CPSC 3720 Project

Requirements Specification Template

Team Members:

Cameron Nickle
Brady Hoeksema
Liam Thorne

Atomic Requirements Shell

This is the Requirement type legend

M = ‘Mandatory’

D = ‘Desirable’

O = ‘Optional’

This is the priority legend

1st is the highest priority

2nd is the desirable priority

3rd is the optional priority

1) Cards
2) Special cards vs standard cards
3) 4 play piles
4) (player) objective pile
5) Discard piles
6) Hand regulator
7) Draw pile
8) Winning condition
9) (OPTIONAL) point versin
10) Randomizer
11) Play checker
12) Completed stacks to redraw pile
13) Help

1.) ID: Cards

Type: M

Use case ID: #1

Description: These are the cards numbered 1 - 12.

Rationale:You need cards 1-12 in order to stack the objective pile

Fit Criterion: How well the stacks interact with each other.

Priority: 1

2.) ID: Special cards vs. Standard cards

Requirement Type: M

Use case ID: #2

Description: Having the set amount of standard cards (1-12) in the pile vs. having the special Skip Bo cards

Rationale: Need the splitting up of the cards to play the cards to play

Fit Criterion: The correct amount of cards are built for each type

Priority: 1

3) ID: 4 Play piles

Requirement type: M

Case ID: #3

Description: The place (stack) where players will play their cards to continue the development of the game and the reduction of their objective pile

Rational: Without the play piles you cannot play your cards and therefore you'll never be able to empty your objective pile and therefore cannot declare a winner

Fit Criterion: For the play piles to be considered successful they need to regulate the counting to 12 and require the players to begin them with a ‘1’ as well as re-set upon hitting 12

Priority: 1st

4.) ID: Objective pile

Type: M

Use case ID: #4

Description: This is the stack that each player has, all are hidden except for one. Once you play a card from this stack you can show the next card on top. You want to get rid of this stack as fast as you can.

Rationale:First one to get rid of this stack either a) wins the game. b) earns a point.

Fit Criterion: How well the stack is able to lose cards when played

Priority: 1

5.) ID: Discard pile

Requirement Type: M

Use case ID: #5

Description: Needing discard pile to discard cards from your hand into, so you can continue to pick up new cards.

Rationale: Needed to play the game

Fit Criterion: Cards can transfer from your hand to the discard pile successfully

Priority: 1

6) ID: Hand regulator

Requirement type: M

Case ID: #6

Description: A checking system that simple checks the number of cards (cells) in the players hand (stack) and makes sure that it is 5 at the beginning of the turn as well as 1 card is discarded at the end of each turn

Rational: Without the hand regulator it will be very easy for the players to cheat their hands and hold onto more than they are allowed to

Fit Criterion: For the hand regulator to be considered successful it needs to mandate the discard at end of turn as well as validate that the player have 5 cards in their hands at the beginning of each round

Priority: 1st

7.) ID: Draw Pile

Type: M

Use case ID: #7

Description: This is the stack that everyone draws their cards from

Rationale:Your playing hand must always have 5 cards. Every turn you draw cards until your playing hand has 5 cards.

Fit Criterion: How well the stack is able to transfer cards to players hand.

Priority: 1

8.) ID: Winning condition

Requirement Type: M

Use case ID: #8

Description: Once the objective pile is empty, the winning condition is met and that player wins the game

Rationale: Without the winning condition, once the objective pile is empty the game would continue on.

Fit Criterion: If implemented correctly, the first player to empty their objective pile wins the game and the game ends.

Priority: 1

9) ID: Point version of the game

Requirement type: O

Case ID: #9

Description: A point mode of the ‘skip -bo’ game which tracks the points of each player throughout the game and when the objective pile is empty for 1 player is empty stops the game and calculates the points of all the players and the player with the most points wins the game.

Rational: This is an optional game mode for players who want to attempt a different version of the game or are ‘more competitive’ in nature

Fit Criterion: The point mode will be considered successful if at the end of the game the player with the highest points is derailed the winner

Priority: 3rd

10.) ID: Randomizer

Type: M

Use case ID: #10

Description: Randomizes or “shuffles” the draw stack

Rationale: So players dont keep picking up the same cards

Fit Criterion: Make sure a player does not get the same hand

Priority: 1

11.) ID: Play checker

Requirement Type: M

Use case ID: #11

Description: Checks if the move the player is trying to complete is valid. Mainly if the card is +1 the card before it.

