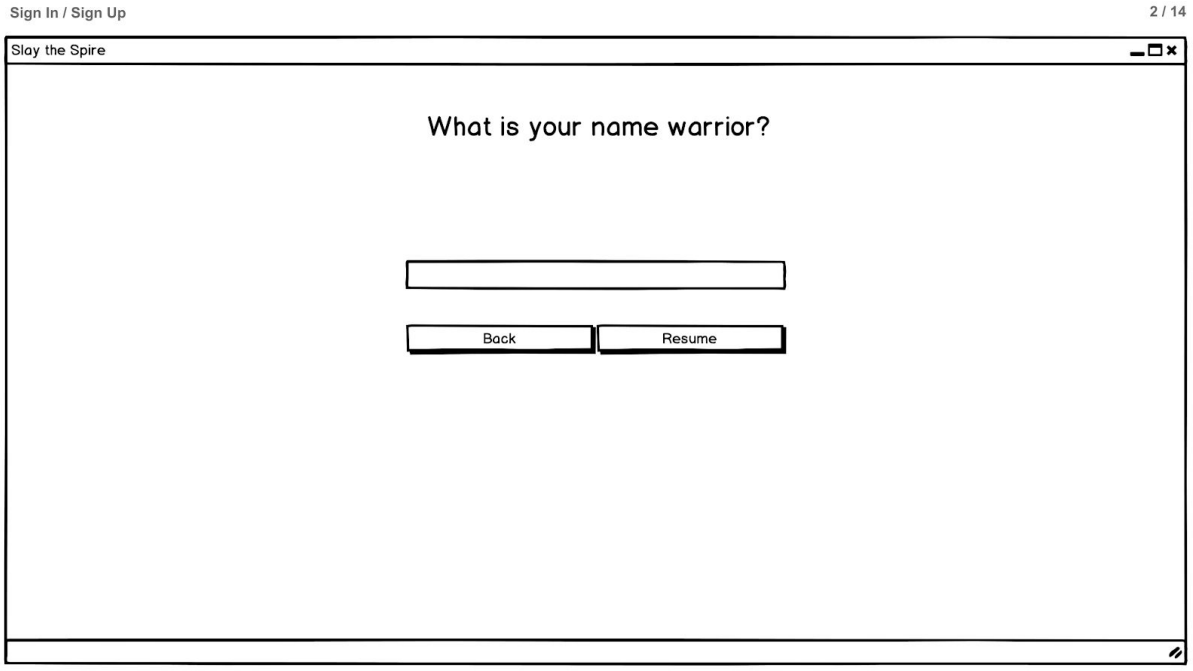


Figure 5.2

To start a new game, the player must enter his warrior's name as in Figure 5.2.

