

Considering Large Student Teams in Game Development Education: A Post-Mortem

José P. Zagal

Division of Games

University of Utah

jose.zagal@utah.edu

ABSTRACT

Having a large (>100) team of students work on a single game as part of their games education experience sounds like a terrible idea. But, is it really? I provide an examination of the reasons why students are encouraged to participate in collaborative game development projects, challenge some of those assumptions, and propose the megateam as an alternate model that might be worth considering. I also present a brief post-mortem of a large (~70 student) team class explicitly designed to provide an educational experience more authentic to working at a large game studio by forcing an organizational structure that foregrounds the content pipeline (and bottlenecks), requires additional communication and coordination, and challenges everyone to maintain a coherent vision for the game they were working on. All of these are common problems identified in game industry post-mortems. While the megateam experience was not without flaws, it demonstrates there is potential for re-imagining the student game project experience such that it highlights a production model (i.e. AAA game development) that is more authentic to what many students aspire to, and may end up participating in. In this way game educators can better prepare students meet their career expectations and help them succeed.

Keywords

Capstone, game education, megateam, game development, post-mortem

INTRODUCTION

A few years ago, a colleague who was teaching a large-scale (>150 students) game development final capstone class, had an idea: what if the entire class worked on the same game? Somehow the idea filtered out to the students, the rest of the faculty in the Division of Games¹ at the University of Utah, and social media. The idea proved divisive amongst the students with feelings of betrayal, accusations that the faculty were seeking fame and attention over the needs of their students, and excitement for an unexpected opportunity. It was also contentious amongst the faculty with many calling it pointless, impractical, and irresponsible. The reaction on social media was the most incendiary – with game industry professionals chiming in that it was an awful idea, that the faculty clearly had no idea what they were talking about, and that the game program was clearly a scam.

To be fair, the idea of over 150 students spending an academic year working on a single game does sound bad. What kind of game would they make? How would you

¹ At the time known as the Entertainment Arts & Engineering program (EAE).

coordinate such an effort? Large studios already struggle shipping games on time, how would a group of novice students pull that off? The challenges seem even greater: students cannot (generally) be “fired”, should not work full-time since they have other classes to attend to, and are not being paid for the privilege – rather it is the opposite. But, if we took this idea seriously, what would it take to make it work, and more importantly, would a student megateam provide the opportunity for novel educational benefits worth pursuing?

In game development education it is common for students to participate in collaborative team projects (e.g. Katchabaw, Elliott, and Danton 2005; Redfield 2021), often in a capstone or final class (e.g. Zagal and Sharp 2011; Phelps, Egert, and Cloutier 2021; Bouchard and Gouglas 2021; Suvajdzic, Coleman, and Smith 2021). Project-based classes offer students opportunities to put into practice their technical skills (eg. programming, animation, art, design) alongside the “soft-skills” necessary to be effective and succeed professionally (e.g. collaboration, communication, coordination) (Brown, Lee, and Alejandre 2009). Some pedagogical reasons for doing this include providing hands-on development experience (Hogue, Kapralos, and Desjardins 2011), allowing students with different backgrounds to collaborate (Kessler, van Langeveld, and Altizer 2009), and providing opportunities for exploring new ideas and creative freedom. These are all seen as contributors for success for students pursuing careers in an industry whose members have a wide range of specific skills (art, programming, writing, design) in addition to soft skills including the ability to communicate across disciplinary boundaries (Parmentier and Picq 2016). In the context of professional game studios we note that it is often the soft skills that are valued most (Bidarra 2011). While specific technical skills are important, a game developer’s soft skills play a major role in their success (e.g. M. McGill 2008; Shahani 2020) with game developers reluctant to hire recent graduates due to their lack of soft skills (Diefenbach 2008). In some cases, soft skills are valued more highly than technical skills (Hewner and Guzdial 2010).

While many reasons support the use of collaborative game development projects in education, we should still examine how this model could be improved. Also, how it might fail to provide some of our students with the authentic game development experience many of us hope to provide in the classroom or studio setting?

