Test Document

Team PJA

April 7, 2019

Table 1: Team

Name	ID Number
Annes Cherid	40038453
Benson Chan	40046280
Carl Neil Cortes	40016567
David Boivin	40004941
Gaoshuo Cui	40085020
Karime LouLou	40027203
Ke Ma	26701531
Kevin McAllister	40031326
Robert Laviolette	27646666
Souheil Al-Awar	26558038

Contents

1 Introduction

This document describes the testing procedures planned and executed on the Codenames project by team *PJA* and their results. The tests described have been carefully planned in the early phases of developement (iteration 1) and have been gradually written and executed as the project neare completion and the necessary portions of code were written.

2 Test Plan

The Test Plan is split into 3 sections, System Level tests, Subsystem Level tests, and Unit Tests. Each of these sections represent a level of integration of the program and have been scheduled in order of integration level, namely Unit tests, Subsystem tests and finally System tests. Some test were not implemented for in the scope of the project like Usability testing, Coverage/White Box test,Performance test and Robustness testing.

System Level: Tests the integration and cohesion of all subsystems to form the finished project. This section also includes usability tests which tests if our program can be interacted with as the requirements specify and make sure said interactions produce the correct output efficiently. These tests are of utmost importance as they regulate whether the program functions according to specification and as such were the first to be conceived yet the last to be executed since all subsystems had to be complete and operational prior. Test execution took place in iteration 3 of development and consist of running the program and testing different situations and user interactions, such as card selection and appropriate reactions based on strategy, control button reactions.

Subsystem Level: Tests the cooperation between low level units to form a subsystem, which in turn performs a major and complex operation for the system as a whole. Such operation include use of different strategies, creation of the program's internal model and controller interaction with model. These tests are designed to ensure that subsystems perform their designated task correctly in preparation for system wide testing. This level of testing was done by using the Facade testing pattern providing a sandbox to test components together. Such tests were planned after the system tests as per the white to black box testing scheme and executed in iteration 2 of development due to the requirement of functioning units.

Unit Test: Tests the lowest level functions of each Unit of code, or class within the code, which make up the components of a subsystem. This level of testing is necessarily designed and executed very early in development (iteration 1) to alleviate concerns of flaws and/or bugs prior to subsystem testing and full blown system testing. It is important to note that not all classes have been thoroughly tested, only important data classes

and linking classes within the model and control portions of the program.

2.1 System Level Test Cases

Test Case 1: Player Card Selection

Purpose
Tests the ability for a user to select a card on their turn

Input Specification
Create a new game with a User and an AI (Random or Sequential). If it is an AI start, then press Next Move Control Button until it is User's turn. Click on a card.

Expected Output
Case 1: User Team Card
It is the User Team's turn to play again.
Case 2: AI Team Card or Bystander
It is the AI's turn to play.
Case 3: Assassine Card
The game ends.
Case 4: Selected Card
Nothing happens.

Traces to Use Cases
The User must be able to play the game as an Operative.

Test Case 2: Next Move Control Button

Purpose
Tests the ability of the AI to make a move.

Input Specification
Create a AI vs. AI new game (Random/Sequential). If Red start, press Next Move until Blue team turn. Press Next Move control button.

Expected Output
Case 1: Blue AI 1 Team Card
It is the AI 1's turn to play again.
Case 2: AI Team Card or Bystander
It is the AI 2's turn to play.
Case 3: Assassin Card
The game ends.

Traces to Use Cases
The User can play against AI with intelligent Random or Iterative strategies.

Test Case 3: Undo Control Button

Purpose
Tests the ability of the user to undo any number of moves.

Input Specification
Create a new game. In any order, press Next Move control button twice and select 2 cards. Press Undo control button.

Expected Output
The game state is exactly what it was before playing the 4th move.

Traces to Use Cases
Part of the Undo/Redo system

Test Case 4: Redo Control Button

Purpose
Tests the ability of the user to redo moves, if there are any.

Input Specification
Create a new game. In any order, press Next Move control button twice and select 2 cards. Press Undo control button twice. Press Redo control button.

Expected Output
The game state is exactly what it was before playing the 4th move.

Traces to Use Cases
Part of the Undo/Redo system.

Test Case 5: Keycard Control Button

Purpose
Tests the ability of the user to see the keycard.

Input Specification
Start a new game. Press the Keycard control button. Find the Black assassin card on Keycard screen. Close Keycard screen. Press the card where the assassin should be according to the keycard.

Expected Output
Assasin card should be flipped and game should end.

Traces to Use Cases
User should be able to see the Keycard.

Test Case 6: Quit Control Button

Purpose
Tests whether the game can be closed.

Input Specification
Start a new game. Press Close control button.

