Acquiring Skills for Academic Success through Project-Based Learning in First-year Engineering

Brian Peach, Darlene Spracklin-Reid, Steve Bruneau
Memorial University of Newfoundland
bpeach@mun.ca, darlenesr@mun.ca, sbruneau@mun.ca

***Abstract** –*

This paper presents the development of a ballista-themed project that comprises part of the pilot of a redeveloped first-year engineering course at Memorial University. The course aims to teach students to "think like an engineer" and provide them with skills and tools to support them throughout their engineering education. Students learn to use tools such as Microsoft Excel and Matlab through the use of meaningful, yet accessible, technical assignments. Students are acquiring requisite engineering knowledge while developing a skill set that will support further learning. The ballista project requires students to design a simple numerical computer model linking the launch and trajectory of a projectile in order to calculate the launch settings required to hit a series of targets. In preparation for the project, instruction is offered on topics including the conservation of mechanical energy, projectile motion and numerical integration. Students compete using an assembled, moderately sized ballista prototype to launch wooden spheres at a castle-like structure. Student use their computer models along with experiment-based model corrections to account for discrepancies in theoretical and actual trajectories.

Keywords: Project, Design, First-Year Engineering, Engineering Tools.

1. INTRODUCTION

A new course module named "Thinking like an Engineer" (TLE) was introduced in Memorial University's engineering program beginning in the fall of 2015. The broad aim of the TLE module was to raise engagement levels of first-year engineering students with project based activities that involved practical applications of engineering and physics. The use of this type of active learning in engineering courses improves traditional teaching by promoting students' participation and engagement [1]. Students develop skills and facts as they progress through the process of solving the problem with a design approach [2].

Using problem-based learning helps students develop their skills and acquire a deeper understanding of theoretical knowledge. Many students come to university knowing how to apply a procedure to a set of variables to get a result, but they often do not understand what the result means. Our challenge as engineering educators is to enhance this type of procedural learning with experiences that result in conceptual learning and skill acquisition [3].

Detailed planning for the TLE began two-months before the start of the fall semester and decisions were made to create 3 computer based case studies to discover how to use Microsoft Excel and Matlab in real-world engineering problems such as engine physics, hydroelectric energy management, and DC circuit analysis. When students engage in problem-based learning that incorporates relevant problems, it provides context and motivation for all the activities that follow [4].

A final course project would incorporate some physical device or system whose function could be modelled with a computer using these newly acquired modelling skills. Ultimately, the selected final project involved modelling and operating a ballista-like device that uses a spring loaded mechanism to launch wooden balls at a structure.

The challenge for the students was to build an appropriate theoretical model based on their understanding of projectile motion and conservation of mechanical energy, which are common topics introduced in high school and first year university physics. In order for first-year students to achieve subsequent academic success and become competent engineers, they need to be introduced to problem solving by a series of graduated problems, be able to develop a realistic mathematical model and be able to make judgements as to the validity of their results [3].

Students were expected to test their theoretical models and, to resolve any discrepancies, they were instructed to use their judgement to apply manual modifications to model parameters such as masses, spring constants, or drag coefficients to force the calculated model to match the actual performance. Students would demonstrate a successful model through a competitive demonstration in which they are presented with targets at x-y locations that they would hit by selecting appropriate launch settings.

2. BALLISTA AND PROJECTILE PHYSICS

2.1. Ballista Design

The ballista consists of resting the wooden projectile onto a firing bar that passes through the center of a barrel. The bar is spring loaded and the ballista is energized by pulling back on a twine cord. Loosening of the upper wing-nuts allows the tube to slide along a track to set the launch angle with markings at 5° increments to use as a guide for setting the launch angle. The ballista can be attached to a firing box using strong shop magnets for stability. This set-up is shown in Figure 1.

