INTEGRATING LIVE CODING AND INTERACTIVE SYLLABUS TO ENGAGE STUDENTS IN AN INTERACTIVE MEDIA ARTS COURSE

Aven Le Zhou, M.S.
Instructor, NYU Shanghai, 1555 Century Avenue, Pudong New District, Shanghai 200122, China
aven.le.zhou@gmail.com
Jace Hargis, PhD
Professor and Director, Center for Teaching & Learning, NYU Shanghai
jace.hargis@gmail.com

ABSTRACT

Teaching coding to a diverse group of students has several major challenges. This paper will suggest two major teaching methodologies that had been shown to be effective in addressing many of the challenges at three small international liberal arts colleges. The teaching techniques used included a Live Coding and Interactive Syllabus. Findings included positive student dispositions during and after the use of the two methodologies; and more meaningful use of coding and discipline specific vocabulary by students. The authors will continue to implement the methods, collect data and further analyze the results in an on-going continuous improvement approach to effective teaching.

Key words: Information Processing, Student Engagement, Syllabus Design, Live Coding

INTRODUCTION

The concept of media art has been discussed in many forms throughout time. Derivatives of this discipline include new media art, digital media art, computational art, interactive art, digital art, Algorithm art, Generative art, and many more. For this paper, the term Interactive Media Art (IMA) is used to describe the work.

WHAT IS DIGITAL MEDIA ART EDUCATION

Historically Digital Media Art is thought of as the combination of arts and technology, resulting in the key skill of integrating the two disciplines. In digital media art education, students are taught the skill sets of developing art concept, and technical skill sets of implementing the idea, build projects. And more ideally, become fluent in combining both, taking advantage of how each can complement each other, resulting in a more powerful approach. However many courses may overly focus on the concept and minimize the practical part; or narrow the scope by teaching focused on the technology aspect of education, when students may not realize the function of the technology. In some cases, students might learn both the art and the technology independently, but still may not be able to integrate the two into a cohesive whole to build a complete project.

In our teaching approach, a major focus was to teach students how to build an authentic, project aligned with the Course Learning Outcomes based on art and technology proficiency. To teach art concepts, it was found that using a project-based approach combined with case studies has been effective (Iwamoto, Hargis, & Vuong, 2016). However, programming as an essential approach to digital media arts project development, is difficult to learn in a similar fashion. To accomplish this task, students usually require a steep learning curve, especially for beginners in digital media art major because art students usually find logic - the core and essence of programming - too strict, abstract (not visually obvious) and perhaps even boring to learn. There are many cases where students early in their career meet but cannot locate the problem, and give up on the problem feeling frustrated. They can view many of videos and feel they are able to complete the project, but they encounter problems quickly and are unable to make progress. Or they learned how to write program code, and then lose interest because they do not know how to apply to art practice.

#1 Live Coding

To overcome the challenges that exist between coding and art, the instructor provides live coding in-class, with real-time demonstrations and visual examples. This approach provides visual reference to the abstract codes and logic, and explain technical concepts more precisely. Students will follow the live coding at the same time they engage in hands-on practices and can access immediate assistance.

#2 Interactive Syllabus

A secondary challenge involved the diverse student group, who usually learn at different paces. The prepared students may become bored; and the confused students are often apprehensive to stop the class and ask questions. Thus, I created and shared an Interactive Syllabus, which allows students immediate feedback and helps the instructor to follow students' progress and adapt class content and peace responsively. With live coding and interactive syllabus, I help students especially beginners to surpass the difficulties in beginning programming study and to accelerate the learning process and hone a solid foundation for future explorations.

Our Main Contribution pursues two methods:

1. When the instructor integrates **Live Coding**, it acts as authentic activity allowing students to practice real-time and subsequently the instructor to identify timely feedback and individualized support (and incidental support, ie. those students near the student with questions could learn from the instructor feedback on the concept); [student says, "I am an artist mind"]
2. Offer an **Interactive Syllabus** as a communication tool to engage students by providing a real-time platform to elicit student concrete ideas on their progress, as the instructor can assist one student's question, which allows every student to observe, review and contribute (more than solutions to problems, but more like a spirit process for "HOW" to approach the problem).