Expected Output
Program should end.

Traces to Use Cases
Easily close the program.

Test Case 7: New Game Control Button

Purpose
Tests the ability of the user to start a new game.

Input Specification
Start application, press New Game control button. Select AI types (Sequential/Random).Press Continue button.

Expected Output
Game should be different.

Traces to Use Cases
Create a new game.

2.2 Subsystem Level Test Cases

Subsystem 1: Hint Subsystem

Purpose
Tests Spymaster hint selection and AI word selection based on given hint.

Input Specification
Using Appendix 2, imported as a list of Cards specifically designed to test hint selection for spymaster and plausible card selection list for AI. Test can be found in *HintsTestFacade.java*.

Expected Output
Spymaster Selected hint "2" and AI selects Card "1".

Traces to Use Cases
AI use of hints to make intelligent decisions.

Subsystem 2: Model Creation Subsystem

Purpose
Tests the creation of the model and all data classes. Making sure the GameBoard is generated correctly.

Input Specification
Generates a GameBoard from predetermined set of cards found in Appendix 3.

Expected Output
Card list matching Appendix 3 and Card at index 15 should be Card[15]: RED - Word: Beef - Revealed: false and starting team is BLUE.

Traces to Use Cases
Insures the model properly represents the game so the AI can use it properly and View can use it to display.

Subsystem 3: Inbox Subsystem

Purpose
Tests if Message and Inbox linking components work together correctly.

Input Specification
Test can be found in *MessageToInboxFacade.java*

Expected Output
Message displaying each message type.

Traces to Use Cases
Ensures proper communication between each MVC part.

2.3 Unit Test cases

DatabaseExtractor unit test

Test name	getTextFromDatabase()
Purpose	Make sure that the class is able to enter the database and extract info from it
test result	Pass

Test name	takeAllLine()
Purpose	Make sure that it take all the hundred words from the data base
test result	Pass

Test name	noNull()
Purpose	Make sure that there is no duplicate word in the database
test result	Pass

Inbox unit test

Test name	getMessage()
Purpose	Make sure that getter and that the function sendMessage() work Properly
test result	Pass .

KeyCard

Test name	rightNumberOfBlueCard()
Purpose	Make sure that the number of blue card is good
test result	Pass

Test name	rightNumberOfRedCard()
Purpose	Make sure that the number of red card is good
test result	Pass

Test name	rightNumberOfBlackCard()
Purpose	Make sure that the number of black card is good
test result	Pass

Test name	rightNumberOfyellowCard()
Purpose	Make sure that the number of yellow card is good
test result	Pass

LogMessage

Test name	getType()
Purpose	Make sure that it's able to retrieve the type data without any problem
test result	Pass

Test name	getCardAffected()
Purpose	Make sure that it's able to retrieve the Card Affected data without any problem
test result	Pass

Test name	getHint()
Purpose	Make sure that it's able to retrieve the right hint Affected data without any problem
test result	Pass

Message

Test name	getMessageType()
Purpose	Make sure that it's able to retrieve the message type without any problem
test result	Pass

Test name	getCardAffected()
Purpose	Make sure that it's able to retrieve the Affected card without any problem
test result	Pass

Outbox

Test name	getReply()
Purpose	Make sure that it's able to retrieve the Reply without any problem
test result	Pass

Test name	getHint()
Purpose	Make sure that it's able to retrieve the hints without any problem
test result	Pass

Reply

Test name	getReplyType()
Purpose	Make sure that it's able to retrieve the Reply type without any problem
test result	Pass

Test name	getCardAffected()
Purpose	Make sure that it's able to retrieve the card without any problem
test result	Pass

Test name	getCardType()
Purpose	Make sure that it's able to retrieve the card type without any problem
test result	Pass

Test name	getBlueScore()
Purpose	being able to retrieve the blue score
test result	Pass

