# Student-Developers and Teacher-Publishers

## A Story of Unintentional PBL Success

by Rob Dieterich
President & Founder, Skyboy Games LLC

This article presents the experiences of the author teaching a pilot educational summer program done in collaboration between George Mason University's Mason Game & Technology Academy (MGTA) and Envision. In this program, the author and a team of teaching assistants taught hands-on skills in computer game design to high-school students during the summer of 2016. While initially struggling with an untested curriculum, the teaching team found success by pivoting to a project-focused strategy. Under this project-focused strategy, teams of high school students worked on self-defined projects while the teaching team played the role of producers, guiding students through their projects. This paper then examines how the experience of the author compares with existing literature on Project-Based Learning (PBL) and the degree to which the results of the pilot MGTA/Envision course compare to the expected outcomes of PBL.

# Student-Developers and Teacher-Publishers

A Story of Unintentional PBL Success

## Intro

From July 11, 2016 to August 5, 2016, I had the opportunity to join the Mason Game & Technology Academy, or MGTA, as a lead instructor. With the organization, I taught two 2-week sessions of a debut course in computer game design to high-school students. This course, called the MGTA Honors Program, was offered as part of MGTA's first collaboration with Envision, an organization that offers "life-changing career exploration and leadership experiences for high-aspiring students" (Envision EMI, n.d.).

In an effort to add direction to this new course using an untested curriculum, a team of teaching assistants and I implemented a system by which teams of students would propose game projects to us which the students would then complete over the course of two weeks with teacher support. This framework was designed to mirror a typical relationship between developers and publishers in the games industry. In this relationship, developers (firms that design and construct video games) will often pitch a concept to a publisher (a firm that funds, markets and distributes games.) Upon receiving approval, also known as greenlighting, developers will construct a video game while taking feedback from the publisher (Wilson & Zackariasson, 2007). During development, a series of milestones are often defined to delineate which game features should be completed by set dates. Additional funding may be tied to these milestones. These milestones can also serve as focal points for contract renegotiation as project / developer / publisher needs see fit. In our classroom model, we established a series of scheduled milestones (called first-playable, alpha, beta, and final) during which each student-developer team would meet with a team of teacher-publishers to receive feedback and guidance on their projects. We used these milestones to set explicit expectations in terms of when certain features were expected to be implemented. We also used these meetings as opportunities to course-correct and adjust projects that were veering in ineffective directions.

In adopting this approach, we found great improvement in terms of student motivation and focus. Once students were focused on projects of their own design, we found it much easier to maintain classroom discipline, especially when a milestone deadline loomed near. It was only later that I happened upon project-based learning (PBL) as a model that potentially matched the approach we had used. It was this late discovery prompted this paper to explore outcomes from research literature on PBL and compare them to the outcomes of the PBL-like class taught by this author.

## Was it PBL?

In his review of PBL research, (Thomas, 2000), instead of providing an exhaustive definition of PBL, establishes criteria by which one can judge whether a project is an instance of PBL. Those criteria are: (1) centrality ("projects are central, not peripheral to the curriculum"), (2) driving question ("projects are focused on questions or problems that 'drive' students to encounter (and struggle with) the central concepts and principles of a discipline"), (3) constructive investigations ("projects involve students in a constructive investigation"), (4) autonomy ("projects are student-driven to some significant degree"), and (5) realism ("projects are realistic, not school-like"). By these criteria, the MGTA Honors Program can be defined as PBL. The game projects simulated the developer-publisher relationship in the games industry (realism), were student-led (autonomy), were focused on having the students find solutions to the particular problems presented by their design (driving question and constructive investigations), and were the central component of the course (centrality).

In an article written for the Buck Institute of Education, (Larmer et al., 2015) list similar criteria as essential elements for PBL project design. In addition to criteria similar to (Thomas, 2000), they add "student voice & choice", "critique & revision", and "public product".  The MGTA Honors Program course fulfilled these elements by having the students define and hold literal ownership of their project and by having the students present their work to each other at the end of every week.

By simulating the developer-publisher relationship common in the games industry, the MGTA Honors Program also integrated well with Envision's established goal of career exploration. This allowed students to explore the developer-publisher dynamic in a safe and time-efficient way.

