Client-Led Game/Simulation Projects' Effects on Motivation and Career Readiness

Ms. Krista Stacey
kjstacey@southalabama.edu
University of South Alabama
Mobile, USA

Dr. David Bourrie
dbourrie@southalabama.edu
University of South Alabama
Mobile, USA

Course Game and Simulation Development
Programming Language C# in Unity
Knowledge Unit Software Development Methods
CS Topics Mixed Reality, Unity Development, Project-Based Learning, Rapid Prototyping, Iteration, Requirements Elicitation, Simulation Design, Educational Technology, Software Documentation, Collaborative Software Development

Synopsis

This instructor-facing submission outlines the implementation of a client-led project-based learning (PBL) model in an upper-level game and simulation development course. The project centered on developing functional AR simulations for real clients. Drawing on professional documentation through Game Design Documents (GDDs), authentic feedback cycles, and self-efficacy theory, students experienced iterative development with external stakeholders. Survey data and focus-group feedback revealed increases in student motivation, enjoyment, and self-efficacy compared to previous faculty-led iterations. The submission includes a conceptual pedagogy model (Figure 1), adaptation pathways for different instructional contexts, and recommendations to integrate client feedback structures without formal external partners.

CCS Concepts

• **Social and professional topics** → *Computing education*;
• **Software and its engineering** → *Software development process management*.

Keywords

Project-Based Learning, Experiential Learning, Simulations, Self-Efficacy, Motivation

ACM EngageCSEdu, December 2025.

ACM ISBN 979-8-4007-2418-3/2025/12
https://doi.org/10.1145/3786353

ACM Reference Format:
Ms. Krista Stacey and Dr. David Bourrie . December 2025. Client-Led Game/Simulation Projects' Effects on Motivation and Career Readiness. In *ACM EngageCSEdu.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3786353

1 Engagement Highlights

This Open Educational Resource (OER) engages students through meaningful, relevant content, and well-structured collaborative learning, two evidence-based practices shown to broaden participation and improve learning outcomes [16]. The targeted course was an upper-level course in game and simulation development in which students created an original interactive experience using Unity mixed reality, specifically Augmented Reality (AR) technologies. It followed a course on computer game design that allows students to create an original game level with free rein of design choices. In the studied course, students worked in teams to develop simulations for real clients (in this case an engineering professor and Instructional Design consultants as lead clients). Each team had a specific area of focus, such as a process or backend, and was expected to deliver a working prototype, a Game Design Document (GDD), and a presentation. The project was relevant to students' future professional contexts, fostering student-centered assessments and iterative feedback loops that support self-efficacy and motivation. The course provided meaningful and relevant content as students tackled real-world AR challenges aligned with the course objectives, increasing intrinsic motivation by linking tasks to professional practice. In addition, structured collaborative learning occurred as teams rotated roles and maintained living GDDs, promoting accountability and peer instruction. Furthermore, there was an opportunity for student-centered assessments, as playtest reflections and GDD revisions allowed choice and self-regulation, tailoring feedback to individual learning paths.

Differentiation strategies include offering scaffolded tutorials on Unity and GitHub for students with varying technical backgrounds, and allowing alternate deliverables (e.g., video walkthroughs) for those needing accommodations. Such strategies fit into the background of cyclic on-boarding, prototype development, playtesting, feedback integration,

and final delivery. Typical team roles included a client liaison or project manager, a lead programmer, an interaction or User-Interface (UI) designer, and a documentation lead responsible for maintaining the GDD, with responsibilities rotating as appropriate so that students could experience multiple perspectives in the development workflow. This sequence is particularly well-suited to upper-level or capstone courses, and instructors in earlier courses may choose to adopt a subset of phases or simplified scenarios while preserving the same engagement structure. The pedagogical model this sequence examines as presented in Figure 1 draws on self-efficacy theory and mirrors professional development workflows, positioning students as active designers collaborating with real clients to achieve educational engagement with the goal of confidence and persistence.

