**AQA**

***A-level Computer Science***

NEA: The Practical Project

**Interactive Chinese Chess**

**Candidate Name: Liu Kwok Hung**

**Candidate Number: 2060**

**Centre Number: 64655**

###### **Teacher:** SORINA BILETCHI

Contents Page

[1. Analysis 1](#_Toc450162355)

[**1.1.** Introduction 1](#_Toc450162356)

[1.1.1. Background 1](#_Toc450162357)

[1.1.2. Problem Definition 1](#_Toc450162358)

[**1.2.** Investigation of User Needs 1](#_Toc450162359)

[1.2.1. Initial Interview (Questionnaire / Observation also, if used) 1](#_Toc450162360)

[1.2.2. Investigation of inputs, outputs, and/or reports needed 1](#_Toc450162361)

[1.2.3. DFD (to level 1) 1](#_Toc450162362)

[1.2.4. Data Sources and Destinations 1](#_Toc450162363)

[1.2.5. Entity-relationship diagram and entity descriptions (if appropriate) 1](#_Toc450162364)

[1.2.6. Class definitions (if appropriate) 1](#_Toc450162365)

[1.2.7. Analysis Data Dictionary (if appropriate) 1](#_Toc450162366)

[1.2.8. Data Volumes (if a data processing problem) 1](#_Toc450162367)

[**1.3.** Constraints 2](#_Toc450162368)

[1.3.1. Hardware and software constraints 2](#_Toc450162369)

[1.3.2. Time constraints 2](#_Toc450162370)

[**1.4.** Limitations 2](#_Toc450162371)

[1.4.1. Areas which will not be included in computerisation 2](#_Toc450162372)

[1.4.2. Areas considered for future computerisation 2](#_Toc450162373)

[**1.5.** Objectives 2](#_Toc450162374)

[1.5.1. General Objectives 2](#_Toc450162375)

[1.5.2. Specific Objectives 2](#_Toc450162376)

[**1.6.** Alternative solutions 2](#_Toc450162377)

[**1.7.** Justification of chosen solution 2](#_Toc450162378)

[2. Design 3](#_Toc450162379)

[**2.1.** Overall system design 3](#_Toc450162380)

[**2.2.** Description of Modular Structure of System 3](#_Toc450162381)

[**2.3.** Definition of data requirements (Design Data Dictionary) 3](#_Toc450162382)

[**2.4.** Database design (Normalised ERD) – if NOT appropriate, take this heading out before the final submission 3](#_Toc450162383)

[2.4.1. Sample of SQL queries (for a database project) 3](#_Toc450162384)

[**2.5.** File organisation and Processing (if only files used, and no database) 3](#_Toc450162385)

[**2.6.** Identification of storage media 3](#_Toc450162386)

[**2.7.** Identification of suitable algorithms for data transformation (in pseudocode) 3](#_Toc450162387)

[**2.8.** Class Definitions (if NOT appropriate, take this heading out before the final submission) 4](#_Toc450162388)

[**2.9.** User Interface Rationale 4](#_Toc450162389)

[**2.10.** UI Sample of Planned Data Capture and Entry Designs 4](#_Toc450162390)

[**2.11.** UI Sample of Planned Output Designs 4](#_Toc450162391)

[**2.12.** Description of measures planned for security and integrity of data 4](#_Toc450162392)

[**2.13.** Description of measures planned for systems security 4](#_Toc450162393)

[**2.14.** Overall Test Strategy 4](#_Toc450162394)

[3. Technical solution 5](#_Toc450162395)

[**3.1.** Forms used 5](#_Toc450162396)

[**3.2.** List of variables and procedures used 5](#_Toc450162397)

[**3.3.** Sample algorithms which demonstrate techniques used 5](#_Toc450162398)

[**3.4.** Annotated program listing 5](#_Toc450162399)

[4. Testing 6](#_Toc450162400)

[**4.1.** A Minimal set of Test Data 6](#_Toc450162401)

[**4.2.** Details of individual tests 6](#_Toc450162402)

[**4.3.** Samples of annotated hard copy of actual test runs for typical, erroneous and boundary data 6](#_Toc450162403)

[5. Evaluation 7](#_Toc450162404)

[**5.1.** Comparison of project performance against general and specific objectives 7](#_Toc450162405)

[**5.2.** Independent feedback 7](#_Toc450162406)

[**5.3.** Analysis of feedback 7](#_Toc450162407)

[**5.4.** Future development 7](#_Toc450162408)

# Analysis

## Introduction

### Background. Problem Definition

Traditional board games are seeing a decrease in popularity in the recent years due to the variety of games that are played on consoles and mobile devices. As a result, I think the traditional board games should have a comeback on these platforms.

In addition, since these traditional board games are not easy to carry around and individual pieces may be lost which can be extremely frustrating, an electronic version of the game should be made in order to solve these problems

## Investigation of User Needs

### Initial research

Questions I asked my potential clients:   
“If I am going to make (code) a game for my upcoming project, what game would you suggest?”

