University of Westminster
Department of Computer Science

6SENG001W Reasoning About Programs Coursework (2018/19)

Module Leader	Klaus Draeger (K.Draeger@westminster.ac.uk)
Module Team	Paul Howells (P.Howells@westminster.ac.uk)
Unit	Coursework
Weighting:	40%
Qualifying mark	30%
Description	Develop a B specification of *a Spaceship & Asteroids 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:	Week starting 22nd October 2018
Due Date	**13:00, Thursday, 6th December 2018**
Expected deliverables	Electronic files: (a) B Specification Structure Diagram (.pdf) (b) *Spaceship & Asteroids Game* B Specification: B machine(s) (.mch) (c) Graph of ProB Animation Session History (.dot) Lab Demonstration of B specification & use of B tools. (See below 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 10 working days (2 weeks). Written feedback and marks 15 working days (3 weeks) after the submission deadline, the week starting Monday 7/1/2019. **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 *Spaceship & 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.*

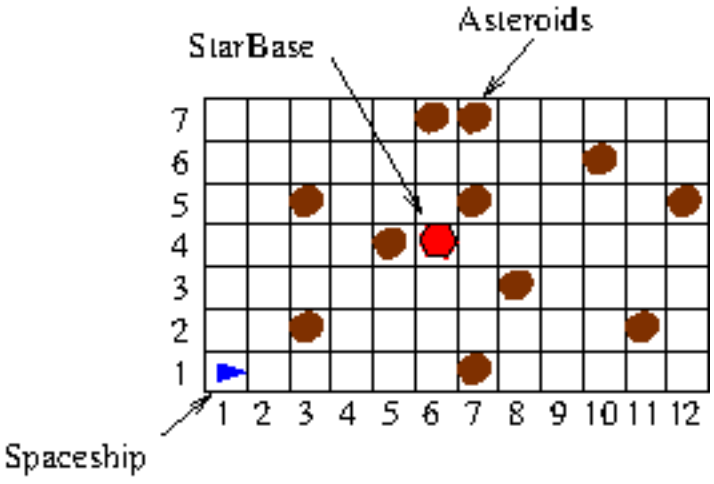

Figure 1. Regions of Space, Spaceship, Star Base & Asteroids

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, & located as shown in Figure 1.
- The Spaceship occupies only one square at a time & can only be in an "empty space square", except when “docked” at 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, Forward (right) & Reverse (left).
- It uses up **5 units of power** when it makes a normal move.
- The spaceship can engage its *warp-drive* & "jumps" 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 many regions it moves.
- If the spaceship crashes into one of the asteroids it **loses 10 units of power**.
- If it has less than the required amount of power to do either a normal move or a warp-drive jump it can not do that type of move.
- 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 1 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 has a reasonable amount of power to start with, but you can vary this when testing the specification in the ProB animator.

2.3 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.

2.3.1 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`
- `MoveForward`
- `MoveBackward (Reverse)`

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.

2.3.2 Warp-drive Spaceship Movement

The movement operation:

- `EngageWarpDrive( newposition )`

where the player enters `newposition` the region of space (i.e. grid co-ordinates) that the spaceship should 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.

2.4 Spaceship's Mission Status

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

- its current location,
- its current power,
- how many asteroid collisions it has had.

2.5 Spaceship's Mission Route

An enquiry operation `MissionRoute` that reports the route (regions of space) that the spaceship has travelled through.

(**Note**: Bonus marks will be awarded if the record of the regions of space indicates the order in which they were passed through, rather than just recording them.)

2.6 Spaceship is Docked at Starbase

An enquiry operation `DockedAtStarbase` that reports whether the Spaceship is "docked" at the Starbase, i.e. its current location is the Starbase.

2.7 Game Status

An enquiry operation `GameStatus` that reports:

- "Game WON" if the game is over & has been won;
- "Game LOST" if over & has been lost; and
- "Game Not Over" if its not over.

2.8 General Requirements

The B specification should use the appropriate features to define the data & operations in any machines that you define. If you use a collection of machines then you must use the appropriate B structuring features to combine them. Bonus marks will be awarded if multiple B machines are **used appropriately**.

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

The specification **must** be **animated** by **ProB**. That is it must *initialise* correctly & all operations can be *animated* successfully & can be used to move the Spaceship from the homebase to the Starbase using a combination of **all of the movement operations**, i.e. it is **not acceptable** to immediately warp to the Starbase.

3. Submission & Lab Demonstration

3.1 Blackboard Submission

The following 3 components are to be submitted via **Blackboard**:

1. The Structure Diagram of the *Spaceship & Asteroids* System, in terms of the abstract machine(s) used to specify it. You must also include as a note with the diagram a "Plain English" explanation 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. **[10%]**

2. The B Specification of the *Spaceship & Asteroids* System, i.e. the abstract machine(s).
   SUBMIT: all B machine plain ASCII Text files: ".mch". **[55%]**

3. A Graph representation of a complete ProB Animation Session history. Using ProB perform an animation sessions that shows how the Spaceship is moved from the start square to the Starbase square using a combination of **all of the movement operations**, including examples of non-successful operations. Note that it is not acceptable to just immediately "warp" to the Starbase. View this Animation Session as a "DOT" graph and then save it.
   SUBMIT: 1 graph file: ".dot" **[10%]**

3.2 Lab Demonstration

After the coursework has been submitted all students must attend a lab demonstration of their B specification using the B tools: Atelier B & ProB, during the normal tutorials sessions:

Date: **Monday, 10th December 2018**
Tutorial 1: 11:00 – 13:00, Lab LG.107.
Tutorial 2: 14:00 – 16:00, Labs 5.113.

The purpose of the lab demonstration is to ensure that the specification is: syntax and type correct as checked by Atelier B, and can be animated and that students can use a selection of ProB's features to analyse their specification. **[25%]**

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 web site & Blackboard.

Criteria	Mark per component	Mark provided	Comments
B Specification Structure Diagram of the *Spaceship & Asteroids* System	10		
B Specification of *Spaceship & Asteroids* System	55		
ProB Animation Session History Graph	10		
Lab Demonstration of: use of Atelier B & ProB on Specification to type check, animate & analyse the B Specification.	25		
Total	**100**		