COMPUTER GAME OF UNO

ZETY MARLIA ZAINAL ABIDIN

FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2004

ABSTRACT

Computer Game of UNO is an interactive computer card game package that provides distinction capabilities from the classic card game. From a manual to digital application, still Computer Game of UNO applies the basic rule of the game. The usage of the system is suitable for all the range of age, and illiterate computer users for a starting practical. This simplicity and less time consuming game is a one-to-one system where only one user interact with minimum one or maximum three computer players.

There are several main functions such as system detecting the input from user and sound manipulation for call "UNO". The login and logout session for user to register their nickname and to exit from game also been provided to ease user. Computer Game of UNO is easily to deliver because it requires less installation process and quick response time for a display, plus the control keys using mouse most of the time make user able to play the game at ease.

Considering computer game is widely used by human from all walks of life, the sense of entertaining and challenging would not be separated in order to give satisfaction and as an exercise the skill in card game.

ACKNOWLEDGEMENT

Through the duration of the project development, many people had been kind in lending helping hands, giving invaluable and encouragement. Here I'd like to thank everyone who helped to see this project to completion.

First and foremost Alhamdulillah for His blessing, and I would like to extend my utmost gratitude to Ms Mas Idayu Md Sabri, my project supervisor who has provided me with unlimited support and guidance through the whole development stage. Not to forget I would like to extend the greatest thanks to Puan Nornazlita Hussin, the project moderator for her suggestions and comments.

Special thank to my dearest family, entire friends Hamsafar, Yuhana Ashikin, Wan Nadia, En. Nazir for the supports, advices, sharing and encouragement through the duration of the project. Besides, I am grateful to be completion of this report depends largely on the resources via the Internet, reference books and past years' thesis report.

CONTENTS

CHAPTER THREE : METHODOLOGY

CHAPTER FOUR : SYSTEM ANALYSIS

CHAPTER FIVE : SYSTEM DESIGN

CHAPTER SIX : SYSTEM DEVELOPMENT AND IMPLEMENTATION

CHAPTER 7 : SYSTEM TESTING

CHAPTER 8 : SYSTEM EVALUATION

LIST OF TABLES

LIST OF FIGURES

1.1 Introduction

Games are a fundamental part of human existence. In ancient times, human only entertained themselves by playing with visible tools but nowadays, we can only sit still and having fun play several of games and do activities with personal computer. The rapid growth of computer game increased as this artificial intelligent are famous and adorable. A 5000-year-old Mancala-like game board, carved from stone, was recently unearthed in the Sahara. The game of Go, popular in Oriental countries, has reportedly been around since 2000 B.C. Backgammon-like games such as Tabula and Nard are talked about in ancient Roman scripts, and even in the Bible. And Tarot decks, initially used to help predict the future, evolved into today's Bicycle playing cards. A decade or two ago, the only games that people spent much time with were professional sports, board games like Monopoly and Chess, paper and dice games such as Dungeons and Dragons, and card games like Poker or Hearts. Some games were for heavy money, some were bone-jarringly competitive, but most were just about good clean fun.

With the advent of computers, games entered a new era. Games became one of the main reasons many people brought these strange beige boxes called computers into their homes. Whether battling through a simple graphical tennis game such as Pong, or a rich, text-only world such as Zork, these were wholly new types of games that could be played anytime against a most formidable opponent: a game designer who had programmed a computer, long ago, showing it how to defeat user.

1.2 Project Overview

This project is designed to develop a computer game package and it is based on one of the most well-liked card game named UNO. It is set up for two players; user and a computer are required to start the game, similar with other card game players may gain or lose possession of symbols either by random processes or by matching some combination allowed by the rules of the game. Essentially, the analysis of combinations and the speed of thinking are the player's primary concern in this card game. UNO is a variant of the card game of Crazy Eights. They share the basic rules, but UNO includes some special action cards that changed the game somewhat. Each player is dealt 7 cards with the remaining ones placed face down to form a "draw" pile. The top card of the draw pile is turned over to begin a "discard" pile. The first player has to match the card in the discard pile either by number, color or word. It is one of the world's most popular family card games, with rules easy enough for kids, but challenges and excitement enough for all ages.

1.3 Project Motivation

Here are the lists of the problems that I have experienced with the classic UNO card game. These problems motivate me to create computer game version of UNO as solution to the problems.

- **Human carelessness**

  As a human being we are prone to make mistake, therefore errors such as cheating and misunderstanding can occur. Therefore, the proposed system is build to avoid the common mistakes and carelessness among players.

- **Missing card pieces**

In the classic UNO card game, small card pieces are very likely to go missing or misplaced eventually. With this proposed project these errors can be avoided as everything will be handled by the system.

- **Time consuming**

In the classis UNO card game everything is done manually. Therefore the game can be very time consuming and slow. With this proposed project, the game will be so much faster because every movement of the cards and player's turn is done by the system.

- **Availability of partner**

For this proposed system, user will have not face the problem with the availability of partner because automatically user will have partner to play with compared to the manual condition where user has to wait for other player.

1.4 Project Objective

There are several objectives to be accomplished in achieving this project. The fundamental objectives of this computer game are:

- To produce a computerized interactive card game based on UNO card game.
- To provide the understanding of UNO rules and implementation to users.
- To provide an attractive surrounding, fun and easy to be handle especially for new user.

- As an exercise and training for decision maker situation.
- To develop an eye-catching interface design.

1.5 PROJECT SCOPE

Scope of project outlines the limitation of the applications. The scope of this project are listed as below

- **User scope**

The recommended ages are from seven years old and up. This means the application is suitable for all the range of ages and background.

- **System scope for Graphical User Interface module**

➢ Development of the GUI

➢ Development of the Cards

- **Setup Module**

➢ Draw cards- each player draws a card. The cards drawn turn by turn by players.

➢ Shuffle and Dealt- shuffle the deck and each player is dealt 7 cards.

➢ Draw pile- place the remaining cards facedown to form a Draw pile. Turn over the top card of the Draw pile to begin the Discard pile. If the top card is Wild or Wild 4 return to deck and picked another card.

➢ Discard pile- match the top card on the Discard pile either by number, colour or word.

- "UNO"- before playing the next to last card, user (human player) will need to click on UNO button for calling "UNO".

1.6 Expected Outcome

Expected outcome of the system

- User friendly Graphical User Interface design suitable for every range of age
- The system will shuffle the deck and deal seven cards to each player
- The user will be able to draw card from deck and to place card at the discard pile
- The user will be able to match the top card on discard pile either by number, color and words.
- The system will be able to detect which player only left one card in hand
- The system will be able to call "UNO" for player who left only on card in hand
- The system will be able to determine the match of the cards on the discard pile with the card placed by user
- The system will determine player who finished all the card in hand first as a winner
- The system will provide the score points for user and computer player.

1.7 PROJECT SCHEDULE

The success of the project will depend critically upon efforts, cares and skills that is applied in initial planning. Thus, it is important to schedule the activities that are required to be carried out and done during the development of entire project. A Gantt chart is an easy way to schedule tasks which show each activity that are needed to be carried out during the project development. The amount of time taken to complete each activity is shown and the bars on the Gantt chart represent each task or activity that is required to be done. The project schedule is as shown in figure 1.1.

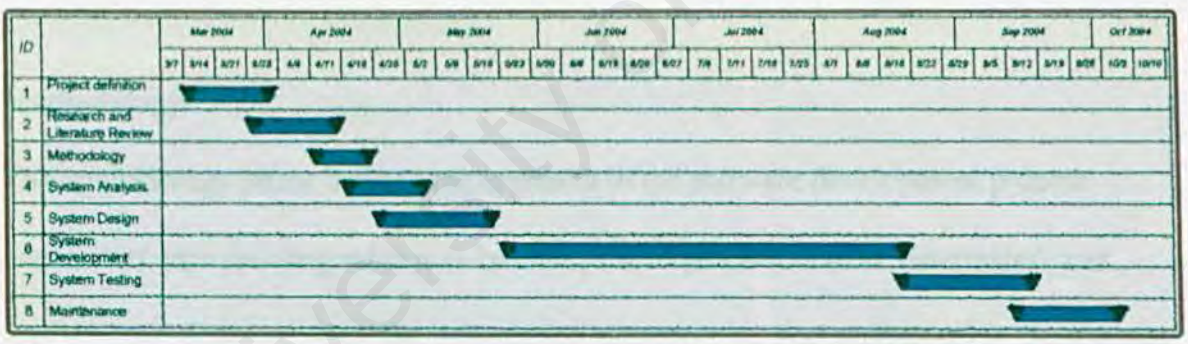


Figure 1.1 Project schedule

In developing this project, these eight phases are taken to manage and accomplish the success of the project. The purpose for each phases are discussed as below:

Project Definition

The first stage is to understand the topic of Computer Game of UNO and identify problems and opportunity. The information of requirements are been monitored and captured to meet the expectation.

Research and Literature Review

Research is done on the existing current system to compare with the proposed system and also the development software and platform. This stage is going to find out system requirement and the suitable software to use to develop the proposed system.

Methodology

The methodology phase defined the approach in the software development process which start with deciding what is to be done, the requirements to be determined, risk analysis and planning. Also, the information gathering method such as using the internet is discussed in detail with other methods.

System Analysis

Requirements of the system are captured and classified into two; functional and non-functional requirements. In this phase every needs to develop the proposed system are discussed and divided into components such as hardware and software requirements to ease the analyzing and to gain more understanding about the system's needs.

System Design

During this phase, the framework of the system is designed to meet the specification. These include the design of the graphical user interface and the specification of system architecture and work flow.

System Development

The development process involved the coding, developing and documenting software. This stage begins after the requirements are captured and the system design is ready to be implemented.

System Testing

This phase is concerned with finding errors that result from unanticipated interactions between sub-systems and sub-system interface problems. It is also concerned with validating that the system meets its functional and non-functional requirements and testing the emergent system properties.

Maintenance

This phase is to ensure that the entire application is tested and the system can work properly.

1.8 Summary

Computer Game of UNO is the computer card game package that contains interactive and easy-to-play card game for all the range of age. It can improve the weaknesses of classis card game and reduce the time consuming and provide the platform of faster understandable card game. This chapter has been clearly clarified the objectives, scope, expected outcome, schedule of the project as well as the importance of this project.

2.0 LITERATURE REVIEW

2.1 Definition and Purpose of Literature Review

Literature review is a study about the knowledge and information needed to develop this system in project. It indicates findings and on the project or thesis, summarization of the findings, analysis of the findings as well as the synthesis of the system proposed. It is also an important process in system development. (**Kendall&Kendall,1999**)

2.2 Background Studies

2.2.1 Computer Game

Computer game research was started by some of the luminaries in computing science history. In 1950, Claude Shannon published his seminal paper that laid out the framework for building high-performance game-playing programs. In 1951, Alan Turing (1953) did a hand simulation of his computer chess algorithm (a lack of resources prevented him from actually programming it); the algorithm lost to a weak human player. Around this time, Arthur Samuel began work on his famous checkers-playing program, the first program to achieve notable success against human opposition. By 1958, Alan Newell and Herb Simon had begun their investigations into chess, which eventually led to fundamental results for AI and cognitive science.

In the half century since Shannon's paper, enormous progress has been made in constructing high-performance game-playing programs. In Shannon's time, it would have seemed unlikely that within a scant 50 years checkers (8 x 8 draughts), Othello, and

Scrabble programs would exist that exceed the abilities of the best human players,(1,2) and backgammon and chess programs could play at a level comparable to the human world champion. These remarkable accomplishments are the result of a better understanding of the problems being solved, major algorithmic insights, and tremendous advances in hardware technology. The work on computer games has been one of the most successful and visible results of AI research. For some games, one could argue that the Turing test has been passed .