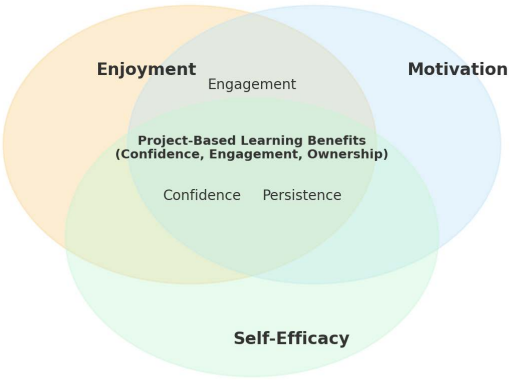

Figure 1: Conceptual Pedagogy

2 Recommendations

This OER was intended primarily for upper-level or capstone courses in game and simulation development where students already have foundational programming experience and perhaps some exposure to game engines. Instructors adopting the full client-led model should be comfortable with Unity, C#, and basic version control practices (e.g., GitHub or Unity Version Control), or should plan to collaborate with technical support staff or teaching assistants. In contexts where instructors and/or students are less experienced with XR or Unity, the same project structure can be implemented using desktop-only simulations or simpler game development environments such as Code.org or GameMaker, while preserving the client-facing, reflection, and documentation components.

Students completed a three-phase project sequence over a 15-week semester:

(1) (On-boarding & Setup): Unity and GitHub workshops, AR headset configuration, and client requirement elicitation (Weeks 1-3)
(2) Iterative Development: A cycle of AR implementation with biweekly sprint demos and GDD updates followed by playtests, reflection assignments, and revisions based on peer/client feedback. (Weeks 4-15)
(3) Final (playable) prototype and pitch presentation. (Week 16)

In the pilot offering, a client project with multiple process-focused components or scenarios served as the focus of the semester. Instructors may choose to implement a single comprehensive scenario throughout the term or assign different scenarios to different teams, depending on enrollment, client availability, and course goals. Each scenario follows the same sequence of proposal, GDD, playtesting reflections, and final pitch, allowing instructors to change scenarios without redesigning the overall structure. When scenarios are used "out of order" or in a subset, instructors should briefly summarize or scaffold any assumed prior work so that students understand the context for each activity. In subsequent offerings, for example, the proposal and GDD was completed in the primary game design course and development in the secondary course. Therefore, students in the secondary course receive a summary of progress and the completed GDD to develop, playtest, and final pitch to the client.

Within each team, clearly defined roles helped support accountability and engagement. Typical roles included:

- Client liaison / project manager, responsible for coordinating with the client and the instructor and tracking milestones.
- Lead programmer, responsible for the core interaction logic and the integration of technical features.
- Interaction or UI designer responsible for user flow, interaction design, and usability considerations.
- Art, audio, or environment lead, responsible for visual and audio assets or sourcing appropriate placeholder assets.
- Documentation / GDD lead, responsible for maintaining the GDD as a living document and ensuring updates after each sprint and playtest.

Roles may rotate during the semester so that students can experience multiple perspectives within the development workflow, or remain stable for courses emphasizing deeper specialization. Roles may also be combined to accommodate smaller groups of students.

Milestones were scaffolded with instructor support while preserving the client as the primary stakeholder. Clients provided written and verbal feedback at two checkpoints, and students submitted GDD updates after each phase. These

checkpoints were used first to verify specifications and second in mid-production to measure expectations. Unity tutorials were integrated into the schedule, but modified to fit the project context. Students used GitHub for source control, Meta Quest 3 headsets for deployment, and were required to test their projects on both desktop and headset platforms. The instructor intervention focused on project management, technical debugging, and facilitating feedback between teams and clients.

As shown in Figure 2, the main adoption hurdle is locating the client project. As the pilot in this study, the client was an NSF-funded professor. The lead author also has a partnership with the University's Instructional Learning Center, where they field requests for educational simulations. These simulations, when chosen, allow the students to be paired with a subject matter expert in the simulation as well as an instructional design specialist that serves as clients. Outside of educational simulations, the University's partnerships with local and government agencies serve as a source of clientele, such as a local steel mill that expressed interest after the pilot study.

One modification for wider adoption is to pair student teams with industry mentors or faculty who act as a client when formal external clients are not available. This maintains the external feedback loop and simulates authentic expectations, even when a live stakeholder may not be present. Industry mentors may also offer domain-specific insights that help contextualize design choices and motivate students to pursue technically sound, usable solutions.