LITERATURE REVIEW

Information Processing

The Information processing model initiates how learners code and decode meaningful information, attend to external stimuli, encapsulate relevant knowledge, skills and dispositions in long term memory, and recall the material upon appropriate cues or prompts (Atkinson & Shiffrin,1971). Further, this theory examines how new information translates to application by connecting and relating to knowledge already stored in the short-term memory store. (Schunk, 1996). This knowledge acquisition involves the application of cognitive, linguistic, motor and social skills (Hargis, 1999). In respect to knowledge acquisition, instructional design of a curriculum that incorporates active learning has been shown to improve the learning and understanding of students. In a metaanalysis of 225 studies, it was found that students in classes with lecturing were 1.5 times more likely to fail than classes with active learning (Freeman, 2014). Active learning strategies such as Live Coding and Interactive Syllabus provides similar environment for connecting students to conceptual frameworks.

Student Engagement

Student engagement as an instructional outcome has been shown to affect a wide variety of dispositions, knowledge and skills important to student success (Hargis, Cavanaugh, Kamali & Soto, 2014). Students are typically eager to engage in authentic, meaningful learning; although, the learning environment are frequently very different from their intrinsic motivation, and career goals. Also, the ways students seek to engage are as different as their ways of processing information (Carini, Kuh & Klein, 2006). A major challenge to teaching well is how are multiple learning opportunities created when students are asked to do more in less time (Kuh, 2001). One method to address these challenges is to provide efficient, active models of instruction, where students can connect their background and ways of learning to their personal career path (Appleton, Christenson & Furlong, 2008). Students have been shown to increase engagement in immersive authentic settings, and project-based learning (PBL) (Salisbury, Umbach & Paulsen, 2009). The goal of these settings is to provide timely authentic experiences that align with learning outcomes, assessments and active experiences. Structuring these approaches in a relevant ways to gather student attention, and ultimate transfer of working memory to long term memory creates a successful experience for deep learning (Umbach & Wawrzynski, 2005).

Syllabus Design

The origin of the word syllabus is from the Latin word "list". In higher education, it is generally thought of as an outline of topics which will be taught. The traditional purpose of a syllabus has been generally viewed as a document to provide information, which students would need to complete a course (Grunert, 1997). Matejke and Lance (1994) suggest three ways a syllabus can be used - as a contract; a communication device; and a cognitive map. If the syllabus contains sufficient detail, it can be an accessible reference for students, when they need pertinent information (Hargis, 2014). In more innovative cases, the syllabus can act as a two-way communication device. During the learning experience, the instructor can add to the syllabus, helpful updates, pertinent resources and class activities. In this way the syllabus behave as learning supplemental resource (Hargis, 2014). Creating an interactive syllabus allows for more functionality, flexibility and is more accessible to a broad range of learners. A visual syllabus can also act as a tool for learner engagement (National Survey of Student Engagement (NSEE), 2013).

Live Coding

Live-coding is an approach to teaching programming by writing code during class as part of the lecture. In these sessions, an instructor thinks aloud while writing code as the students follow along by observing and writing code as well (Adalbert, Soosai, Jignesh, Richard & Rosenfeld, 2018). Adalbert et. al (2018) found that livecoding

(1) makes the process of programming easy to understand for novice programmers;
(2) helps students learn the process of debugging; and
(3) exposes students to good programming practices."

Similar approach to using live coding have shown positive results on student perception, disposition and achievement (Paxton, 2002; Rubin, 2013).

