Centre Number						Candidate Number				
Surname										
Other Names										
Candidate Signature										

General Certificate of Secondary Education
June 2014

Computer Science 4512/CB3

Component 1: Practical programming

Candidate Booklet

Scenario 3: Game Application

BotMod

For candidates entering for the 2014 examination
To be issued to candidates on or after 1 September 2012

This scenario is one of four available. Each of the four scenarios is available in a separate candidate booklet. You must complete ***two*** *of the four scenarios.*

- You have approximately 25 hours in which to complete this scenario.
- Before starting work on the problem, read the whole of this Candidate Booklet thoroughly. You can ask your teacher to explain anything in this booklet, except Computer Science specific terms, that you do not understand.
- There are restrictions on when and where you can work on this problem. Your teacher will explain them to you. For example, you should only do work that you intend to hand in for marking when a teacher is present, so that he or she can confirm that the work is your own. The Candidate Booklet must **not** be taken outside your school/college.
- You may need to use the Internet to research certain parts of the problem. This does not have to be within the 25 hours recommended time.
- You will need to complete and sign a Candidate Record Form which your teacher will provide.

Information

You will also be marked on your use of English. It is important to:

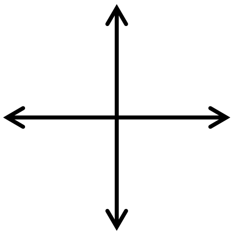
- make sure that all your work is legible
- use correct spelling, punctuation and grammar
- use a style of writing which suits the person you are writing for
- organise your information clearly, so that you make yourself understood
- use Computer Science terms where they are needed.

Scenario 3: BotMod

BotMod is a game where a player has to guide a robot through a landscape to collect people and carry them to a base camp before the robot's power runs out. The landscape is represented by a 10 by 10 grid and contains a mixture of grassland, rocks and ice.

The robot always starts in the top left square of the grid. The player can move the robot one square at a time in any of the directions shown in **Figure 1**.

Figure 1



The robot **cannot**:

- make diagonal moves
- move beyond any edge of the grid.

When the robot moves into a square containing a person:

- if the robot has space in its passenger bay, the person is automatically picked up
- if there is no space in the passenger bay, the person remains in the square and is not picked up.

When the robot moves into the square containing the base camp all of the people in the robot's passenger bay are automatically unloaded.

The robot starts with 150 power units.

Before the player begins controlling their robot they should modify it by choosing the:

- type of traction they want (wheels, tracks or skis)
- size of the passenger bay (large, medium or small).

These modifications affect how the robot operates as it moves around the different types of terrain in the landscape. Different choices mean that different amounts of power will be used by the robot as it moves around the landscape. The size of the passenger bay also determines the maximum number of people that the robot can carry at one time. Full details are given in **Tables 1 and 2** on **page 5**.

After each move made by the robot the number of:

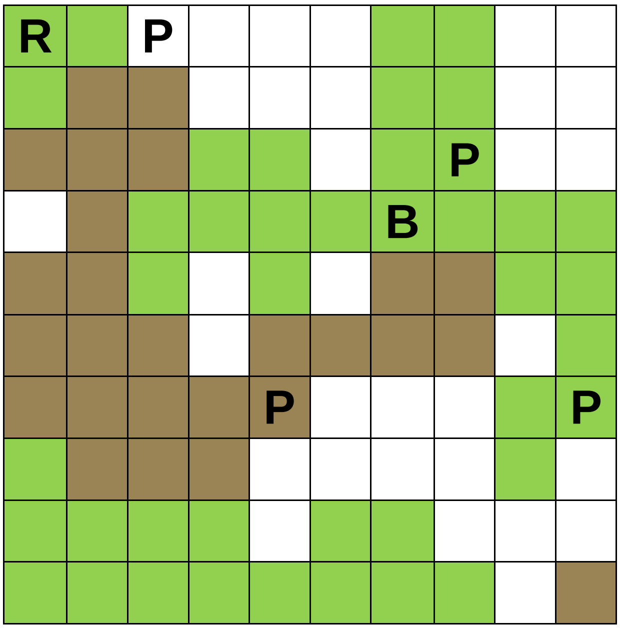
- passengers being carried by the robot
- power units the robot currently has

should be displayed to the player.

The player wins the game when all of the people are dropped off at the base camp before the robot runs out of power.

The landscape and initial positions of the robot, people and base camp are shown in **Figure 2**.