2.2.2 History of Card Game

Card games were co-created with playing cards themselves and may have been invented by the Chinese when they began shuffling paper money into various combinations. Though where and when cards originated is uncertain, China does seem the most likely place to have invented cards, and the 7th to the 10th century the earliest probable time playing cards appeared [1].

English playing-cards are known and used all over the world everywhere where Bridge and Poker are played. In England, the same pack is used for other games such as Whist, Cribbage, Rummy, Nap and so on. But in other European countries games such as Skat, Jass, Mus, Scopa, and Tarock are played, using cards of totally different face-designs many of them with roots far older than English cards. The history of these national and regional patterns has only recently become the concern of students and collectors.

As many travelers to more southerly parts of Europe can tell, the familiar suits of Hearts Spades Diamonds and Clubs give way to quite different sets of symbols: Hearts

Leaves Bells (round hawkbells) and Acorns in Germany; Shields 'Roses' Bells and Acorns in Switzerland; Coins Cups Swords and Clubs (cudgels) in Spain and Mediterranean Italy; Coins Cups Swords and Batons in Adriatic Italy. In the latter region, in particular, local packs of cards have a decidedly archaic look about them - which reflects the designs of some of the earliest cards made in Europe [2].

2.2.3 Other Card Game

2.2.3.1 Solitaire

1. Foundations. (Top of layout)

A. Build up in suit from Ace to King.

B. Complete these piles to win the game.

2. Tableau. (Below foundations)

A. Build down by alternate color.

B. The topmost card is available for play on the foundations, tableau or reserve Piles.

C. Any built unit of cards can be moved to another tableau pile as long as there are enough free reserves to hold them.

D. Empty spaces can be filled by any card, or legal group of cards.

3. Reserves. (Bottom of layout)

A. These are the "cells".

B. Cards in these piles can be played on the foundations and the tableau.

C. Each reserve can hold one card.

2.2.3.2 Hearts

1. Object: The main object of the game is not to pick up tricks that contain hearts in them

2. Players: Anywhere from three to seven players can play this game. Four players are the ideal number and there are no partners.

3. Cards: Use a standard deck of 52. Depending on the number of players, the following cards need to be discarded so all players can be dealt the same number of cards.

4. Deal: Choose the first dealer by low cut. Deal the whole pack, face down, one at a time in clockwise rotation. After each hand, the deal passes to the player on the dealer's left.

5. Play: The player to the left of the dealer leads first. (Variant: the opening lead is made by the player holding the 2 Clubs .) Each player after the lead must follow suit if he can. If he cannot, he may play any card he desires. Tricks are won by the highest card played of the suit led. The winner of a trick leads the next. Hearts cannot be led until hearts have been broken (a heart has been discarded on a previous trick). Obviously, if a player only has hearts in his hand, he can lead a heart even if they have not been broken. Variation: Hearts cannot be broken on the first trick. This rule takes away from strategy relating to the suit of clubs, so most players would rather avoid this variation.

6. Scoring: Once the hand is over, players get one penalty point for each heart card in the tricks they took. The winner of the game is the one with the fewest number of points when any player reaches the designated ending score.

2.2.3.3 Old Maid

1. To Play: Shuffle and deal 3 - 6 cards to each player.
2. Players take turns drawing a card from a player to their left.
3. If a player draws a card that matches one in his or her hand, he/she reads the two matching words in order to keep the pair.
4. Play continues until all the cards are matched, except for the one odd card. The player who holds that card at the end wins the game.

2.2.4 History of UNO

A man named Merle Robbins was a barber in Ohio and he was obsessed with playing cards. Every spare moment he had would be spent playing games with his wife. He was at work, happily clipping someone's hair in his hairdresser in 1971 when he had an idea. This idea was Uno. Little did he know when he was introducing the game to his family that his creation would later sweep the nation. The family together invested around £6,000 and put together 5,000 sets of the game.

These sets were sold from Robbins' shop in Ohio and from other small, local outlets. A few years later, Robbins sold the rights to the Uno brand to a fan in Illinois, also in the USA. She was, of all people, a funeral parlour director. The sales of Uno shot through the roof as a company, International Games was formed to market Uno. Then, reasonably recently, Mattel bought International Games and own the rights to Uno. They have created other games, but not as successful as the original[3].

2.2.5 The Game of UNO

Uno is a card game that involves trying to dispose of the original shuffling which you have been handed. There have since been many spin-offs of the original game, which include Uno Stacko, the Uno computer games and the original card game. The game is a relatively simple one, and although the age ranges on the pack suggest that the game is intended for children of seven plus, plus anybody over the age of six should be able to handle the easy rules relatively well. It's great to play if have got friends round, since the game allows you to play with anything from two to ten players. The best number to play with is four, as you can easily keep your eyes on what's going on and it doesn't take an hour for your turn to come round.

2.2.6 Rules of UNO

- Content of cards

1) Blue
2) Green
3) Red
4) Yellow

- The Special Action Cards

1) Skip card- The next person in line to play after this card is played loses his/her turn and is "skipped". If this card is turned up at the beginning of play, the first player is skipped. This card only is played on a matching colour or on another Skip card.
2) Reverse card- This card reverses direction of play, from left to right and vice versa. If this card turned up at the beginning of play, the player to the right is now played first, and play then goes to the right instead of left. This card only be played on matching colour or on another Reverse card.
3) Draw 2 card- When you play this card, the next person to play must draw 2 cards and forfeit his/her turn. If this card is turned up at the beginning of play, the first player must draw 2 cards. This card may only be played on matching colour or on another Draw 2 card.
4) Wild card- When you play this card you may change the colour being played to any colour to continue play. You may play a Wild card even if you have another

playable card in hand. If this card is turned up at the beginning of play, the person to left of the dealer chooses the colour to begin play and plays the first card.

5) Wild Draw 4 card- This card allows you to call next colour played and requires the next player to pick 4 cards from the Draw pile and forfeit his/her turn. You can only play this card when you don't have a card in your hand that matches the colour of the card previously played. If turned up at the beginning of play, you may return to the deck and picked another card.
6) End of game- When someone plays his last but one card, he must say "Uno". If he misses to do this, he gets a penalty of four cards from the draw pile. The player, who gets rid of all his /her cards, is declared the winner.

2.2.7 The Strategy of UNO

2.2.7.1 Manual play of UNO

1) Player to the left of dealer plays first. Play passes to the left to start.
2) Match the top card on the Discard pile either by number, color or word. For example, if the card is Green 7, you must play a green card or any color 7. Or, you may play any Wild card or a Wild Draw 4 card. If you don't have anything that matches, you must pick a card from the Draw pile. If you draw a card you can play, play it. Otherwise, play moves to the next person.

3) Before playing your next to last card, you must say "UNO". If you don't say UNO and other player catch you with just one card before the next player begins their turn you must pick four more cards from the Draw pile. If you are not caught before the next player either draws a card from the Draw pile or draws a card from their hand to play, you don't have to draw extra cards.
4) Once a player plays their last card, the hand is over.

2.2.7.2 Computer game version of UNO

1) Open file or downloaded into your computer.
2) The game is started by user login and click at Start button.
3) User will always plays first, as user need to match with the top card on the Discard pile either by number, color or word. Or user may play Wild card or a Wild Draw 4 card.
4) If user doesn't have anything that matches, user must pick a card from the Draw pile and if user draws a card that can play, just go on. Otherwise, play moves to the next computer player.
5) If user (human player) left only one card in hand he/she need to click on UNO button to declare call "UNO", otherwise system will automatically drawn him/her four cards.
6) Once player plays their last card, the hand is over and system will declare the winner.

2.3 Analysis Studies

In this section, several of existing UNO card game are being research and analysis to explore the system advantage and drawback of the system been delivered to user.

2.3.1 Case study 1 : Manual card

Uno is a variant of the card game Crazy Eights. They share the basic rules, but Uno includes some special action cards that change the game somewhat. Shuffle the cards and deal seven to each player. The rest of the cards are placed on the table, this is the draw pile. Turn over the top card and place it on the board, this is the first card of the discard pile. The active player must play a card on the discard pile. It must match the top card on either suit or number. For example, if the top card is yellow 3, he can play any yellow card or any card numbered 3, or he may play a Wild card. If he has no card to play, he must draw the top card from the draw pile. If this card matches then he may play. Then the next players turn.

- **Strengths** :

1) Players may recognize who among them are cheaters and then can probably try new methods and various ways to play.
2) The game would be more interactive because players communicate with one another as they can manage strategies and manipulate other players

- **Weaknesses** :

1) Missing card pieces or misplaced because UNO card is small and light.
2) As everything done manually and this caused time consuming so the game may be slower and players will get bored easily.
3) The cards will easily scratch or damage because it keeps on passing to players hands.

2.3.2 Case study 2 : Handheld UNO

Handheld of UNO use the same concept with the classic method, but this small electronic machine provide some features that somewhat different from the classic card such as the card design and complete the users needs even opponent. It requires three AA batteries and the scores are calculated in points.

- **Strengths** :

1) This game is ultimately playable anywhere and anytime.
2) The game will be clear from any cheating situation compared to manual method because player is challenging a machine
3) Player will not face the card damage because this is digitally used.

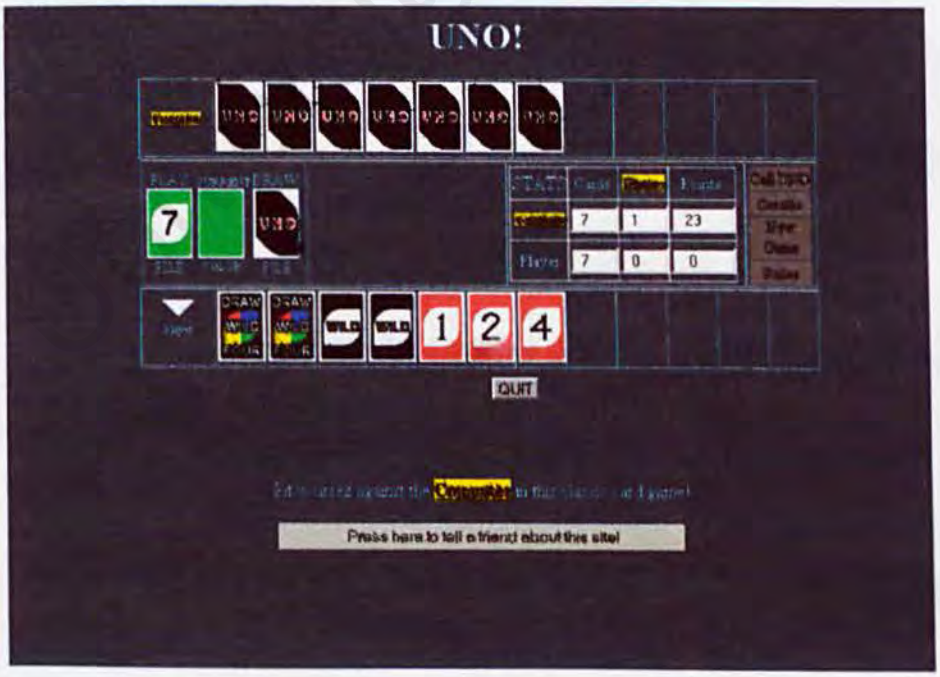
- **Weaknesses :**

1) It can't be played in sunlight because the LCD will flicker. The LCD works fine in room lighting and well filtered sunlight but outside on a sunny day, the LCD will need to be held at an angle.