TEACHING ACTIVITIES

While integrating Live Coding and Interactive Syllabus as teaching method, several interactive media arts classes were conducted in the past three years. The Live Coding methods was first used is Fall 2016 through the Summer of 2019 at three different private universities in Asia. The Interactive Syllabus was developed as the need arose from teaching these courses and used in classes taught in the summer of 2019 in a Sino-America Joint university in Asia. Approximately 57 students participants (21 male and 35 female students) were taught using the Live Coding method; and 15 students (11 male and four female students) among them are taught for the Interactive Syllabus (Table 1).

Table 1. Course information, frequency and demographics.

Course Title	TEACHING METHOD **Live Coding**	TEACHING METHOD **Interactive Syllabus**	Students Participants		Teaching Time
			Male Student	Female Student	
Video Game Design	yes	yes	7	4	Summer 2019
Intro to Creative Programming	yes	yes	4	0	Summer 2019
Digital Art	yes	no	1	12	Spring 2019
Make Design Interact	yes	no	2	12	Fall 2017
Creative Coding & Making	yes	no	7	7	Fall 2016

All classes are taught to students from art and design background and with minimal understanding of technology, and are regarded as an introduction to interactive media arts courses with a focus on productive outcome. All classes are project-based, more specifically, self-directed projects. Students will develop their concept and ideas, learn necessary technical skills and build their own projects from scratch to completed.

In all five classes, students are encouraged and required to develop the project from their own original ideas and creative questions. Instructor works with students from the beginning of the class to look into students' background and interests, seeking for possible ideas that could connect to both students' personality and the course contents. Students will dive deep into their own previous experiences, values and opinions, and take advantage of background research and references as inspiration and discover their own creative and innovative ideas and directions of projects development. With projects ideas and direction of developments in mind, students will define the challenge and take the initiative to seek for solutions. Instructor will build a collaborative productive environment both in and out of the classroom with students. Student can then solicit and provide feedback to peers, adjust and improves their own projects. Students will learn and develop their conceptual and technical development skills along with the projects development.

The projects will be evaluated in different aspects, including the progressive efforts and collaborative participation in previously described process; and the conceptual and technical developments; and the project quality and perfection; the documentation and presentations; and punctuality of the projects submission.

1) The progressive efforts will be evaluated from students' motivation and investment during the process such as the challenges they choose and take, and their initiatives in solution seeking; and whether they

embrace and take positive influence from possible temporary failures; and how do they improve their project from peers' feedback. The projects development will benefit much from a collaborative environment, mostly from fellow students and instructor's feedback. The collaborative participation is also included in the project assessment, and require students to provide peer feedback to colleagues and solicit feedback from both peers and instructor.

2) The conceptual and technical developments also plays an important role in the project evaluation. Creativity and originality of the project count most of the concept development evaluation. Projects should reflect original thoughts and include students' personality such as their personal opinion and the connection between project concepts and their background and experiences. Students are also required to use references as background research and inspiration. For technical developments, projects demonstrate good understanding, complex and fluent application of technological concepts will gain more positive comments. The technical developments of project will be assessed with its complexity of programming technology adopted in development and whether they meet expectations of the project concepts.
3) The final delivery of the project should be completed working prototype that can be run live with minimal errors. And all the components should have been well polished as to perspectives of art, design and technology. The project runs in live smoothly and demonstrates good user interaction design, programming techniques and takes good usage of graphics, sound effects and etc. will be evaluated as high quality work.
4) The overall quality of the self-directed project will be assessed with all the progressive presentations such as project concept presentation, in-class development workshop and final project presentations as well as all the documentations for these processes. In the presentations and documentations, students should include the coherence of project from concept to result, with the initial ideas, the background research and inspiration, the progressive efforts, interpretation of results and proper citation of references.
5) All of the submissions should be turned-in on time, the full mark of punctuality is 10. A one-day delay of delivery will cause a one point reduction in the total grade.