Figure 2



🟩 R	🟩	⬜ P	⬜	⬜	⬜	🟩	🟩	⬜	⬜
🟩	🟫	🟫	⬜	⬜	⬜	🟩	🟩	⬜	⬜
🟫	🟫	🟫	🟩	🟩	⬜	🟩	🟩 P	⬜	⬜
⬜	🟫	🟩	🟩	🟩	🟩	🟩 B	🟩	🟩	🟩
🟫	🟫	🟩	⬜	🟩	⬜	🟫	🟫	🟩	🟩
🟫	🟫	🟫	⬜	🟫	🟫	🟫	🟫	⬜	🟩
🟫	🟫	🟫	🟫	🟫 P	⬜	⬜	⬜	🟩	🟩 P
🟩	🟫	🟫	🟫	⬜	⬜	⬜	⬜	🟩	⬜
🟩	🟩	🟩	🟩	⬜	🟩	🟩	⬜	⬜	⬜
🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬜	🟫

Key

Colour of square	**Type of terrain**
Green	Grassland
White	Ice
Brown	Rocks

Letter	**Object**
R	The robot
P	A person
B	Base camp

Turn over for the tasks

Turn over ►

Tasks

1) Develop a start menu for the game. This must give the user the options of modifying their robot, playing the game or exiting the program.

2) Develop the part of the program where the player modifies their robot. The player should be able to select their choices and return to the start menu when they have finished. They should be able to make the following choices:

   - type of traction they want (wheels, tracks or skis)
   - size of the passenger bay (large, medium or small).

3) Develop the program so that when the player chooses to play the game from the start menu the landscape and initial positions of the robot, people and base camp are displayed (as shown in **Figure 2**).

4) Develop the part of the program to enable the player to move the robot as described in **Figure 1**.

5) Develop the program so that the robot automatically picks up a person when it moves into the square they are occupying if there is enough room in its passenger bay. It should also automatically drop off all its passengers when it moves into the square containing the base camp.

6) Develop the part of the program that calculates and displays the power units and number of passengers after each move. At the start of every game the robot should have 150 power units. Refer to **Tables 1 and 2** for how these values should change after each move.

7) Develop the part of the program that checks if the player has won or lost the game.
   a. The player has won if all of the people have been taken to the base camp.
   b. The player has lost if the number of power units runs out.
   c. If the power units run out **on** the last move the player has still won.

   When the player finishes a game a relevant message should be displayed and they should be returned to the start menu.

8) Extend the program by creating more levels for the game that increase the difficulty for the player. Each time a player wins a game, they move up a level and the robot starts with fewer power units than on the previous level.

Table 1

Traction type	Effect when moving onto grassland	Effect when moving onto rocks	Effect when moving onto ice
Wheels	Costs 1 power unit	Costs 2 power units	Costs 3 power units
Tracks	Costs 3 power units	Costs 3 power units	Costs 3 power units
Skis	Costs 3 power units	Costs 3 power units	Costs 1 power unit

Table 2

Size of passenger bay	Maximum passengers at any one time	Power Cost
Large passenger bay	3	Costs 2 additional power units per move
Medium passenger bay	2	Costs 1 additional power unit per move
Small passenger bay	1	Costs 0 additional power units per move

Turn over for an example of the game in action

Turn over ►

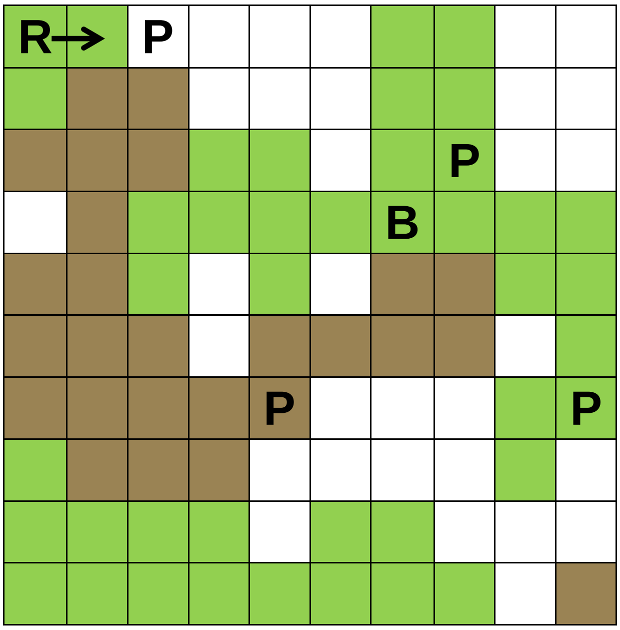
An example of the game in action

The player has made the following modifications to the robot:

