Team PB-J

CODENAMES GAME

Team Members

Name	ID Number
Ashesh Patel	40018519
Christophe Savard	40017812
Benjamin Thérien	40034572
Daniel Thibault-Shea	40073133
Shereece Victor	40105094
Michael Wilgus	29206388
Rezza-Zairan Zaharin	40003377
Steven Zanga	40000797
Mottel Zirkind	27206151

Table of Contents

Project Analysis and Development Plan

1.1 Introduction

The purpose of this document is to detail the high-level requirements and features of the Codenames Game developed by team PJ-B. The Codenames Game is a game of 2 to 8 human and AI players. This project is an adaptation of the Codenames Game designed by Vlada Chvátil and published by Czech games.

The specifics of how the Codenames Game fulfills these needs will be detailed in the use cases which more will be detailed in the upcoming design phase

1.2 Purpose

This document will describe the specifications entailed by the development of Codenames in compliance with the requirements of COMP 354. It will outline the high-level requirements encompassing user interfaces, product functions, user descriptions, assumptions and dependencies, constraints, specific requirements and an analysis model. The analysis model will hold use case diagrams, class diagrams, sequence diagrams and state transition diagrams.

1.3 Scope

This document only addresses the high level requirements of Codenames that the design phase entails. The analysis model will contain UML diagrams used to serve multiple purposes. The Use case diagrams will give an overview of the functions of the project and how users will interact with it. The class diagrams will show the interactions between different objects in the game.

1.4 Definitions and Abbreviations

1.4.1 Definitions

Board

The main playing area will be composed of a 5 by 5 grid of words. Each cell of the grid are representative of codenames of agents. Each cell in the grid will be defined in this document as a *Card*.

Card

The card will hold two values indicative of it's state: hidden or revealed. In its hidden state, it will present a word taken from the game. In its revealed state, it can either be a blue or red team's *Spy* card, a *Civilian* card or the **Assassin** card.

Assassin
The Assassin is a card type that when revealed, it would cause the team that revealed it to lose.

Spy

The Spy is a card type that belongs to either the red or blue team. Once all spy cards of a team are revealed, the team wins.

Civilian

The Civilian is a card type that occupies leftover space on the board that is not occupied by *Spy* cards or the *Assassin* card. When revealed, it only skips the revealing team's turn.

1.5 Overview

The rest of this document outlines the problem description and the development plan.

The problem description will describe the game user's interfaces, product functions, user descriptions, assumptions and dependencies, constraints, specification requirements, and the analysis model.

Problem Description

2.1 Project Purpose, Scope, and Objectives

The objective of this project is to simulate a multi-user tabletop game named Codenames game by by Vlada Chvátil and published by Czech games. The rules of the game will intimately follow Vlada Chvátil with slight variations to accommodate the digital conversion. This project will be a multi-user game of up to four players; these players can be either human or computer players.

By working on this project, Team PJ-b can experience the software engineering process entailing the difficulties in managing a group project.

The project will be worked on over 3 iterations whereby they are the requirement phase, the design phase, and the test phase.

2.1.1 User interface

As this is the requirement phase, we are keeping the interface to a minimum design. They will not preface any visual grace but present the most useful information at the time. The visual representation of the game is as shown below:

Board

The board is represented by a 5 by 5 set of cards, each possessing a word that when revealed, it will change in color to present what is the card type. When revealed, the card will be of a specific color to determine its value:

- Red is a red spy.
- Blue is a blue spy.
- Black is *the Assassin*.
- Peach is a Civilian.

Round

On the top left, the "*Round*" text represents the number of rounds that have passed in the game in accordance to the numerical value next to it.

Phase

On the top left, the "*Phase*" text details the current player in action. This value would be either one of these at a time:

- Blue Spymaster
- Blue Operative
- Red Spymaster
- Red Operative

Scorekeeping

In the middle top, Two sections with the text "*Red*" and "*Blue*" denotes the number of spies revealed for each team to the number of cards left to revealed. This is done in the format of:

"Number of spies revealed / Total Number of spies"

Guesses

On the top right, the text "*Guesses*" denotes the number of guesses left to be made by the *Operative* given to by the *Spymaster*. The value is formatted in:

" Number of Guesses Left / Maximum Number of Guesses "

Clue

On the top right, the clue given by the *Spymaster* will be displayed here.

Undo/Redo

On the bottom left, Two buttons of "*Undo*" and "*Redo*" functions as a point to click to control the turns elapsed. The *Undo* button reverses a turn so long as it is made. The *Redo* button replays a turn that was affected by *Undo*.

Next Move

On the bottom right, the "*Next Move*" button serves as a platform to click to progress the game forward by a turn.