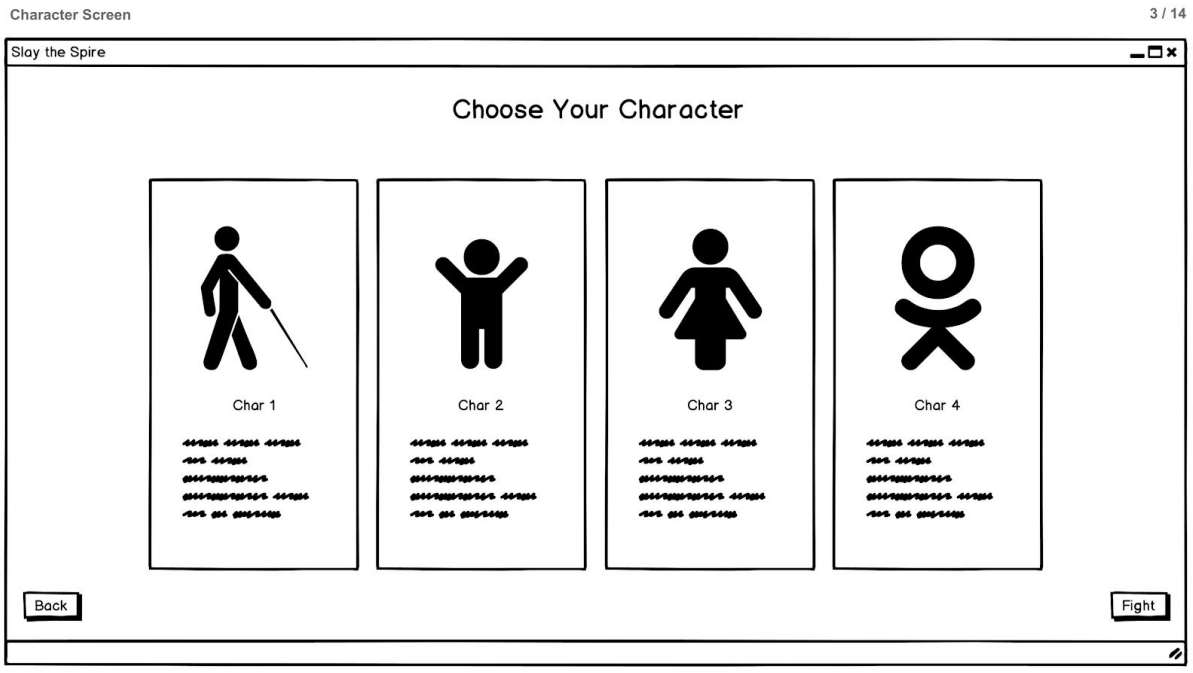


Figure 5.3

Then a character type must be selected as in Figure 5.3.

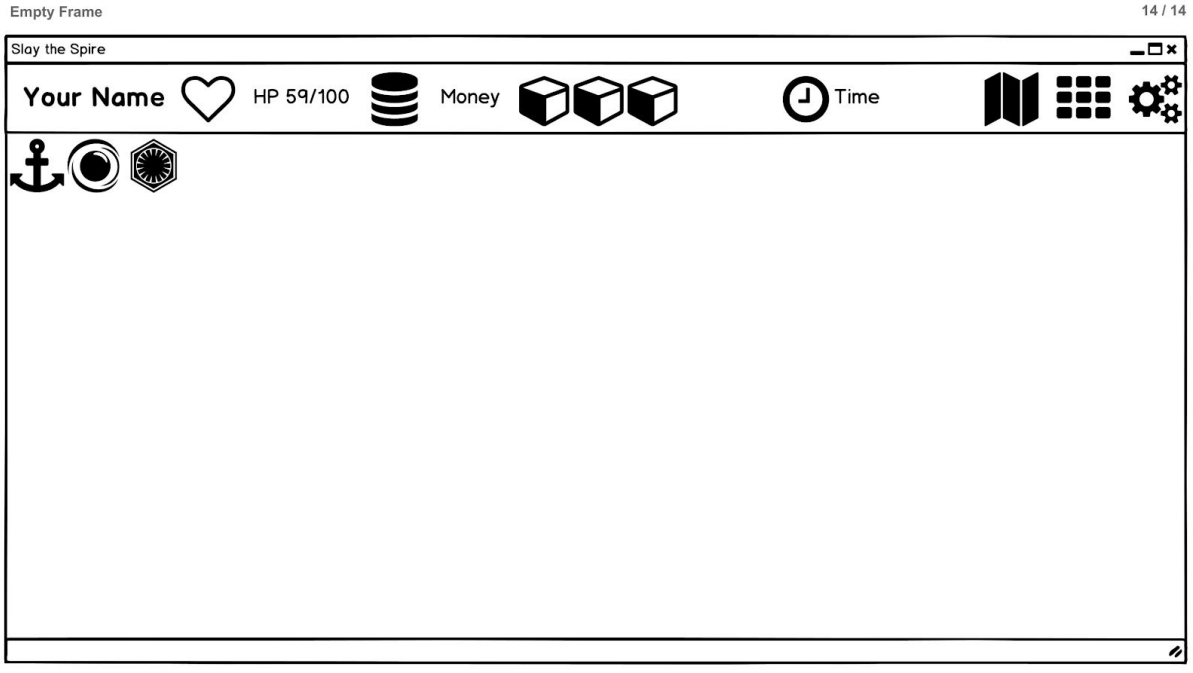


Figure 5.4

In in-game screens there is a default frame at the top of the screen that shows some features of your character such as remaining HP (Health Point), gold, potions and passed time. Also map, your deck and options can be reached through buttons in this frame as in Figure 5.4.