Another useful adaptation is to use a hybrid model that starts with simulated client scenarios in the first half of the term, allowing students to gradually develop communication protocols and task management skills. By scaffolding the introduction of client interaction, instructors can help ensure that students build confidence and technical fluency before adding real-time feedback cycles. This can ease the transition for students new to team-based or real-world projects.

This model is particularly effective in upper-level or capstone courses where students are expected to demonstrate synthesis of technical and design skills in a collaborative context. It also encourages the development of professional soft skills, such as communication, time management, and problem-solving, in novel environments. Although the model was implemented using Unity and AR headsets, its core structure (client alignment, iterative GDDs, and reflection checkpoints) can be easily adapted to traditional software/game development or non-immersive simulation projects. It can also be deployed in the game development curriculum as an example of serious gaming.

One lesson learned from this implementation in the future is, especially with Unity, to enforce version control down to the editor used. When teams integrated their pieces of the project, the Unity editor and plug-in differences would cause the entire project to break, setting the teams back. It should be noted that part of on-boarding should include how to use version control resources such as Unity Version Control or GitHub for a minimum loss of time in this area, as well as a discussion of versions of plugins and libraries that all teams are expected to use. Another lesson learned is to make sure that one person on each team has the responsibility to communicate with the other teams to ensure that the final product is more uniform in design and depth. Some teams produced processes that were more aesthetically pleasing, while others produced processes that were more interactive.

As another lesson learned is the need to make these outcomes more visible to both students and instructors, we recommend including a brief summative analysis and postmortem at the end of the term. In our OER set, this takes the form of a written assignment in which each team member analyzes another group's client project and then reflects on their own project's goals, team process, and use of feedback. Instructors at programs with strong assessment requirements (e.g., ABET) may find that this assignment provides a natural point of evidence for communication, teamwork, and professionalism outcomes, and it can be adapted as either an individual or team deliverable.

Finally, because students work on client-defined projects, instructors should clarify institutional policies regarding intellectual property (IP) and the academic nature of the work. In this implementation, projects were framed as prototypes created within a course context, with students retaining authorship and clients receiving non-production demonstration artifacts for training or exhibition purposes. Making these expectations explicit can help students understand the value of their work and prepare them for future client engagements beyond the classroom. Some students asked and received written documentation from the institution clarifying their IP ownership and that they could pursue further development of their individual IP outside of the delivered project. Students were also welcome to include their work in their portfolios, social media profiles, etc.

3 Background

It is proposed that software development education should equip students with the ability to analyze problems efficiently, work collaboratively on projects, and document processes systematically [13]. In addition, industry-specific knowledge, including language selection and development strategies, is critical for student preparation [13]. In game and simulation development, there are unique mathematical, visual and engineering components in addition to traditional software requirements [9]. Furthermore, language choice is often dictated by the game engine, such as Unity with C# [2].

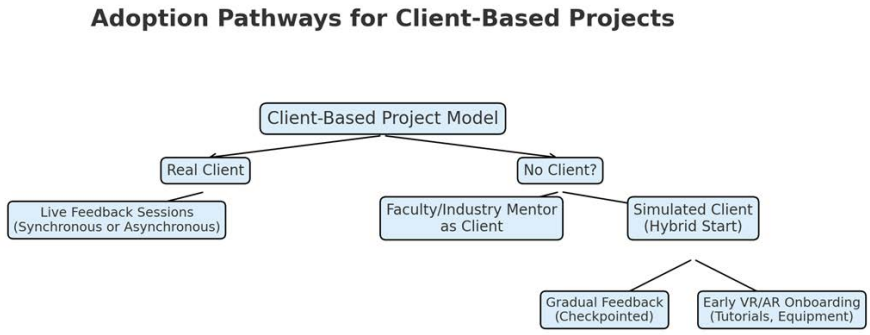

Figure 2: Suggested Adoption Decision Tree

In such settings, the critical technical decisions for students shift from choosing a language to selecting appropriate interaction patterns, data structures, and architectural designs within the constraints of a given engine and deployment platform.