## About MGTA

The Mason Game & Technology Academy (MGTA) is an organization that offers educational programs in computer game design to students age 9-18 in collaboration with the Virginia Serious Game Institute and the Mason Computer Game Design Program. MGTA operates in the Fall, Spring, and Summer offering computer game design-focused courses that integrate STEM (science, technology, engineering, and mathematics) skills. MGTA leverages its close association with GMU to produce its courses. Mason faculty develop and teach MGTA courses and are often assisted by university students majoring in Computer Game Design or Computer Science. It is worth noting that the artistic nature of computer game design combined with the technical requirements makes a strong argument for MGTA representing an ideal STEAM (science, technology, engineering, arts, and mathematics) program.

During the summers, MGTA offers intensive programs that run for one or two weeks. The MGTA Honors Program which I taught, was one of these two-week programs. This program was launched in partnership with Envision.

## About Envision

Envision is a collection of youth education and leadership programs. It is run by Envision EMI, LLC, a for-profit education company founded in 1985. Envision offers a variety of education programs for elementary to college-age students. The programs are short, intensive courses ranging from 5-14 days and include topics such as medicine, law, journalism and youth leadership.

In thirty years of operation, Envision boasts a total attendance of over 800,000 students. Student applicants are selected based on having a GPA of 3.0 or greater and having demonstrated leadership potential through extracurricular activities. Student applicants must also write essays related to future career aspirations.

## Structure of the MGTA Honors Program

The MGTA Honors Program, executed in Summer 2016, was a summer program focusing on teaching high-school students the basics of computer game design using 3D workflows and tools. Each session of the course lasted two weeks with class meetings every weekday from roughly 9 AM to 5 PM. Students would either be day students commuting from the local area or residential students staying at the GMU dormitories. Commuting students would either be signed up for 1 week or 2 weeks of participation. Residential students were always signed up for 2 weeks. Our section was generally composed of two-thirds residential students with only a handful of commuting students not returning for the second week. We also didn't have new students join us for the second week in any given session, so we were able to retain a high level of student cohesion between the first and second weeks of our sessions. I taught two sessions of the MGTA Honors Program for a total of four weeks working for MGTA. There were three sections of the MGTA Honors Program run this year, each with their own team of teachers and assistants. Each class had about 30 students per session for a total of around 90 students per session.

The teaching team for each classroom was made up of a lead instructor assisted by several instructors and teaching assistants. As a graduate student with significant games industry experience, I was had the privilege to be classified as a lead instructor. On my team, I had three other teachers / teaching assistants during the first session and four during the second. Lead instructors, were (with the exception of myself) made up of faculty from GMU's Computer Game Design program. Instructors were generally recent graduates of the Computer Game Design program while instructor assistants were undergrad GMU students.

## Pivoting to a Project-Based Classroom in the First Session

Before opening MGTA for summer courses, all the staff met for an orientation session. In addition to handling logistics related to payroll and policies, it also served as the first opportunity for the collected MGTA Honors Program staff to meet and discuss their approach for the new program. It was during this meeting that the team assigned development of the first week and second week syllabi to particular team members. These team members took the time to develop syllabi that they then sent out to all of the team members for review in the weeks leading up to the summer courses.

While the MGTA Honors Program team had worked to prepare for teaching ahead of time, there was still a lot of work that the various teaching teams had to do on-the-fly. As the old saying goes: "No plan survives first contact with the enemy." As is natural for a new program, no one knew exactly what to expect going into the classroom. In the beginning, my team experienced significant difficulty trying to apply the original syllabus to our class. Student skill levels varied significantly which made it hard to apply one-size-fits-all lectures and activities in a way that significantly engaged a majority of the class. While I can only speak to the experiences within my own classroom, I did hear about similar trouble being experienced by the teaching teams in the other classrooms.

During our first day, we focused the class on creating paper prototypes eschewing the computer-based aspect at first in order to focus on game design fundamentals. Our primary activity of the day centered around iterations of 1000 Blank Cards, a cooperative game design activity in which students create rules for a card game on-the-fly. We had students split into four groups and play multiple iterations of 1000 Blank Cards, each time having them refine rules in order to make their game more playable. Later, we had the various teams switch decks and play each others' games in order to cross-pollinate ideas and to see how the interpretation of rules can change between different groups.

