LEGO SPACE CHALLENGE

Version 1.0
BSc Computing for Games

Dr Michael Scott & Dr Mark Zarb

Introduction

Welcome to the LEGO Robot Olympiad: *Space Challenge*! In small teams (typically, 4–5 people), you will **design** and **build** a LEGO EV3 robot and **write** a program that will command the robot on a simulated space mission.

Computing professionals, of every specialism, are expected to have strong interpersonal skills. Software is not only too complex for so-called 'bedroom programmers' to develop on their own, but it is also created for, and with the support of, a wide variety of stakeholders in almost every domain! It is, therefore, important to develop these skills in an applied and relevant way. Playing with LEGO is a fun and creative way to achieve this. You will, therefore, leverage our LEGO robotics kit to exercise these skills while also developing a basic understanding of software development. Further to this, you will break the ice with your new peers and get to know them.

> *"I didn't fail. I found a thousand ways how not to make a LEGO brick..."*
>
> *— Patrick Henry*

This activity is formed of several parts:

(A) **Build**, as a **group**, a robot that will:
  i. **conform** to the basic robot template;
  ii. **implement** the basic sensors and actuators that are available;
  iii. and *optionally* **demonstrate** your creativity.
(B) **Implement**, as a **group**, a program for the robot so that it will:
  i. **detect** the distance to an obstacle;
  ii. and **avoid** colliding with an obstacle when moving.
(C) **Design and Build**, as a **group**, a draft robot that will:
  i. **implement** more sophisticated and advanced behaviours;
  ii. **demonstrate** your ingenuity **and** creativity;
  iii. **fulfil** at least **one** requirement of the LEGO Space Challenge.
(D) **Design and Build**, as a **group**, a final robot that will:
  i. **revise** any issues raised by your tutor and/or your peers.
(E) **Present**, as a **group**, a practical demo of the robot to your tutor that will:
  i. **fulfil** more than **one** requirement of the LEGO Space Challenge;
  ii. as well as **demonstrate** your interpersonal skills.

Assignment Setup

This assignment is an **induction task**. No course credit is awarded for such tasks, but they are nevertheless great opportunities for learning.

Fork the GitHub repository at:

`https://github.com/Falmouth-Games-Academy/lego-space-challenge`

Download the resources in this repository.

Please also ensure that you have access to: a LEGO EV3 robot kit and the associated cables; as well as a computer with the LEGO Mindstorms EV3 Education-Edition software.

What will you build with your EV3 lego set?

Part A (Session 1)

Part A consists of a **single formative submission**. This work is **collaborative** and will be assessed on a **threshold** basis.

To complete Part A, build your robot and then demonstrate your robot at the end of the session. If acceptable, this will be signed-off.

You will receive immediate **informal feedback** from your **peers**.

Part B (Session 2)

Part B is a **single formative submission**. This work is **collaborative** and will be assessed on a **threshold** basis. The following criteria are used to determine a pass or fail:

(a) The robot is cable of moving;
(b) The robot stops when it approaches an obstacle.

To complete Part B, implement a program for your robot. Then, demonstrate your robot at the end of the session. If acceptable, this will be signed-off.

You will receive immediate **informal feedback** from your **peers**.

Part C (Sessions 3—4)

Part C is a **single formative submission**. This work is **collaborative** and will be assessed on a **threshold** basis. The following criteria are used to determine a pass or fail:

(a) The robot fulfils the requirements of at least one space challenge;
(b) Evidence for emerging programming knowledge **and** interpersonal skills.

To complete Part C, re-design and re-program your robot. Then, demonstrate your robot at the end of the session. If acceptable, this will be signed-off.

You will receive immediate **informal feedback** from your **tutor**.

Part D (Session 5)

Part D is a **single formative submission**. This work is **collaborative** and will be assessed on a **threshold** basis. The following criteria are used to determine a pass or fail:

(a) Enough work is available to hold a meaningful discussion;
(b) Some evidence of programming knowledge **and** interpersonal skills.

To complete Part D, prepare a small-scale demo of the robot. Ensure that you attend the scheduled demonstration session and complete a peer-review.