3 Test Results

3.1 System Tests

Test Case 1: Player Moves

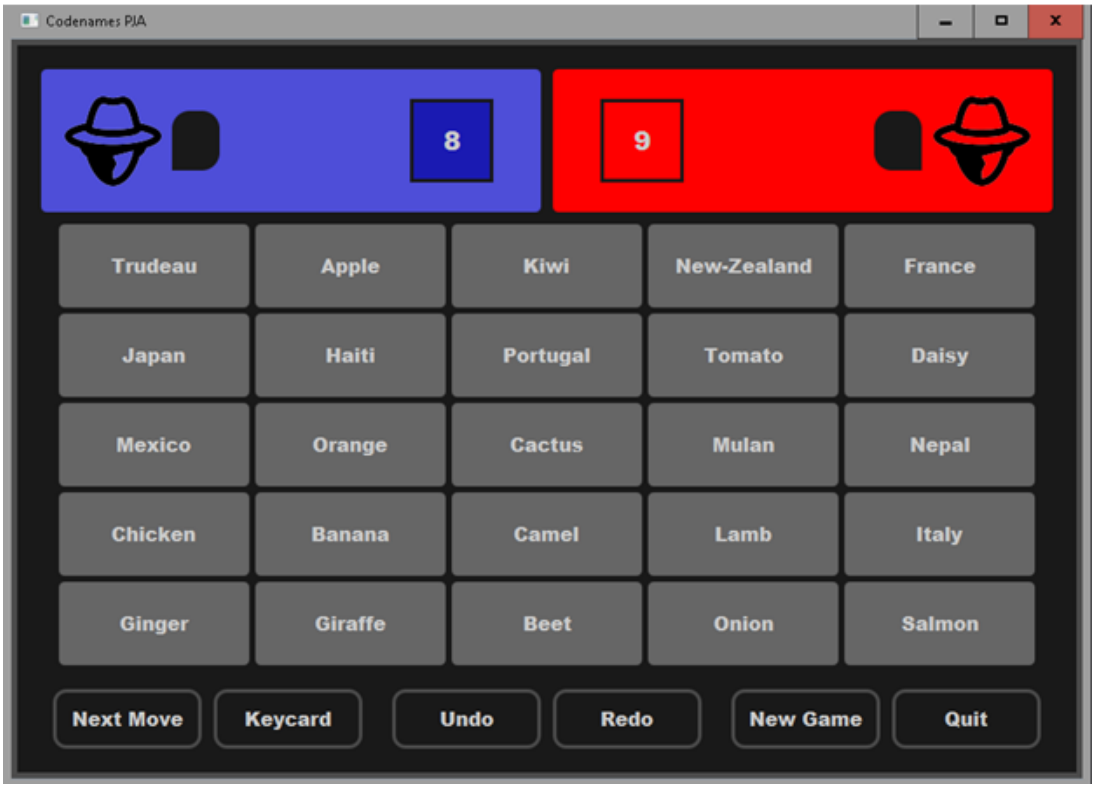

Figure 1: Starting board

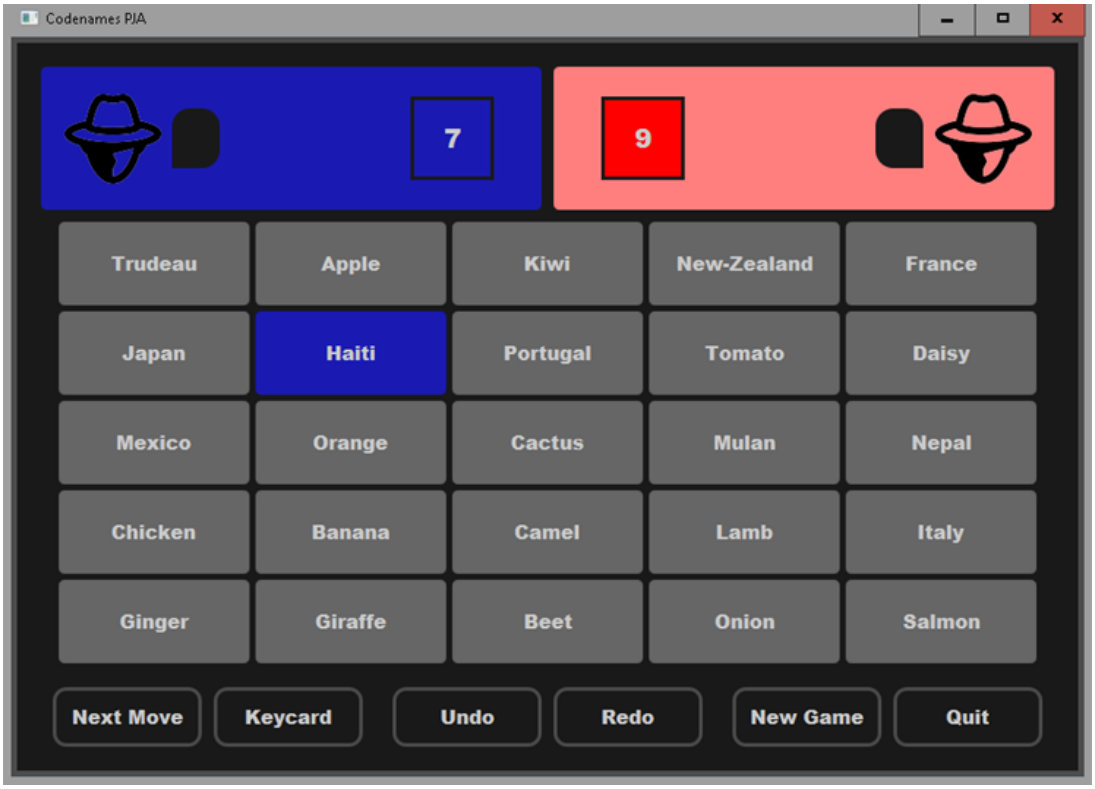

Figure 2: Blue picks blue card, score updates, same turn

Figure 3: Blue picks blue card, score updates, same turn