Fig. 1. TLE Ballista System

The design of the ballista was aimed at achieving the following objectives:
(i) The ballista design had to depend on reasonably simplified physics to be accessible to first-year students.
(ii) Firing of the ballista was reasonably accurate and repeatable such that students did not resort to blaming any competition results on "luck"
(iii) Each team was supplied with their own ballista so they may practice. To this end, consistent manufacturing quality of each ballista was preserved.
(iv) For the students, the focus of the project was not about design but more-so around the computer modelling of the ballista. For this reason, a complete system should be offered; however, to avoid stifling all design creativity however, some degree of student modifications to the system were permitted

2.2. Ballista Physics

Students were instructed to use principals of conservation of energy to determine projectile launch velocity. The initial energy state is the fully loaded position in which there is only energy stored in the spring. The final energy state is the point at which the bar reaches the upper stops and the projectile is lobbed out of the tube. At this state, there is kinetic energy in the moving components, gravitational work from the elevation change, and left-over spring energy from the preloaded stretching of the naturally closed system. Working through the associated equations, students can obtain a launch velocity equation:

$$v = \sqrt{\frac{4k(L^2 + 2Lp)}{m_{system}} - 2g\sin\theta_0}$$

Students were then expected to know or determine system parameters such as spring constants k [M/T^2], the preload stretch p [L], total system mass m_{system} [M] and gravitational constant g [L/T^2] to define how the launch velocity v [L/T] is a function of the pullback length L [L] and launch angle θ_0 [rad]. Students were also encouraged to consider the reasonableness of their launch model by neglecting the less important gravity and preload effects with a more simplified expression:

$$v = 2L\sqrt{\frac{k}{m_{system}}}$$

These solved equations were not directly handed to the students and were instead discovered by working with the energy equations in a classroom tutorial setting. While this may appear trivial from an onlooker, the students struggled with this task. Most students have never written equations about a physical system in their hands – they are only familiar with seeing labeled sketches and solving word problems with key phrases that point them to the appropriate section of their formula sheets. It was also a surprisingly significant struggle for them to determine appropriate values of k, p, and m_{system} – they expected the numbers to be conveniently available to them on paper! In the end, estimated values were provided, but it was stressed to the students that they must devise a plan for measuring exact values tailored for their individual ballista units.

2.3. Projectile Physics

In high-school and first-year physics students receive instruction on ideal 2D projectile motion where gravity provides a constant vertical acceleration and there is no acceleration in the horizontal plane – reasonable for dense/heavy objects at low speeds where air resistance is negligible. The light wooden balls used for this project will be shot at speeds such that air resistance exerts an appreciable effect resulting in non-constant accelerations that cannot be resolved by analytical calculus techniques accessible to the students. It is for this aspect of the project that students must develop a computer program with basic numerical integration and in an algorithm as follows:

(i)   Set a pullback length and launch angle and compute the initial launch velocity and resolve it into x-y coordinates.
(ii) Set the initial x-y position of the projectile which will be dependent on a desired reference frame for measuring target locations on the castle.
(iii) Use the velocity to compute the drag force on the projectile.
(iv) Use the trajectory angle to resolve the drag force into x-y components for which x-y accelerations can be obtained using Newton's Second Law.
(v)   Use the rectangle rule with a fine time step to update the x-y velocities of the projectile based on the x-y accelerations.
(vi) Use the rectangle rule with a fine time step to update the x-y positions of the projectile based on the x-y velocities.
(vii) Use the new x-y velocities to determine a new drag force and trajectory angle and repeat steps (iv) to (vii).

With this algorithm in force, the projectile trajectory can be easily plotted in x-y space. Students were required to execute this algorithm in Microsoft Excel and Matlab.

2.4. Shot Solver

At this stage, students are only equipped with a means of plotting the trajectory of prescribed shot settings – they have not yet developed any functionality for inputting a desired target and having the computer solve for the required shot settings. At the current stage of their education, the students do not appear to have any appreciation for the challenges associated with solving a forward problem versus solving an inverse problem. For the most part, their experience with math and physics is centralized around simple algebraic equations which can be manipulated to explicitly solved for a singular unknown value. They become frustrated when it is explained that their program cannot be effectively executed backwards, using a desired position to inverse calculate through the drag force algorithm to determine a required initial speed that can be used to find a pullback length and launch angle.

The most basic strategy they are instructed to use is to impose a target onto their trajectory plot and to then guess and check pullback lengths or launch angles until the projectile visually intersects the target. The observed typical response from the first-year students is that this is the 'wrong' way to solve this problem, like it is somehow 'cheating'; however, by the end of the project most students abandon any more sophisticated techniques and decide that they like this approach once they realize that with only a few guesses they can hone in with a good shot.

