University of Westminster
School of Computer Science and Engineering

6SENG005W	Formal Methods – Coursework (2023/24)
Module leader	Klaus Draeger
Unit	Coursework
Weighting:	50%
Qualifying mark	30%
Description	Checking acyclicity of directed graphs
Learning Outcomes covered in this assignment:	This assignment contributes towards the following Learning Outcomes (LOs): LO1, LO2, LO3, LO4
Handed Out:	October 2023
Due Date	13:00, Tuesday, 9th January 2024
Expected deliverables	**A zip file containing** **(1) A structure diagram (.pdf format)** **(2) A specification using the B language (.mch format)**
Method of Submission:	Electronic submission on 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 very simple version of the old *Asteroids arcade game*, using the B tools Atelier B & ProB.

Figure 1. gives the layout of the regions of space (a rectangular grid shape), the *Spaceship* is represented by the blue triangle, its starting position is its home base (1, 1).

The aim is to move the *Spaceship* from *its home base* through space using the various movement operations to get to the *Starbase (6,4), avoiding the Asteroids.*

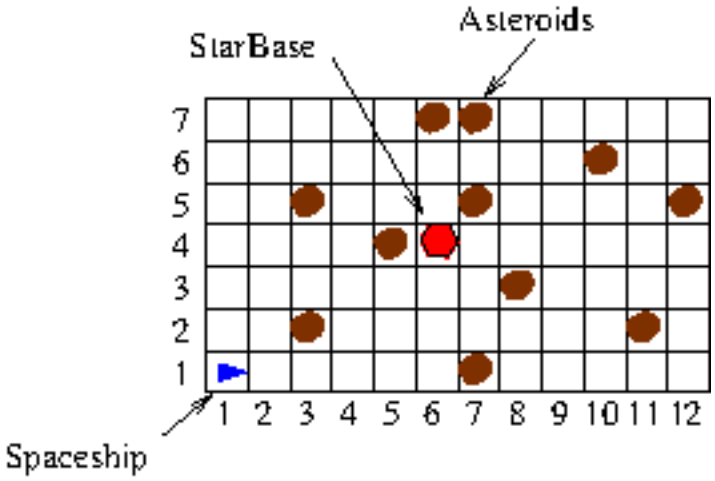

Figure 1. Regions of Space, Spaceship, Star Base & Asteroids

Notes

- Space is made up of regions (squares of the grid) 12 wide by 7 high.
- The regions of space are populated by **11 asteroids**, each in one region of space, and located as shown in Figure 1.
- The Spaceship occupies only one square at a time which must be either an "empty space square" or the Starbase in its square (6,4).

  For example, the Spaceship can be in region (5, 3), but not (8, 3) as it is occupied by an asteroid.
- The Spaceship is initially in its *homebase*, i.e. the bottom left square (1, 1).
- The spaceship can make a normal move, i.e. from one region of space (grid square) to an adjacent one, in one of four directions: Up, Down, Left and Right.
- It uses up **5 units of power** when it makes a normal move.
- The spaceship can engage its *warp-drive* to “jump” to any region of space, except one occupied by an asteroid. It must not travel outside known space, i.e. outside the grid.
- It uses up **20 units of power** when it engages its warp-drive, no matter how far it moves.
- If the spaceship crashes into one of the asteroids it **loses 10 units of power** and bounces back into the square it came from.
- It cannot do a move for which it does not have the required amount of power.

- The state of the game is one of the following:
    - the spaceship docks at the Starbase, in which case the game has been **Won.**
    - the spaceship is **not docked** at the Starbase & **can not move** because it has run out of power, in which case the game is **Lost.**
    - otherwise the game is **not over**.

2 Develop a B Specification of the Regions of Space, Spaceship & Asteroids

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

2.1 Sets and Constants

Any sets and constants that are required to define the data and state of the spaceship, space, asteroids and their properties.
(Hints: Represent space and the asteroids as relations. What is the relationship between space, "empty space" & the asteroids locations?)

2.2 System State

The state variables required to represent space, asteroids and the spaceship. Including the state invariant and initialisation.

You can assume that the spaceship starts at its homebase, has no power yet, has not had any collisions and it has only visited the regions of space its homebase is located in.

2.3 New Game

To start or re-start the game use the *NewGame(power)* operation. This should reset the spaceship to the initial state, except that it sets its power level to the value of the *power* parameter.