You will receive immediate **informal feedback** from your **tutor and peers**.

Part E (Session 6)

Part E is a **single summative submission**. This work is **collborative** and will be assessed on a **criterion-referenced** basis. Please refer to the criteria specified at the end of the brief for further detail.

To complete Part E, revise the robot design and its program based on the feedback you have received. Then, prepare a practical demonstration of robot and showcase your robot.

You will receive immediate **formal feedback** from your **tutor**.

Additional Guidance

It is critically important that you do not neglect your individual roles in the development process. Programming in groups means that you work together. Pair programming is a great aid in this respect—switching between driver and navigator. It is a great opportunity to develop your technical communication skills and overcome common misconceptions about programming. It should not, however, be treated as a 'free ride'.

There are no marks or grades, and no course credit for this activity. This does not mean that it is unimportant! Rather, it is a critical juncture where you become a member of our community and make friends. Our aim is to make you a little more comfortable. Settling into university life is not easy. Most people find it hard, and some will find it more challenging than others. Peer support is critical. This induction task is, therefore, an opportunity to have fun and to get to know each other.

Play! Experiment! You are *supposed* to make mistakes! If you are *not* making mistakes then you are *not* learning! The most fun you can have with robots is seeing them do stupid things and spectacularly mess up. Then, trying to work out *why*!

Not everyone on the course will come in with a programming background. Please permit everyone on your team the opportunity it `drive'. Be patient, constructive and helpful. Just because someone isn't as comfortable with computer code now, they will catch up and you may even depend on their experience in the future (e.g., composition, illustration, writing, etc.).

Our emphasis on peer-review and peer feedback does sometimes raise alarm. However, this is standard practice in the industry. Furthermore, the only way to learn how to review code effectively is by actually reviewing code. Critical evaluation is a core learning outcome for the course. Your tutor will guide you through the process and provide advice. With practice, it will become clear what is satisfactory by discussing the quality of work with your peers and your tutor during the peer review sessions.

FAQ

- **What is the deadline for this assignment?**
  Falmouth University policy states that deadlines must only be specified on LearningSpace. Please examine the assignment area where you located this document.
- **What should I do to seek help?**
  You can email your tutor for informal clarifications. For informal feedback, make a pull request on GitHub.
- **Is this a mistake?**
  If you have discovered an issue with the brief itself, the source files are available at:
  `https://github.com/Falmouth-Games-Academy/bsc-assignment-briefs`.
  Please raise an issue and comment accordingly.

Additional Resources

- Guzdial, M.J . and Ericson, B. (2015) Introduction to Computing and Programming in Python: A Multimedia Approach, 4th Edition. Pearson: New York.
- Martin, R.C. (2008) Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall: New York
- http://guide.agilealliance.org/guide/pairing.html
- http://www.pairprogramming.co.uk/
- http://www.pythontutor.com/