The analytical rubric (see Appendix) used for the Self-Directed Projects is as follows:

- **Progressive Efforts, Motivation, Investment (15%)**
  - Takes challenges and shows initiative in seeking solutions
  - Embraces temporary “failed events”
  - Adjusts/improves the project based on the feedback
  - Significant time and effort
- **Collaborative Participation, Engagement, Contribution (15%)**
  - Solicits feedback
  - Provides peer feedback to colleagues
  - Does not hesitate to ask questions for troubleshooting
- **Conceptual Development, Creativity, Originality (20%)**
  - Original thoughts
  - Level of including personality, personal voice and connection in projects
  - Respects and use of sources as background research, inspiration and reference
- **Technical Development, Understanding, Complexity and Fluency (20%)**
  - Complexity of the subject, concepts or technology
    - (Excellent, Good, Weak) Complexity of programming concept attempted and meets expectations in (Excellent, Good, Weak) level
  - Evidence of understanding, application and/or analysis of technology.
    - (Excellent) taken more than (number) of the programming concepts and techniques taught in class and applied in Creative and Original format
    - (Good) used sample codes as a starting point and successfully developed the project from the reference in creative and Original format
    - (Weak) used some sample codes with minimal manipulation and adoption
- **Project Quality, Perfection (10%)**
  - Live Demonstration of a completed projects
    - (Excellent) Project runs stable with no errors
    - (Good) Project runs smoothly with minor error, however it clearly explains how it works.
    - (Weak) Project runs with major error and cannot run live
  - Polishing every component from the perspective of art, design and technology professionally.
    - Use of (Excellent, Good, Weak) user interaction design, programming techniques,

graphics, sound effects, etc. to enhance user experience and/or visual aesthetic

- **Presentation & Documentation (10%)**
    - Coherence of project demonstration from concept to result
    - Initial Idea, Research and Inspiration
    - Effort and Perseverance
    - Interpretation of Results, References and Correct Citation
- **Punctuality (10%)**
    - Progressive and final presentation. Documentations need to be submitted on time.
        - (Excellent) All submitted on time (10 points)
        - (Reduction) 1 day delay equal to 1 point off and the minimal is 0

Creative Coding & Making, Fall 2016
This class is conducted in New Media and Communication program at a university in China. The program offers a three-years' college diploma which teaches students new media arts and communication design. This program is very practical driven with a strong focus on teaching students practical skills so they can get into the job market right after graduation. Students are recruited to this program with strong fine art background but usually with minimal science knowledge such as math or physics. The class included 14 sophomore students, seven male students and seven female students and they have learnt design concept and digital design software skills such adobe suit .etc in their first year of study. The class meets three hours per week and lasts for 16 weeks and consisted of 14 weeks' classes for teaching and last two weeks for projects developments. All 14 students finished their individual project at the end of this class.

Make Design Interact, Fall 2017
This class was taught at a university in Shanghai, China with 14 students, which included two male and 12 female students in the 16 weeks' course. All students are sophomores with similar background to Creative Coding and Making, they joint the program with strong skills and experiences of fine art and studied general practical design skills such as graphic design, video shooting and editing, and design software such as adobe suite in their freshman year. And then they started this class in their second year but with minimal understanding of programming. The class was designed based on students' design background and regarded as an introductory course for the major. Thirteen students finished the class and their final project in the class. One student left the program in the middle of the class.

Digital Art, Spring 2019
This class was taught in a two years' master program at a sino-american university in Shanghai, China. It's a first-year introductory class in Cultural & Creative Industry Management major. The school and major has a strong focus on combine culture, creation with business and technology. Students came from various different backgrounds such as Economics, History, Media & Communication, Graphic Design, etc. Most students just finished their undergraduates and continued to study the program and this course right after. There are 13 students, one male and 12 female students in this class. The course runs for seven weeks, for three hours per week. Students are divided into four groups by themselves and all finished their final projects and were curated into an open art exhibition along with other course project.