2) The voice are repetitive. Sound down does work although it works best on a used set of batteries. It should not be played without sound because the visual act strong enough for play in this mode.

2.3.3 Case study 3: Internet version of UNO

http://www.referencetable.net/game/uno/unoA



This UNO online game is one of most popular card game because this addictive and hands-on game is fast moving and everything done from shuffling to showing scores by the system. Human player will challenge the computer one to one as the computer will show its skill and strength to defeat human. Users may download this game everywhere and can call tell to another user about this game site.

- **Strength :**

1) Super fast game-play with same time turn mode. No more waiting so the game much faster than before.
2) All are new cards and controlled by the system.
3) Have fun feature for everyone including bright 2D graphic.
4) Can be played against friends, with challenging computer opponent or over the internet.

- **Weaknesses :**

1) The computer player draws the card and action card too fast until human player unable to look at those placed cards.
2) There is no sound as to call "UNO", just a simple click button and if player forget click at the button player have to pick another card from the deck.

2.3.4 Proposed System

The proposed system is a stand alone that requires only one user and one computer player. The system implies the basic rules of UNO but for call "UNO" user (human player) need to click UNO button.

- **Strength :**

1) It is easy to download and uncomplicated to open the file
2) Make ease the user to call "UNO" as the system provide UNO button to call "UNO".
3) As the starting point for children and new computer user to make use of personal computer because the system provides simple instruction and understandable input and output.

- **Limitations**

1) The proposed system only provides the score points for winner.
2) It is non-network system so user only manages to play with computer.
3) The proposed system is set up only for one human player and one computer player.

2.4 Development Software

2.4.1 Macromedia Flash MX (**Macromedia Flash MX,2002**)

Macromedia Flash is the standard for producing high impact web experiences. It enables authors to build attractive and effective web interface and applications, beautiful graphic and engaging narrative animation. It has a timeline that can contain multiple layers. The timeline metaphor excels at letting an author manipulate and synchronize media elements over time. Interactivity, branching and logic can be added by using the macromedia action script language. Besides that, flash also grown as a product for creating e-learning application.

Why Macromedia Flash so widespread:

1. The high impact vector file format produced by it. Vector-based files tend to be much smaller than files that store information pixel-by-pixel such as bitmaps.
2. The flash file format produces beautiful graphics and animation that have additional advantage of being resolution-independent.
3. Not only support text, graphics and sophisticated animation but also can include MP3 audio.
4. Creating graphics and animation that are valuable for illustrating facts, concepts and procedure. The advanced features such as gradients, alpha channel, and media let developers create that is beautiful and captivating.

5. Action Script in flash is marvelous language that can communicate with the flash movie.

2.4.2 Macromedia Director 8.5 (**Macromedia Director 8.5 Shockwave Studio,2002**)

Macromedia Director Shockwave Studio is the solution for developing engaging internet destination and powerful multimedia. The director interface is a timeline that can contain up to 1000 tracks. Media element are added to the timeline and drag-and-drop behaviors and Lingo scripting are used to add interactivity and logic. Media assets and code are organized and maintained in groups called casts.

Why Macromedia Director so widespread :

1. Director excels in media handling and synchronization, critical for learning application with complex simulations.
2. Support industry-standard format of all types of media, including text, graphics, animation, AIFF, WAV, SND and MP3 format audio files and video.
3. The only macromedia product that support true 3D graphics
4. The scripting language called Lingo, which gives developers robust programming capabilities and extensive control over media. Lingo programmers can develop behaviours that provides reusable code and simplify many authoring task
5. Library palette in director is a very useful function for non-programmer background developer to add the behaviours to the objects.

2.4.3 Adobe Photoshop

Adobe Photoshop is a state of the art tool with its comprehensive set of retouching, painting, drawing and web tools. Photoshop complete any image-editing task efficiently. Besides, Photoshop is a technical tool, for advanced user, it can almost create any texture, any 2D image and can finely retouch on a photo. As Photoshop become more robust and rich features in every latest released, which makes it the best image editing software in market.

2.4.4 Sound Forge 6.0 **(Sound Forge 6.0,2002)**

Sound Forge is well popular as powerful audio editor with the features such as powerful audio process, real-time audio effects, and manipulating audio.
Here are some features of Sound Forge:

1. Support multiple file format-15 import file format and 17 export file format, such as WAV,MPEG and MP3
2. Sophisticated background rendering
3. Over real-time audio effects-include acoustic mirror, amplitude modulation, chorus, vibrato and reverb.

2.4.5 Adobe Illustrator 10 **(Adobe Illustrator,2002)**

Adobe Illustrator is another Adobe product. It is industry-standard vector graphic software. Here are some feature and the advantages of the software:

1. Balance image quality and file size
2. Can get an accurate on-screen view before going further
3. Enable colors effect by the tolls such as natural-looking transparency, clipping mask and opacity mask, solid, patterns and gradient tools
4. Integrated with other Adobe product such as layer mask, transparency and compound shapes preserved in Adobe Photoshop

2.5 Programming Language

2.5.1 Visual Basic

Visual Basic is the fourth generation language for designing and building application with graphical user interface (GUI) [4]. Visual basic is an extremely powerful, full-featured application development tool that exploit in window-based system. It is easy to use and can be used to create several application such as Client application, ASP web-based application, Database interface application, and Server side application. Visual Basic hides away some complicated part of traditional programming to programmer. Programmer need only to plan program's logic and the design of the interface and needless to worry about the memory allocation, initialization and building of interface component and window-based system.

Visual Basic provides scripting language call VB script to help develop interactive and dynamic web pages. VB scripting programming is a light weight scripting language that provides programming functionality based on the Visual Basic programming. Window-based applications are event-driven and they need good graphical user interface (GUI) support which makes window-based application well supported in Visual Basic. Event driven means the code remain idle until an event is called upon respond to some event. Visual Basic has strong influence in other development software where certain element has been "borrowed". For example, Visual Basic programmer will find action script in Flash is easy to catch up.

2.5.2 C ++

C++ is a superset of C developed by Bjourne Stroustrup at Bell Laboratories. C++ provides a number of features that spruce up the C language. More important it provides capabilities for Object Oriented Programming. Objects are essentially reusable software component that model real item inn real world [5]. Object Oriented Programming is one of the methods which are able to allow a programmer to define a group of variables and functions, which known as class. The object concept allow programmer to reuse this set of variables and functions in any part of the program. Besides classes, C++ does have features that C and VB do not really have, example C++ templates, operator overloading, inheritance, virtual function and polymorphism. With this features, it allow a C++ programmer more degree of freedom in programming. C++ is recognized as one of the technical programming language in the sense of command and machine operation awareness to a programmer. In another words, a C++ programmer need to understand more clearly on the structure and flow of an operation in order to write the coding.

C++ is not user friendly as VB in window-based system. A C++ programmer requires to write from memory allocation to window initialization in order to write a simple "Hello World" program in a window-based system where it take a few line of code in VB.

2.5.3 Java

Java is a simple, object-oriented, network-savvy, interpreted, robust, secure and multithreaded dynamic language. Java omits many rarely used, poor understood features of C++. These omitted features primarily consist of operator overloading (although Java language does not have method overloading). To run Java software, a computer needs an operating system containing Java Virtual Machine (JVM).

Java also supports multimedia with the use of sound, images, graphics, and video. Java capabilities also include drawing two-dimensional shapes with the use of Java 2D API classes, drawing three-dimensional shapes with Java 3D classes and also support graphics. Applets are Java programs that can be embedded in Hypertext Markup Language (HTML) documents. When a browser loads a web page containing an applet, the applet downloads into the web browser and begins execution. Java allows multiple concurrent threads of execution to be active at once. The Java.lang package provides a thread class that support method to start, run and stop a thread and check on its status[6].

2.6 Other Game Development Tools

Below is a table showing some of the option that can create the game. Normally a looked more challenge tool is able apply programming language such as C/C++

inside the game. However, the more looked realistic game that in the market now use the software that can support the 2D/3D graphics.

Table 2.1 Game Development Tools

Tools	**Language**	**Pros**	**Cons**
Adventure Game Studio	None	Create point-and-Click adventure game	User can't create arcade/platform style games nor straight RPG's
BCC 5.5	C/C++	Its a freeware 32bit compiler to create DOS or windows game. Can be used for both DirectX and OpenGl	Its harder to get help if user are stuck using Borland. User have to register with Borland using user name and address to download
VB 6	Visual basic	Easy to learn and fast development time	No pointers and Object Oriented Programming not as fully implemented in C++.Slower than

			C++ in terms of runtime.

2.7 System Development Tools Analysis

Appropriate system development tools must be chosen to reduce the project of the risk of being over budget or system failure plus delivered late. Based on the literature review that has been conducted previously, an analysis was done on the tools in order to choose the best tools to develop the game system.

2.7.1 Adobe Photoshop 6.0

Adobe Photoshop is among the top software used for images or any graphics editing. With the used of drawing concepts by vector and layers, those finest images can be produced referring to the creativity and idea of development. Adobe Photoshop is well planned to be used for the purpose of graphic image editing that will be used during the system development such as to design the UNO cards, buttons and other images. The special features of Adobe Photoshop are listed as below:

- **Layers**

Using the layers method the images that developer want to produce will be easier and more effective to be processed in modification and editing process.

- **Supporting the Used of Vector**

It is to ease the process of producing vector shapes such as square, circle, ellipse and polygon shapes that can be modified all over again.

- **Supporting the Text Vector**

Text that been produced can be integrated together with the images that will be created.

- **Other Special Effects**

Some of the special effects that guaranteed the attractiveness that can be formed such as shadow effects, visible effects in 3D, light effects and brightness.

2.7.2 Java

Java is an Object Oriented Programming language that uses may common elements from other OOP language, such as C++, but is added some enhancement to make programming Java easier. Like any other language, Java has particular syntax, a structure of programs, and many supporting applications. The Java Developer's Kit (JDK) contains of the tools necessary to create applications (or web applets) using the Java programming language.

Simple

The simplicity of Java is enhanced by its similarity to C and C++ Java has simplified C++ programming by both adding features beyond those found in C++ and by removing some of the features that make C++ a completed and difficult language to master. Java is simple because it left out of the unnecessary feature of other high-level language programming language.

Object Oriented

In its approach to object-oriented, Java follows more closely along the lines of language than C++. Just like C++, Java uses classes to organize code into logical modules. At runtime, a program creates objects from classes. Java classes can inherit from other

classes. Experts for its primitive data types, everything in Java are an object. Java's support for object-oriented does not include multiple inheritance[7].

Multi-threaded

A multi threaded program is on that is written such that it performs more than one task at a time. Java programs can contain multiple threads of exclusion, which enables program to handle several tasks concurrently. An application has at least one thread, which represents the program's main path of execution.

Secure

Closely related to Java's robustness is its flow on security. Because Java does not use pointers to directly access memory locations, as is prevalent in C and C++. Therefore, programs cannot gain access to areas of the system for which they have no authorization. Thus, Java has a great deal of control over the code that exist within the Java environment.

Platform independence

Platform independence is another way of saying that Java is architecture neutral. Platform independence is the capability of the same program to work on different

operating system, Java is completely independent. This means that Java do not care what system they are running on.

2.7.3 Sound Forge

Sound Forge can be used to record voices and sounds. With the available sound card that able to record the sound and one microphone, the sound can be recorded and stored in the computer with the specific format. Thus, Sound Forge able to support multiple format -15 import files and 17 export file formats such as WAV, MIDI and MP3. With the features such as audio process and manipulating audio, Sound Forge is one of the most convenience and easy software to use.