In keeping with standard industry practices, the course emphasized documentation via the GDD, described as a living document that encompasses all areas of a development team's work and is updated throughout the project life-cycle [4]. The course employed Agile methodologies, particularly Scrum and rapid prototyping, as are common in game development education [6]. These methodologies are used not only to manage the scope but also to encourage reflection and iteration, the core values of Agile as a mindset rather than a fixed process [1]. Within this framework, teams are expected to negotiate trade-offs between scope and feasibility, justify design choices in their GDDs, and adjust their plans based on stakeholder feedback rather than following a static, instructor-prescribed specification.

This project-based model was also guided by previous studies that outlines flexible implementation strategies for PBL in software development curricula and highlighting differences between faculty-led, student-led, and client-led project sourcing [11]. When properly implemented, PBL enhances student satisfaction and confidence (self-efficacy) [10, 15]. In this course, students reported the highest levels of motivation and ownership when engaged in real client feedback, consistent with other reviews of the experiential learning literature [7]. Earlier courses used faculty-defined projects, in which the instructor specified the core problem and constraints. In the client-led model described here, external partners define goals and domain requirements, while students assume responsibility for interpreting those requirements, proposing technical and design solutions, and reconciling conflicting constraints-more closely mirroring the role of a creative lead or technical designer on a professional team.

This approach is further supported by Bandura's (1977) theory of self-efficacy, which posits that mastery experiences, modeling, persuasion, and emotional states all contribute to individuals' belief in their ability to succeed [3]. When applied to computing, this definition has been adapted to include guided mastery and social persuasion [8]. In addition, self-efficacy should be treated as formative, meaning that external conditions such as feedback can shape it [12]. Furthermore, combining instructor feedback with additional feedback mechanisms positively influenced students' self-efficacy [16]. Likewise, in this course, client feedback functioned not just as commentary but as validation of student contributions, leading to greater perceived capability.

Enjoyment, which often correlates with motivation, is also central to this framework. Students in client-based PBL environments perceive the experience as more engaging and feel better prepared for future work in systems design [5]. Enjoyment and intrinsic motivation are linked in educational contexts where students are given authentic challenges and are able to see the impact of their work [14].

Therefore, this pedagogical approach centers on the application of PBL within a game/simulation context that reflects current industry expectations, as well as prepares students for collaborative, iterative, and stakeholder-responsive development workflows. The course structure draws explicitly on these pedagogical models to integrate both faculty oversight and authentic client engagement. Furthermore, this approach aligns with established computing education best practices by integrating iterative development, reflective assessment, and authentic stakeholder engagement—key components emphasized in Computer Science education research and curriculum guidelines.

4 Outcomes and Reflection

Students in the course reported increased enjoyment, confidence, and professional relevance. Survey data showed an

average increase of over 1.0 out of 5 points on a Likert-scale in motivation and technical confidence compared to previous faculty-led projects. In open-ended responses, students contrasted this model with earlier, instructor-defined projects by emphasizing the weight of working for a real client and the need to justify design decisions beyond simply meeting a rubric. Post-focus group discussions revealed that working with a client contributed to a deeper sense of responsibility, and several students cited feedback loops as the most meaningful difference from past experiences. For example, one student remarked that they preferred critical feedback from clients because it signaled that the work was being taken seriously. Another stated, "This felt more like you were applying what you were taught instead of just redoing what you were taught."

Challenges such as coordinating schedules, learning GitHub, and deploying across multiple VR devices were noted. These are consistent with real-world conditions and can serve as learning opportunities when paired with instructor support. Teams that faced and overcame deployment errors or versioning issues reported higher confidence in their abilities by the end of the course. Instructors should be prepared to support student teams with version control, hardware access, and communication planning. Informal and formal feedback points allow students to calibrate their progress and reinforce the connection between feedback and iteration, a critical habit in professional development. The structured playtest reflections in particular prompted students to move from noticing basic usability issues toward articulating whether project goals, learning objectives, and client expectations were being met, helping to surface the kinds of analytic and communication skills expected in client-facing work.

These outcomes reflect a positive shift in student self-efficacy, particularly in their ability to apply technical skills in a collaborative, iterative, and externally validated context. Taken together, the data suggest three primary outcome areas: affective gains (greater enjoyment and motivation), technical growth (increased confidence deploying and iterating within a mixed reality toolchain), and professional development (improved comfort communicating with stakeholders, coordinating within teams, and responding to external feedback). Although the findings are promising, the results are based on a small cohort (n = 15) and should be interpreted in the context of a single-semester pilot implementation.