Introduction to Creative Programming, Summer 2019
This class was taught as a one-week summer intensive class to high school students in a sino-american university in Shanghai, China. The class is part of a secondary Academy program and was designed to provide high school students experiences of university studies. The class runs from nine am to five pm and lasted for five days with a closing ceremony at the end. There are four male students all at grade 10, two are from Chinese international high school , one is from American high school and one is from Chinese local high school. There is one student had learnt about software programming in advanced level courses and the other three students had no previous programming experiences. All students finished their individual projects.

Video Game Design, Summer 2019
This class was taught in a sino-american university in Shanghai, China as introductory experiences of university studies. It's also part of the secondary program as the Introduction to Creative Programming course. There are 11 students, seven male and four female students in this class, among which there are four students at grade nine, two students at grade ten, four students at grade 11 and one student at grade 12. All students do not have previous experiences in programming or design. The class is designed to teach students basic concept of video game design and practical skills, so they can build their own video games. Eleven students divided themselves into three groups and have finished three group projects.

METHODS

Project-Based Learning (PBL)

To teach course concepts, it was found that using a Project-Based Learning (PBL) approach combined with case studies have been effective. Significant PBL research (Iwamoto, Hargis, & Vuong, 2016; Johansen, Scaff, & Hargis, 2009; Brown & Hargis, 2008; Hargis, 2007) have shown that this pedagogical approach is effective in many disciplines. Projects were demonstrated using first hand material and insights and professional colleagues' projects.

A major focus for this student was to teach students how to build an authentic, project aligned with the Course Learning Outcomes, and based on art and technology proficiency. The project analysis, coupled with a case study was modelled and reinforced throughout the term with projects from the instructor. Ultimately, one of the outcomes was to promote, monitor and measure self-directed/self-regulated learning as students completed their projects.

Live Coding and Interactive Syllabus

To effectively teach this type of technology, a “Live Coding" environment was offered where students follow the instructor who is coding real time in class, showcasing their thoughts, mistakes, approach and how they manage their thought process. Secondly, in conjunction, a shared interactive syllabus was created, where students can access and collaborate. Finally, this approach modeled the use and implementation of the tools by asking students to share progressive presentations, which align with industry standards and are open to peer critical feedback and remediation through dialogue.

Hardware Setup: First, the hardware environment for the course included content such as computers, Internet access, code editing software (Integrated Development Environment, IDE), web browser and Google Suite access. The hardware can be arranged in a computer lab PCs or school laptops connected to the internet and with software prepared in advance. Or in our case, students were allowed to bring in their own laptops. It is worth pointing out, that all of the IDE and Google Suite that were used in classes are open source or commercial software with free (trail) access.

The classroom is offers ample power supplies and Wifi that connected to the Internet with access to Google service. After all of the computers are connected, students will log into their Google accounts. The instructor will share access to the interactive syllabus as a Google Document. The instructor will grant all students with the editing access to the shared syllabus and emphasize that they should be careful when editing.

Software Setup: Within the interactive syllabus, the instructor can include all the links to different resources such as the link to download all the softwares, links to Google slides used in class, etc. With the links and instruction of how to setup the IDE, students can manage to download and install the specific version of the IDE. The class will then be working from the same software, coding environment setup. In our case, Processing IDE and Atom was run on Mac OSX or Windows. Both Processing and Atom are open source free software and require minimal setup.

When explaining the concept of programming technology, the instructor will open the IDE on the instructional computer and project to the main screen. The instructor will start to code and demonstrate examples of previously mentioned concept. The instructor will code some examples from the beginning and explain line by line at the same time he is coding. This approach is usually very slow and most students should be able to follow the live example. With hands on practices, students apply the concepts while viewing the immediate execution and result of their own code. The instructor will post the example code in realtime to the interactive syllabus so the student behind schedule can reference to and improve their own code. Students are also required to post not only their own code but also their results such as graphics in a screenshot format. Students can then view each other's work, and take reference from each other in real time.