2.7.4 Macromedia Flash

A Flash movies is an animation and interactive graphic vector. Flash can be used to create navigation control, animated symbols, animation with timeframe and complete with synchronized sounds. Flash movies is full with small vector graphics that can be downloaded quickly and consistence scaled according to user's screen size. The purposes usages of this software are to save the entire storage areas and to produce quick download time whenever the user asks for helped instantly.

2.8 Summary

In this chapter, all the requirements for the system are captured and map out. These requirements will provide a strong foundation to the following chapter. Software is justified not to limit the requirements but to know the limitation that will pose by it.

3.1 Introduction

He who does not lay his foundation beforehand may be great abilities do so afterwards, although with great trouble to the architect and danger to the building.

-Niccolo Machiavelli

The Prince

The essence of the software development process that consists of analysis, design, implementation, testing and refinement is to transform users' needs into a software solution that satisfies those needs. However, some people view the software development process as interesting but feel it has little importance in developing software. It is tempting to ignore the process and plunge into the implementation and programming phases of software development, much like the builder who would bypass the architect.

3.1.1 Methodology - The Spiral Model

The Spiral Model was designed to include the best features from the Waterfall and Prototyping Models, and introduces a new component - risk-assessment. The term "spiral" is used to describe the process that is followed as the development of the system takes place. Similar to the Prototyping Model, an initial version of the system is developed, and then repetitively modified based on input received from customer evaluations. Unlike the Prototyping Model, however, the development of each version of

the system is carefully designed using the steps involved in the Waterfall Model. With each iteration around the spiral (beginning at the center and working outward), progressively more complete versions of the system are built. [8]

Risk assessment is included as a step in the development process as a means of evaluating each version of the system to determine whether or not development should continue. If the customer decides that any identified risks are too great, the project may be halted. For example, if a substantial increase in cost or project completion time is identified during one phase of risk assessment, the customer or the developer may decide that it does not make sense to continue with the project, since the increased cost or lengthened timeframe may make continuation of the project impractical or unfeasible.

The Spiral Model is made up of the following steps:

- **Project Objectives**. Similar to the system conception phase of the Waterfall Model. Objectives are determined, possible obstacles are identified and alternative approaches are weighed.
- **Risk Assessment**. Possible alternatives are examined by the developer, and associated risks/problems are identified. Resolutions of the risks are evaluated and weighed in the consideration of project continuation. Sometimes prototyping is used to clarify needs.
- **Engineering & Production**. Detailed requirements are determined and the software piece is developed.

- **Planning and Management**. The customer is given an opportunity to analyze the results of the version created in the Engineering step and to offer feedback to the developer.

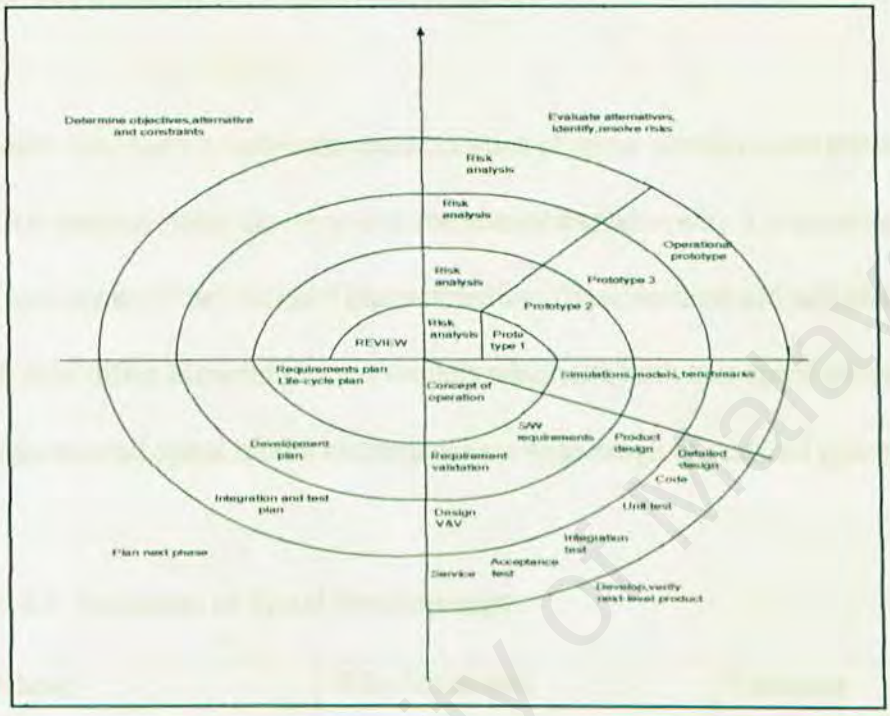


Figure 3.1 Spiral Model

3.1.2 Problems/Challenges associated with the Spiral Model

Due to the relative newness of the Spiral Model, it is difficult to assess its strengths and weaknesses. However, the risk assessment component of the Spiral Model provides both developers and customers with a measuring tool that earlier Process Models do not have. The measurement of risk is a feature that occurs everyday in real-

life situations, but not as often in the system development industry. The practical nature of this tool helps to make the Spiral Model a more realistic Process Model than some of its predecessors.[9]

3.1.2.1 The Invariants of Spiral Development

This table lists the six invariant characteristics of spiral development process models. The first column states the invariant; the second explains why it is invariant; and the third lists some of the "variant" characteristics. These variants are sub-characteristics which may differ between process models which are both true spiral process models. Thus the overall spiral model encompasses a wide range of realized processes.

Table 3.1 Invariants of Spiral Development

Invariant	**Why invariant**	**Variants**
1. Concurrent rather than sequential determination of key artifacts--Ops Concept, Requirements, Plans, Design, Code--in each spiral cycle	-Avoids premature sequential commitments	1a. Relative amount of each artifact developed in each cycle 1b. Number of concurrent mini-cycles in each cycle
2. Each cycle considers critical stakeholder	-Avoids commitment to alternatives that are	2a. Choice of risk resolution techniques:

objectives and constraints, product and process alternatives, risk identification and resolution, stakeholder review, and commitment to proceed	risky or unacceptable to stakeholders -Avoids wasting effort on unusable alternatives	prototyping, simulation, modeling, benchmarking, reference checking, etc. 2b. Level of effort on each activity within each cycle
3. Level of effort on each activity within each cycle driven by risk considerations	- Avoids too little or too much of each activity - Avoids overkill or belated risk resolution	3a. Choice of methods used to pursue activities: MBASE/WinWin, Rational RUP, JAD, QFD, ESP, . . . 3b. Degree of detail of artifacts produced in each cycle
4. Degree of detail of artifacts produced in each cycle driven by risk considerations	- Avoids too little or too much of each artifact - Avoids overkill or belated risk resolution	4a. Choice of artifact representations (SA/SD, UML, MBASE, formal specs, programming languages, etc.)
5. Managing stakeholder life cycle commitments via the LCO, LCA, and IOC Anchor Point Milestones	-Avoids analysis paralysis, unrealistic expectations, requirements creep,	5a. Number of spiral cycles or increments between anchor points 5b. Situation-specific

	architectural drift, COTS shortfalls or incompatibilities, unsustainable architectures, traumatic cutovers, useless systems	merging of anchor point milestones
6. Emphasis on system and life cycle activities and artifacts rather than software and initial development activities and artifacts	-Avoids premature suboptimization on hardware, software, or development considerations	6a. Relative amount of hardware and software determined in each cycle 6b. Relative amount of capability in each life cycle increment 6c. Degree of productization (alpha, beta, shrink-wrap, etc.) of each life cycle increment

3.2 Justification

The Spiral Model approach that will be adapted in the proposed project encompasses the activities in determined objectives, alternatives and constraint, evaluate alternatives, identify, resolve risks, develop, verify next-level product, and plan next phase. Each of loop in the spiral is split into four sectors :.

a) Objectives setting

Specific objectives for that phase of the project are defined. Constraints on the process and the product are identified and a detailed management plan is drawn up. Project risks are identified. Alternative strategies, depending on these risks, may be planned.

b) Risk assessment and reduction

For each of the identified project risks, a detailed analysis is carried out. Steps are taken to reduce the risk. For example, if there is a risk that the requirements are inappropriate, a prototype system may be developed.

c) Development and validation

After risk evaluation, a development model for the system is then chosen. For example, if user interface risks are dominant, an appropriate development model might be evolutionary prototyping. If safety risks are the main consideration, development based on formal transformation may be the most appropriate and so on.

d) Planning

The project is reviewed and a decision made whether to continue with a further loop of the spiral. If it is decided to continue, plans are drawn up for the next phase of the project.

3.3 Information Gathering Method

For information gathering method, there is no underlying standard or procedure to be followed strictly as each single project is unique and data gathering may be vary suit the needs of each particular project. However, there are a certain number of methods that are commonly used in gathering information such as collecting hard data like written documents or reports, interviewing using questionnaire, observation and sampling.

As for this project, due to cost and tight schedule constraints as well as difficulties in finding and getting domains experts whom are willing to help, method such as interviewing becomes intermediary who obtained the user's requirement from the real estate company. The main data sources for system analysis were written documents, reference books, observation and other source from Internet.

3.3.1 Written Document and Reference Books

For the purpose to gather information about user's need, system requirement and also technical requirement I have done some through printed documents such as books, magazine and journal. I have to go to library to find some books, magazines and journals that related to computer card game to learn about the

background and function that can help me to understand more about the system proposed. Furthermore, I also gained more information and updated scenario about today computer technology from Pc magazine and from newspaper such as In-Tech from the Star.

3.3.2 Internet Research

Other than written documents gathering method, one of the likely research was also done via the Internet. The result from the research are some elaborated in chapter 2 which comparison have been made among other system that are similar with the proposed system. Besides, Internet is used as main resource for referring to any ambiguities that might arise during the entire development period. I get a lot of software, tools, programming languages, game development tools and computing knowledge from these researches as it provides me with various help file so I can learn more about the software and programming tools that will be applied in the proposed system.

3.3.3 Brainstorm

During the requirement elicitation, I try to generate as many ideas as possible without any analysis until all the idea have been exhausted. Plus, I study the feasibility of the requirement identified in this stage.

3.4 Summary

The methodology used in the development process is the Spiral Model that also known as the risk-assessment component that produced by the combination of Waterfall and Prototyping Models. The Spiral Model implies the similarity in Prototyping Model but the development of each version designed using the steps in Waterfall Model. Other information gathered are by Internet research, written document and brainstorming to expand information findings of the proposed system.

4.1 Requirement Definition

Requirement of the system is the description of the features or functions that the system should provide as well as the constraints of the system. System requirements are generally classified into two; functional requirement and non-functional requirement.

4.1.1 Functional Requirement

The functional requirement describe the functionality in services that the system is expected to provide. It also refers as features that the project should have function.

User Mode

User mode is a module that system will be able to detect user (human player) to start the game.

System Detection

System detection is a module where system will be able to handle the algorithm of the game play. User will click at the deck and system begins to identify to match the card according to the same number first, then according to the color or else the same words. If the card doesn't have the similarity of number, color or words user need to pick another card from the deck. At the end of the game system also able to identify the winner who has finished the entire card at first.

Mouse Clicked

This is a module on input where system is able to detect input from user by clicking the button provided.

Object Manipulation

Object here refers to cards, and object manipulation is a module where system is used to manipulate objects in the system. The cards will be able to shuffle, draw and deal as the programmer favors.