While starting with paper prototyping exercises for making games is a time-honored tradition, I suspect our adherence to paper prototyping for the entirety of the first day had unforeseen consequences regarding student discipline during the first session. Because we eschewed official computer usage for the first day, our students' first experiences touching the classroom computers ended up being for playing games during some break time (a situation that was further exacerbated when we found out at the last moment that lunch would be delayed by one hour.) I suspect that, because the student's first experience with the class computers involved playing games on them, we set a precedent for playing games on the computers that overrode the official use of the computers (running the software needed to make games.) As a result, we found ourselves spending significant effort to coach students away from playing games during the course of the entire first session. When not watched, many students would quickly start running a game instead of working on their projects or following lectures. During the second session, we took steps to avoid this early introduction to playing games on the class computers and had significantly less trouble getting students off games as a result.

During the second day of the first session, our team focused on tutorials for various game asset creation tools. For these tutorials, we set up the expectation that all students would follow along with the instructor-led tutorial and recreate the same results at their own workstations. This approach was moderately successful with some students gaining new skills that they would demonstrate later in the session. That being said, we still had many students who did not bother to follow along with the demonstration. Some students already knew the particular material being taught and quickly started doing something else during the demonstration. Other students had focus problems and resisted teacher efforts to get them back on track. Even for students who did follow along, the tutorial often had to pause to help students who were having trouble following along. While we had met with more success on the second day compared to the first, our teaching team wasn't quite satisfied with how the classroom was operating.

On the third day, we decided to start splitting the classroom into project pairs (or trios where necessary) in order to focus the students' learning efforts. That morning, we asked our students to split into pairs and come up with a pitch for a game they would be making for the rest of the session. Once they had their pitch ready, each team would then take turns presenting the idea to a pair of instructors playing the role of a faux publisher for their game. We didn't place any particular limits on the kind of game to be pitched. As the publishes, the teaching team was able to steer the pitched games in directions that would help them be completed in the time allowed. In particular, we would steer students away from particularly time-consuming features like network play and complicated melee combat. We would also prioritize tasks for the student teams so as to focus their efforts and prevent endless iteration on non-essential features. By allowing the students to pitch their own ideas, most of the students became instantly more invested in their classroom efforts. We were surprised by how much easier it became to manage the class once the student teams had a concrete goal to focus their efforts.

While students were more focused once they had goals of their own design declared, that's not to say we didn't have some discipline and organizational problems. We still had to coax students away from playing games during class time. With the student pairings though, we did have some groups begin to police themselves and discourage time-wasting by their partners. With the disparate set of pitched game ideas, we also dedicated nearly 100% of teacher effort to floating between groups and helping them implement necessary features. This approach required all of the teaching team to use their debugging and game development creativity to their fullest in order to keep the student teams on track. While this approach worked well for us, I attribute it almost entirely to the problem-solving skills of my teaching team. Had they been less capable game developers, I doubt this approach would have worked nearly as well as it did.

Once we had students focused on their projects, it was a natural extension to have them present their projects to each other every Friday. We found that having an approaching deadline in which they had to show their work to their peers helped motivate some of our less disciplined groups. In such cases, those groups would suddenly put in a magnitude greater effort in face of a looming deadline. Given this and the overall success of the student project pitching solution, we opted to largely discard the established syllabus and shift further toward a project-oriented approach for the second session.

## Completing the Transition to a Project-Based Classroom

Based on the results of the first session, our teaching team opted to double-down on the project-based approach for managing our classroom. We began by establishing project teams on the first day of the session. Next, we set a series of deadlines the students had to meet modeled roughly on the milestones that may be set in a typical developer-publisher relationship. Finally, we eliminated mandatory follow-along tutorials in favor of specific subject-matter lectures that students could opt-to follow as they deemed appropriate for their projects.