In addition to supporting student motivation and technical growth, this model provides all students with authentic engagement opportunities, regardless of prior industry connections. By integrating real-world client collaboration into the curriculum, all students from all backgrounds gain early exposure to professional feedback and expectations that are often gate-kept by networking barriers. As one student said, "I never thought I'd be able to work on a real-world project like this before graduating. It made everything we've learned feel applicable." At the same time, the present study focused on students' perceptions of motivation, self-efficacy, and professional readiness rather than directly measuring learning gains from the simulations themselves, a direction that is taken up further in the discussion of limitations and future work.

5 Limitations and Future Work

While the findings demonstrate promising gains in student motivation, enjoyment, and self-efficacy, several limitations warrant consideration. First, there was a small sample size (n = 15): All data derive from a single semester of the course with 15 participants. This limited cohort constrained the generalization of the results and increased the risk that idiosyncratic factors (e.g., specific client personalities, cohort dynamics) influenced outcomes.

Second, implementation occurred at one university, within a single discipline (game and simulation development). Institutional culture, resource availability, and student demographics may differ elsewhere, potentially affecting engagement and learning outcomes. In particular, this implementation took place in a setting with access to XR-capable lab machines and headsets; institutions with limited hardware, lab time, or network infrastructure may face additional barriers to adoption that were not captured in this study.

Third, the assessment focused on pre-post gains within one semester. There was no tracking of long-term retention of skills or sustained changes in self-efficacy beyond course end, nor was there a measure of transfer of collaborative workflows to subsequent projects. Similarly, the study did not directly assess learning gains attributable to the simulations themselves (e.g., domain knowledge in the clients' subject areas); outcomes were framed in terms of motivation, self-efficacy, and perceived professional readiness rather than content mastery.

Fourth, motivation and self-efficacy metrics rely on student surveys and focus groups, which may introduce response bias. Future work should triangulate with objective measures (e.g., code repository logs, client feedback scores, usability metrics). The Final Analysis and Post-Mortem assignment can be a source of qualitative data about student decision-making, teamwork, and perceived career readiness. Systematic analysis of these reflections, alongside survey and focus-group data, could provide a more nuanced understanding of how client-led projects shape students' professional identities over time.

Finally, this model centered on an AR project. While AR provided engaging, contextualized tasks, instructors interested in other modalities (VR, desktop simulations) should

pilot and validate engagement practices in those environments. Because the project is intentionally open-ended and multidisciplinary, task difficulty can vary significantly depending on the client, topic, and team composition, and some components (e.g., concept art, pitching) draw on skills outside traditional CS training. In this pilot, expectations for these components were framed in terms of clarity of communication and meeting of client base expectations rather than artistic polish, but the impact of different scaffolding strategies for these non-CS elements was not systematically studied. In addition, the client-centric framing raises broader questions about student labor, IP, and topic selection. In this implementation, projects were treated as course-based prototypes with students retaining authorship and clients receiving non-production demonstration artifacts, but institutional policies and community partnerships vary widely, and so the ethical implications of client selection and use of student work were beyond the scope of the present analysis.

For future work, there should be larger, multi-site studies where this OER is replicated with larger enrollments across varied institutions and disciplines to test scalability and contextual adaptability. Students should be followed into subsequent courses or capstone projects to assess sustained impacts on professional skills and self-efficacy. Analytics during development such as commit frequencies and issue resolution times should be integrated to complement self-reports and better quantify collaborative behaviors. Future studies should also incorporate direct measures of learning outcomes tied to the client domains (e.g., pre-post content assessments or performance tasks) to examine how the simulations support conceptual understanding. The framework should be adapted to other simulation contexts, including desktop-only or low-resource settings, to facilitate comparison of relative engagement benefits, technical hurdles, and equity implications for students with differing levels of hardware access. In addition, disaggregated analyses to explore differential effects on student populations can shed better light on reasonable accommodations for all learners and on how group dynamics, peer evaluation mechanisms, and explicit guidance around IP and client expectations influence students' sense of agency and fairness.