In Microsoft Excel, the students are encouraged to create two cells called *x-target* and *y-target* and use a VLOOKUP function with *x-target* in the x-y position data to output the corresponding *y-position* of the projectile with the current launch settings. They can then create a cell called *miss* which is the difference between *y-position* and *y-target*. If *miss* is less than say 2cm, it means that the projectile will hit the target. The students can then use the Microsoft Excel Solver add-in to adjust either the launch angle or pullback length to minimize *miss*. This technique is simple enough that students can use it and it introduces them to the VLOOKUP function and Solver add-in which they believe are 'pretty cool'.

Using Matlab, the instructional focus of the course has been restricted to using vector variables, if-else statements, and for loops and, therefore, the students were expected to develop a program using only these features. To add a shot-solver onto the projectile trajectory algorithm from section 2.3, students were encourage to run the program through a set of launch angles or pullback lengths (say 10cm to 20cm in 0.1cm increments) and save the loop index of the shot that has the lowest *miss*. Students who develop this functionality appeared to prefer the Matlab program over the Excel program, however, more students use the Excel program simply because it feels less foreign to them. Students were also casually shown that Matlab can be used to generate a graphical user interface application in which they can show the trajectory and have a slider bar to adjust the pullback length and watch the trajectory update as shown in Figure 2. They could also have fields to enter targets which could be coded to auto-populate on the plot. One student team thought this was particularly fascinating and successfully completed this challenge. It was apparent that this enhanced their appreciation of engineering.
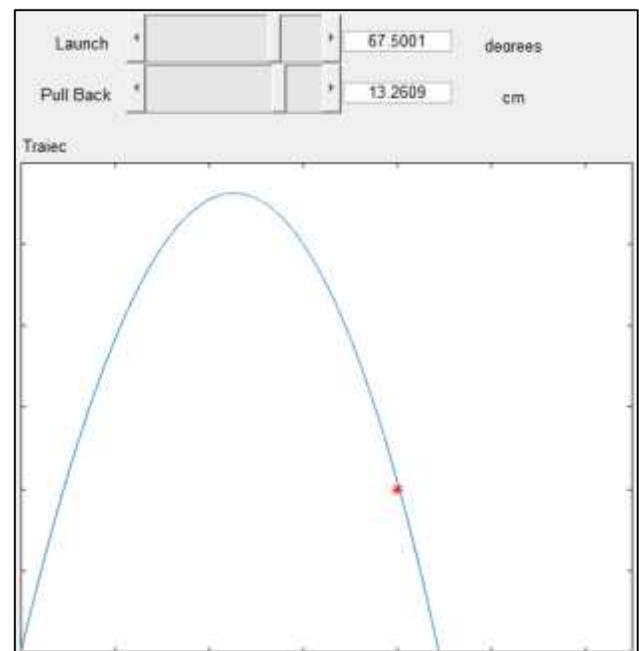

Fig. 2. One student team followed a suggestion to create a Matlab GUI App incorporating a shot-solver technique of marking a target and guessing an angle/pullback to hit it.

3. COMPETITION

3.1. Castle Target Design

A set of targets were themed as elements of a castle, including flags, doors, turrets, and standing pieces as shown in Figure 2. Most targets were worth 5 points; however, sinking a projectile into a turret was worth 10 points and hitting the inner keep door was worth 20 points (which can only be reliably accessed by shooting through a small window). In total, students were permitted 10 shots however they were only given 5 minutes to perform all set-up, measurement, and shooting (averaging 30 seconds per shot). Given this time constraint, students were aware that they had to develop and practice all strategies before the competition and develop any contingency plans. Students were also offered a x3 bonus on their first 3 shots and a x2 multiplier on shots 4 and 5 to encourage them to carefully plan/practice without relying on in-field corrections.

