University of Westminster
School of Computer Science and Engineering

6SENG001W Reasoning About Programs
Coursework 2 (2021/22)

Module Leader	Klaus Draeger (K.Draeger@westminster.ac.uk)
Module Team	Paul Howells (P.Howells@westminster.ac.uk)
Unit	Coursework 2
Weighting:	50%
Qualifying mark	30%
Description	Develop a B specification of *a Snakes & Ladders Game*, using the B tools Atelier B & ProB. Compute pre-conditions for implementation fragments.
Learning Outcomes Covered in this Assignment:	The coursework assesses learning outcomes: LO1, LO2, LO3 & LO4.
Handed Out:	November 2021
Due Date	**13:00, Tuesday, 11 January 2022**
Expected deliverables	Electronic files: (a) B Specification Structure Diagram (.pdf) (b) *Snakes & Ladders Game* B Specification: B machines (.mch) (c) Graph of ProB Animation Session History (.dot) (d) Screenshot of Atelier B type check (.jpeg/.jpg) (e) A document containing the Hoare Logic part. All files should be compressed into a single ZIP archive. The ZIP archive should be named using your surname & "cwk", e.g. "howells_cwk.zip".(See section 3 for full details.)
Method of Submission:	**Online via Blackboard**
Type of Feedback and Due Date:	Verbal feedback in tutorial(s) before the assessment is submitted. Sample answers of the assessment after 15 working days (3 weeks). Written feedback and marks 15 working days (3 weeks) after the submission deadline.

	All marks will remain provisional until formally agreed by an Assessment Board.

Assessment regulations

Refer to section 4 of the “How you study” guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:

http://www.westminster.ac.uk/study/current-students/resources/academic-regulations

Coursework Description

1. Introduction

This coursework requires you to develop a B specification of a Snakes & Ladders *board game*, using the B tools Atelier B & ProB.

Figure 1. gives the layout of the board you must use for the game. The aim of the game is to throw a single dice & move up the board. In the game if you land at the bottom of a ladder then you move to the top of it. If you land at the head of a snake then you move down to its tail. The goal is to complete the game by landing on the last square, i.e. 100.

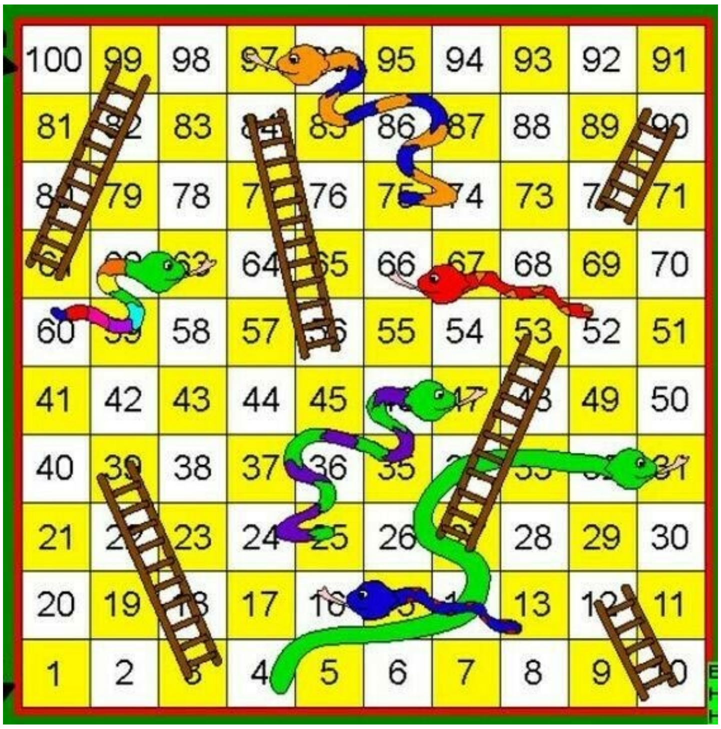

Figure 1. The Snakes & Ladders *board*

Notes

- The board is made up of 100 squares, starting form 1 up to 100.
- There are 6 snakes on the board. For example, the green snake's head is in square 31 & its tail is in square 4. If you land on its head square 31 then you move down to its tail square 4.
- There are 6 ladders located on the board. For example, the first ladder's bottom is in square 3 & its top is in square 39. If you land on its bottom square 3 then you up to its top square 4.
- The head of a snake is must always be higher up the board than its tail, similarly a ladder's bottom must always be lower down the board than its top.
- A square cannot be both a head & a tail square for either the same snake or different snakes. Similarly a square cannot be both a bottom & a top square for either the same ladder or different 's.
- In other words, a square can only contain: nothing (a normal square), or 1 snake's head, or 1 snake's tail, or 1 ladder's bottom, or 1 ladder's top.