Most frequent response from my clients  
 “Just make whatever you want to make”

July 2016

“Why don’t you make something like [League of Legends]? –Justin

\*Although League of Legends is my favourite game, its complexity of its functionality is too much for me to accomplish and to complete by the end of April.

“Hangman (with animation) seems pretty fun.” – A Year 9 pupil from my boarding house

\*The idea itself is interesting but the code behind it will be too simple for me to achieve enough marks in the exam in terms of complexity.

August 2016

“(Cantonese) If you are making a game, why don’t you make Chinese Chess? You used to love the game when you were a child.” –Mum

\*The major reason why I have decided to choose to do a Chinese Chess game as it reminds me of my childhood when I play with my family members a lot, which I eventually got better than them and win prizes in some competitions.

General Responses:

\*My reflection on their opinion(s)

### Investigation of inputs, outputs, and/or reports needed

Input with be mouse clicks as all other data (chess pieces) will be created during runtime as a new player is declared  
  
Output will be the form itself while chess pieces and possible locations are always shown on the board and when a player makes a move respectively

### DFD (to level 1)

### Class definitions (if appropriate)

Form1: The display of the board that the players will be able to click on.

Game: Set up board and both players (red and black)

Board: A board of a 8\*9 array which stores the details of the chess piece located at the point, display a picture box and a target box after a chess piece is clicked.

Player: Locate chess pieces on the board at the start of the game according to the side of the player

ChessSelected: Display possboxes at locations according to the type of the chess piece (picture box) clicked.

MoveChess: Move the chess piece to the location and display it where the possbox is clicked. Then make the picture box at the original location invisible.

If the new location of the chess piece is different from the original location, pass the turn to the other player

## Objectives

### General Objectives

1. To create an electronic version of Chinese Chess
2. The game can be played by two users on the same device

### Specific Objectives

1. Input names of both players before the game starts
2. The user will be able to decide the number of rounds that he/she wants to play
   * The user can leave the input blank (Total rounds = 255)
   * The user will need to repeat the input action if number is out of range or if the input contains non-numbers (string)
3. Players’ name will be display on screen
4. Players’ score will be display on screen
5. Start a new Game
6. Generate Locations (an array) on screen which contains the following details
   * Occupied
     + Make the Picture box at the location visible when it is true
   * Side
     + Side of the chess piece located on it
   * Value
     + Value of the chess piece located on it
   * PicBox
     + Display the image of the chess piece located on it
7. Generate array of possboxes (target) on the screen which is only visible when after a chessc
8. Create picture boxes to display chess pieces for both players.
9. Chess pieces of different values\* should have a specific move pattern to follow
10. All chess pieces should not be able to move out of the board (Validation).
11. Chess pieces on different sides might have a boundary moving area within the board  
    E.g. Soldiers, Elephant, Advisor, General.
12. Each player takes turns to move their chess pieces (1 move per turn).
13. During a player’s turn, if one of the pieces is being selected, possible locations
    * for the piece to move to should be put into a list
    * displayed on the board (using graphics/picture boxes)
14. When a possible location is clicked, the specific chess piece should move to its new location
15. Both Generals should not be able to move to a location which faces another General directly on the board.

## Justification of chosen solution

# Design

## Design overview/Introduction

The aim of the project is to create an electronic version of Chinese Chess that can be played on computers or mobile devices. Forms are used so user can access the chess pieces and move them correspondingly with the graphical display of image boxes.

## Modular Structure of the System

Form1 for user interface, immediately after execution, a New Game is created

After names are input for both players, chesses are allocated to them using the player class

Chess pieces are generated (using the ChessPiece class) and displayed on the board

When a valid chess piece is clicked, a New Move will be created and the chess piece can perform certain moves according to their values

## My complex algorithms (in pseudocode (?))

## Class Definitions (if appropriate)

(Explain how does the classes function / how subroutine works)

Form1 – Graphical display of board (The user interface), Timer, Current player (Red/Black)

Game – Starting a game by creating New players (Player1 = True, Player2 = False)

