**School of Computing Science and Digital Media**

**Coursework Assignment**

|  |  |
| --- | --- |
| Surname | Seddon |
| First name | Liam |
| Matriculation Number | 1705071 |
| Contact phone number | 07503945044 |
| Course + Year | Computing Application Software Development Year 3 |
| Module Co-ordinator | Dr. K. Hui |
| Module Number + Name | CM3038 Artificial Intelligence for Problem Solving |
| Coursework Title | Solving the River Crossing Puzzle |
| Coursework Part | Resit Part 1 of 1 |
| Due Date | 2nd August 2019 (16:00) |
| Feedback Due Date | 2nd September 2019 |

*The University operates a Fit to Sit Policy which means that if you undertake an assessment then you are declaring yourself well enough to do so. Further details are available at:* [*www.rgu.ac.uk/academicregulationsstudentforms*](http://www.rgu.ac.uk/academicregulationsstudentforms)

**Declaration** \*\* *This* ***must*** *be affirmed by adding your name below with the date of submission*

**I acknowledge that by submitting the work, accompanied by this front cover, I take responsibility for the ownership of the submitted work.**

**I confirm:**

* **that the work undertaken for this assignment is entirely my own and that I have not made use of any unauthorised assistance**
* **that the sources of all reference material has been properly acknowledged.**

|  |  |
| --- | --- |
| Student Signature | Liam Seddon |
| Date Submitted | 2/8/19 |

|  |  |
| --- | --- |
| **Marker's Comments** | |
| **Marker** | **Grade** |

## \*\* An extract from the University Regulations

## 6. Academic Misconduct

Refer also to Schedule 3.3 of this Regulation for guidance on this procedure.

6.1 **Academic Misconduct** is defined as any attempt by students to gain an unfair advantage in assessments and examinations. Examples of academic misconduct include plagiarism, cheating, falsifying data, collusion, bribery or attempted bribery, personation or any other activity intended to provide an unfair advantage.

(i)**Plagiarism** is the practice of presenting the thoughts or writings of another or others as original, without acknowledgement of their source(s). All material used to support a piece of work should be carefully referenced and should not normally be copied directly unless as an acknowledged quote. Text translated into the words of the individual student should in all cases acknowledge the source.

(ii)**Cheating** includes:

1. the taking of any unauthorised material into an examination;
2. obtaining copy of “unseen” papers in advance of an examination;
3. communicating or attempting to communicate in any way with another student during an examination;
4. copying or attempting to copy from another student during an examination or in the production of coursework;
5. wilful deception in any element of an examination or assessment.

(iii) **Falsification of data** consists of the misrepresentation of the results of experimental work or the presentation of results from fictitious work.

(iv) **Collusion** is the representation of unauthorised group work as that of an individual student.

(v) **Bribery** is the paying, offering or attempted exchange of an inducement for information or material intended to advantage the recipient in an examination or assessment.

(vi) **Personation** consists of a substitute taking the place of a student in an examination.

**A student who aids and abets a fellow student to commit academic misconduct shall be deemed to have committed academic misconduct and will be dealt with accordingly.**

**Design**

For my design of this River Crossing Puzzle I split my code up into 7 different classes. These are “RiverCrossing” which is the main class, Person State which will hold all the states of the people including, how much each person weighs and who can drive the boat. I also have another class called “PersonAction” this class is to get the person working, tracking them whether they are north or south of the river and is also the class that increments the cost of the program. My fourth class is the “Cost” class, this is just to set the initial state of the cost to 0 each time it’s ran. My fifth class is the “Boat” class this class takes in the max weight for the boat and the capacity. My sixth class is the “AStar” class, this class is the one that runs the search and has my hreustic function in it, without this the program wouldn’t execute properly. And my seventh and final class is the “Person” class, this class sets up getting the weight of the passengers, their name and if they can drive the boat or not.

For my code, I used arrays to store the node data, I also used Booleans, an example of using them is asking whether the person can drive or not, this making it true or false or 1 or 0. I’ve also used integers to store numbers for the cost and also string .

for my heuristic I used the BestFirstSearchProblem, this herustic is admissabe because I think it has managed to reach the most optimal as it doesn’t continue running after the code is done, the hreustic function is only based on the weight in the boat and uses that as the cost. For my test runs ill have them in my appendix.

**Appendix**

run:

~~~~~~~~~~~~~~~~~~RIVER~~~~~~~~~~~~~~~~~~

Dave(30)[false] Claire(50)[true] Betty(90)[false] Adam(100)[true] boat

Moving Claire(50)[true] - Adam(100)[true] - SOUTH\_TO\_NORTH

Claire(50)[true] Adam(100)[true] boat

~~~~~~~~~~~~~~~~~~RIVER~~~~~~~~~~~~~~~~~~

Dave(30)[false] Betty(90)[false]

Moving Claire(50)[true] - NORTH\_TO\_SOUTH

Adam(100)[true]

~~~~~~~~~~~~~~~~~~RIVER~~~~~~~~~~~~~~~~~~

Dave(30)[false] Claire(50)[true] Betty(90)[false] boat

Moving Claire(50)[true] - Betty(90)[false] - SOUTH\_TO\_NORTH

Claire(50)[true] Betty(90)[false] Adam(100)[true] boat

~~~~~~~~~~~~~~~~~~RIVER~~~~~~~~~~~~~~~~~~

Dave(30)[false]

Moving Claire(50)[true] - NORTH\_TO\_SOUTH

Betty(90)[false] Adam(100)[true]

~~~~~~~~~~~~~~~~~~RIVER~~~~~~~~~~~~~~~~~~

Dave(30)[false] Claire(50)[true] boat

Moving Claire(50)[true] - Dave(30)[false] - SOUTH\_TO\_NORTH

Dave(30)[false] Claire(50)[true] Betty(90)[false] Adam(100)[true] boat

~~~~~~~~~~~~~~~~~~RIVER~~~~~~~~~~~~~~~~~~

Nodes visited: 70, Cost: 470

BUILD SUCCESSFUL (total time: 11 seconds)

Run build for river crossing problem.

River Crossing