6 Materials

The accompanying OER materials are designed to support scaffolded student development from concept to functional AR simulation. These materials include:

- Proposal assignment with rubric to initiate alignment with client goals
  - OER Simulation Proposal Assignment.pdf
  - OER Simulation Proposal Rubric.pdf
- Game Design Document (GDD) template and rubric, emphasizing iterative planning and technical communication
  - OER Simulation GDD Template.pdf
  - OER Simulation GDD Rubric.pdf
- Peer playtest reflection assignments and rubrics (2 phases), supporting user testing and design iteration
  - OER Playtest1 Reflection Assignment.pdf
  - OER Playtest1 Reflection Rubric.pdf
  - OER Playtest2 Reflection Assignment.pdf
  - OER Playtest2 Reflection Rubric.pdf
- Final pitch assignment with a client-facing feedback form to simulate professional delivery and evaluation
  - OER Final Pitch Assignment.pdf
  - OER Client Pitch Rubric And Feedback Form.pdf
  - OER Final Analysis.pdf
  - OER Final Analysis Rubric.,pdf
  - OER README.pdf

Each rubric uses multiple performance levels (for example, exemplary, proficient, developing, and insufficient) with brief descriptors that clarify how full and partial credit are assigned for each criterion. This structure is intended to make expectations more transparent to students and to support consistent grading across teams, especially for 10-point criteria where each point carries substantial weight.

The GDD template includes prompts for learning objectives, target audience, and content grounding, requiring teams to identify authoritative sources or subject matter experts for the domain they are simulating. An additional prompt asks students to consider the vulnerability of their intended learners and any sensitive aspects of the topic, and to briefly describe how their design addresses these considerations. For components that draw on non-CS skills, such as concept art and pitching, rubric language emphasizes clarity of communication, alignment with design goals, and appropriateness for the intended audience rather than artistic polish or professional-level production values.

The playtest reflection assignments guide students to document usability findings, assess whether the stated learning objectives and client expectations are being met, and plan specific revisions. Instructors may also adapt these reflection prompts into brief mid-semester and end-of-semester self- and peer-evaluation activities, allowing students to rate their own contributions and their teamwork practices using the same criteria emphasized in the project rubrics.

The Final Analysis and Post-Mortem assignment helps to culminate the work completed over the project. In this written deliverable, students select another team's client project and analyze its alignment with client needs, target learners, design and technical decisions, and perceived effectiveness. They then compare that project to their own and complete a post-mortem reflection on their team's goals, process, use of feedback, and professional growth. This assignment is intended to "close the loop" on the project sequence by making explicit connections between the concrete artifacts (proposal,

GDD, playtests, and final pitch) and broader course outcomes such as communication, teamwork, and career readiness.

6.1 Resource Type

This submission is best categorized as a **Project** resource. It includes a multi-phase, client-driven assignment in which students develop an AR simulation aligned with real-world goals. The project spans several weeks or an entire semester and includes deliverables such as a Game Design Document (GDD), playtesting reflections, and a final live presentation for external stakeholders. Scaffolding is provided through assignments and rubrics to support student progress from concept to deployment.

7 Auxiliary Materials

(1) Unity Learn – Create with VR: A self-paced tutorial series introducing VR development fundamentals using Unity and the XR Interaction Toolkit. https://learn.unity.com/pathway/create-with-vr
(2) Bond, R. (2013). *Game Development Essentials: Game Interface Design* – Companion site with project samples and templates. https://www.cengage.com/c/game-development-essentials-game-interface-design-2e-bond
(3) Creative Commons Attribution 4.0 License https://creativecommons.org/licenses/by/4.0/

8 Acknowledgments

The client was the PI of NSF Award #2427766 (ME-REG: Microelectronics Experimental Research Experiences for Graduates). The instructor received a stipend for the study under the University of South Alabama Scholarship of Teaching and Learning (SoTL) fellowship, as well as a small technology grant from the University of South Alabama's Innovations in Learning Center (ILC).