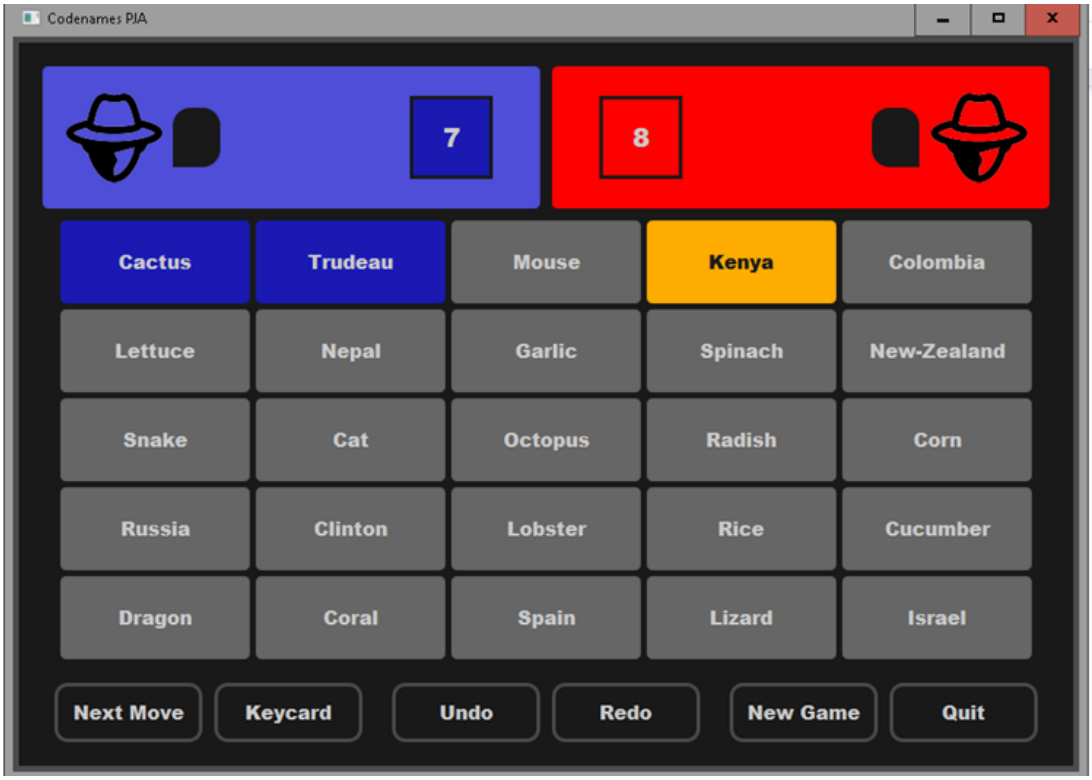

Figure 4: Blue picks yellow card, red turn

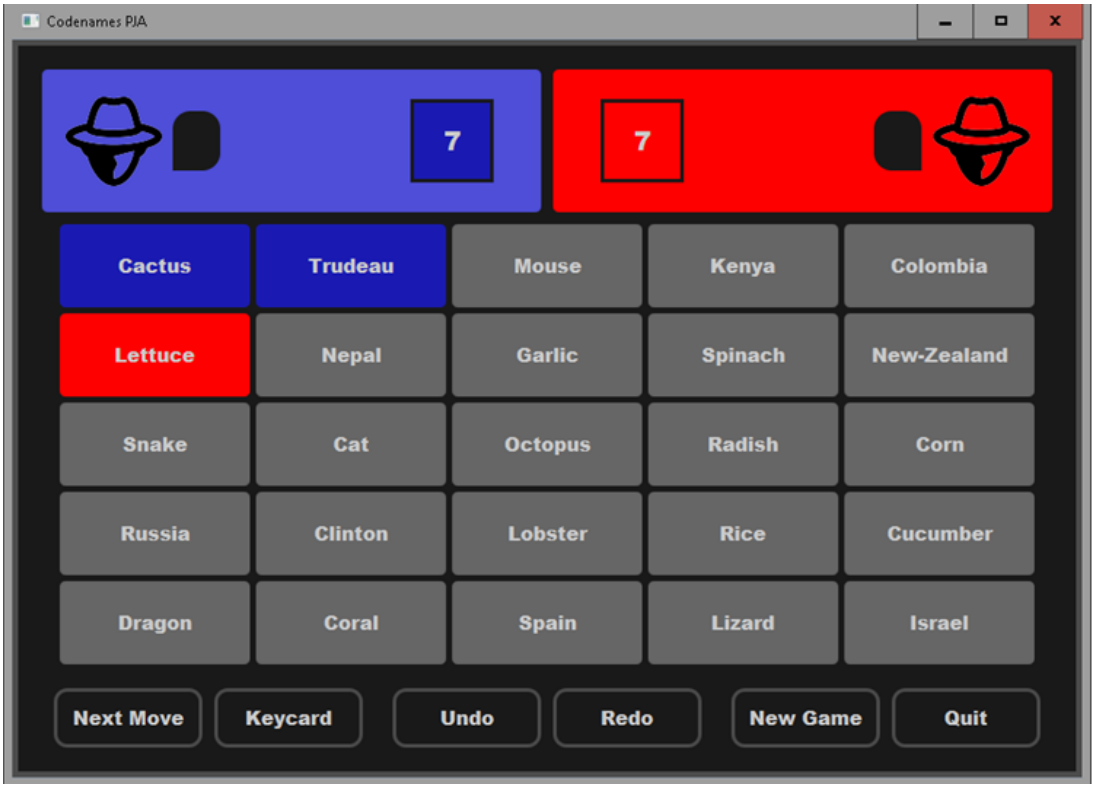

Figure 5: Red picks red card, score update, same turn

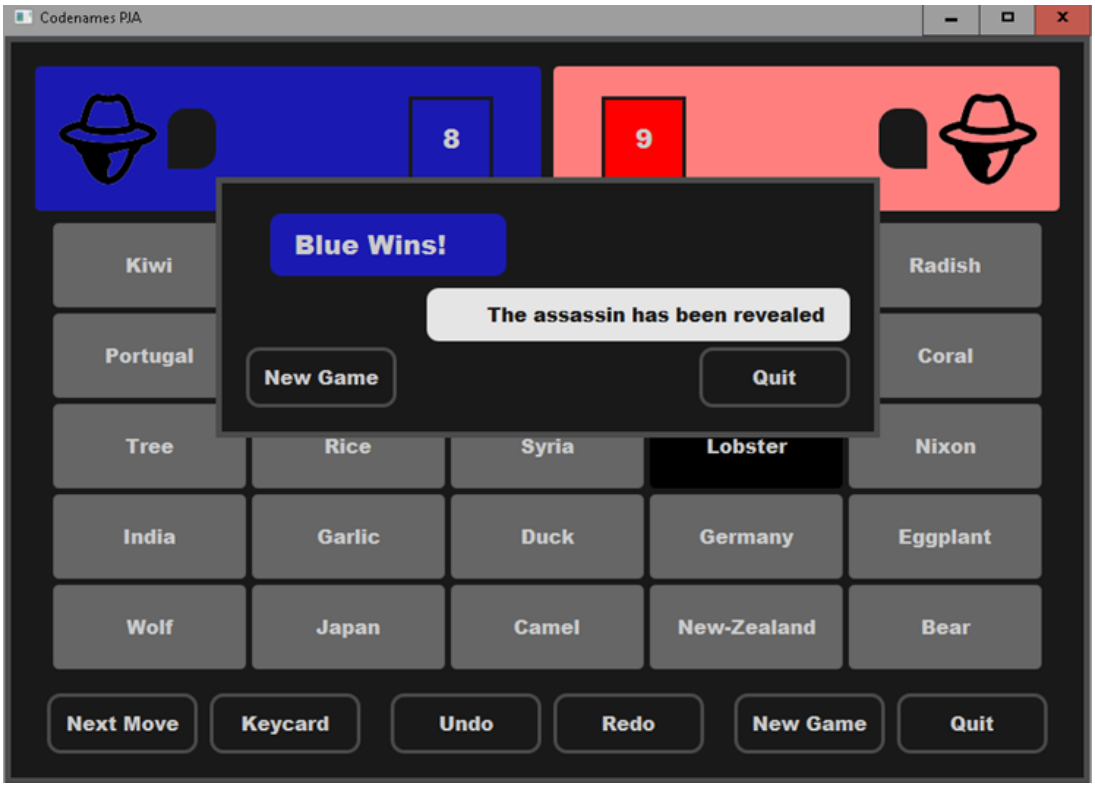

Figure 6: Red picks black card, blue wins

