University of Westminster
School of Computer Science and Engineering

6SENG005W Formal Methods Referred/deferred Coursework (2023/24)

Module Leader	Klaus Draeger (K.Draeger@westminster.ac.uk)
Module Team	Alexander Bolotov, Tom Oliver
Unit	Coursework
Weighting:	50%
Qualifying mark	30%
Description	Develop a B specification of *a Snakes & Ladders Game*, using the B tools Atelier B & ProB.
Learning Outcomes Covered in this Assignment:	The coursework assesses learning outcomes: LO1, LO2, LO3, LO4.
Handed Out:	June 2024
Due Date	**13:00, 9th July 2024**
Expected deliverables	Electronic files: (a) B Specification Structure Diagram (.pdf) (b) *Snakes & Ladders Game* B Specification: B machines (.mch) All files should be compressed into a single ZIP archive.
Method of Submission:	**Online via Blackboard**
Type of Feedback and Due Date:	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 and 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 die and move up the board. If you land at the bottom of a ladder, you move to its top. If you land at the head of a snake, you move down to its tail. The goal is to complete the game by landing on the last square, i.e. 100.

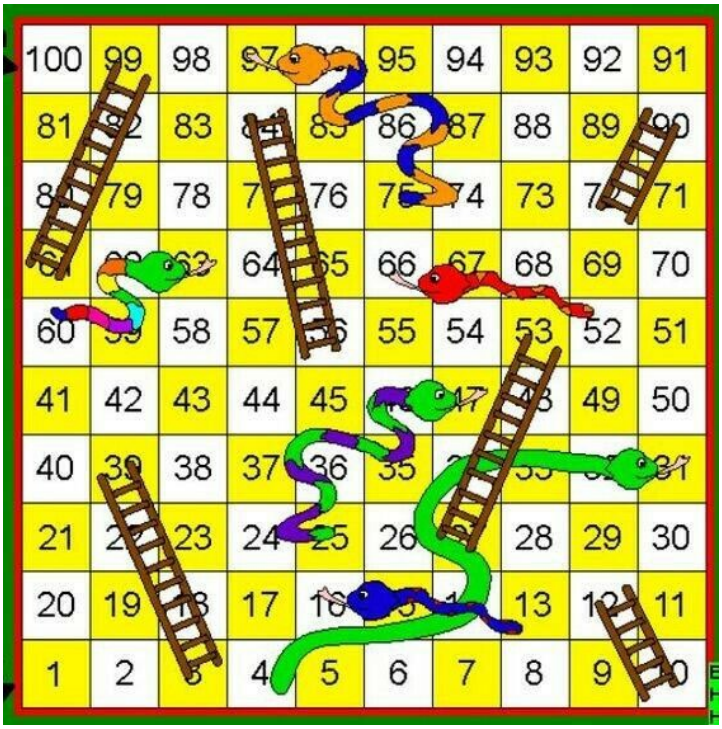

Figure 1. The Snakes & Ladders *board*

Notes

- The board is made up of 100 squares, starting from 1 up to 100.
- There are 6 snakes on the board. For example, the green snake's head is in square 31 and 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 and its top is in square 39. If you land on its bottom square 3 then you up to its top square 39.
- The head of a snake is must always be further up the board than its tail, similarly a ladder's bottom must always be further down the board than its top.
- A square cannot be both a head/tail/top/bottom for more than one snake or ladder.
- For obvious reasons:
  - a snake's head cannot be in the first row or the last square,
  - a ladder's bottom cannot be located on the top row.
- The state of the game includes the following:

- the current position on the board,
- the die value last thrown (note that die rolls are simulated by user choices),
- 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
- the list of squares that have been visited in the order visited. This should include all empty squares landed on, for a snake or ladder encounter this would be two (its head and tail squares or its bottom and top squares, respectively)
- 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 **(given as an input parameter)** to the current position on the board then takes whatever action is appropriate for the result. That is either:

- do not move as the new square is above 100
- move onto a snake’s head and down to its tail
- move onto a ladder’s bottom and up to its top
- land on the last square and win, or
- just move to 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 after 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 able to be animated by ProB. That is, it must *initialise* correctly, and all operations need to work correctly to play the game to completion.

3. Blackboard Submission

The following components are to be submitted via Blackboard:

- The Structure Diagram of your Snakes & Ladders game B machines as a pdf file. **[20%]**
- The B Specification of the Snakes & Ladders game as mch files. **[80%]**

NOTE: All files should be compressed into a single ZIP archive. **ONLY ZIP format archives will be accepted.**

Coursework Marking Scheme Overview

The Coursework will be marked based on the following main component marking criteria.

Criteria	Mark per component
Structure Diagram of Snakes & Ladders Game	20
B Specification of Snakes & Ladders Game	
Sets and constants	15
System state	15
Move operation	20
GameStatistics operation	10
VisitedSquares operation	10
NewGame operation	10
Total	**100**