References

[1] Pekka Abrahamsson, Outi Salo, and Jussi Ronkainen. 2002. Agile Software Development Methods: Review and Analysis. *VTT Publications* 478 (2002), 107.
[2] Zulfiqar Ali and Muhammad Usman. 2016. A framework for game engine selection for gamification and serious games. In *2016 Future Technologies Conference (FTC)*. IEEE, San Francisco, CA, USA, 1199–1207. doi:10.1109/FTC.2016.7821753
[3] Albert Bandura. 1977. Self-Efficacy: Toward a Unifying Theory of Behavioral Change: Psychological Review. *Psychological Review* 84, 2 (Jan. 1977), 191–215. https://libproxy.usouthal.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=eric&AN=EJ161632&site=eds-live
[4] Bob Bates. 2001. Game Design: The Art and Business of Creating Games. doi:10.5555/580517 Archive Location: world.
[5] Cathy Bishop-Clark. 2006. Problem-based Service Learning in a 200-level Systems Analysis and Design Course. *Journal of Computing Sciences in Colleges* 21, 3 (2006), 174–180.
[6] Jeremy Gibson Bond. 2014. *Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#*. Addison-Wesley Professional, Boston, MA. Google-Books-ID: 40T1AwAAQBAJ.
[7] Jeffrey Scott Coker, Evan Heiser, Laura Taylor, and Connie Book. 2017. Impacts of Experiential Learning Depth and Breadth on Student Outcomes. *Journal of Experiential Education* 40, 1 (March 2017), 5–23. doi:10.1177/1053825916678265
[8] Deborah Compeau and Christopher A. Higgins. 1995. Computer self-efficacy: development of a measure and initial test. *Management Information Systems Quarterly* 19, 2 (June 1995), 189–211. doi:10.2307/249688
[9] Ben Kenwright. 2016. Holistic game development curriculum. In *SIGGRAPH ASIA 2016 Symposium on Education*. ACM, Macau, 1–5. doi:10.1145/2993352.2993354
[10] Mark J. W. Lee, Sasha Nikolic, Peter J. Vial, Christian Ritz, Wanqing Li, and Tom Goldfinch. 2016. Enhancing Project-Based Learning Through Student and Industry Engagement in a Video-Augmented 3-D Virtual Trade Fair. *IEEE Transactions on Education* 59, 4 (Nov. 2016), 290–298. doi:10.1109/TE.2016.2546230
[11] Jens Liebehenschel. 2019. Adaptable Concept for Projects in Computer Science Degree Programs – An Experience Report. In *2019 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, Dubai, UAE, 45–50. doi:10.1109/EDUCON.2019.8725157 ISSN: 2165-9567.
[12] George M. Marakas, Richard D. Johnson, and Paul F. Clay. 2007. The Evolving Nature of the Computer Self-Efficacy Construct: An Empirical Investigation of Measurement Construction, Validity, Reliability and Stability Over Time: Journal of the Association for Information Systems. *Journal of the Association for Information Systems* 8, 1 (Jan. 2007), 15–46. https://libproxy.usouthal.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=24979091&site=eds-live Publisher: Association for Information Systems.
[13] Cheryl Noll and Marilyn Wilkins. 2002. Critical Skills of IS Professionals: A Model for Curriculum Development. *Journal of Information Technology Education: Research* 1 (2002), 143–154. doi:10.28945/352
[14] Jae-Eun Oh, Yuet Kai Chan, and Kyulee Viviane Kim. 2020. Social Media and E-Portfolios: Impacting Design Students' Motivation through Project-Based Learning. *IAFOR Journal of Education* 8, 3 (2020), 41–58. https://eric.ed.gov/?id=EJ1272529 Publisher: International Academic Forum ERIC Number: EJ1272529.
[15] Jeremy Straub, Scott Kerlin, and David Whalen. 2017. Teaching software project management using project based learning (PBL) and group projects. In *2017 IEEE International Conference on Electro Information Technology (EIT)*. IEEE, Cedar Rapids, IA, 016–021. doi:10.1109/EIT.2017.8053323 ISSN: 2154-0373.
[16] Katharina Alexandra Whalen, Alexander Renkl, Alexander Eitel, and Inga Glogger-Frey. 2024. Digital re-attributional feedback in high school mathematics education and its effect on motivation and achievement. *Journal of Computer Assisted Learning* 40, 2 (2024), 478–493. doi:10.1111/jcal.12889 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jcal.12889.