- Traction type: wheels
- Passenger bay size: large

Number of passengers: 0
Power units: 150

Move 1 (robot moves onto grassland)

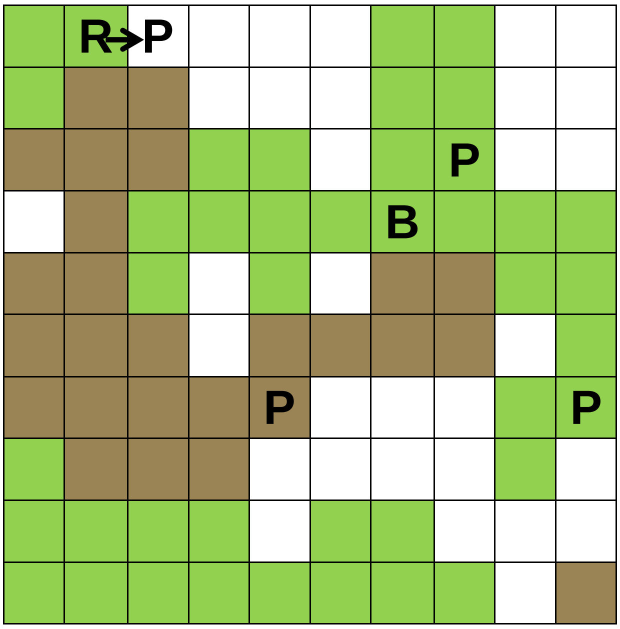


🟩 R	🟩 →	⬜ P	⬜	⬜	⬜	🟩	🟩	⬜	⬜
🟩	🟫	🟫	⬜	⬜	⬜	🟩	🟩	⬜	⬜
🟫	🟫	🟫	🟩	🟩	⬜	🟩	🟩 P	⬜	⬜
⬜	🟫	🟩	🟩	🟩	🟩	🟩 B	🟩	🟩	🟩
🟫	🟫	🟩	⬜	🟩	⬜	🟫	🟫	🟩	🟩
🟫	🟫	🟫	⬜	🟫	🟫	🟫	🟫	⬜	🟩
🟫	🟫	🟫	🟫	🟫 P	⬜	⬜	⬜	🟩	🟩 P
🟩	🟫	🟫	🟫	⬜	⬜	⬜	⬜	🟩	⬜
🟩	🟩	🟩	🟩	⬜	🟩	🟩	⬜	⬜	⬜
🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬜	🟫

Wheels cost 1 power unit
Large passenger bay costs 2 power units

Number of passengers: 0
Power units: 150 – 1 – 2 = 147

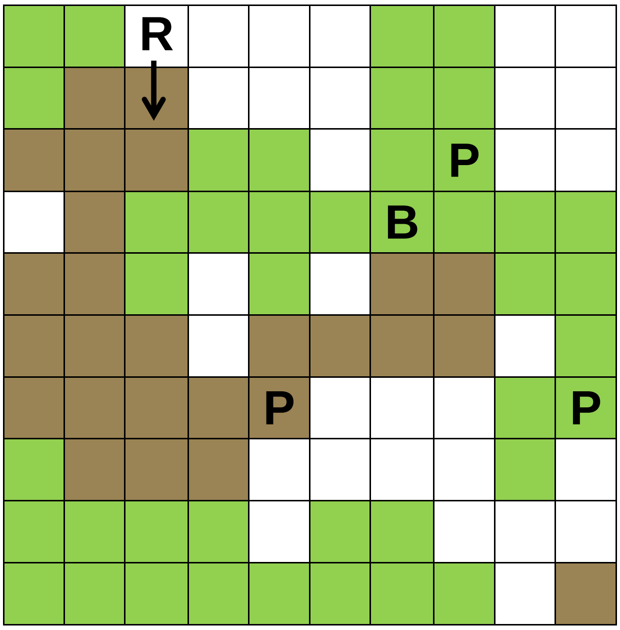
Move 2 (robot moves onto ice to collect a person)



Wheels cost 3 power units
Large passenger bay costs 2 power units

Number of passengers: 0 + 1 = 1
Power units: 147 – 3 – 2 = 142

Move 3 (robot moves onto rocks)