Since we saw a significant increase in student discipline after establishing project teams in the first session, we opted to establish project teams as early as possible for our second session. Like the first session, our first major activity was still 1000 Blank Cards[1]. This time around, we shortened the activity and capped off the morning by having the students split into project teams and begin working on pitches. To help direct their efforts, I presented an example of a single-page pitch document to the class as a model for how they might present a game idea. We encouraged the students to create these one-page documents on the workstations in an attempt to establish a precedent of using the classroom computers for project work.

After lunch, we had teams take turns presenting their idea to a two-person team of teachers playing the role of the publisher. We called this the Greenlight meeting and used it to vet and tune the initial ideas presented by the student teams. We didn't place any particular limits on genre for the pitch. Because our team had a fairly broad amount of programming experience, we weren't particularly worried about facing implementation problems that could stump the team. I think a team with less general implementation experience would likely benefit from limiting available project genres in order to limit the breadth of the problem domain. Similarly, we allowed teams to choose either Unity 5[2] or Unreal Engine 4[3] as we had teachers confident in both engines.

In addition to the Greenlight deadline, we established several other deadlines over the course of the session. These deadlines consisted of a presentation of their projects to the class on each Friday and three private meetings between the team and the "publisher." Following along with the developer-publisher narrative we named the deadlines: First-Playable Check, First-Playable Presentation, Alpha Check, Beta Check, and Final Presentation. As lead instructor, I was present for each of the private meetings. I rotated the other instructors in as my partner to allow each of them the experience of

---

[1] See https://en.wikipedia.org/wiki/1000_Blank_White_Cards

[2] https://unity3d.com/5

[3] https://www.unrealengine.com/

discussing projects with the student teams. We used the private meetings to set expectations for what we wanted each student team to implement in their game for the next deadline. This allowed us to quickly address teams that were floundering as well as suggest additional implementation areas for teams that were performing ahead of expectations. We found most of the teams to be proactive in identifying trouble areas and suggesting new features to implement. The frequent build checks also allowed us to practice the process for creating builds in both Unity and Unreal. Having students acclimated to producing builds for presentation was helpful in ensuring the smooth operation of the Friday project presentations.

Because we had students begin working on their project almost immediately, we ended up shifting our classroom's teaching style almost exclusively to on-the-fly consultations between teachers and particular project teams. We did conduct some tutorial lectures to the entire class, but we didn't require the entire class to follow along. Instead, we encouraged the student teams to decide for themselves whether they would benefit from listening to a particular lecture. In general, we had roughly one-quarter to one-third of the class listening to any particular lecture while the rest of the students were focused on their particular projects.

As most of our teaching to the students was composed of one-on-one interactions between teachers and student teams, our teaching approach relied heavily on the creativity and flexibility of the individual members of the teaching team. As an industry veteran, I was confident in my ability to solve almost any implementation problem the student teams encountered. I was pleased to find that my teaching team was similarly capable in addressing student needs. The teaching team was willing to tackle novel implementation challenges and use all the tools at their disposal to help students solve problems. As we had teachers fluidly moving between groups, there was always the danger that teachers might undo the work done by other teachers, but we didn't run into particular problems in that respect. Teachers were generally respectful of the solutions established by other teachers and worked with, not against, code designed by others.

## Expected Results of PBL

Research on the outcomes of PBL have found that students exhibit a positive attitude towards the methods used in the project (Kaldi et al., 2011) as well as the material studied (Tseng et al., 2013). In addition, (Kaldi et al., 2011) found that students "initiated interactions that often led to clarifications and exchange of ideas with other group members". This can be taken as an indicator for student initiative likely encouraged by the student-led nature of PBL. Publications such as (Ergül & Kargın, 2014) and (Baran and Maskan, 2010) found improvements in student performance versus control groups in science education, indicating that students experience more than just an increase in positive attitude towards the subject matter engaged.

There are, of course, trade-offs when implementing PBL. In particular, (Thomas, 2000) mentions the critical role teachers must play in a PBL context by "shaping opportunities for learning, guiding the students' thinking, and helping them construct new understandings." In this role, significant content knowledge and project management skills become required to support student motivation throughout a PBL project.