The key concept of live coding is to demonstrate problems which 1) can be solved with a systematic approach and fluently; and 2) could also be difficult, perhaps frustrating to solve but eventually accomplished with sustained effort. Sometimes the class will naturally run into challenging scenarios, and the instructor can take the opportunity to demonstrate solution strategies. When a challenge does not present itself, the instructor can insert hurdles. The instructor can prepare more difficult problems for students to solve and encourage students to raise questions so they can discuss frequent questions for solutions.

After students are ready for more challenging tasks, the instructor will share more examples, scaffolding prior concepts. In addition, the instructor can share certain problems which guide the class towards the learning outcomes. The instructor can walk around the class to offer one-on-one assistance for just-in-time learning. When the problem is easy, students produce various solutions and the interactive syllabus provides an ideal destination to share their efforts with peers and seek comments. When the problem becomes more difficult, students frequently are confused on a particular concept and cannot complete the problem. Students can post their problem to the interactive syllabus and the instructor can summarize common problems and bring to the attention of the entire class, which can create an efficient method for timely formative assessment.

When a problem is easy to solve, the instructor will demonstrate the fluent process of how to solve the problem and explain the general thoughts focussing on how he approached the solution. Another powerful aspect of this approach is when the instructor cannot describe an immediate solution and talks through the process of problem solving, so students can view authentic problem solving. The students and instructor will analyse the problem together, take inspiration from the discussion and try different possible solutions. It is important to show students how those obstacles can be resolved by explaining the process so they can learn to solve problems on their own.

Another advantage of using live coding as an engaging method is to offer problems that are not prepared in advance. To show the natural process of coding and solving problems with integrated short-term failed events, the instructor can encourage students to raise questions or find reference that are interesting to them from their prior experiences. The instructor and the students then select the most interesting questions and create possible alternative solutions together. All attempts and solutions and efforts will be part of the interactive syllabus, shared with everyone in real time, which provides a clear process for referencing.

DISCUSSION

The results from this study further informed the instructor on the teaching methods; how to attend to student knowledge, skills and disposition; and connect with students during class. The author applied the idea of live-coding, interactive syllabus and Google Suite. It was found to be convenient to collect student's response in real time, and their conceptual understanding before and after the lecture. This approach worked effectively, it was easy to share code and demonstrate in class, both from the instructor and students viewpoint. Plus, by using an interactive, collaborative syllabus, the result was complete documentation of the learning experience.

LIMITATIONS

The major variables in this study included the small number of students in the classes; private, liberal arts setting; and at the initial stages of developing data collection instrumentation. Variables were minimized by teaching more sections before summarizing the methodology; asking more questions along the way; and discussing the methodology with the university Center for Teaching & Learning. Ultimately, variables were accounted for by realizing this is a preliminary study advancing the live coding methods originally published in 2002, attempting to determine if these methods were generalizable in our setting.

REFERENCES

Adalbert G., Soosai R., Jignesh, M., Richard H., & Rosenfeld, E. (2018). Role of Live-coding in Learning Introductory Programming. In Proceedings of 18th Koli Calling International Conference on Computing Education Research, Koli, Finland, November 22–25, 2018.

Appleton, J., Christenson, S., & Furlong, M. (2008). Student engagement with school: Critical conceptual and methodological issues of the construct. Psychology in the Schools, 45(5), 369- 386. doi:10.1002/pits.20303.

Atkinson, R. C., & Shiffrin, R. M. (1971). Human memory: A proposed system and its control processes. Advances in Research and Theory, 2, 89-195.

Boyer, E. (1990). Scholarship Reconsidered: Priorities of the Professoriate. NY Carnegie Foundation for Advancement of Teaching.

Brown, S., & Hargis, J. (2008). Undergraduate research in art history using project-based learning. Journal of Faculty Development, 22(2), 152-158.

Carini, R., Kuh, G., & Klein, S. (2006). Student engagement and student learning: Testing the linkages. Research in Higher Education, 47(1), 1-32. doi:10.1007/s11162-005-8150-9.