Action Card

Action Card module is a module where system able to run the special functionality of the Action Cards during the game play. For example, system will able to detect the action to be taken if players placed or draw one of the Action cards.

Score Calculation

This module defined the system will able to show the score table for user if they win the game and the scores.

4.1.2 Non-Functional Requirement

A non-functional requirement is a description of the features, characteristics and attributes of the system as well as any constraints that may limit the boundaries of the proposed solution [10]. The non-functional requirements are listed as below:

Simplicity and User Friendly

The game is meant to develop for it simplicity, in the sense of the game play where player can simply understand how the game is running. User friendly means for it control keys using mouse most of the time to click to be well organized and able to play the game at ease.

Faster and Smoother Game Play

Faster and smoother means that the game can be able to play smoothly at the minimum requirement specification and the response time between user input and display are fast. No lagging on a proper system.

Entertaining and Challenging

The game must have some sense of entertainment and challenging in order to bring interest and excitement to the user. The game must be able to give player sense of satisfaction for the time spent on playing the game.

Easily to Implement

The game is easily to deliver where less installation process required. The game will be able to execute easily without install or register extra data to the operating system.

4.2 Hardware and Software Requirement

Hardware and software requirement describe constraints on computer and peripheral equipment, their capacity and time available for use, and development support software. Hardware and software requirement need to be decided to determine the feasibility of performance requirements. Both of them are divided into developers side and user side requirements.

4.2.1 User Side Requirements

Table 4.1 User Side Requirements

Hardware Requirements	Software Requirements
• Pentium 233 MHz and higher • Random Access Memory : 32 MB and above • 16 MB Graphic Card • Standard input and output • Other standard computer peripherals	• Windows 98/Me/XP operating system • Any operating with graphic user interface

4.2.1 User Side Requirements

Table 4.1 User Side Requirements

Hardware Requirements	Software Requirements
• Pentium 233 MHz and higher • Random Access Memory : 32 MB and above • 16 MB Graphic Card • Standard input and output • Other standard computer peripherals	• Windows 98/Me/XP operating system • Any operating with graphic user interface

4.2.2 Developer Side Requirements

Table 4.2 Developer Side Requirements

Hardware Requirements	Software Requirements
• Pentium or AMD with 500 MHz and above • At least 64 MB RAM • Other standard computer peripherals • Screen solution 800x600 pixel and High Color (16 bit) • Plug in for Macromedia software	• Windows 98, 2000, Me, XP operating system • JCreator LE • Adobe Photoshop 6.0 • Java 2 SDK

4.3 System Development Tools Analysis

Appropriate system development tools must be chosen to reduce the project of the risk of being over budget or system failure plus delivered late. Based on the literature review that has been conducted previously, an analysis was done on the tools in order to choose the best tools to develop the game system.

4.3.1 Adobe Photoshop 6.0

Adobe Photoshop is among the top software used for images or any graphics editing. With the used of drawing concepts by vector and layers, those finest images can be produced referring to the creativity and idea of development. Adobe Photoshop is well planned to be used for the purpose of graphic image editing that will be used during the system development such as to design the UNO cards, buttons and other images. The special features of Adobe Photoshop are listed as below:

- **Layers**

  Using the layers method the images that developer want to produce will be easier and more effective to be processed in modification and editing process.

- **Supporting the Used of Vector**

  It is to ease the process of producing vector shapes such as square, circle, ellipse and polygon shapes that can be modified all over again.

- **Supporting the Text Vector**

  Text that been produced can be integrated together with the images that will be created.

- **Other Special Effects**

  Some of the special effects that guaranteed the attractiveness that can be formed such as shadow effects, visible effects in 3D, light effects and brightness.

4.3.2 Java

Java is an Object Oriented Programming language that uses may common elements from other OOP language, such as C++, but is added some enhancement to make programming Java easier. Like any other language, Java has particular syntax, a structure of programs, and many supporting applications. The Java Developer's Kit (JDK) contains of the tools necessary to create applications (or web applets) using the Java programming language.

Simple

The simplicity of Java is enhanced by its similarity to C and C++ Java has simplified C++ programming by both adding features beyond those found in C++ and by removing some of the features that make C++ a completed and difficult language to master. Java is simple because it left out of the unnecessary feature of other high-level language programming language.

Object Oriented

In its approach to object-oriented, Java follows more closely along the lines of language than C++. Just like C++, Java uses classes to organize code into logical modules. At runtime, a program creates objects from classes. Java classes can inherit from other

classes. Experts for its primitive data types, everything in Java are an object. Java's support for object-oriented does not include multiple inheritance.

Multi-threaded

A multi threaded program is on that is written such that it performs more than one task at a time. Java programs can contain multiple threads of exclusion, which enables program to handle several tasks concurrently. An application has at least one thread, which represents the program's main path of execution.

Secure

Closely related to Java's robustness is its flow on security. Because Java does not use pointers to directly access memory locations, as is prevalent in C and C++. Therefore, programs cannot gain access to areas of the system for which they have no authorization. Thus, Java has a great deal of control over the code that exist within the Java environment.

Platform independence

Platform independence is another way of saying that Java is architecture neutral. Platform independence is the capability of the same program to work on different operating system, Java is completely independent. This means that Java do not care what system they are running on.

4.4 Summary

In this chapter, all the requirements for the system are captured and map out. These requirements will provide a strong foundation to the following chapter. Software is justified not to limit the requirements but to know the limitation that will pose by it.

5.1 System Design

Design is the creative process of transforming the problem into solution; the description of a solution is also called design. Design is a two part iterative process. First, we produce a conceptual design or system design that tells user exactly what the system will do. Once the customer approves the conceptual design, we translate the conceptual design into a much more detailed document, the technical design that allows system builders to understand the actual hardware and software needed to solve the problem. The process is iterative because in actually, the requirements, Proposing possibilities to the users and documenting the design for the programmers. (**Pfleeger,2001**)

In the system design phase, information collected earlier will be used to accomplish the logical design for the system. The accurate data-entry procedures are designed so that data going into the system are correct. In addition, effective input is provided to the system by using techniques of good form and screen design.(**Kendall & Kendall,1999**)

5.2 System Functionality Design

The arrangement of the structure chart shows tells us the relation between modules in this project. It is used to show the activities that make up the system. It gives the meaning that the requirements into system functionality can be used. The main thing of this design is it focuses on the system architectural design and user interface design.

Structure Charts using System Structuring and Modular Decomposition

Structure chart are used for procedural programs to illustrate the following information about a program in a visual format:

- Partitioning of a program into named modules (functions)
- Top-down hierarchy and organization of modules
- Linkage between modules

The figure below shows the main system structure chart of Computer Game of UNO

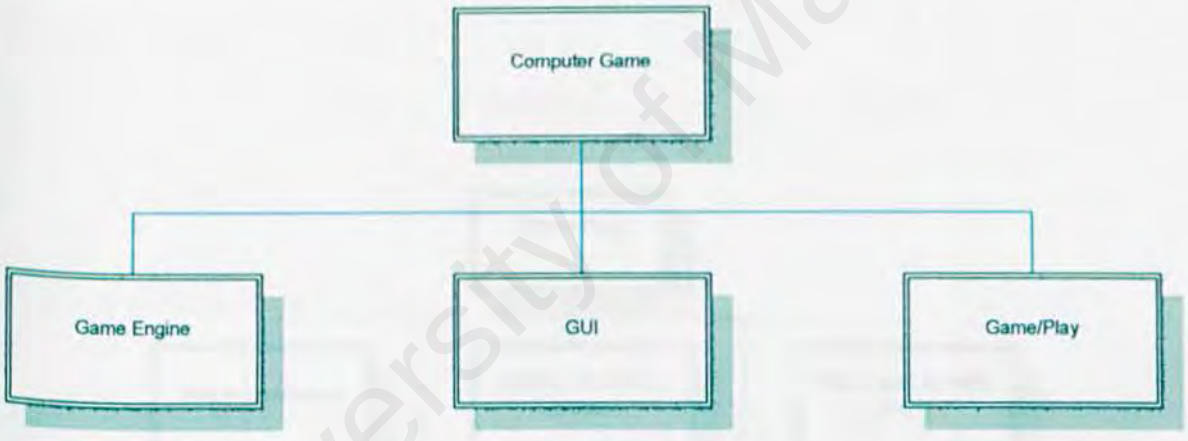


Figure 5.1 : Structure Chart of Computer Game of UNO

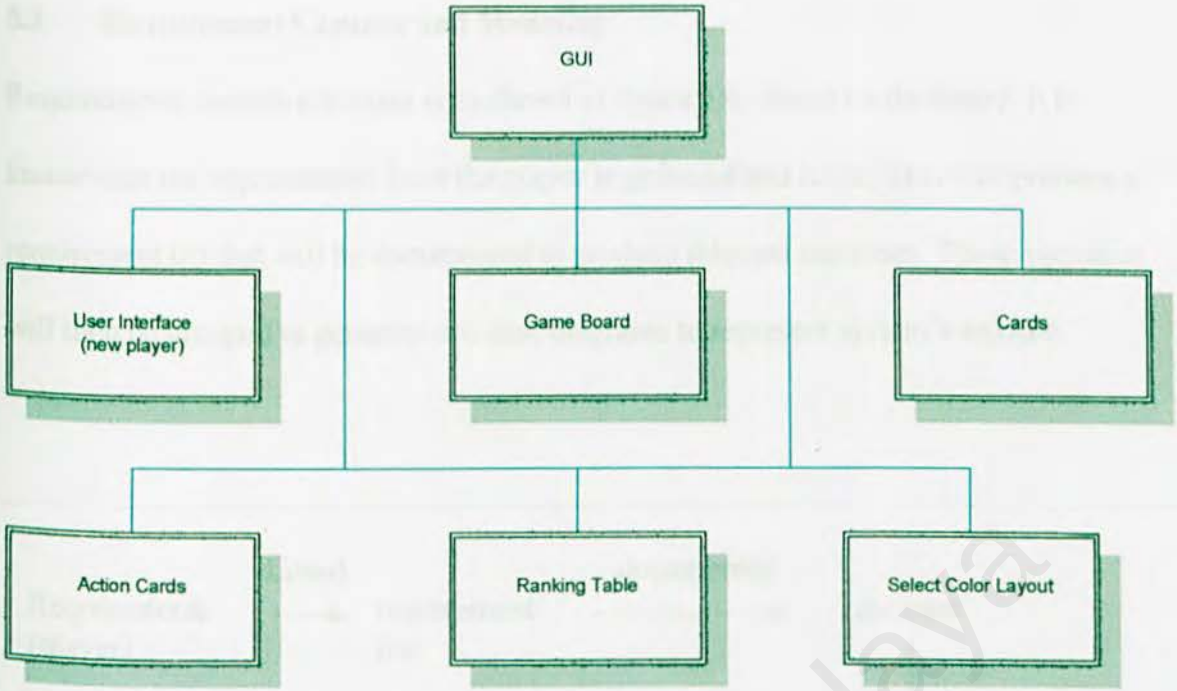


Figure 5.2 : Structure Chart of Graphical User Interface

Figure 5.3 : Structure Chart of Game/Play

5.3 Requirement Capture and Modeling

Requirements capture activities is as shown in figure 5.4. Based on the figure, it is known that the requirements from the player is gathered and listed. This will produce a requirement list that will be documented to produce relevant use cases. Those use cases will then be grouped to generate use case diagrams to represent system's module.

