# Assignment 2

## Camel Up Game

This project will allow you to implement a computer game, with some simple graphics. It is a game where you will implement some simple artificial intelligence and incorporate it in the computer player.

This project is worth 7% of your final grade. You must do this project on your own. The submission deadline is the last teaching day of semester, **Friday 11 October 2019 at 4pm**. Your submission will be marked out of 12 in the lab, then your C# code will be marked out of 8 after you submit your program.

## Project Specification[1]

You have been asked to write a computer version of the Camel Up board game, by Steffen Bogen. It is a camel race with a couple of twists, where you bet on which camel wins and which camel loses. The object of the game is to gain the most money by backing the right camel to win a leg or even the entire race.

In Camel Up, the players bet on a camel race that is run in 4 legs. Each leg, players bet on which camel will come first, as well as betting on the overall race winner and loser. Players roll dice of a random colour, to move the camels forward, and can place tiles on the racetrack to enhance or delay the camels movement. The interesting feature of the race is that when two, or more, camels land on the same square, they stack up, and when a lower camel in the stack moves, it carries all camels stacked on top of it as well.

On any turn there are 4 possible moves, and the player can only take one of them. The moves are:

1. take a bet for this leg. This is done by taking one of the leg bet cards, giving a payout of 5 or 3 or 2 for a coloured camel if they are in first place. The card pays 1 if the coloured camel is in second place, and -1 if the camel is third or later.

2. place a bet for overall winner or loser. This is done by placing a coloured camel card on either the winning stack, or the losing stack. After the race is finished, the first person to place a card of the correct colour will get 8, the second 5 and the third 3, and after that the correct colour gets 1. All wrong colours have to pay 1.

[1] Rules from https://boardgamegeek.com/boardgame/153938/camel

3. place an oasis/desert tile on the track. The tile can only go on a blank square, i.e. it does not have a tile or camel already on it, that is not square 1, and does not have a tile on an adjacent square.

4. or roll a dice to move a camel. The dice colour is chosen randomly (from the camel colours that have not yet moved this leg) and moves the camel (and any camels stacked on top of it) forward 1, 2 or 3 squares. If they land on a tile then they move forward (oasis) or backward (desert) one space, and give 1 dollar to the player who placed the tile. The person who rolled the dice gets 1 for this action.

A leg ends when each coloured dice has been rolled (each camel has moved). Players are paid out depending on their leg bets and the position of that coloured camel. If camels are stacked the camel on top is ahead of the one below.

The game ends when a camel crosses the finish line, even if it is not the end of a leg. First score as if it was the end of a leg, then the overall winner and overall loser is scored. Winner has the most points/money.

The full rules are in the Camel_Up_Rules.pdf file.

Your program should implement (a simplified version of) the game where the user plays against a computer player.

## Robot player

A computerised version of this game will need a robot player to play against. At each turn the robot player needs to choose one of the four actions: take a bet for this leg, place a bet for overall winner or loser, place an oasis/desert tile on the track, or roll a dice to move a camel. The robot player must only make legal moves.

A purely random robot player would not be that interesting to play against. So, your robot player should try to have some artificial intelligence, or ‘heuristics’, such as only playing an oasis tile if there are over 3 camel dice left in the pyramid. Is the computer a ‘safe’ player, taking only bets for camels that are ‘obviously’ going to win, or are they ‘greedy’ taking bets that have a higher payout (or picking earlier) that might be riskier. Or is the computer player only taking moves that immediately pay money (moving camels and placing tiles) and not that interested in betting at all. Getting the computer to play differently depending on what the camels are doing would be ideal.

## Tasks

Write a program that satisfies the specifications described above. It should make use of the programming features that you have learnt so far, in particular using classes, subclasses and methods to structure your program.

The project requires you to both get it verified in the lab and to hand in the source code. Half of the grade will be based on how the program runs and half on the structure, code style, documentation and user interface/interaction.

We suggest that you build a simplified version of the game first, and then add in more features as time permits. Concentrate on getting the objects and simple turn play right before adding fancy graphics and controls. It is often easier to control both players, adding a computer player once you have a basic game going. Once you have a stupid computer player, then add more intelligence and heuristics (preferably into player subclasses).

If your program is structured well, you should be able to let the user choose which opponent they want to play, and the different behaviours are provided by setting which subtype of objects are used.

Note: There is a card game version, and a second edition, of Camel Up. You are welcome to implement either of these games if you prefer.

# COMPX102-19B

## Assignment 2 Hand-in due 4pm 11th October 2019

Compress (Zip) the Visual Studio folder with your program code and submit it via Moodle.

### Student Declaration of Originality

I declare that the program which I have had verified and submitted in Moodle is entirely my own work. I have not worked together with any other people. I have suitably acknowledged (referenced) any parts of other programs that I have used. I understand that if I have breached the above conditions I will be sent to the University Disciplinary Committee.

Note: This project will only be marked if this Declaration of Originality is signed.

**Name:** ____________________________________

**ID Number:** ________________________________

**Signed:** ___________________________________

**Date:** _____________________________________

### Functionality and Usability (to be demonstrated in the lab)

| | |
|---|---|
| Displays 'board' and coloured camels | _____ /1 mark |
| Random coloured dice is rolled and moves camel | _____ /1 mark |
| Stacks of camels move when bottom camel moves | _____ /1 mark |
| Can place Oasis and Desert tiles on correct squares | _____ /1 mark |
| Oasis and Desert tiles affect movement. Pay owner | _____ /1 mark |
| Leg bets can be taken and payout at end of leg | _____ /1 mark |
| Coloured cards can be placed on winner/loser stacks | _____ /1 mark |
| Game end and payouts implemented. Winner chosen | _____ /1 mark |
| Random robot player implemented | _____ /1 mark |
| 'Smarter' robot player implemented | _____ /1 mark |
| Usability (layout, choice of controls, feedback, etc.): | _____ /2 marks |
| **Functionality Total:** | _____ **/12 marks** |
| Bonus: Game resizes when change form size | _____ /+1 mark |
| Bonus: Multiple players, either robot or human | _____ /+1 mark |

### Coding Style

| | |
|---|---|
| Code Style (object design, docs, contracts, etc.): | _____ /8 marks |
| **Total:** | _____ **/20 marks** |