Rationale: Invalid moves will lead to cheating

Fit Criterion: Only the next card can be played. If the card you are trying to play

Priority: 1

12) ID: Completed stacks to redraw pile

Requirement type: D

Case ID: #12

Description: When the draw pile becomes empty you need to refill it to continue playing the game and to do this your take the previous stacks of 12, shuffle, and then place them into the draw pile again

Rational: To maintain the game flow you need to have a draw pile and to avoid excessive stack use it is best to ‘reshuffle’ used cards but is not mandatory option for the game

Fit Criterion: If the maxed piles of cards is ‘shuffled’ and sent back into play without needing to create new cards it will have been successful

Priority: 2nd

13.) ID: Help

Type: O

Use case ID: #13

Description: Displays rules of the game.

Rationale: First time players can get a grasp of the game

Fit Criterion: Displays a document for the game

Priority: 3

Goals of the Project

To take a given set of players and then deal them the cards ranging from 1-12 along with specialty cards. These cards become the objective pile which when empty will spike the win condition of the given mode. At the beginning of each turn each player must have 5 cards and play in the given play piles suitably till the win condition is spiked.

For us to have considered the project successful we need to ensure that the project meets 3 desired criteria, those being:

1) Replay-ability
2) Reliable
3) And generally fun to play on its own

For us to know that we have achieved this goal we will be looking for user review under the circumstance that the player has completed one game and that it did not experience any unforeseen complications. Via sliding pole system.

2. The Stakeholders

The Player's

3.**Steve Jaws** is a lawyer from Queens, NY and is 39 years of age. His favourite past times are Golfing and Skiing. Loves a good smoked salmon. Favourite music is Death Metal. Goes Switz Alps for skiing trips. Has boat loads of money. Would play the game very strategically.

1.Gary Snail is an exotic Veterinarian from Sydney, Australia and is in his late 60's. His favourite past times are surfing and collects rocks. Loves some tasty oysters. Favourite music is Techno and Disco. He camps on the beach for his weekends off. Middle class citizen. He would play the game to forget about his boring life.

2.Kayle Tri is a nurse from London, Ontario and is 25 years of age. Her favourite past times are reading. Loves a good italian pizza. Favorite music south chicago drill. Really likes to explore big cities. Broke and paying off student loans. Would play to have fun and relax.

Development Team (or Other Stakeholders)

Development Master: Removes impediments to developments super twins team. Prioritizes issues. Buffer between team and other distractions. Sets meetings and enforces rules.

Development Super Twins: They develop software and tests however the software developed is tested by the other twin to afford 'fresh eyes' on the software being developed

3. Constraints

Solution Constraints

1.) Description: Must use C++98

    Rational: Must be able to run on the Uleth Linux lab environment

    Fit Criterion: Compiles and runs

2.) Description: Must use GitLab as version control

    Rational: Developers only have university access to GitLab and therefor it must run through the University GitLab server pipeline.

    Fit Criterion: The project can be marked by the marker.

3g. Budget Constraints

1.) Description: Time
    Rational: We have two weeks of development time.
    Fit Criterion: Reasonably functional within two weeks

2.) Description: Varying schedules of developers
    Rational: Time needs to be coordinated in order to facilitate the development process
    Fit Criterion: A “standup” every three days.

4. Naming Conventions and Terminology

4a. Glossary of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project

Objective Pile: The initial hand that is dealt to every player, to which the player can only present one card to play. For the player to win the pile must be gone.

Special Card: Unnumbered card that can be used as any value.

Standard Card: Numbered card that is 1 - 12.

Play Piles: Communal pile that every player can use from their objective pile or hand and work towards achieving a victory.

Discard Pile: The four piles that any player can discard into and then use later to facilitate playing off of their playing pile.

Draw Pile: The pile of which every player will draw from at the beginning of their turn.