2.2 Product Functions

Every function below has to support system functions, such as a click of a button or revealing images when necessary. The following functions will be a part of the Codenames game.

2.2.1 Introduction

The user is first introduced to the game. Since human players are currently not being implemented into the game, only the "Start Game" button is visible.

Input:

- The user clicks the "*Start Game*" button.

Action:

- 4 computer players are created: 2 for each team with each either being a *Spymaster* or *Operative*.
- 25 words are randomly picked from a database.
- The starting team is randomly picked.

Output:

- The teams are defined.
- A pool of words in use for the game are kept in an array.
- A turn order is decided.

Validity Check:

-

2.2.2 Board

To be updated.

2.2.3 Game

There will be two teams of **red** and **blue**. Each team possesses a pair of players each playing either the role of Spymaster of Operative. Since each team follows the same path:

  - Team *Spymaster* reveals clue.
  - Turn passes to Team's *Operative*
  - Team *Operative* makes guesses based on clue given.
  - Turn passes to opposing team's *Spymaster*

Input:
To be updated.

Action: To be updated.

Output:
To be updated.

Validity Check:
Sequence of the players is to be followed according to the order set earlier established. When a team finishes their turn, the next team's turn becomes active.

2.3 Constraints

Assuming that a majority of player's PCs will run by the Windows OS, the project must be written in this platform that supports GUI.Thus, the decision has been made to use :

- JAVA as the programming language.
- SQLite as chosen for data storage.
- JUnit for unit testing.
- Eclipse as the integrated development environment(IDE).

2.4 Analysis Models

2.4.1 Use Case Diagrams

The following diagarams will help provide an overview of the functions in the game. They describe the action that a player can perform, as well as the interaction between some of the system functions which are not directly controlled by the player.

These use case diagrams are included due to their importance in defining the user-to-software interactions, the requirements, and the scope of the system.

The section below will detail the actors involved in the game:

User

The User represents the human player. For this iteration there is only one User. The User is able to control the chronological flow of the game via the use cases detailed below which are "*Undo*", "*Redo*", "*Next Turn*", and "*Start Game*".

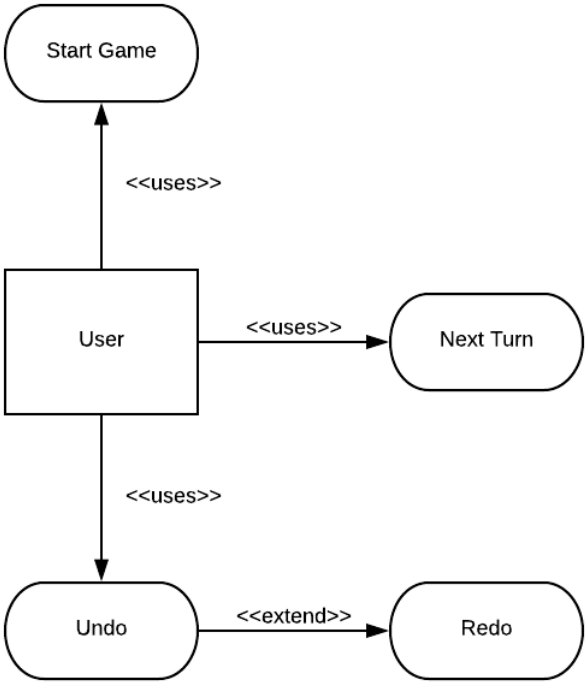
Spymaster

The Spymaster is involved in the game by doling out clues. As there are only two teams, there will only be two Spymasters in the game. The Spymaster hands out clues via the "*Reveal Clue*" use case detailed below.

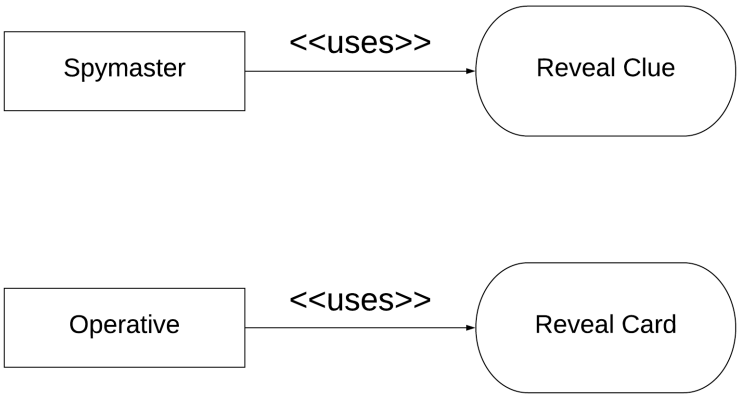
Operative

The Operative is involved in the game by revealing cards in accordance to the clue and number of guesses given to by the Spymaster actor. As there are two teams, there will be only two Operatives in the game. The operative reveals cards through the use case "*Reveal CARD*" use case as detailed below.