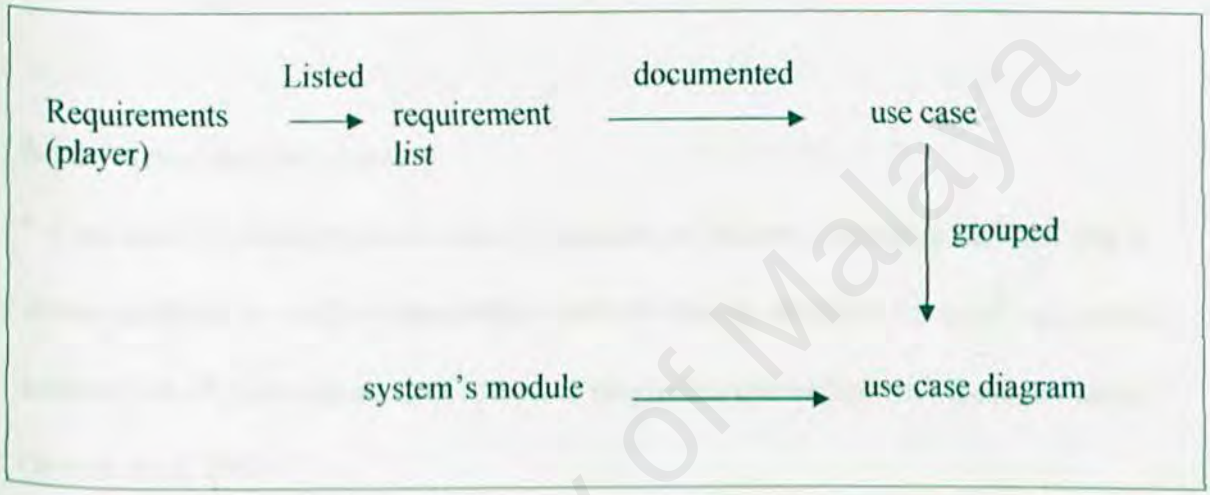


Figure 5.4 Requirement Capture

5.3.1 Requirement List

The requirement list is used to list down the requirements of the system's actor (player). This list consists of two columns, one column to define the requirements and the other one is to show which use case shall provide the required functionality.

Table 5.5 Requirements list for player

Requirements	**Use case**
1) To type in username	Login

2) To start the game	Start Game
3) To draw the cards	Draw Card
4) To place the cards	Place Card
5) To exit the game	Exit Game
6) To display the score points	Game Result

5.3.2 Actor and Use Cases

" A use case is a description of a set of sequences of actions, including variants, that a system performs to yield an observable result of value to an actor. An actor represents a coherent set of roles that users of use cases play when interacting with these use cases " **(Booch et al, 2001)**

Typically, an actor represents a role that a human, a hardware device or another system plays with a system. The actor of this system is as shown below

Table 5.6 Actor of the game system

Actor	**Description**
Player	A player is a user of the system that plays the game. A player is required to click

	Login button to register his/her username. Then player need to click Start button to start the game.

Use case diagrams are used to model the behavior of a system, a subsystem, or a class. Use case diagrams show a set of use cases, actors and their relationships. Relevant use case descriptions are presented in the tables. Based on the use case diagrams that have been drawn, the system has been identified to contain three modules.

- Player Identification
- Game System (UNO)

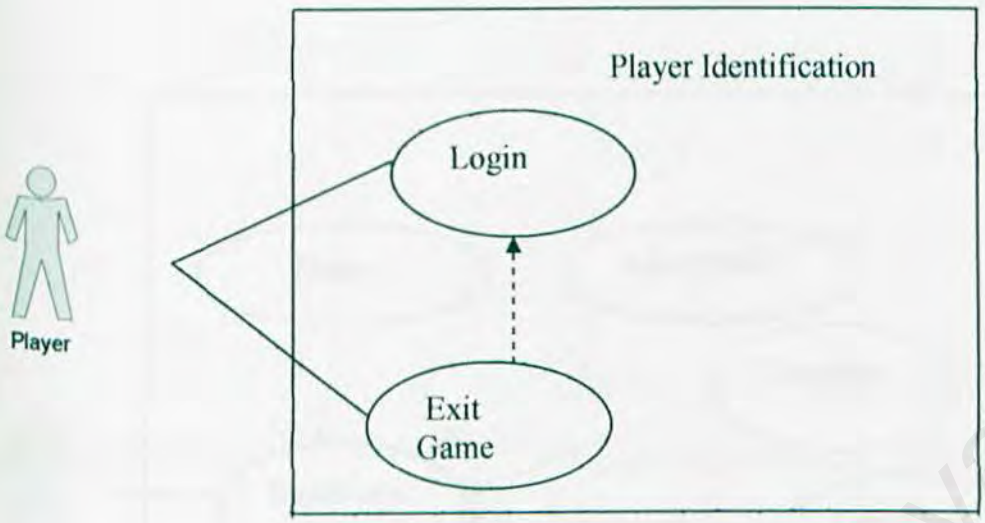


Figure 5.7 : Player Identification Use Case Diagram

Table 5.8: Description of Player Identification Use Case Diagram

Use case	**Description**
1) Login	**Pre-condition:** None **Basic course:** The player of the game must type in his/her username. **Post-condition:** None
2) Exit Game	**Pre-condition:** Started the game **Basic course:** The player clicks the exit window at the page and the system removes player from the system. **Post-condition:** Exit the game

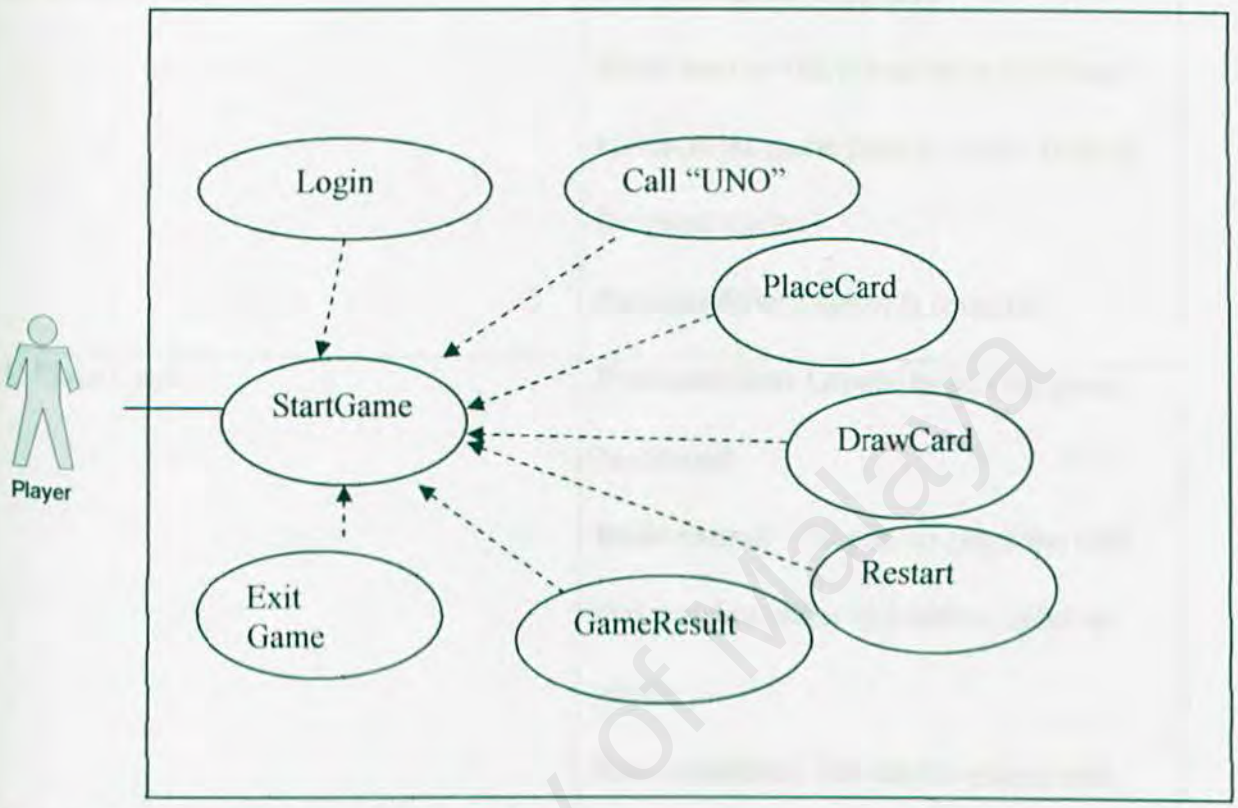


Figure 5.9 : Game System Use Case Diagram

Table 5.10 : Description of Game System Use Case Diagram

Use Case	**Description**
1) Login	**Pre-condition:** None **Basic course:** The player register his/her username to be displayed in the game page **Post-condition:** Logged in
2) Start Game	**Pre-condition:** Logged in **Basic course:** The player starts playing the

	game by clicking the "Start " button **Post-condition:** Game starts
3) Restart Game	**Pre-condition:** Logged in **Basic course** The player click the "Start" button at the game page to restart playing the game again. **Post-condition:** Game is restarted
4) Place Card	**Pre-condition:** Logged in and the game has started **Basic course:** The players place the card that matches either by number, color or words. **Post-condition:** The card is placed and game is played
5) Draw Card	**Pre-condition:** Logged in and the game has started. **Basic course:** The players draw the card from the deck pile. **Post-condition:** The card is drawn and game is played
6) Call "UNO"	**Pre-condition:** Logged in and the game has started

	Basic course: The player click "UNO" button when played second last card. **Post-condition:** "UNO" has been called
7) Game Result	**Pre-condition:** Logged in and the game is finished **Basic course:** The score points will be displayed for both players. **Post-condition:** Game result is displayed
8) Exit Game	**Pre-condition:** Logged in and the game finishes **Basic course:** One player has finished all cards in hand. **Post-condition:** Cards are finished and game is over. Player just close the window to exit the game.

5.4 Use Case Realization

In this section, each defined use cases are analyzed in more details. By using the interaction diagram, the use case realization can be performed. An interaction diagram shows an interaction, it consists a set of objects and their relationships. There are basically two types of interaction diagram which is the sequence diagram and collaboration diagram.

Collaboration diagram will be used to realize the use case of the system. A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Collaboration diagram helps with transition from use case to class diagrams. There are two types of classes that participate in the collaboration; the boundary classes and control classes. Boundary classes model interaction between the system and its actions. The control classes represent the coordination, sequencing transactions and control of other objects.

The use case realization for all the use cases are as shown in the collaboration diagrams.