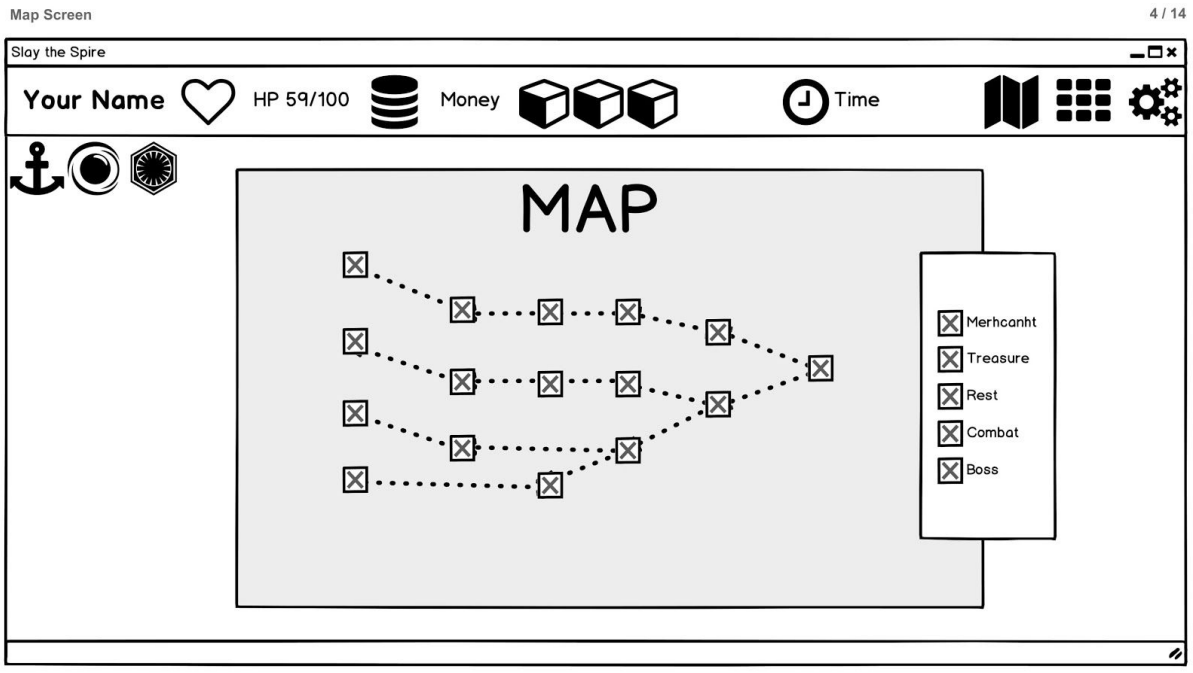


Figure 5.5

The player must climb to the top by fighting. The map guides the player on their way. There are different types of vertices in the path, Combat and Boss vertices are common fighting places but beyond that there are some vertices that holds treasures and also locations to rest as shown in Figure 5.5.