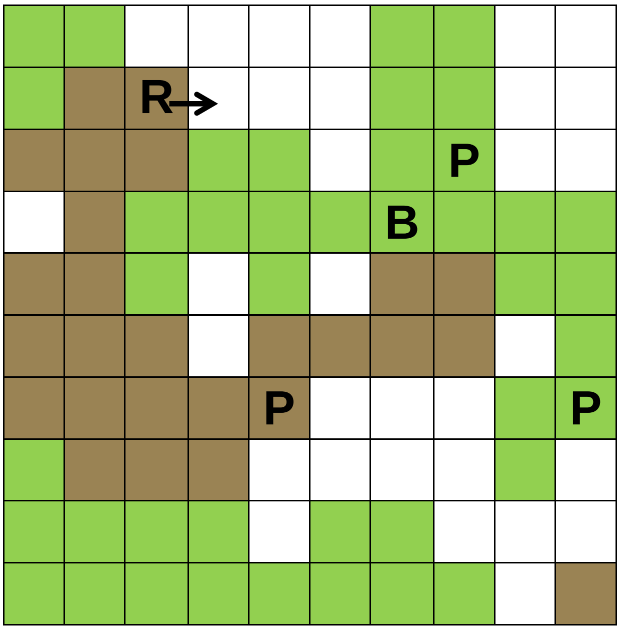


Wheels cost 2 power units
Large passenger bay costs 2 power units

Number of passengers: 1
Power units: 142 – 2 – 2 = 138

Turn over ►

Move 4 (robot moves onto ice)



Wheels cost 3 power units
Large passenger bay costs 2 power units

Number of passengers: 1
Power units: 138 – 3 – 2 = 133

In addition

1. **Your Portfolio**

Remember that you are looking to provide an application that will allow a person to fully play the game.

You are free to use whatever software tools and techniques are available to you.

What your teacher will be looking for and how to provide that evidence for your Portfolio

In preparing you for this unit of work, your teacher will have provided you with more information about the section headings below.

Part 1 – Design of solution

Design of solution (0–9 marks available)
What you must do
• Show an understanding of what the problem involves with reference to the user's needs. • Produce an overview plan that shows how the problem is to be solved. • Produce pseudo code (or suitable alternative) showing the main blocks within the proposed solution.

Part 2 – Solution development

Solution development (0–9 marks available)
What you must do
• Show evidence of an understanding of how the final solution meets the needs of the user. • Produce annotated code that demonstrates an understanding of the programming techniques used.

Part 3 – Programming techniques used

Programming techniques used (0–36 marks)
What you must do
• Show an understanding of the programming techniques used and how the different parts of the solution work together. • Explain/justify the choice of programming techniques used to create a solution that has been coded efficiently. • Show evidence for the purpose and use of data structures. • Show the techniques used (appropriate to the language used) within the code to make the solution robust.

Part 4 – Testing and evaluation

Testing and evaluation (0–9 marks available)
What you must do
• Produce a test plan that shows the expected tests, test data and expected results. • Show that the planned tests have been carried out and provide a record of the actions taken. • Evaluate how the final solution meets the needs of the user.

Turn over ►

2. Organising your Portfolio of work

Your Portfolio is where you keep the evidence that you have produced.

You should imagine that the Portfolio is to be used by another person who is interested in how you produced your solution. It is to help them to do something similar. It is important that you organise work for the Portfolio as shown below.

- You must keep all the work you produce for the organiser in hard copy in a Portfolio (or save your work electronically in folders which you will later copy onto a CD or DVD). Your teacher will have instructed you what to do.

- If you are putting hard copy printouts in your Portfolio make sure that you number each page and fasten it all together. Take your work out of any plastic sleeves before you hand it in to your teacher for marking.

- Each page should have your name, centre number and candidate number clearly shown on it.

- When you have completed this scenario, if you are putting your work on a CD or DVD, put the work for each heading on page **9** of this booklet in a separate folder. Each folder must be clearly named *(for example, 'Design of solution', 'Solution development' etc)*. Inside each folder the work must have a filename (*for example, 'What the problem involves', 'Control statements' etc*) which should be a final version for each heading. The CD or DVD should have your name, centre number and candidate number clearly shown on it. Your teacher will have advised you what to do.

END OF CANDIDATE BOOKLET