Winning Condition: Condition where the player wins i.e objective pile is empty

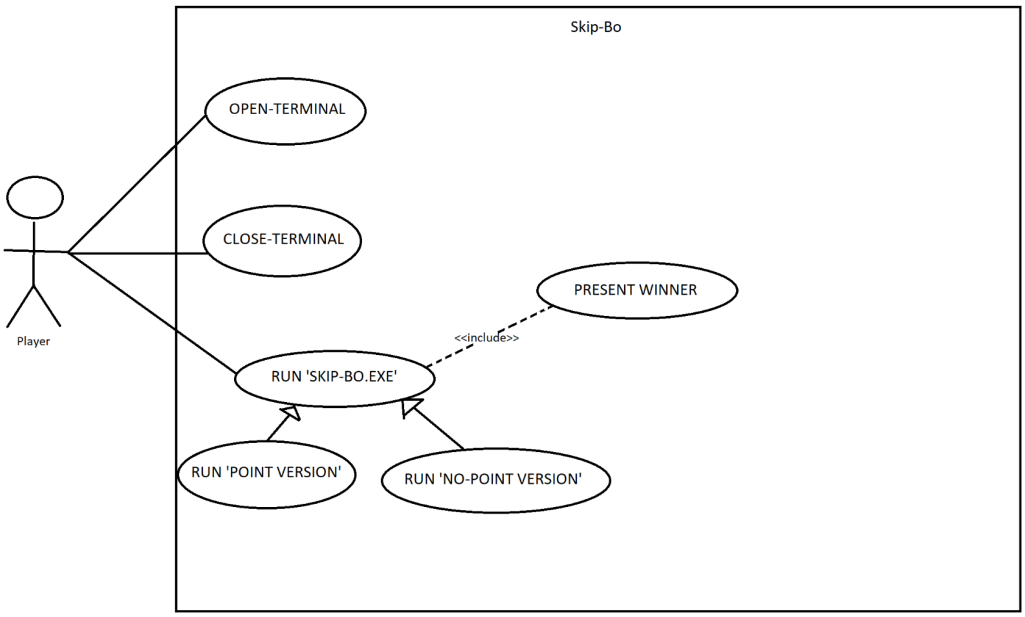
Point Version: Where the points of each pile are kept track of and assigned to a player.

5. Assumptions

1. Assuming that the developers know what a stack is.
2. The developer knows what conditions are.
3. The knowledge on how to use Linux/Unix terminal.
4. Knows Git commands.
5. Knowledge on random number generators

6. The Scope of the Product

Use Case Diagram



Use case 1: Playing the point version of the game, select the point version and play
Use case 2: Playing the no-point version, select the no-point version and play

7. Functional Requirements

This is the Story Point legend

1. **Less than 30 min**
2. **Less than 3 hours**
3. **Less than 6 hours**
5. **Less than a day**
8. **Less than 3 days**
13. **More than 3 days**

Priority	M
Use Cases	1, 2
Rationale	Having the standard set of cards (1-12) as well as the special Skip Bo cards
Fit Criterion	The correct amount of cards are created and they interact with the

	player correctly
Story Points	2

Priority	M
Use Cases	3, 4, 5, 7
Rationale	These are stacks of “cards” that the game needs in order to function correctly
Fit Criterion	The players are able to receive cards from piles and put cards into piles
Story Points	8

Priority	M
Use Cases	6
Rationale	These are the individual hands that players will play from
Fit Criterion	The player is able to lose these cards and gain cards. Must maintain five cards at the beginning of each turn
Story Points	8

Priority	M

Use Cases	8
Rationale	Check if player has won the game
Fit Criterion	The first player to empty their objective pile wins the game and the game ends.
Story Points	5

Priority	O
Use Cases	9
Rationale	Tracks the points of each player throughout the game and when the objective pile is empty for 1 player is empty stops the game and calculates the points of all the players and the player with the most points wins the game.
Fit Criterion	the end of the game the player with the highest points is derailed the winner
Story Points	13

Priority	M
Use Cases	10
Rationale	Shuffles the “draw stack” so that everyones hands are different every time

Fit Criterion	Draw a random amount of hands and make sure no player ends up the same hand they drew before
Story Points	2

Priority	M
Use Cases	11
Rationale	Checks if the move the player is trying to complete is valid. Mainly if the card is +1 the card before it.
Fit Criterion	The player is stopped from making an illegal move
Story Points	5

Priority	D
Use Cases	12
Rationale	When the draw pile becomes empty you need to refill it to continue playing the game and to do this your take the previous stacks of 12,

	shuffle, and then place them into the draw pile again
Fit Criterion	If the maxed piles of cards is ‘shuffled’ and sent back into play without needing to create new cards
Story Points	8

Priority	O
Use Cases	13
Rationale	Displays rules of the game.
Fit Criterion	Type a command into the terminal and the rules or a link to a website is outputted
Story Points	2

Non-functional Requirements

Priority	D
Use Cases	1, 3-7, 10

Rationale	Performance should be thought of during the development process. That way players are not waiting long periods of time in between hands.
Fit Criterion	Players wait less then a second between hands
Story Points	2