Test Case 2: Next Move

Figure 7: Blue picks blue card, same turn

Figure 8: Blue picks yellow card, red turn

Figure 9: Red picks black card, blue wins

Test Case 3: Undo

Figure 10: Before pressing Undo

Figure 11: After pressing Undo

Test Case 4: Redo

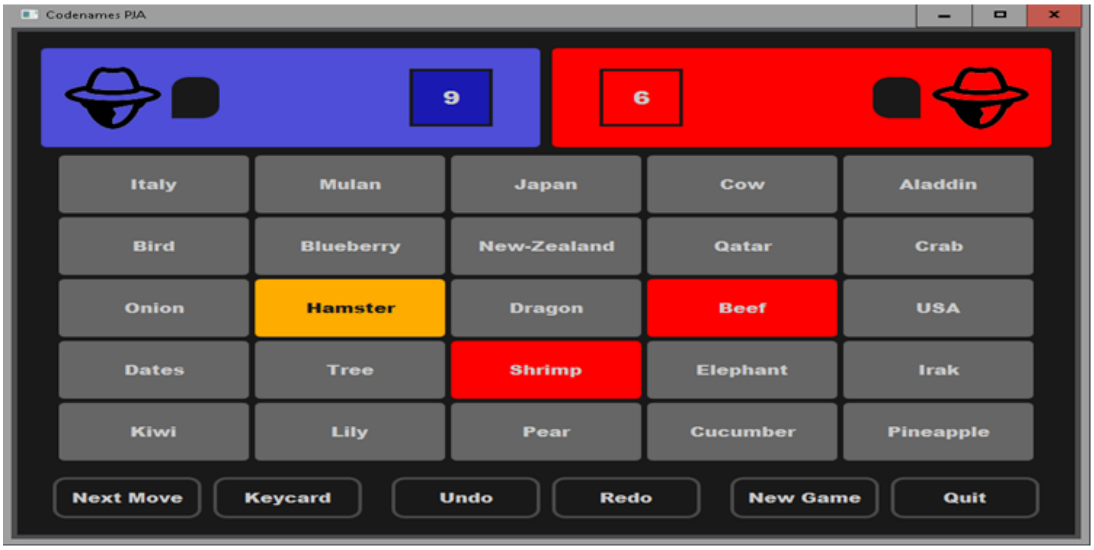