Marking Rubric

Criterion	Weight	Refer for Resubmission	Basic Competency	Basic Proficiency	Novice Competency	Novice Proficiency	Professional Competency
Activate Communications	5% ‡	No algorithm has been implemented successfully. The source code does not compile or there are serious logical errors.	At least one algorithm has been implemented successfully. There are many obvious logical errors, more than one of which is significant.	At least two algorithms have been implemented successfully. There are several obvious logical errors, at least one of which is significant.	At least three algorithms have been implemented successfully. There are some obvious logical errors, which are not significant. The brief has been satisfied.	At least three algorithms have been implemented successfully. There are few obvious logical errors, which are cosmetic and/or superficial. The brief has been satisfied.	At least three algorithms have been implemented successfully. There are no obvious logical errors. The brief has been satisfied.
Assemble Your Crew	5% ‡	No insight into the appropriate use of programming constructs is evident from the source code. No attempt to structure the program (e.g. one monolithic function).	Little insight into the appropriate use of programming constructs is evident from the source code. The program structure is poor.	Some insight into the appropriate use of programming constructs is evident from the source code. The program structure is adequate.	Much insight into the appropriate use of programming constructs is evident from the source code. The program structure is appropriate.	Considerable insight into the appropriate use of programming constructs is evident from the source code. The program structure is effective. There is high cohesion and low coupling.	Significant insight into the appropriate use of programming constructs is evident from the source code. The program structure is very effective. There is high cohesion and low coupling.
Free the MSL Robot	5% ‡	There are no comments in the source code, or comments are misleading. Most variable names are unclear or inappropriate. Code formatting hinders readability.	The source code is only sporadically commented, or comments are unclear. Some identifier names are unclear or inappropriate. Code formatting is inconsistent or does not aid readability.	The source code is somewhat well commented. Some identifier names are descriptive and appropriate. An attempt has been made to adhere to thhe PEP-8 formatting style. There is little obvious duplication of code or of literal values.	The source code is reasonably well commented. Most identifier names are descriptive and appropriate. Most code adheres to the PEP-8 formatting style. There is almost no obvious duplication of code or of literal values.	The source code is reasonably well commented, with Python doc-strings. Almost all identifier names are descriptive and appropriate. Almost all code adheres to the PEP-8 formatting style. There is no obvious duplication of code or of literal values. Some literal values can be easily "tinkered" in the source code.	The source code is very well commented, with Python doc-strings. All identifier names are descriptive and appropriate. All source code adheres to the PEP-8 formatting style. There is no obvious duplication of code or of literal values. Most literal values are, where appropriate, easily "tinkered" outside of the source code.
Launch the Satellite	5% ‡	No creativity. The work is a clone of an existing work with mere cosmetic alterations.	Little creativity. The work is derivative of existing works, with only minor alterations.	Some creativity. The work is derivative of existing works, demonstrating little divergent and/or subversive thinking.	Much creativity. The work is somewhat novel, demonstrating some divergent and/or subversive thinking.	Considerable creativity. The work is novel, demonstrating significant divergent and/or subversive thinking.	Significant creativity. The work is highly original, with strong evidence of divergent and/or subversive thinking.
Return the Rock Samples	5% ‡	GitHub has not been used.	Source code has rarely been checked into GitHub.	Source code has been checked into GitHub at least once per week. Commit messages are present. There is evidence of engagement with peers (e.g. code review).	Source code has been checked into GitHub several times per week. Commit messages are clear, concise and relevant. There is evidence of somewhat meaningful engagement with peers (e.g. code review).	Source code has been checked into GitHub several times per week. Commit messages are clear, concise and relevant. There is evidence of meaningful engagement with peers (e.g. code review).	Source code has been checked into GitHub several times per week. Commit messages are clear, concise and relevant. There is evidence of effective engagement with peers (e.g. code review).
Secure Your Power Supply	5% ‡	At least one part is missing or is unsatisfactory.	Submission is timely. Enough work is available to hold a meaningful discussion. Clear evidence of programming knowledge and communication skills. Clear evidence of reflection on own performance and contribution. Only constructive criticism of pair-programming partner is raised. No breaches of academic integrity.				

Criterion	Weight	Refer for Resubmission	Basic Competency	Basic Proficiency	Novice Competency	Novice Proficiency	Professional Competency
Initiate Launch	5% ‡	At least one part is missing or is unsatisfactory.	Submission is timely. Enough work is available to hold a meaningful discussion. Clear evidence of programming knowledge and communication skills. Clear evidence of reflection on own performance and contribution. Only constructive criticism of pair-programming partner is raised. No breaches of academic integrity.				
Creative Flair & Ingenuity	25% ‡	At least one part is missing or is unsatisfactory.	Submission is timely. Enough work is available to hold a meaningful discussion. Clear evidence of programming knowledge and communication skills. Clear evidence of reflection on own performance and contribution. Only constructive criticism of pair-programming partner is raised. No breaches of academic integrity.				
Collegiality	40%	At least one part is missing or is unsatisfactory.	Submission is timely. Enough work is available to hold a meaningful discussion. Clear evidence of programming knowledge and communication skills. Clear evidence of reflection on own performance and contribution. Only constructive criticism of pair-programming partner is raised. No breaches of academic integrity.				