In this article I challenge some assumptions underlying the use of collaborative game projects by presenting, in postmortem form, my experience teaching a game design class. In this class of approximately 70 students, everyone collaborated on a single game project. By presenting the intended design of the class, how it was structured, and also discussing its failures, I hope to provide a foundation for deeper critical reflection on the class game projects that are so commonly used across games education.

This paper does not represent the results of an empirical research study and as such there are no traditional “findings” to report. Rather, it is a challenge to consider how we can (individual institutional circumstances permitting) imagine and implement different environments to (hopefully) provide learning opportunities for students that are different from those we already support or that perhaps emphasize certain learning outcomes in ways that we currently struggle to achieve. The games industry’s production practices have changed over the years, thus we should also consider what those changes can, or should, mean for games education.

AUTHENTICITY IN GAMES EDUCATION

A common goal in educational practice is to provide learning activities and experiences that are “authentic”. This includes games education. Shaffer and Resnick (1999) argue

that there are different uses of the terms “authentic learning” that are each mutually reinforcing, and that educators should try to design educational interventions that support them all. The four uses they identify are *real-world authenticity* (when the materials and activities align with the non-classroom world), *authentic assessment* (assessment is aligned with what students really should be learning), *personal authenticity* (when what is learned is aligned with what students want to know), and *disciplinary authenticity* (when methods of inquiry are aligned with disciplinary practices) (Shaffer and Resnick 1999).

In games education, especially when its orientation prioritizes the professional over the academic, classes where students work in teams developing a game are a focus for the kinds of authentic learning experiences Shaffer and Resnick recommend. For the purpose of this article I will discuss *real-world authenticity* and *personal authenticity* (Shaffer and Resnick 1999).

Most projects-based classes strive for *real-world authenticity* because students design and develop games similarly to what occurs in the games industry. Students often pitch ideas to faculty panels and/or game industry guests (e.g. Altizer and Zagal 2014) and manage their work using industry standard development methods (e.g. agile, scrum) (e.g. Kristiadi et al. 2019; Trier and Treffers 2021). Additionally, students use the same tools as industry professionals including game engines (e.g. Unreal, Unity) (e.g. Comber et al. 2019), art applications (e.g. Photoshop, Maya), programming languages (e.g. C++, Python, Lua) (e.g. M. M. McGill 2009; Chapter 4, Keogh 2023), and other support tools (e.g. version control software, communication support tools like Slack, etc.). Towards the end of the development process students are often encouraged to publish their games – making them publicly available on industry-standard digital game storefronts such as Valve’s Steam. Student game projects are also authentic to industry practice in that they are highly collaborative across different disciplines: artists, engineers, designers, producers, etc. all working together. This collaboration can happen within the same program or across academic departments.

The authenticity of making a game in a team is important because there are things that are simply not learned from other sources such as game development textbooks, game postmortems, and scholarly accounts of game development (Whitson 2020). This is partly because “learning to make games is emergent, embodied, and context-dependent” (Whitson 2020). As game educators, we want students to use their skills and face the kinds of problems they would encounter in an industry context. Common problems include technical and creative challenges. However, many of the problems in game development are related to soft skills including effective cross-discipline communication (Petrillo et al. 2009), collaboration (Berg Marklund et al. 2019), and coordinating dependencies (Petrillo et al. 2009).