Figure 12: Before pressing Redo

Figure 13: After pressing Redo

Test Case 5: New Game

Figure 14: New Game prompt

Figure 15: After hitting comfirm

Test Case 6: KeyCard

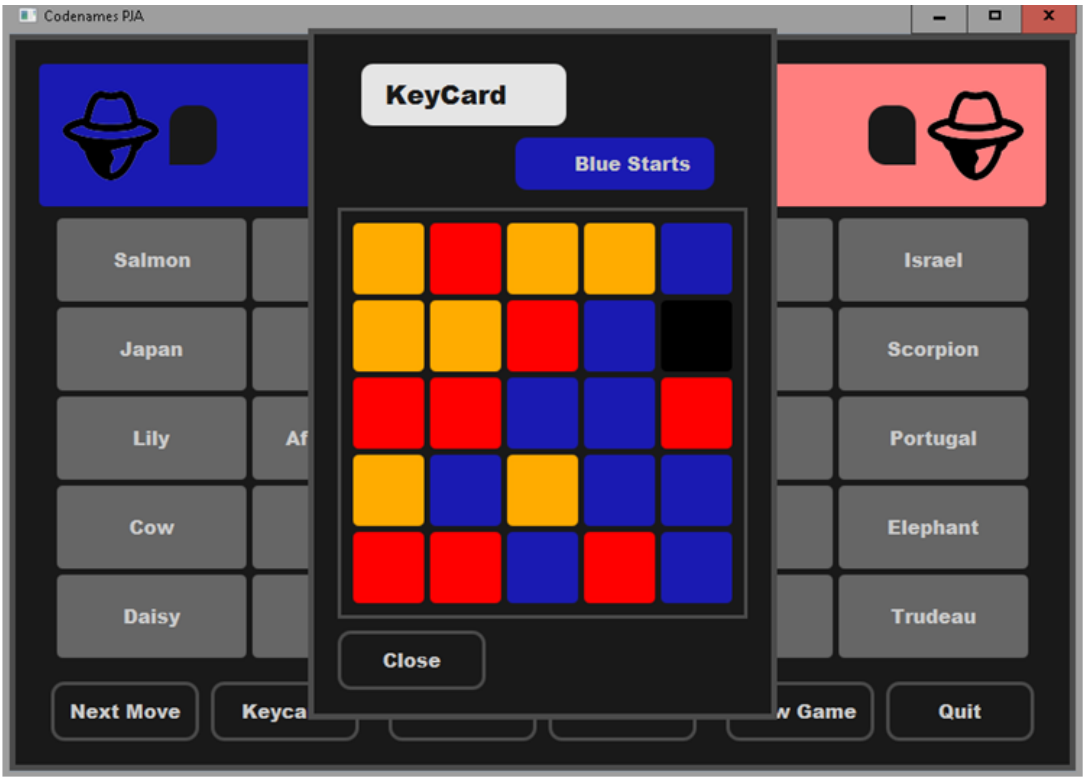

Figure 16: KeyCard popup

3.2 Subsystem Tests

Hint Subsystem:

Output:
AI hint testing...
Expected hint:2, expected word:1

Picking hint... AI picks hint: 2
Picking word... AI picks word: 1

Result: Pass

Model Creation Subsystem:

Output:
Testing Game Generation...

Card List:
Card[0]: YELLOW - Word: Radish - Revealed: false
Card[1]: BLUE - Word: Tomato - Revealed: false
Card[2]: BLACK - Word: Giraffe - Revealed: false
Card[3]: BLUE - Word: Germany - Revealed: false
Card[4]: RED - Word: Ginger - Revealed: false
Card[5]: RED - Word: Daisy - Revealed: false
Card[6]: YELLOW - Word: Calamari - Revealed: false
Card[7]: RED - Word: JFK - Revealed: false
Card[8]: BLUE - Word: Russia - Revealed: false
Card[9]: BLUE - Word: Haiti - Revealed: false
Card[10]: BLUE - Word: Bird - Revealed: false
Card[11]: BLUE - Word: Israel - Revealed: false
Card[12]: YELLOW - Word: Snake - Revealed: false
Card[13]: RED - Word: Peru - Revealed: false
Card[14]: RED - Word: Shrimp - Revealed: false
Card[15]: RED - Word: Celery - Revealed: false
Card[16]: BLUE - Word: Portugal - Revealed: false
Card[17]: YELLOW - Word: Tunisia - Revealed: false
Card[18]: YELLOW - Word: Irak - Revealed: false
Card[19]: YELLOW - Word: Mulan - Revealed: false
Card[20]: BLUE - Word: Brazil - Revealed: false
Card[21]: YELLOW - Word: India - Revealed: false

Card[22]: RED - Word: Wolf - Revealed: false
Card[23]: RED - Word: Dragon - Revealed: false
Card[24]: BLUE - Word: Potato - Revealed: false

Getting Card 15 information:
Card[15]: RED - Word: Celery - Revealed: false

Starting Turn is Blue: true
Result: Pass

Inbox Subsystem:

Output
Testing sending a message to the inbox...
Test1: Message detail: NEW$_{G}AME_{BR}ANDOM_{RR}ANDOMaffectingoncard\#1$
$Test2 : Messagedetail : NEW_{G}AME_{BR}ANDOM_{RN}EXTaffectingoncard\#2$
$Test3 : Messagedetail : NEW_{G}AME_{BN}EXT_{RR}ANDOMaffectingoncard\#3$
$Test4 : Messagedetail : NEW_{G}AME_{BN}EXT_{RN}EXTaffectingoncard\#4$
$Test5 : Messagedetail : NEXTaffectingoncard\#5$
$Test6 : Messagedetail : UNDOaffectingoncard\#6$
$Test7 : Messagedetail : REDOaffectingoncard\#7$
$Test8 : Messagedetail : SELECTaffectingoncard\#8$

Result: Pass

3.3 Unit Tests

DatabaseExtractor Unit

getTextFromDatabase(): Pass
takeAllLine(): Pass
noNull(): Pass

Inbox Unit

getMessage(): Pass

KeyCard Unit

rightNumberOfBlueCard(): Pass
rightNumberOfRedCard(): Pass
rightNumberOfBlackCard(): Pass
rightNumberOfYellowCard(): Pass

LogMessage Unit

getType(): Pass
getCardAffected(): Pass
getHint(): Pass

Message Unit

getMessageType(): Pass
getCardAffected(): Pass

Outbox Unit

getReply(): Pass
getHint(): Pass

Reply Unit

getReplyType(): Pass
getCardAffected(): Pass
getCardType(): Pass
getBlueScore(): Pass

4 Log

Beson Chan 40046280

Date	bf Description	bf hours
April 3	Caught up on the meeting Because absent	1h
April 3	Worked on the Qa testing	3h
April 6	Worked on the Qa testing	4h
April 7	converting into latex	1h
	total	9h