User Use Case



Spymaster and Operative Use Case



2.4.2 Use Cases Details

Use Case 1: Start Game

Description	The Player commences the game
Actors	User
Pre-Conditions	None
Basic Path	1. The User clicks "Start Game". 2. 4 Dummy players are generated and each are: • Deposited into a team. • Distributed a role. 3. The board is initialized. 4. The card layout is randomized. 5. The team turn order is randomized. 6. The first turn order is given to the leading team's Spymaster.
Alternative Paths	None
Post-Conditions	• The board has been initialized. • 4 Dummy players are generated with their own team and role. • The first player can play his/her turn.
Related Use Cases	
Used Use Cases	None
Extending Use Cases	None

Use Case 2: Next Turn

Description	The game moves forward a turn.
Actors	User
Pre-Conditions	• The Board is initialized. • The game has not ended.
Basic Path	1. The user clicks the button, "Next Turn". 2. The current acting player plays his/her turn. 3. The turn passes to the next acting player.
Alternative Paths	None
Post-Conditions	A player has acted on his/her turn.
Related Use Cases	
Used Use Cases	None
Extending Use Cases	None

Use Case 3: Undo

Description	The game reverses to a state before the current turn.
Actors	User
Pre-Conditions	• The user clicks the button,"Undo". • The Board is initialized. • At least a turn has been made.
Basic Path	1. The game reverses the actions done for the current turn. 2. The turn is passed to the previously acting player.
Alternative Paths	None
Post-Conditions	A player has has his/her turn redacted.
Related Use Cases	
Used Use Cases	None
Extending Use Cases	Redo

Use Case 4: Redo

Description	The game repeats action redacted by the "Undo" use case.
Actors	User
Pre-Conditions	• The user clicks the button "Redo" • The Board is initialized. • The "Undo" use case has been used at least once.
Basic Path	1. The game proceeds a turn with action undone by the use case "Undo". 2. The turn is passed to the next acting player.
Alternative Paths	None
Post-Conditions	A player has has his/her turn redone.
Related Use Cases	
Used Use Cases	None
Extending Use Cases	None

Use Case 5: Reveal Clue

Description	The Spymaster issues a clue
Actors	Spymaster
Pre-Conditions	• The Board is initialized. • The turn belongs to the Spymaster. • The team Operative has not started his/her turn. • There are still cards on the board to reveal.
Basic Path	1. The Spymaster issues a clue with a number of guesses. 2. The turn is passed to the team's Operative as detailed in the use case, "Reveal Card".
Alternative Paths	Alternative 1: 1. After step 1, if the clue not considered valid by the system (i.e. One or more words contains the clue word), the turn is passed to the Spymaster. 2. The turn is also passed for the team Operative as detailed in the use case, "End Turn"
Post-Conditions	• A clue has been revealed • The Spymaster's turn has ended • The turn is passed to the team's Operative.
Related Use Cases	
Used Use Cases	Reveal card, End Team's turn.
Extending Use Cases	None

Use Case 6: Reveal Card

Description	The Operative reveals a card
Actors	Spymaster
Pre-Conditions	• The Board is initialized. • The team's Spymaster • The turn belongs to the Operative. • The team Operative has not started his/her turn. • There are still cards on the board to reveal. • The number of guesses given by Spymaster is at least one.

Basic Path	1. The Operative chooses to either:
	  • Continue to make guesses, proceed to step 2.
	  • End his/her turn, proceed to step 4.
	2. The Operative reveals a card on the board.
	3. The number of guesses for the Operative is depleted by one.
	4. If there are still more than one guess left and there are cards to be revealed, repeat step 1.
	5. The turn is passed to the enemy Spymaster detailed in the use case, "Reveal clue".

Alternative Paths	Alternative 1: • After step 2, If card revealed is an *Enemy Spy* or *Civillian*, the turn is passed to the enemy Spymaster detailed in the use case, "Reveal clue". • If all of the enemy team's *Spy* cards are revealed, the game ends with the enemy team winning. Alternative 2: • After step 2, If card revealed is the *Assassin* card, the game ends with the enemy team winning. Alternative 3: • After step 2, If all of the Operative's *Spy* cards are revealed, the game ends with the Operative's team winning.
Post-Conditions	• A card is revealed on the board. • The turn is passed to the enemy Spy.
Related Use Cases	
Used Use Cases	Reveal Clue
Extending Use Cases	None

2.4.3 Class Diagrams

a couple letters

Full Class Diagram a couple letters

Simplified View a couple letters

Hierarchical View a couple letters