- For obvious reasons:

    - a snake's head cannot be in the first or last squares,
    - a ladder's bottom can not be located on the top row.
- The state of the game includes the following:
    - the current position on the board,
    - the dice value last thrown,
    - the number of moves,
    - the number of snakes gone down,
    - the number of ladders gone up,
    - the list of squares visited in the order visited.

2 Develop a B Specification of the Snakes & Ladders *board game*

Your B specification, i.e. collection of 1 or more B machines, should include the following elements.

2.1 Sets and Constants

Any sets, constants and properties that are required to define the data and state of the board and state of a player playing the game.
(Hint: you should not try to represent the board as a grid.)

2.2 System State

The state variables required to represent: the player's position on the board, the number of snakes and ladders that have been encountered, the number of turns taken. In addition the list of squares that have been visited in the order visited. This should include all empty squares landed on, for a snake encounter this would be two, its head and tail squares, for a ladder encounter this would be two, its bottom and top squares. The most recent dice value that was thrown.

Including the state invariant and initialisation.

2.3 Operations

The system has the following four operations.

2.3.1 Move

A move operation that adds the last thrown dice value to the current position on the board then takes what ever action is appropriate for the square landed on. That is either: go down a snake, go up a ladder, do not move as the new square is above 100, land on the last square and win, or just landed on a normal square.

The move operation must report its outcome:

- a message indicating that the move either went down a snake, up a ladder, threw too high a dice value to finish, finished and won, landed on a normal square,
- the square it occupies as a result of the move.

2.3.2 GameStatistics

Outputs the current statistics for the game: current position on the board, the number of snakes and ladders encountered, number of turns taken.

2.3.3 VisitedSquares

Outputs the *list* of squares that have been visited in the order visited.

2.3.4 NewGame

Reinitialises the game so that a new game can be started and outputs a message stating a new game has started.

2.4 Specification Structure & General Requirements

The specification structure can be developed as a:

- single B machine that contains all the state and operations, or
- three B machines, representing the *board, dice, game*.

The B specification should use the appropriate features to define the data and operations in your B machines.

The specification must be syntactically and type correct, as checked by using the Atelier B tool.

The specification must be animated by ProB. That is it must *initialise* correctly and all operations can be *animated* successfully and used to play the game to completion.

3. Write and analyse an implementation fragment using Hoare Logic

- Write a While Program modelling a single move in the "Snakes and Ladders" game. It should use variables **roll** and **position** to represent the given dice roll and current player position, respectively, and correctly update position taking into account the snakes and ladders.
- Compute the pre-condition for reaching the target in two legal moves. For this, you will need to analyse the program **roll:=firstRoll; P; roll:=secondRoll; P** where **P** is your program from the first bullet. Write a proof outline to compute the pre-condition for the post-condition **position=100 & firstRoll>0 & firstRoll<7 & secondRoll>0 & secondRoll<7** with respect to this program. Note that you will need to simplify the intermediate pre-conditions as much as possible, otherwise their size will get out of hand.

4. Blackboard Submission

The following 4 components are to be submitted via Blackboard:

(1) The Structure Diagram of your Snakes & Ladders game B machines.
SUBMIT: 1 ".pdf" file. **[10%]**

(2) The B Specification of the Snakes & Ladders game.
SUBMIT: the B machines ".mch" file as is. (**DO NOT submit it as a Word file.**) **[50%]**

(3) Examples of successfully using the 2 B tools Atelier B and Prob.

For Atelier B: a single screen shot showing Atelier B's main window the successful type checking of all machines in the specification, i.e. the "specification box" with the "green TC" circle.

For ProB: a Graph representation of a complete ProB Animation Session history. Using ProB perform an animation sessions that shows a game being played from start to finish, that includes going down at least one snake and up at least one ladder. View this Animation Session as a "DOT" graph and then save it.

SUBMIT:

- ◦ 1 Atelier B screen shot in either “.jpg” or “.jpeg” format.
- ◦ 1 ProB animation graph ".dot" file as is. **(DO NOT open using Word.)** **[10%]**

(4) Your Hoare Logic analysis.

- A text document containing the while program for a single game move.
- A text document containing the proof outline for the pre-condition analysis **[30%]**

NOTE: All files should be compressed into a single ZIP archive. **ONLY ZIP format archives will be accepted.** The ZIP archive should be named using your surname & "cwk", e.g. "howells_cwk.zip".

Coursework Marking Scheme Overview

The Coursework will be marked based on the following main component marking criteria, the full marking criteria details will be published on the module's Blackboard site.

Criteria	Mark per component	Mark provided	Comments
B Specification Structure Diagram of the *Spaceship & Asteroids* System	10		
B Specification of Snakes & Ladders Game	50		
Atelier B type check screen shot, ProB Animation Session History Graph	10		
Hoare Logic	30		
Total	**100**		