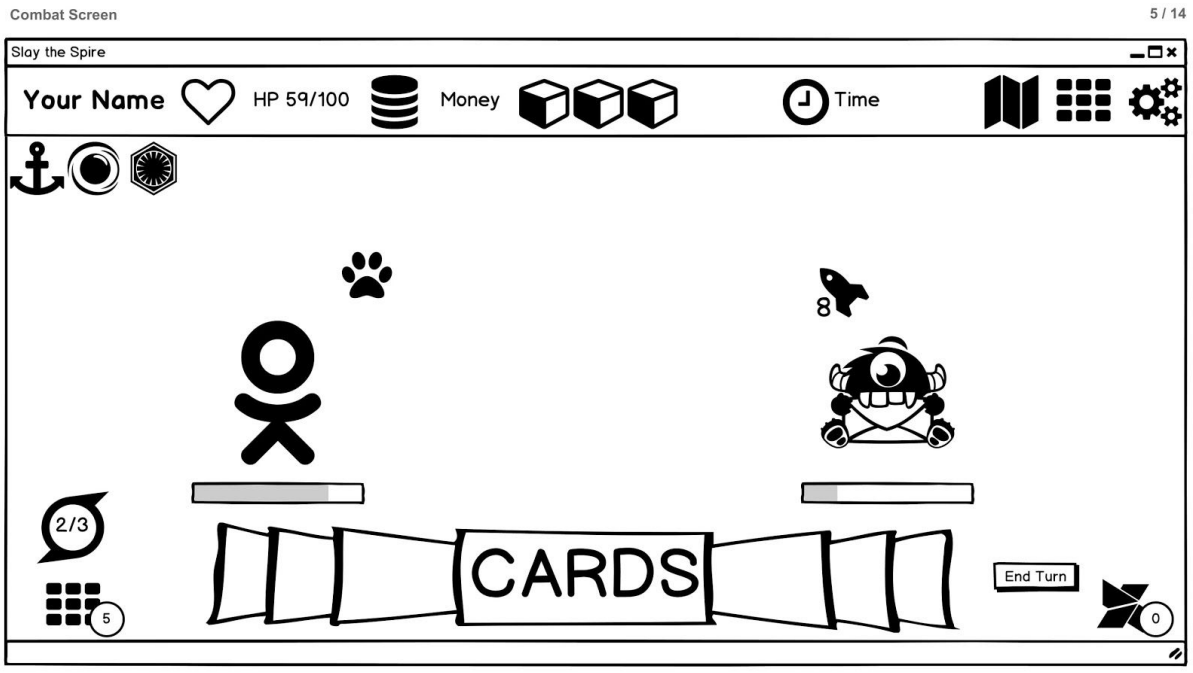


Figure 5.6

The combat screen is the player's fight arena. The enemies are placed at the right part of the screen, the hand can be seen at the bottom. After each turn, remaining cards go to the discard pile (at bottom right) and the player draws new cards from the draw pile (at left bottom). Also each card has an energy cost which is deducted from the player's energy (near the draw pile). Lastly, the current and maximum HP of both the player and the enemies can be seen at the bottom of the figures.

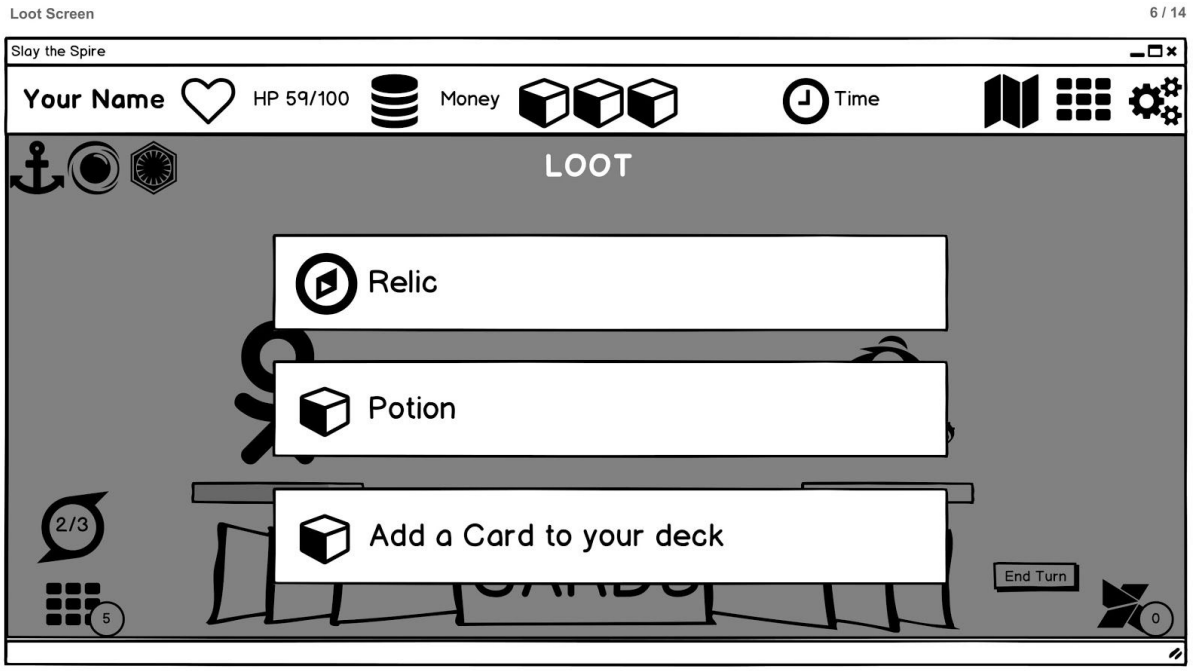


Figure 5.7

After the player defeats the enemy/boss, he will be rewarded. Gold is the main currency in the game to use in market. Card reward adds a new card to your deck. Potions or relics have various effects described in text.