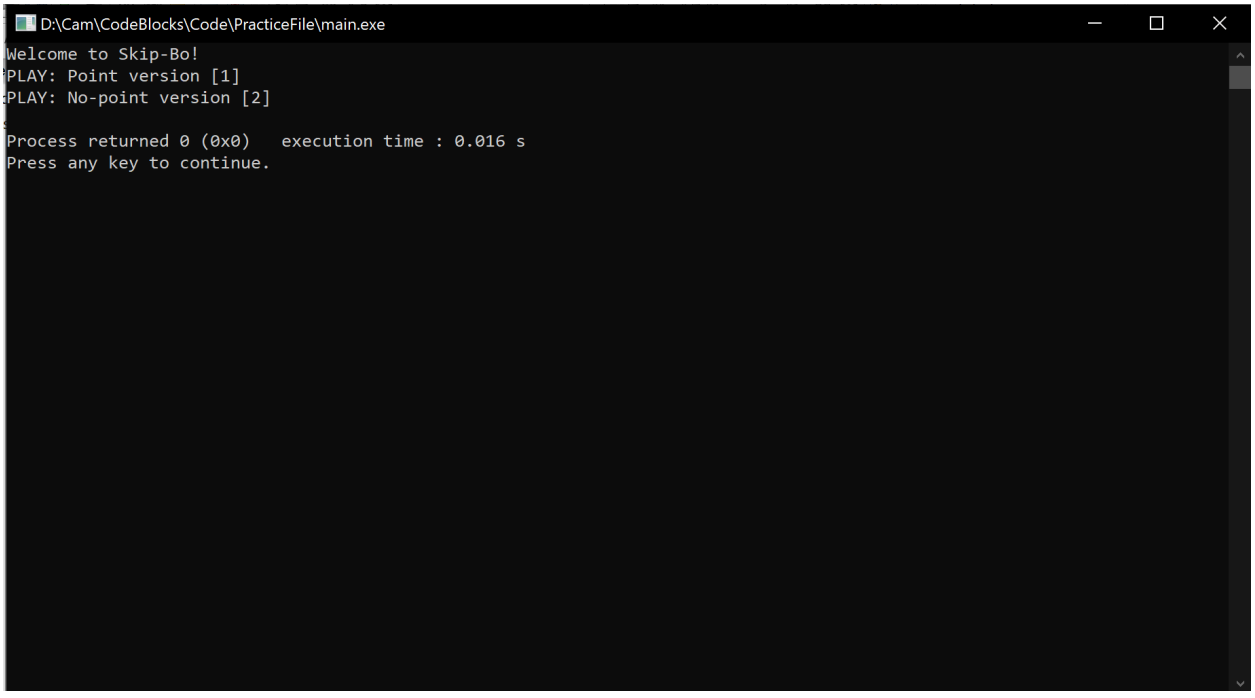
Priority	O
Use Cases	1-13
Rationale	Maintainability if developers decide to take the game live and add new features
Fit Criterion	Well organized and highly readable
Story Points	1

Priority	O
Use Cases	1-13
Rationale	Scalability can be considered depending on how many players you

	would like to play with. More or less cards may need to be created.
Fit Criterion	For more players, more cards. For less players, Less cards
Story Points	5

8. Look and Feel Requirements

We will be using a standard Bash shell/OS Terminal that will take text input from the user and output standard text.



The appearance of a Bash shell/OS Terminal is as follows. This is a high-fidelity interface of the game.

9. Risks

1.) People
    a.) Member of group dropped the course

i.) If someone that is in the team suddenly drops the course, then the remaining members of the team need to shift their focus onto the mandatory functions of the game that will get it operational.

b.) No show team member
   i.) If a team member is MIA and no one can contact them, the other team members need to all contact Dr. Anvik and explain the situation, and similar to a member dropping the course, shift focus to mandatory parts of the system.

c.) Member is unfamiliar with the technology
   i.) If a member is unfamiliar with the technical skills needed for the project, it is up to that member to learn the skills needed and still remain on time for the project.

d.) Sickness
   i.) If one of the team members becomes ill, or contracts COVID-19, then it is their responsibility to let their fellow team members know and provide them with any information they may need to take on their work. The other members will need to accommodate this and help/take over the sick person's work.

2.) Requirements
   a.) Project is too large for the timeframe
      i.) If the deadline is approaching and time is running out, if needed, multiple members should converge onto the parts of the program that absolutely need to get done. Hopefully, any potential issues will be overcome easily if two minds are on the isse.

   b.) Major design change needed
      i.) If a rework of the design is needed, then all members should sit down together and figure out the changes needed so everyone is on the same page. After this is figured out, then members will decide what changes they need to make to their code, or that they believe they can implement easily.

3.) Tools
   a.) Loss of access to primary programming machine
      i.) If one of the members is unable to use their personal computer to complete their portion of the assignment, then they need to make arrangements to work in the labs at the university.

   b.) Lack of internet
      i.) Similar to “Loss of access to primary programming machine”, it will be the team member’s that cannot access the internet to make arrangements to complete their work in the university labs.