Kevin McAllister

Date	bf Description	bf hours
March 20	Group Meeting	1h
March 22	Coding	3h
March 27	Group meeting	2h
April 3	Group meeting	2h
April 4	bug fixing	1h
April 5	Coding	2h
April 6	testing facade	2h
April 7	Final changes	5h
	total	18

Gaoshuo Cui

Date	bf Description	bf hours
March 19	Qa	1h
March 21	list test plan	1h
April 3	Group meeting	2h
April 4	testing	2h
	total	6h

Ke Ma

Date	bf Description	bf hours
March 29	jUnit	2h
April 3	Group meeting	2h
April 6	Create strategy test	2h
April 7	build facade	4h
	total	10h

Souheil Al-awar

Date	bf Description	bf hours
March 20	Group meeting	1h
March 27	Group meeting	2h
April 4	coders meeting	4h
April 5	coding	2h
April 5	coding	2h
	total	10h

Souheil Al-awar

Date	bf Description	bf hours
March 20	Group meeting	1h
March 27	Group meeting	2h
April 4	coders meeting	4h
April 5	coding	2h
April 5	coding	2h
	total	10h

Annes Cherid

Date	bf Description	bf hours
March 27	Group meeting	2h
April 4	coders meeting	4h
April 5	coding	2h
April 6	coding	2h
April 7	doc	1h
	total	11h

Annes Cherid

Date	bf Description	bf hours
March 27	Group meeting	2h
April 4	coders meeting	4h
April 5	coding	2h
April 5	coding	2h
	total	10h

Annes Cherid

Date	bf Description	bf hours
March 20	Group meeting	1h
March 27	Group meeting	2h
April 4	coders meeting	4h
April 5	coding	2h
April 6	coding	2h
April 7	coding	4h
	total	15h

Karim Loulou

Date	bf Description	bf hours
March 20	Group meeting	1h
March 27	Group meeting	2h
April 3	Group meeting	2h
April 4	coders meeting	3h
April 5	coding	3h
April 7	Documentation	8h
	total	21h

5 References

5.1 Description of Input Files

Appendix 1: CodenamesDatabase.txt

This input file is filled with the words used inside the program and has a format:

{word: *word1* hints: *hint11, hint12, hint13, ...* }

{word: *word2* hints: *hint21, hint22, hint23, ...* }
...

The file contains 100 words where each word has at least 3 hints and each hint relates to a minimum of 2 words. Hints are not considered words, though a word can be used as a hint.

The CodenamesDatabase.txt can be found at:

COMP354PJA/client/src/CodenameDatabase.txt

Appendix 2: Hint Test Card List

Though this is hardcoded in the HintTestFacade.java file, it is required to properly test AI and Spymaster use of hints.

{word: *0* hints: *1* }
{word: *1* hints: *1, 2* }
{word: *2* hints: *1, 3* }
{word: *3* hints: *1, 2, 3* }
{word: *4* hints: *2, 4* }
{word: *5* hints: *3, 2* }

Appendix 3: Game Generation Card List

A list of cards used to test game generation

Card[0]: RED - Word: Beet - Revealed: false
Card[1]: BLUE - Word: Duck - Revealed: false
Card[2]: BLUE - Word: Germany - Revealed: false
Card[3]: RED - Word: Pig - Revealed: false
Card[4]: BLUE - Word: Tunisia - Revealed: false
Card[5]: BLUE - Word: Onion - Revealed: false
Card[6]: BLACK - Word: Apple - Revealed: false
Card[7]: BLUE - Word: Cucumber - Revealed: false
Card[8]: BLUE - Word: Irak - Revealed: false
Card[9]: YELLOW - Word: Radish - Revealed: false
Card[10]: YELLOW - Word: Russia - Revealed: false
Card[11]: BLUE - Word: Australia - Revealed: false
Card[12]: RED - Word: Rice - Revealed: false
Card[13]: RED - Word: France - Revealed: false
Card[14]: BLUE - Word: Lobster - Revealed: false
Card[15]: RED - Word: Beef - Revealed: false

Card[16]: BLUE - Word: Crab - Revealed: false
Card[17]: YELLOW - Word: Eggplant - Revealed: false
Card[18]: RED - Word: Calamari - Revealed: false
Card[19]: RED - Word: Mushroom - Revealed: false
Card[20]: YELLOW - Word: Kiwi - Revealed: false
Card[21]: YELLOW - Word: Ginger - Revealed: false
Card[22]: RED - Word: Bear - Revealed: false
Card[23]: YELLOW - Word: Cat - Revealed: false
Card[24]: YELLOW - Word: New-Zealand - Revealed: false

Appendix 4: Results for System Tests

Because of issues using latex with images, the System tests have been included as a .docx document, which can be found:

COMP354PJA/docs/SystemTest.docx