Fig. 3. TLE Castle of Cram-a-lot

3.2. Launch Base

A primary launch base was placed somewhere between 12 and 15 feet from the castle and secondary base was placed somewhere between 6 and 9 feet from the castle as depicted in Figure 4.
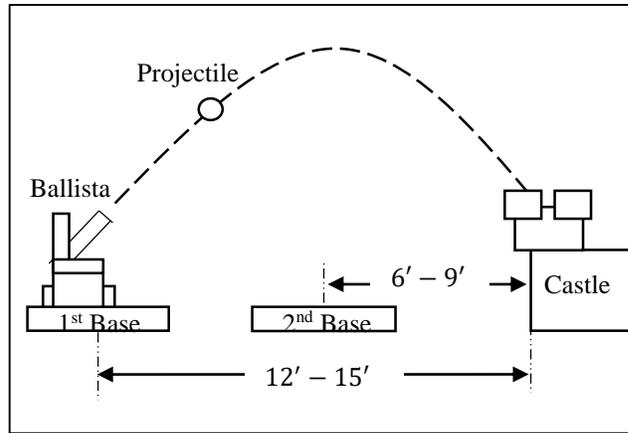

Fig. 4. Competition Set-up

The secondary base could only be used for the last 5 shots however the primary base could be used for all 10 shots. It was very interesting to note that not a single team had planned to do any of their shots from the closer base even though it would be much easier to hit the targets! The psychology of this would be interesting to study, however the general reason when asked was that students had only practiced at a far distance, therefore they did not want to 'risk' shooting at a closer distance. During the competition, the close base was used by some, but only as a reactive measure to a remarkably poor performances shooting from the far base. During the competition, the instructor prompted some struggling teams to move up close which resulted in a series of successful shots and an 'a-ha' moment in realizing how much smarter this was.

3.3. Final Competition Strategies

Another general observation was that by the time of the competition, the majority of students distrusted their computers and favored an intuition-based strategy for operating their ballista (often to their detriment). Their attitude was that the computer program was just something that was needed to satisfy the requirements of the final report. Some members of the team would practice random shooting and attempt to memorize where a shot goes from a given power and angle without performing any calculations. It was communicated and expected that students would spend practicing efforts to adjust system parameters so that any final computer models would accurately predict actual performance. One may consider forcing students to demonstrate how they calculate each shot, however, perhaps they will learn a more valuable lesson from executing an inferior strategy.

4. PROJECT DELIVERY

4.1. Project Timeline

The delivery of the project comprised of a 6 week period with formal 2-hour in class preparation sessions on weeks 1 and 3 followed by the grand tournament in week 5. Throughout the project period students had access to the castle so that they could test their programs and practice with their groups. This timeline is represented in Figure 5.

Week 1
- In Class: Introduction, hand-out ballistae, instruct launch modelling
- Lesson: Analytical Modelling

Week 2
- Out of Class: Establish team roles, devise methods to measure parameters
- Lesson: Appreciate fine tactile details

Week 3
- In Class: Instruct drag modelling, target measuring, and shot-solving
- Lesson: Numerical Modelling

Week 4
- Out of Class: Test and practice
- Lesson: Appreciate and adjust for actual/theoretical discrepancies

Week 5
- In Class: Grand Tournament
- Lesson: At the end of the day, calculations are used for something real

Week 6
- Out of Class: Finish Report
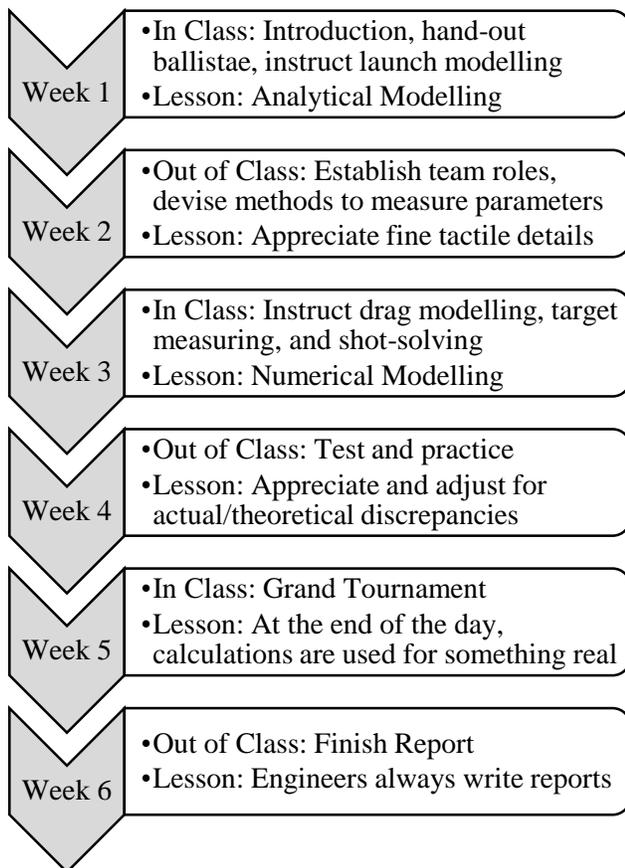- Lesson: Engineers always write reports