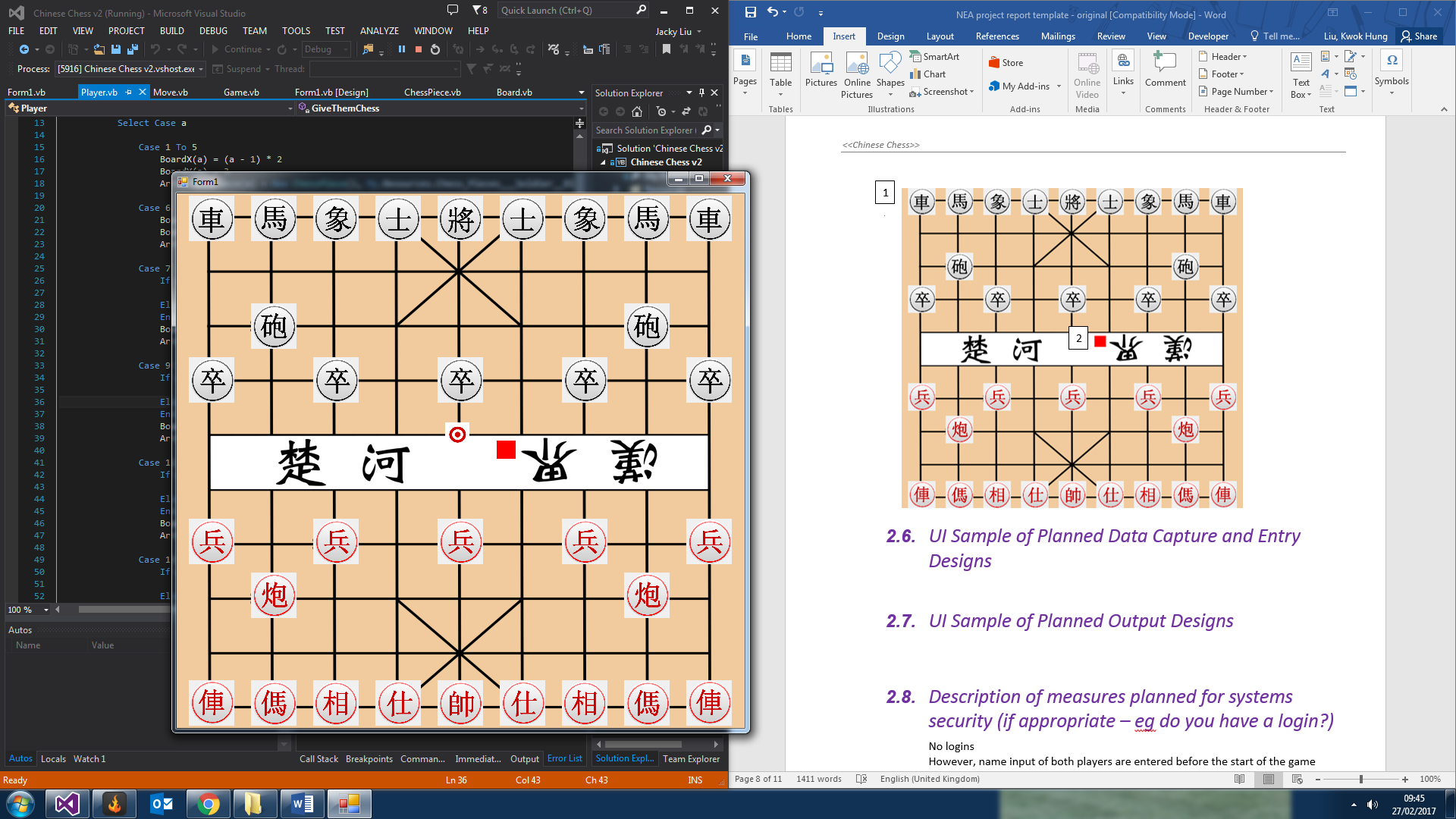
Player – Allocating chess pieces (16 of them) to players and giving the chess pieces their value (how they can move), side (Red/Black), locating them on the board and displaying their corresponding images

Board – Setting constants such as how far away the joints are from each other, the position of the top left joint (0,0), how big is the board and set up the possible location (invisible) red dots

Move – Managing how chess pieces of different values move differently. When a chess piece is clicked while the side of it matches current player, possible locations to move to should be shown on board. If a click on a red dot follows, the chess piece will move to the red dot’s location, current player will change to the opposite value and all the red dots shown will become invisible again.

## User Interface Rationale

1



3

4

2

1. Black chess piece (Chariot) at (0,0)

2. Red chess piece (Soldier) at (2,6)

3. Possible location red dots when black chess piece at 4,3 is clicked

4. Image box that changes colour when a move is done by the current player

## UI Sample of Planned Data Capture and Entry Designs

All chess pieces that are still alive (on the board) are displayed by image boxes

Possible location (red dots) remain invisible until a chess piece is clicked and will become invisible again when the player finishes a move

## UI Sample of Planned Output Designs

## Description of measures planned for systems security (if appropriate – eg do you have a login?)

No logins

However, name input of both players are entered before the start of the game

# Technical solution

## Complex code reference (page numbers)

### Description: ….. page…..

## Annotated program listing, starting with a screenshot of the form

# 

# 

# Testing

## Test plan (Finish by Easter)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no | Purpose | Description | Test Data | Expected Outcome | Actual Outcome |
| 1 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Samples of annotated hard copy of actual test runs for typical, erroneous and boundary data (if appropriate) or link to video of project testing

# 

# Evaluation

## Comparison of project performance against general and specific objectives

## Independent feedback

## Analysis of feedback

## Future development