Freeman, S., Eddy, S.L., McDonough, M., Smith, M.K., Okorafor, N., Jordt, H., & Wenderoth, M.P., (2014). Active learning increases student performance in science, engineering, and mathematics. Proceedings of the National Academy of Sciences (PNAS), 111 (23), 841.

Grunert, J. (1997). The course syllabus: A learning-centered approach. Bolton, MA: Anker Publishing.

Hargis, J. (2000). The Self-regulated learner advantage: Learning science on the Internet. Electronic Journal of Science Education, 4(4).

Hargis, J. (2007). Teaching project-based assessment in 12 days in a developing country. Journal of Excellence in College Teaching, 18(3), 129-142.

Hargis, J. (2014). A ten year study of faculty classroom observations. Transformative Dialogues: Teaching and Learning Journal, 7(2).

Hargis, J., Cavanaugh, C., Kamali, T. & Soto, M. (2014). A federal higher education iPad mobile learning initiative: triangulation of data to determine early effectiveness. Innovative Higher Education, 39(1), 45-57.

Iwamoto, D., Hargis, J., & Vuong, K. (2016). Effect of Project-Based Learning pedagogical model on achievement through the evaluative lens of student perceptions. International Journal for the Scholarship of Technology Enhanced Learning, 1(1), 24-42.

Johansen, D., Scaff, C., & Hargis, J. (2009). Interdisciplinary project-based model for enhanced instruction of marketing courses.International Journal for the Scholarship of Teaching and Learning, 3(1), Article 22.

Kuh, G. (2008). Excerpt from high-impact educational practices: What they are, who has access to them, and why they matter. Association of American Colleges and Universities, Washington, DC

Matejka, K., & Lance B. K. (1994). Designing a great syllabus. College Teaching 42,115-117.

National Survey of Student Engagement. (2013). A Fresh Look at Student Engagement - Annual Results 2013. Bloomington, IN: Indiana University Center for Postsecondary Research.

Paxton, J. (2002). Live programming as a lecture technique. Journal of Computing Sciences in Colleges 18, 2, 51–56.

Rubin. M. (2013). The effectiveness of live-coding to teach introductory programming. In Proceeding of the 44th ACM technical symposium on Computer science education. ACM, 651–656.

Salisbury, M., Umbach, P., Paulsen, M., & Pascarella, E. (2009), Going global: Understanding the choice process of the intent to study abroad. Research in Higher Education, 50(2), 119-143. doi:10.1007/s11162-008-9111-x.

Schunk, D. H. (1996). Learning theories. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Umbach, P., & Wawrzynski, M. (2005). Faculty do matter: The role of college faculty in student learning and engagement. Research in Higher Education, 46(2), 153-184. doi:10.1007/s11162- 004-1598-1.

APPENDIX

Analytical Rubric for Self-directed Projects

Progressive Efforts, Motivation, Investment 15%	Takes challenges and shows initiative in seeking solutions / Embraces temporary “failed events” / Adjusts and improves the project based on the feedback / Significant time and effort
Collaborative Participation, Engagement, Contribution 15%	Solicits feedback / Provides peer feedback to colleagues / Does not hesitate to ask questions for troubleshooting
Conceptual Development, Creativity, Originality 20%	(Excellent, Good, Weak) Original thoughts / (Excellent, Good, Weak) Level of including personality, personal voice and connection in projects / (Excellent, Good, Weak) Respects and use of sources as background research, inspiration and reference

Technical Development, Understanding, Complexity and Fluency 20%	Complexity of the subject, concepts or technology / Evidence of understanding, application and/or analysis of technology
Completive Project, Quality, Perfection 10%	Live Demonstration of a completed projects / Polishing every component from the perspective of art, design and technology professionally
Presentation & Documentation 10%	Coherence of project demonstration from concept to result / Initial Idea, Research & Inspiration / Effort and Perseverance / Interpretation of Results, References and Citation
Punctuality 10%	Progressive and final Presentation, documentations need to be submitted on time.