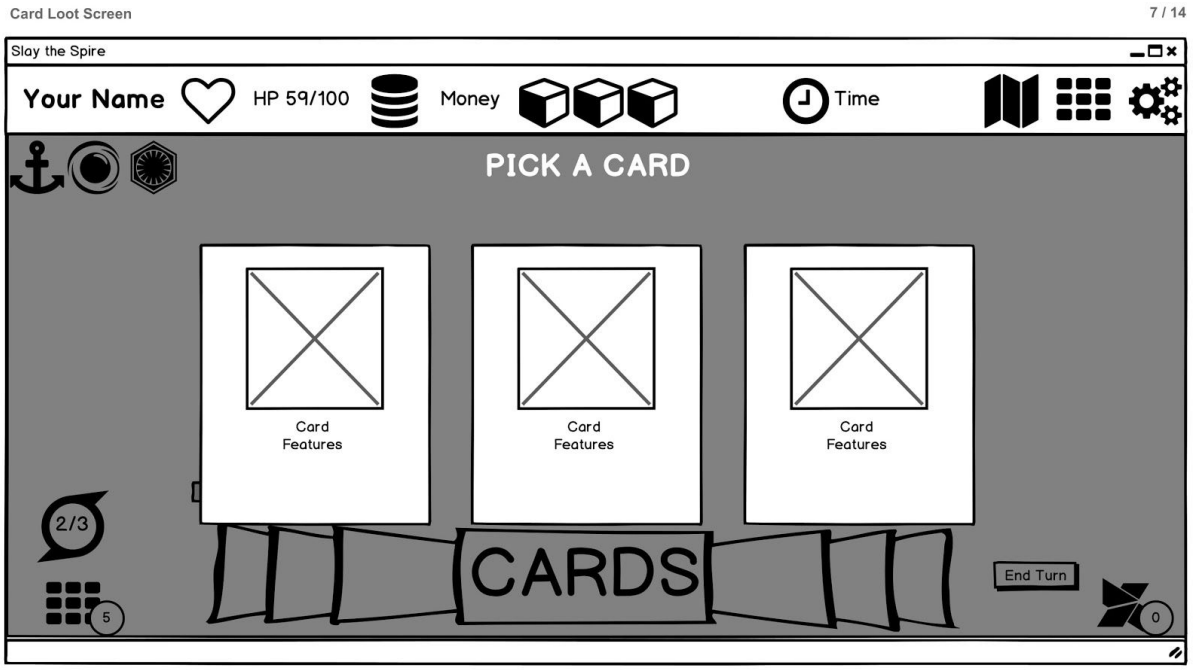


Figure 5.8

The card to add to your deck generally chosen by player from 3 random cards as in Figure 5.8.

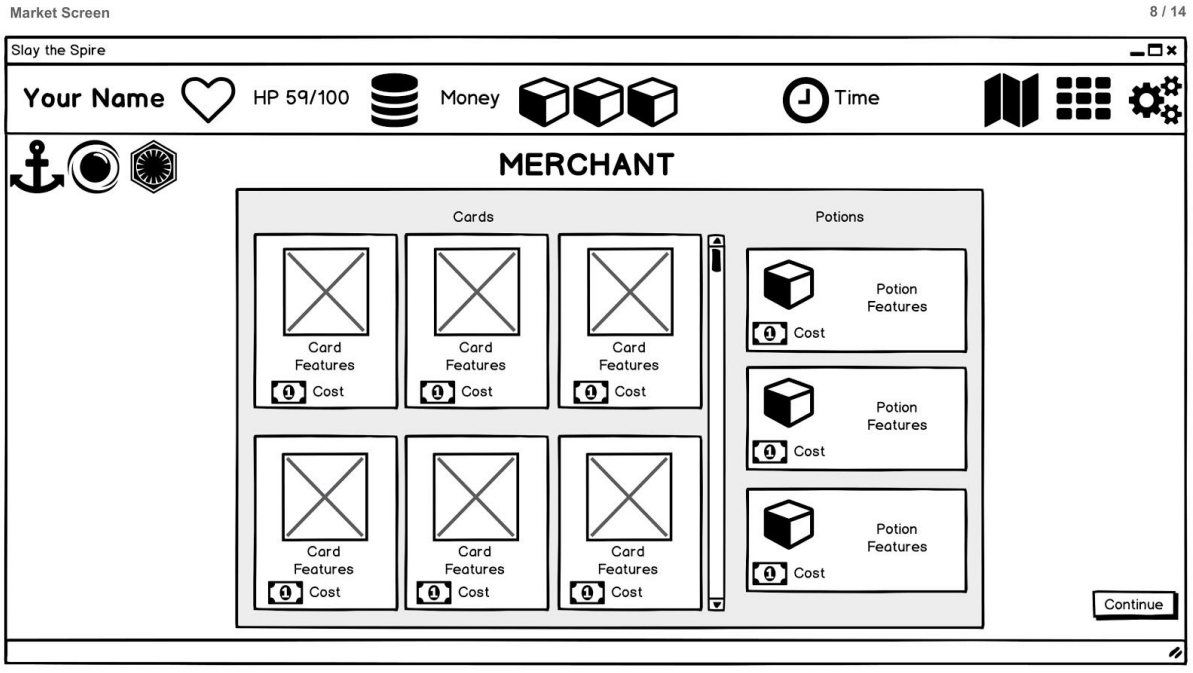


Figure 5.9

In the merchant screen, the player can buy different cards and potions in exchange for gold.