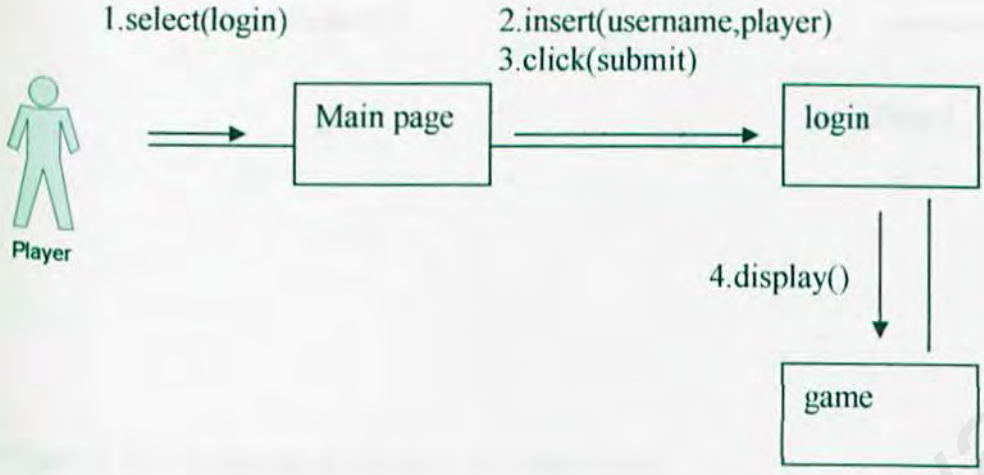


Figure 5.10 Collaboration Diagram for Login

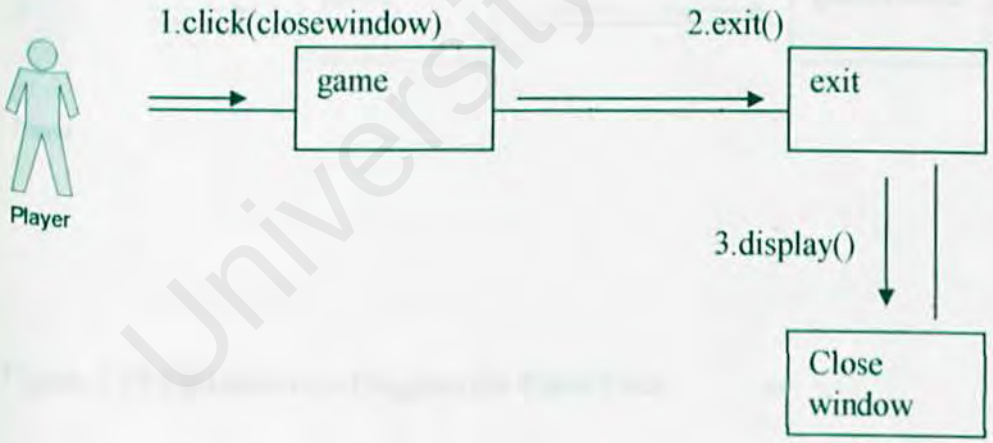


Figure 5.11 Collaboration Diagram for Exit Game

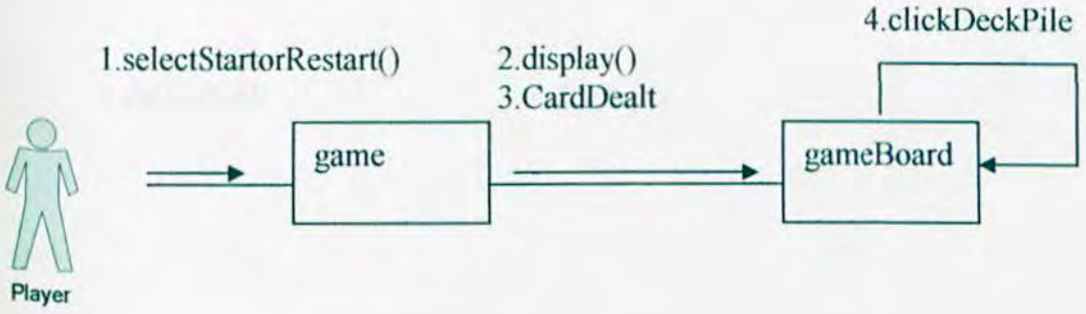


Figure 5.12 Collaboration Diagram for Start Game

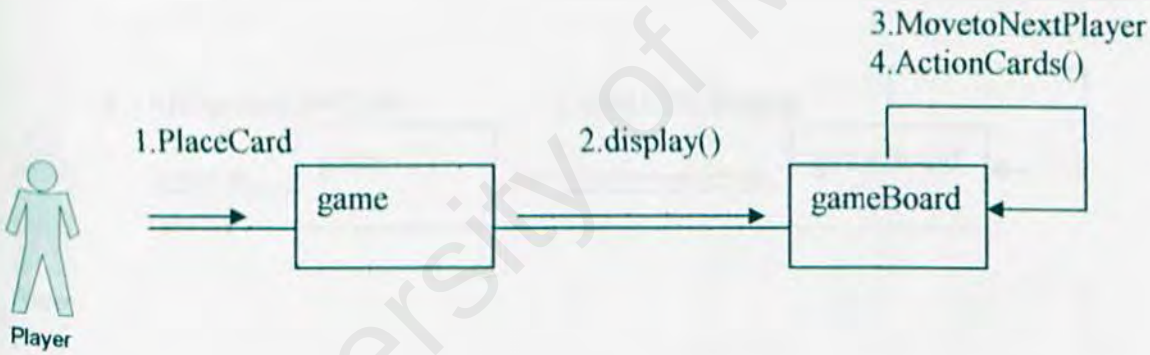


Figure 5.13 Collaboration Diagram for Place Card

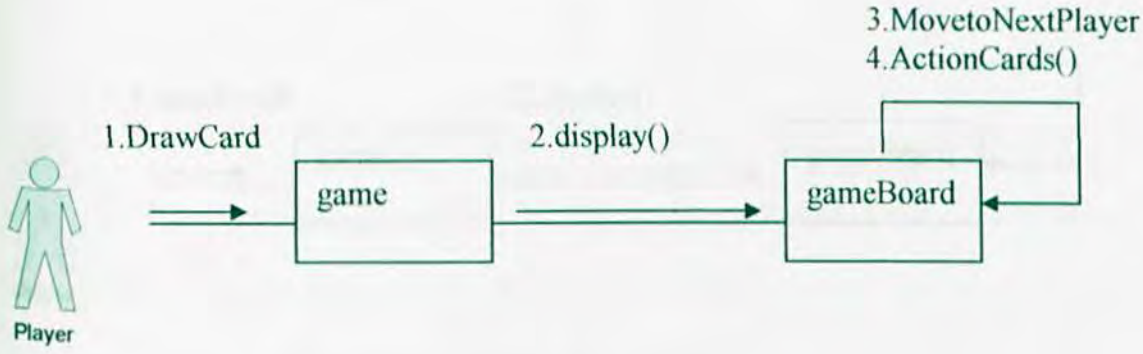


Figure 5.14 Collaboration Diagram for Draw Card

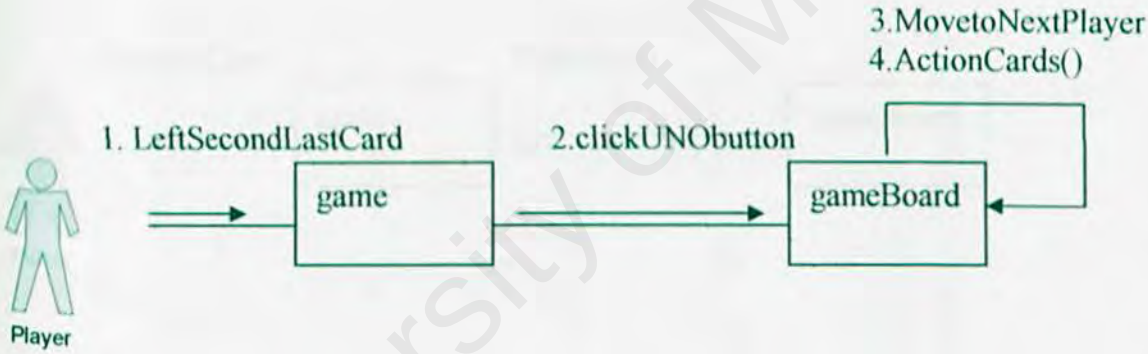


Figure 5.15 Collaboration Diagram for Call UNO

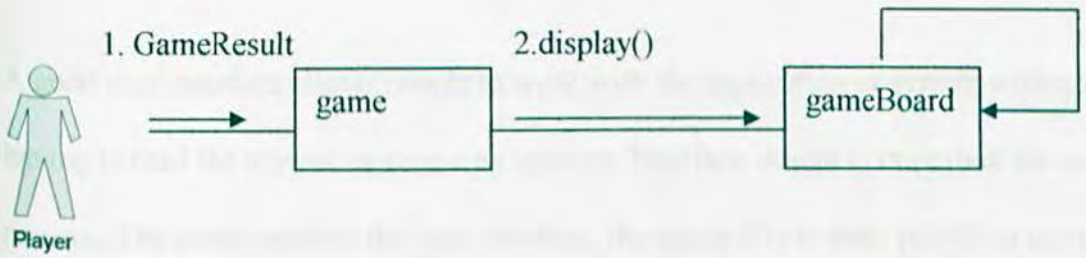


Figure 5.16 Collaboration Diagram for Game Result

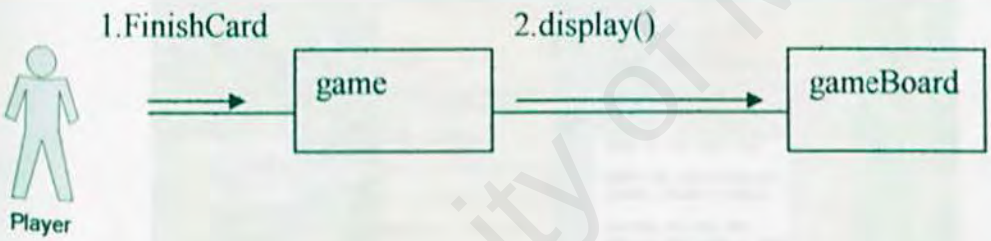


Figure 5.17 Collaboration Diagram for Finish Game

5.5 User Interface Design

A good user interface allows people to work with the application or system without having to read the manual or receiving training. Interface design is important for several reasons. The more intuitive the user interface, the easier it is to train people to use it, to reduce the training costs [**Kendall & Kendall,1999]**

The better interface, the less help people will need to use it, reducing the support costs as well as increasing the user's satisfaction with the work that the designer had done.

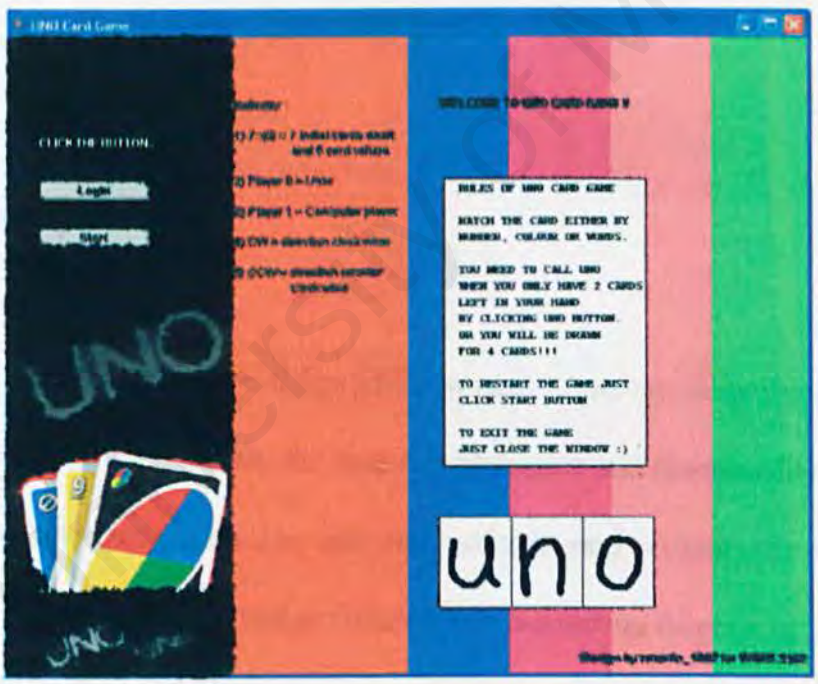


Figure 5.18 Main page of Computer Game of UNO

The main page of Computer Game of UNO consists of :

- **Text area of Indicator and UNO Rules** – this text area section display the indicator and rules to ease user getting understand before start playing the game.
- **Button Login** – this button is the Login button for user to click and to enter the system
- **Button Start** – this button is the Start button where user need to click this button to start the game after enter their nickname.

