CAB201 Class Assignment – Card Games Part A

Due Date: 2nd June, 2015
Weighting: 40%

Specification Version 1.0 (1st May 2015)

In this assignment, you have the option of working in pairs. If anyone in the class wishes to work alone, they will not receive any special consideration if they are unable to complete the assignment in time. Programming pairs need to be registered with Mike Roggenkamp by email before COB 19th May, 2015.

Given 4 weeks to complete this assignment, ***there will be no extensions granted under any circumstances.*** *Plan to submit by 30thMay, 2015. Need to start working on this part of the assignment before 8th May.*

If illness or other exceptional circumstances prevent you from working on this assignment during the four weeks of May, email Mike Roggenkamp, m.roggenkamp@qut.edu.au, for possible alternative assessment arrangements.

Table of Contents

1. Introduction

This assignment aims to give you a "real world experience". You have been hired to complete as much as possible a project that has not been fully specified at this stage. You have been given (in your mind) an impossible delivery date. There is some supporting documentation to the project. In order to fulfil your contract you must deliver the prototype program which fulfils the stated functionality by the required date.

By the way, this is called the **Class Assignment** because it shows the use of *multiple C# classes* (contrasted to your first two 10% assignments, where all your code was in a single class).

2. The Task

You have been hired you to implement a prototype that is an early stages of the development. The programmer who was to implement the prototype has left the company at short notice and has left behind the prototype project folder which is basically empty apart from containing the shell of three projects which will eventually

hold various classes. The prototype project folder is **GUI Games** and it contains the following three (3) project folders: **GUI Games**, **Game_Class_Library** and **Shared Game Class Library.** Open **Solution Explorer** to see three projects.

The Company wants you to implement and complete the prototyping of two card games. The Company requires you to:

(a) Use **Windows Forms**, rather than other GUI technologies such as WPF, XNA, web-pages, game engines, etc. To keep this program relatively simple, do not use advanced techniques such as MVC or layered architectures. (If you don't know what some of these acronyms mean, that's fine. Just ignore them.)

(b) Develop GUI layouts that are very similar to the ones in the screenshots shown in the documents, **GUI Notes on xxx_Game_Form.docx**. That is, don't go making up a GUI of your own design. As well do not waste time looking for Icon and Background images for your forms. There are no marks for having "good-looking" forms. Your GUI layout on each form should look fairly similar to the screenshots, though you are not expected to have the locations and size, of each control, correct to the last pixel. The GUI layouts will be explained with the release of **Part B** of this assignment.

(c) Develop the game logic which will play each of the Card Games according to the specified rules which will be explained with the release of **Part C** of this assignment.

3. Shared Game Class Library

In of this assignment you will implement the classes which will be part of the **Shared Game Class Library** project folder.

These classes will define the basic Game Objects which will be used by the actual Game Logic classes. The Game Logic classes will be contained within the **Game_Class_Library project** which will be implemented in **Part C.**

The basic Game Object classes are **Card class**, **Hand class** and **Pile class**.

- **Card class** represents a playing card that has a face value (Ace, King, Queen etc) and a suit value (Clubs, Diamonds, Hearts and Spades)

- **Hand class** is used to hold the playing cards that have been dealt during a card game. A hand may be that dealt to a player in a multi-player game or part of the layout of a single player game as in games like Hearts, Free Cell and various forms of Solitaire.

- **CardPile class** holds a collection of playing cards. Objects in this class may be used for different purposes. E.g. one **CardPile** object could hold a full deck of playing cards (52 cards – does not include Joker cards) at the start of a game, while another **CardPile** object could be used to hold cards that have been discarded by the players during a game. The number of cards in a **CardPile** object will change, as a game is played. This class is **not** used to hold the cards that each player has – see **Hand Class** above

Each of these classes has a UML class diagram in **Shared Game Class Library UML Diagrams.docx.** This document contains additional information for the implementation of each of the classes.

The **Shared Game Class Library** currently has an empty class named **Class1.cs** this is merely a placeholder. You can rename this class to be **Card.cs** and use it to implement the **Card class.** You will add the other 2 classes yourself.

4. Where to Start

Implement the **Card class** first, followed by the **Hand class** and then **the CardPile class** according to the information supplied in **Shared Game Class Library.docx.**

All classes will contain a class comment, all methods will have a method comment which will include a post condition. A precondition is required whenever the method has a parameter.

5. Notes

Though all care has been taken in the production of this specification, there may be a need to notify by class email any alterations/clarifications to this specification. **Check your QUT email daily, even if you have redirecting this to your own email check QUT directly in case your email provider blocks the QUT group email.**

Part B will be release during Week 10 and **Part C** will be released at the start of Week 11.

Working in Optional Pairs
Ensure that both people in the group are involved and are responsible for doing some part of the assignment.

Do not forget to register your group via email to Mike Roggenkamp before 19th May, 2015.

To register your group supply full name and student number of each person and the name for your group eg. Dream Team or 2 Blokes (apologies to "3 Blokes", a local Games company formed by ex students). The group name simplifies the recording of your assignment mark. If no name is supplied, one will be generated "randomly" for you.

6. Submission Details

Will be provided by the 19h May 2015 on Blackboard.