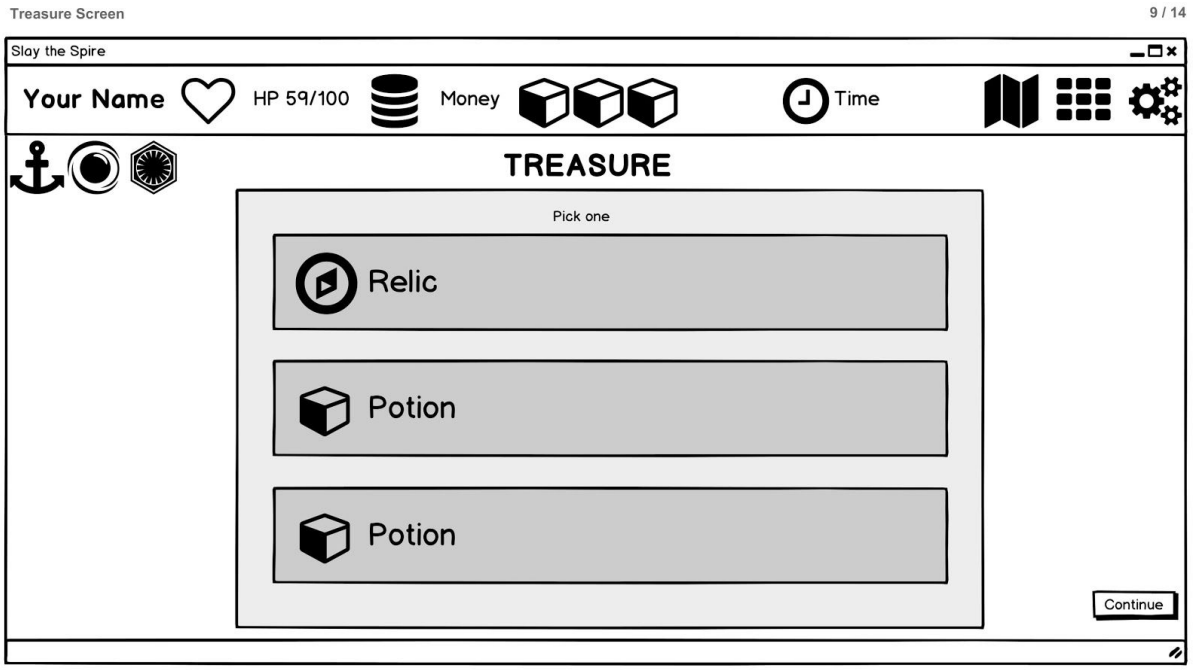


Figure 5.10

Treasure rooms contain a chest that gives you potions or relics. Relics are permanent strong buffs. Potions are more common extras.