Fig. 5. Activity Timeline

The authors would encourage any first-year engineering educator to borrow this timeline and invent their own project that embodies a combination of accessible analytical-numerical-physical elements. For example, consider a robot competition, an egg drop competition, a Kub Kar rally, etc.

4.2 Final Reporting

For their final report, students were expected to include the following elements:
(i) Executive Summary, Introduction, and Team Roles.
(ii) Strategy – including details such as preferred shot solver technique, how the team decided to measure

their shots, how will power and angle be controlled, and whether any device modifications will be used.
(iii) Excel/Matlab Development – including thorough details of how the programs function and the degree of advantages and disadvantages in each.
(iv) Experimental Testing.
(v) Tournament Predictions, Outcome, and Conclusions
(vi) Appendices including computer formatted sketches and typed engineering calculations of the launch velocity derivation and projectile trajectory algorithm development.

In general, students have had minimal exposure to preparing technical analysis-based reports on a computer and therefore had received training with the earlier case study components of the TLE module. By the end of the course, the improvement of the students reporting abilities is evident and the quality of the final reports were quite impressive.

5. CONCLUSION AND RECOMMENDATIONS

Engineering students must be effective problem solvers to be successful in their courses, and to be competent engineers after graduation. Problem-based learning can help students to become effective problem solvers by giving them experience with making sensible assumptions and applying design techniques [3]. It can also facilitate the development of the types of skills that are required to be successful in their program of study.

First-year problem-based engineering courses have been shown to emphasize teamwork, engineering problem solving, documentation, data collection and analysis, use of appropriate computer tools, and communication skills [5]. The Ballista Project was incorporated to help students achieve these outcomes, and to improve engagement of students in a first-year course. Anecdotally, students appeared to be more engaged and motivated, had improved communication skills and had acquired analytical skills through the use of Matlab and Excel. Further study is required to determine if, and the degree to which, students have improved in the areas.

Lastly, based on this anecdotally perceived success, it is recommended that it would be highly beneficial to students if first year engineering educators consider developing similarly natured projects to administer in their respective engineering programs. The students synthesized analytical and numerical modelling knowledge, hands on skills, computer programming skills, report writing skills, and teamwork to grow their appreciation for engineering at an early stage in their development.

Acknowledgements

References

[1] I. Estévez-Ayres, C. Alario-Hoyos, M. Pérez-Sanagustín, A. Pardo, R, Crespo-García, D. Leony, and C. Delgado-Kloos, "A Methodology for Improving Active Learning Engineering Courses with a Large Number of Students and Teachers through Feedback Gathering and Iterative Refinement," *International Journal Of Technology And Design Education,* vol. 25, no. 3, pp. 387-408, 2015.

[2] B. D. Jones, J. R. Chittum, S. Akalin, A. B. Schram, J. Fink, C. Schnittka and C. Brandt, "Elements of Design-Based Science Activities That Affect Students' Motivation," *School Science And Mathematics,* vol. 115, no. 8, pp. 404-415, 2015.

[3] M. Peters, "Using Cognitive Load Theory to Interpret Student Difficulties with a Problem-Based Learning Approach to Engineering Education: A Case Study," *Teaching Mathematics And Its Applications,* vol. 34, no. 1, pp. 53-62, 2015.

[4] H. S. Barrows and I. L. M. Tamblyn. Jones, *Problem-Based Learning: An Approach to Medical Education.* New York, NY: Springer, 1980.

[5] F. Aloul, I. Zualkernan, G. Husseini, A. El-Hag, and Y. Al-Assaf, "A Case Study of a College-Wide First-Year Undergraduate Engineering Course," *European Journal Of Engineering Education,* vol. 40, no. 1, pp. 32-51, 2015.