## Course Results as Reported by Envision

Based on pre-session and post-session surveys, Envision Academic Affairs reported the following regarding the performance of the MGTA-Envision collaboration courses:

- A significant majority of students (98%) attending the program rated the quality of their overall program experience as a 4 or 5 on a five-point scale.
- A significant majority of students (98%) attending the program rated the program as having a positive impact on them as a 4 or 5 on a five-point scale.
- A significant majority of students (96%) attending the program rated the quality of their GMU Professor as a 4 or 5 on a five-point scale.
- A significant majority of parents (94%) rated the relevance of the program curriculum as a 4 or 5 on a five-point scale.
- A significant majority of parents (93%) rated the quality of their overall program experience as a 4 or 5 on a five-point scale.
- A significant majority of students (82%) reported that the program helped them identify skills and knowledge needed to be successful in a career as a result of the conference experience.
- A significant majority of students (78%) reported that they were Interested in Applying to George Mason University.
- A significant majority of students (76%) reported that they improved their creativity skills as a result of the conference experience. (Envision Academic Affairs, 2016)

## Conclusions

The high satisfaction numbers reported by (Envision Academic Affairs, 2016) mirror the expected outcome of positive student attitude indicated in PBL literature. In addition, the survey numbers regarding curriculum relevance and help identifying career-relevant skills indicate results that match the goal of career exploration expressed by Envision. It is also worth noting that parents showed similarly positive attitude towards the quality of the program.

While (Thomas, 2000) mentions challenges in PBL implementation related to content knowledge and project management, the GMU faculty and students acting as MGTA staff were well-equipped to meet this challenge. As students and faculty of the Computer Game Design program at GMU, the MGTA staff were significantly versed in the domain-specific knowledge of game development and, thus, were able to offer advice and guidance to students based on their own experiences as game developers. Also, because a significant number of courses in the Computer Game Design program involve hands-on projects, the students and faculty came to MGTA well-equipped with project management skills honed by first-hand experience executing projects under strict deadlines. In this way, the MGTA staff proved to be a team that was well-prepared to mitigate the technical difficulties of PBL while allowing the students to enjoy its benefits. This, along with its placement as a summer program, enabled the MGTA Honors Program to fulfill the needs and expectations of students, parents, and staff alike.

# References

Baran, M.& Maskan, A. (2010). The effect of project-based learning on pre-service physics teachers'electrostatic achievements. Cypriot Journal of Educational Sciences, 5, 243-257

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning. Educational Psychologist, 26(3-4), 369–398. http://doi.org/10.1080/00461520.1991.9653139

Envision Academic Affairs. (2016). 2016 Mason Game & Technology Academy Overview of Student and Parent Survey & Assessments.

Envision EMI. (n.d.). Envision Career & Leadership Programs. Retrieved from https://www.envisionexperience.com/

Ergül, N. R., & Kargın, E. K. (2014). The Effect of Project based Learning on Students' Science Success. Procedia - Social and Behavioral Sciences, 136, 537–541. https://doi.org/10.1016/j.sbspro.2014.05.371

Kaldi, S., Filippatou, D., & Govaris, C. (2011). Project-based learning in primary schools: effects on pupils' learning and attitudes. Education 3-13, 39(1), 35–47. http://doi.org/10.1080/03004270903179538

Larmer, J., Mergendoller, J., & Boss, S. (2015). Gold Standard PBL: Essential Project Design Elements. Buck Institute for Education.

Marx, R. W., Blumenfeld, P. C., Krajcik, J. S., & Soloway, E. (1997). Enacting Project-Based Science. The Elementary School Journal, 97(4), 341–358.

Thomas, J. W. (2000). A review of research on project-based learning. Retrieved from http://www.newtechnetwork.org.590elmp01.blackmesh.com/sites/default/files/dr/pblresearch2.pdf

Tseng, K.-H., Chang, C.-C., Lou, S.-J., & Chen, W.-P. (2013). Attitudes towards science, technology, engineering and mathematics (STEM) in a project-based learning (PjBL) environment. International Journal of Technology and Design Education, 23(1), 87–102. http://doi.org/10.1007/s10798-011-9160-x

Wilson, T. L., & Zackariasson, P. (2007). Relationships in Video Game Development. Retrieved from http://www.academia.edu/download/4518777/2007rms.pdf