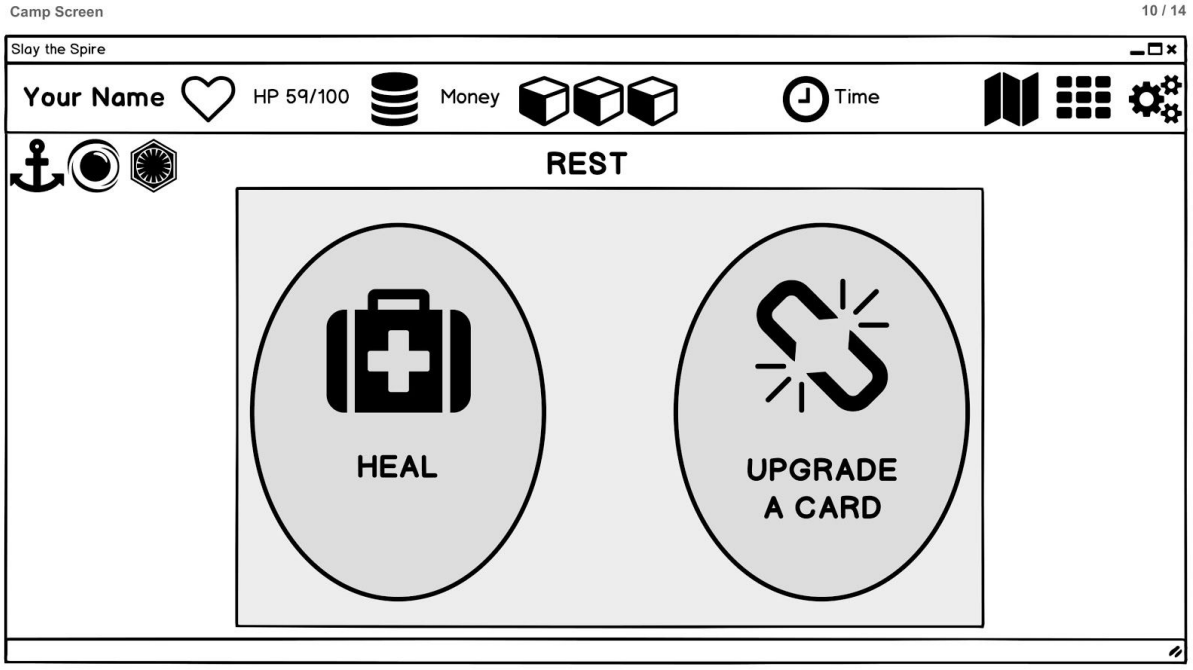


Figure 5.11

In rest vertices, the player has two options. They can either rest and heal a percentage amount of their HP or upgrade a card in their deck.

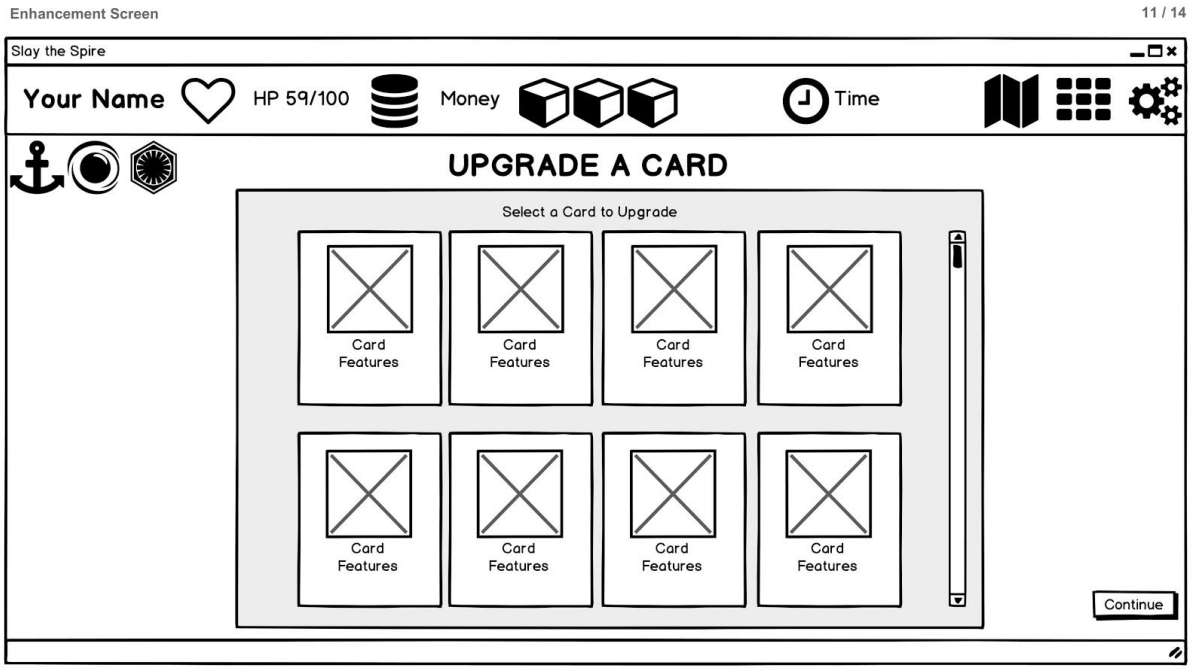


Figure 5.12

If the player chooses the upgrade option, he must pick a card from his deck to upgrade.