2.4 Spaceship Movements in Space

Note that all movement operations must report the outcome of an attempted movement. That is, either it was successful, failed due to space boundary issues, failed due to an asteroid, or failed for some other reason.

2.4.1 Normal Spaceship Movements

The following operations are the basic movements that all move the spaceship one region (square) in the appropriate direction in space and uses up 5 units of the spaceship's power:

- *MoveUp*
- *MoveDown*
- *MoveLeft*
- *MoveRight*

Note that If the move results in the spaceship hitting an asteroid the spaceship remains in its current location, but its **power is reduced by 10 units**.

If any attempted movement cannot be performed because of the boundary of space or insufficient power then **an error is reported**.

2.4.2 Warp-drive Spaceship Movement

The movement operation:

- *WarpDrive(newposition)*

where the player enters the *newposition* parameter**,** the region of space (i.e. grid co-ordinates) that the spaceship should warp jump to. Engaging the warp-drive **uses up 20 units** of the spaceship's power.

If the warp-drive cannot be used because the destination region input is either not within the known regions of space or is occupied by an asteroid or if there is insufficient power to use the warp-drive then an **appropriate error message should be reported**.

2.5 Spaceship's Mission Status

An enquiry operation *MissionStatus* that reports the current status of the spaceship:

- the game status (WON, LOST, or NOT_OVER),
- its current location,
- its current power reserves,
- how many asteroid collisions it has had.

2.6 Spaceship's Mission Route

An enquiry operation *RegionsVisited* that reports the regions of space that the spaceship has travelled through.

2.8 General Requirements

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

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 move the Spaceship around the regions of space, e.g. from the homebase to the Starbase using a combination of **all** of the movement operations, including the warp jump.

3. Submission & Lab Demonstration

3.1 Blackboard Submission

The following 3 components are to be submitted via Blackboard:

1. The Structure Diagram of your *Spaceship & Asteroids* System B machine. You must also include as a "plain English" description of the "state invariants" of the system. Examples of Structure Diagrams can be found in the lecture notes and in the tutorial exercises.
   SUBMIT: 1 ".pdf" file. **[20%]**

2. The B Specification of the *Spaceship & Asteroids* System.
   SUBMIT: the B machine ".mch" file as is. (**DO NOT submit it as a Word file.**) **[80%]**

Coursework marking scheme:

Criterion and range	Indicative mark	Comments
Structure Diagram (0-10 marks)	8-10	A structure diagram has been submitted and includes all the parts of the machine(s) and relations between them.
	4-7	A diagram has been submitted but parts are incomplete or missing.
	0-3	Diagram is missing or fundamentally incorrect.
Invariant Explanation (0-10 marks)	8-10	A clear and correct explanation of the invariant has been provided.
	4-7	An explanation has been provided but is incomplete or incorrect, or just restates the formula.
	0-3	Explanation is missing or fundamentally incorrect.
Sets and Constants (0-15 marks)	11-15	All necessary sets and constants have been defined, including their properties. Types of constants are suitable.
	6-10	Sets, constants and properties are incomplete or not entirely correct.
	0-5	Major parts are missing or wrong.
Variable (0-15 marks)	11-15	All necessary variables have been defined, including their invariants and initialisation. Types are suitable.
	6-10	Variables, invariants and initialisation are incomplete or not entirely correct.
	0-5	Major parts are missing or wrong.
NewGame operation (0-10 marks)	8-10	Operation is defined correctly and does what is expected.
	4-7	Operation is mostly correct.
	0-3	Operation is missing or inherently incorrect.
Normal movement operations (0-10 marks)	8-10	Operations are defined correctly and does what is expected.
	4-7	Operations are mostly correct.
	0-3	Operations are missing or inherently incorrect.
WarpDrive operation (0-10 marks)	8-10	Operation is defined correctly and does what is expected.
	4-7	Operation is mostly correct.

	0-3	Operation is missing or inherently incorrect.
MissionStatus operation (0-10 marks)	8-10	Operation is defined correctly and does what is expected.
	4-7	Operation is mostly correct.
	0-3	Operation is missing or inherently incorrect.
Route operation (0-10 marks)	8-10	Operation is defined correctly and does what is expected.
	4-7	Operation is mostly correct.
	0-3	Operation is missing or inherently incorrect.