5.6 Summary

This chapter is mainly about the design of the system. There are many things to be taken for the consideration for example, the system architecture and functionality, and of course the user interface design. Use case realization is used to clarify the requirements captured and collaboration diagram is chosen as an interaction diagram to emphasize the structural organization of the objects in the system.

6.0 SYSTEM DEVELOPMENT AND IMPLEMENTATION

System development is the development, installation and testing of system components while system implementation deals with the delivery of the system into production. In general, during system implementation, the designs that have been discussed in the previous chapter are converted into commands that are understandable by computer by using one or more programming language.

6.1 Software Implementation

Several topics will be discussed as below:

- Software Tools
- Coding Standard
- Coding Approach
- Debugging Mechanism
- Exception Handling

6.1.1 Software Tools

The implementation of the system requires a few of tools to be conducted. The tools have been used for the development of the system are as shown in the table.

Table 6.1: Software Tools

Type of Tool	**Tool**
Compilers, interprets and runtime support	Java 2 Software DK (J2SDK)
Visual editor	JCreator LE
Code editor	JCreator LE
Other supporting tools	NetBeans IDE 3.6, Paint

6.1.2 Coding Standard

The coding standard that has been used includes the naming standard. Naming standard is followed to ensure it is easier to understand and make it easier to trace respective requirements. The naming standard that have been used are as stated as below :

- The name of the variable should start with lowercase letter. The subsequent words are combined together by capitalizing the first letter. This will make the program easier to understand. For example, the name for the variable customer name can be customerName.

- The name of a class most starts with an uppercase letter and followed by lowercase letters. The next word can be combined by capitalizing the first letter. For example, the class name for hello world can be :

```
public class HelloWorld {
}
```

- The name of the functions/methods declared by using a lowercase letter for the first word and then the next word can be combined by capitalizing the first letter. As an example, the calculate salary method can be defined as :

```
public void calculateSalary() {
}
```

6.1.3 Coding Approach

A top down approach is chosen to develop the UNO game system. By using this approach, the modules of the system are broken into functions and procedures.

6.1.4 Debugging Mechanism

The codes that were generated will be most likely to contain errors. Therefore, a proper debugging mechanism must be implemented as a lot of time and effort are required for fixing the errors. There are two kinds of errors that need to be fixed as the system is being developed.

- Compilation errors
- Runtime errors

6.1.5 Exception Handling

Exception handling is one of the debugging mechanism that focus on detecting and responding to unexpected events at runtime. Using this mechanism, some potentially troublesome codes are enclosed within a try and catch clause. An exception is a piece of code that is executed when something abnormal happens to the code. There are several exception classes that are being provided by Java programming language. The syntax for handling an exception takes the form :

```
try {
  //code to throw the exception
  }catch(exceptiontype exception_name){
  //code to handle the thrown exception

  }
```

If an exception is thrown, the statesment after the thrown exception in the try block will not be executed. Instead, the program will try to catch the thrown exception by executing the catch block that matches the exception.

6.2 Component Diagram

A component is a physical and replaceable part of the system that conforms to and provides the realization of set interfaces. Therefore, a component diagram shows a collection of related components. In Java, the program codes that were written down must be compiled so that Java byte codes will be generated. Byte codes are sets of instructions that look like machine codes, but they are not specific to any one processor. The byte codes are then interpreted by a Java byte code interpreter and Java Virtual Machine (JVM) will executed the byte codes.

6.3 Summary

In this chapter of development and implementation, system is elaborated in several sections such as software tools, coding approach and coding standard. Here, the system coding are assembled in packages which classes and extension of classes remain in one package. Objects were called using naming standard and function method is put in lowercase letter and can be combined with the next letter. A top-down coding approach is used to ease developer and debugging mechanism used to fewer errors.

7.0 SYSTEM TESTING

System development involves system testing. The objectives of conducting system are error detection, error removal, error tracking and also regression testing. During the system testing, the system is tested for errors and bugs so that they can be removed from the program. This testing is not done to demonstrate that there are no errors in the system but it is conducted to detect any errors that may still exist.

The types of testing that are conducted are:

- Unit testing
- Integration testing
- System testing

7.1 Unit Testing

Unit testing focuses on verification effort on the smallest unit of software design which is the software or module. Each component or module is treated as a stand alone entity and tested individually to ensure correct operation.

7.2 Integration Testing

After conducting unit test, the individual classes are combined into a system. The purpose of software integration is to adequately test whether or not the software actually runs as one program. There are several techniques available that can be used for integration testing. They are such as bottom-up integration, incremental integration, top-down integration and big-bang integration.

The approach that has been chosen for integration testing of the system is incremental integration testing. Minimal system component are integrated and tested. Then, other components are added to the minimal system component and are tested after each added increment.

7.3 System Testing

A system test is a series of different tests conducted to uncover the system's limitations and measure its capabilities. The objectives are to test an integrated system and verify that it meets specified requirements. There is several system test that can be used. They are as stated below:

- Functional testing
- Performance testing
- Usability testing
- Acceptance testing
- Installation testing

7.3.1 Functional testing

In this test, the integrated system is checked to determine whether it performs its functions as specified in the requirements. Use cases can be used to guide the functional testing that is to be conducted. Table 7.1 shows an example of the test cases for Player use cases.

Table 7.1 Test Case for Player Use Cases

Use case	**Activities Tested**	**Expected Result**	**Result**
Login	User click login button	JDialog appear for user to type in his/her username	As expected
	After type in username and press "Submit"	Username being put in game session and can start play	As expected
	Enter nothing and press "Submit"	Username is null in game session and can start play	As expected
	Enter nothing and close the JDialog	Username is null in game session and can start play	As expected

7.3.2 Performance testing

Performance tests are conducted to ensure that the system response time meets user expectations. During these tests, response time of the system is measured.

A performance test was conducted on the game applet to evaluate the time used to download the game applet. Timer is set equals to 3 seconds for system to move every component in computer player's hand to the Deck pile and for shuffle method it takes every millisecond according to recent timer and randomly pick any cards in the array of cards.

7.3.3 Usability testing

Usability testing is conducted to determine how easy is it for the user to figure out what to do without referring to the user manual or training courses. This test is done by asking a few users to try the developed system to get their opinion about the system's usability.

7.3.4 Acceptance testing

Acceptance testing is done to confirm that the system is developed according to the user requirements and it is ready for operational use. During this test, the users of the system will compare the system to its requirements. This type of testing will be only performed when the system is delivered to the customer.

7.3.5 Installation testing

Installation testing is the last test that needs to be performed to allow users to exercise system functions and to document the problems encountered during system installation. This kind of testing will be conducted only when the system is delivered to the customer.

7.4 Summary

System testing is among the major phase in finalizing the system functionality and requirements work or not. The elements of software quality are assured by implementing several types of testing begin by functional until the installation testing. Before this masterpiece is come to an end, it has been tested by some of users and for the system, it generates to detect errors and failure so it will be errors free when delivered to user.

8.1 System Evaluation

This chapter will discusses about the problems encountered during the system development, solutions applied to the problems, the strengths of the system being developed, the system's limitation, future enhancement that can be done to the system and an overall conclusion for the project being developed.

8.2 Problems Encountered and Solutions Applied

During the system development, several problems were encountered, starting from the inception phase to the transition phase. These problems results in a slow system development process. Listed below are the problems encountered and the solutions that have been applied to them.

- **Lack of experience in developing a full system**

  There have been difficulties in developing the full system (card game) as the system needs to be developed by only one person. The role of system developer, system analyst, system designer and also system programmer are played by the same person. During our three years of studies, we have not been exposed to develop a full system all by ourselves. As a result, many problems arise during the system development. However, this problem was resolved by obtaining advises and opinions from the experienced system developers. The object-oriented programming language (Java) that has been chosen to develop the system was studied to gain further understanding on the language.

- **Time constraint**

Time constraint is the major problem in developing the system. Lack of experience in system development has caused too much time is spent on doing researches and this has results in the time constraint problem. To overcome this problem, the time schedule for the system development is rearranged.

- **Lack of experience in developing computer game**

The developer has no experience at all in developing the computer game. to solve the problem, developer has surf the internet for information and sample. Besides that, the developer had also been referring to Java books for better understanding on game functions.

8.3 System strengths

The system that has been developed has some strength if compared to some other existing system. The strengths are:

- Reusability

Since the game system was developed using an object oriented approach, most of the classes are reusable and this can make the future enhancement easier to be implemented.

- Player username

Compared to other existing game, registration for username is rarely found. This developed game had created the username registration in order to display human player's name.

- Scoring points

This developed game had already stated the values of the cards without user need to refer to other page for the score. So the game can avoid any delay action from user.

- Number of card left

The best part of this system is it can automatically displays the number of cards left in player's hand both human and computer. This method will ease the game flow as human player can quickly detect when to call for "UNO".

8.4 Future Enhancement

Some future enhancements that are proposed for the developed system are as listed below:

- Select number of Player

As this developed game is mainly for one human player and one computer player at one time, for future enhancement it may require more than two players and user can select number of players for him/her to play with.

- Help Page

Help page will be setup to assist user to view an example movement of the game. This will ease new user to expose the technique of the UNO card game and get understand immediately.

- Network system

For future enhancement this UNO card game will be develop as a network game so the game will enable be played via server and connect more than one human player where they can meet online to defeat each other.

8.5 Summary

In conclusion, the game system has been developed using an object-oriented approach, which is the spiral model process. The object-oriented programming language that has been chosen to develop the game system is the Java programming language.

This computer card game may enable two players at one time. This game enable player to test their skill in planning and strategizing.

While developing the game system, highest attention needs to be focused to the efforts of passing and calling objects in classes.

REFERENCES

1. [4] Sellapan,P.(2001).Petaling Jaya:Sejana Publishing
2. [5] Harvey&Paul Deitel (2000).*C How To Program.* Prentice Hall,Inc
3. [9] Tucker, William, "*Chapter 4: Defining the Needs for Models and Simulations*", Applied Modeling & Simulation: *An Development & Operation Integrated Approach to.* Cloud, D. and Rainey, L., editors, U.S. Air Force Academy, 1996.
4. [10] Pfleeger, Shari Lawrence. (1998). Software Engineering: *theory and practice*. Prentice Hall.

Internet Resource

1. [1] (http://inventors.about.com/library/inventors/blpuzzles.htm), 20/03/2004.

2.[2](http://inventors.about.com/gi/dynamic/offsite.htm?site=http://www.thehouseofcard.com/card%5history.html), 21/03/2004.

3. [3] (http://www.cs.uno.edu/~adlai/history_cs.pdf), 22/03/2004.

4. [6] (http://www.webopedia.com/TERM/J/Java.html), 02/04/2004.

5. [7] (http://www.webopedia.com/TERM/S/Smalltalk.html), 05/04/2004.

6. [8] (http://osiris.sunderland.ac.uk/rif/linda_spence/HTML/contents.html) 20/04/2004.

7. (http://java.sun.com/docs/books/tutorial/), 05/09/2004.

BIBLIOGRAPHY

1. Timothy Budd (2000). *Understanding Object Oriented Programming with JAVA*, Updated Edition, Berkeley, California: Addison-Wesley Longman, Inc.
2. Mary Campione, Kathy Walrath, Alison Huml, and the Tutorial Team (1999). *The Java Tutorial Continued: The Rest of JDK*, Berkeley: California: Addison-Wesley Longman, Inc.