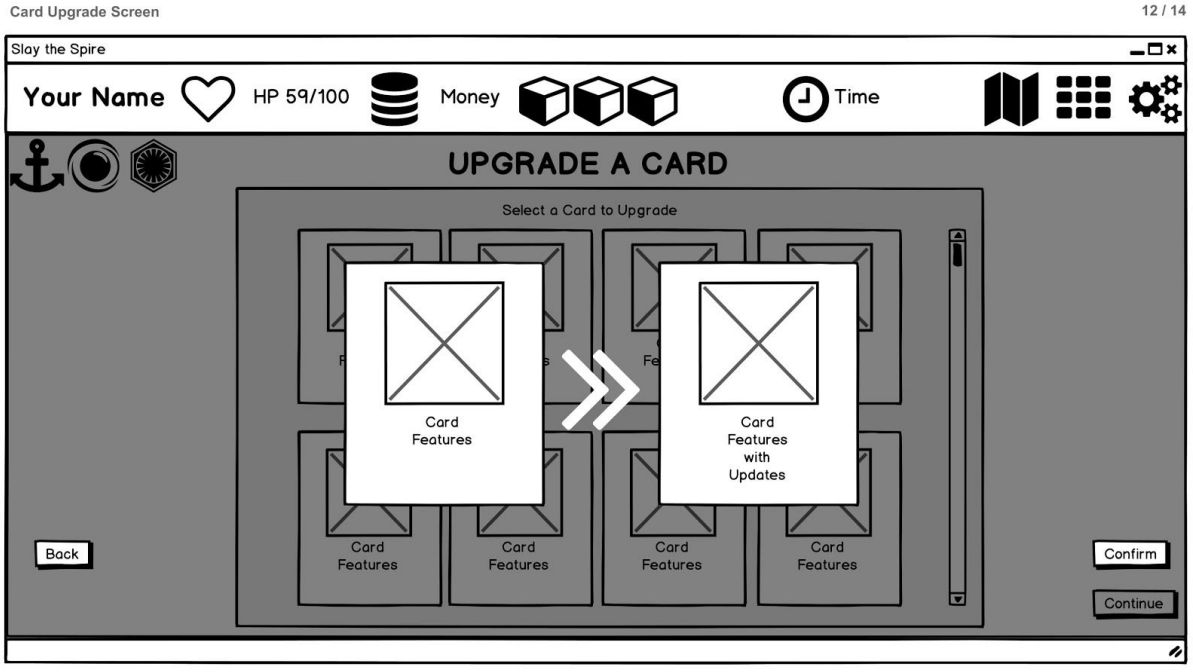


Figure 5.13

The current feature and upgraded version of the card will be shown when the player picks a card as in Figure 5.13.

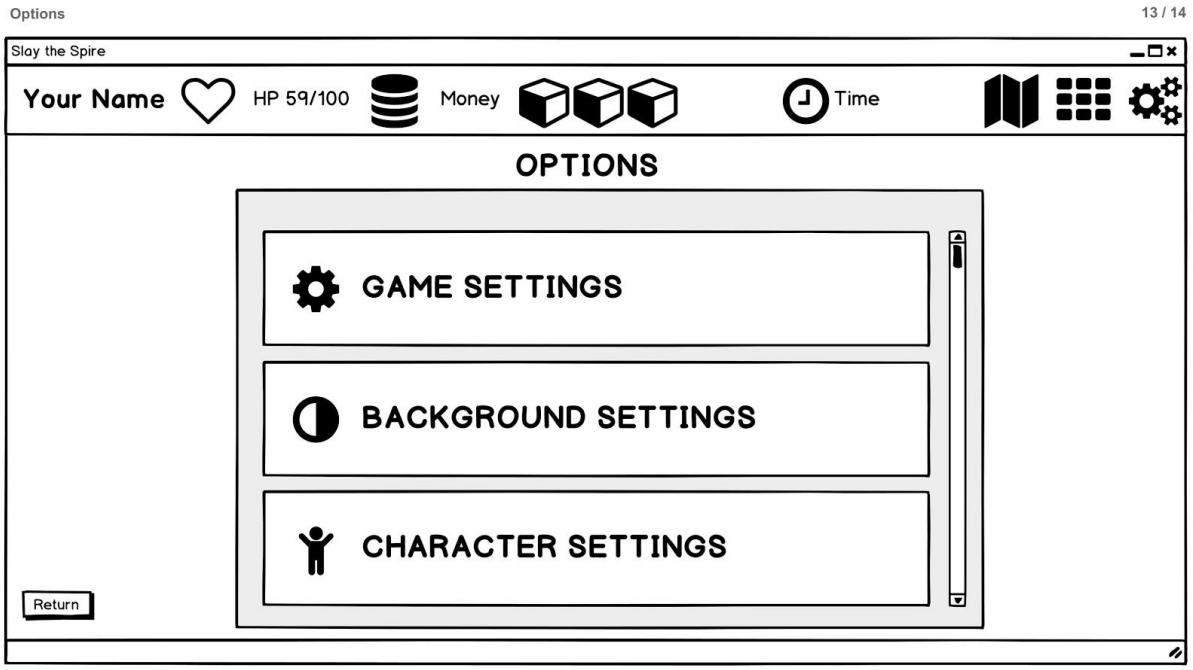


Figure 5.14

In the options screen there are two different settings. One is about game settings which allows the user to change technical options such as graphic and sound, the other shows different themes and gameplay choices.