The practice of game development for educational purposes mirrors that of the industry even when there are significant differences. For example, the educational context is often without financial constraints (e.g. no need to sell games for profit) and scheduling/timing issues might be more restrictive (e.g., students split their time with other classes and academic term deadlines are stricter than industry launch dates). Thus,

Assumption 1: *Student game projects are authentic to professional game studios at least along the dimensions of skills and soft-skills.*

Games education programs are also *personally authentic* (Shaffer and Resnick 1999) because students’ interest in these programs generally comes from a personal interest in games (i.e. they like playing games) and they want to participate in the industry that makes them (e.g. Chapter 4 in Keogh 2023). Students often have a desire for “dream

jobs” at AAA game companies that make the games they love and enjoy (Bulut 2020). While some students have entrepreneurial aspirations – e.g. start their own company to develop and release their own dream game –most are interested in the financial stability that working at large established companies (supposedly) provides. Therefore,

Assumption 2: *Most game students want jobs at large (AAA) studios working on popular titles.*

I think it’s fair to think of student game projects (and the classes those projects are developed in) as authentic to game studios in many ways. However, what kind of game studios are student game projects most like (i.e., authentic to)?

A study of game capstone classes found that the average class size was 15 students with a “reported worst-case scenario with 70+ students for a single instructor to manage.” (Zagal and Sharp 2011). Similar and/or smaller numbers have been reported in other cases (Jones 2000; Decker, Egert, and Phelps 2016). Furthermore, most classes have multiple teams of students working on different projects. Therefore, student game projects are mostly like small independent game studios in terms of their size and not like AAA studios.

One of the perceived benefits of student projects is their creative potential – students are free to explore their interests and creative aspirations in terms of the games they choose to make. Student rarely develop sequels to existing games nor license existing intellectual property. They are free to explore the expressive boundaries of games as a medium. Thus, student teams are similar to independent game studios. While indie game studios have financial considerations, the creative freedom they aspire to is one of the common explanations for their existence (Garda and Grabarczyk 2016).

Smaller indie studios often find themselves in situations where members must wear “many hats” (Hill-Whittall 2015) – team members might do art and game design, someone else handles programming and production, a third person might do narrative design while managing social media. Student teams are no different and they provide an excellent context in which to learn about work that is central to game development but often ignored or sidelined (quality assurance, community management, marketing, etc.). Therefore,

Assumption 3: *Student game projects are more like independent game studios than other kinds of studios due to their size and creative freedom.*

There is a contradiction when we consider these three assumptions. Students get an educational experience that is authentic (for some important variables) to working at an independent game studio when what they mostly want is to work at a larger AAA studio. This might be a problem if:

Claim 1: *There are fundamental differences between independent game studios and AAA studios in terms of the skills and knowledge required to be successful at them as a professional.*

For example, while all teams will have to deal with the challenges of effective communication and coordination, these challenges will take a different form and need to be addressed in different ways when a team is large. In support of Claim 1 (above) I offer a few examples illustrating how large studios are different from smaller independent ones. In large studios:

- A. It is rare for a single person to develop a game element from concept all the way to final implementation. Rather, many people work on different steps of the development – each handing off their work to the next person until it’s ready to be included in the game. For example, a game’s level might start with a level designer who then passes it to an artist, who then hands it off to an animation/VFX/lighting specialist, before finishing with a sound designer for the final touches (Lemarchand 2021, 86–87; Karlsson, Brusik, and Engström 2022).
- B. Team sizes change significantly over the development of a game – generally pre-production and early creative work requires a small team, while production sees teams grow in orders of magnitude with post-production (polish, release, further support) seeing the team shrink in size.
- C. Due to the scale and size of the production needs of AAA games, most developers do not have a sense of the entirety of the game – e.g., the entire storyline, all of the levels, etc. Instead, most people on the team share a vision of the game and work on their “part of the game” in a way that is aligned with and supports that vision. A common problem at large companies is misaligned teams where “people may establish different/diverging visions on the game design and development” (Politowski et al. 2020).
- D. It is rare to have team-wide communication and coordination (i.e. everyone on the team talking to anyone on the team). Instead, communication and coordination is limited and mirrors the structural organization of the team (e.g. sub-teams might communicate all-to-all, but across sub-teams communication is channeled through leads or a producer who liaises and coordinates). Communication and coordination problems, both among and across teams, are commonly reported in industry post-mortems (Petrillo et al. 2009; Politowski et al. 2020).
- E. There is diversity experience, knowledge and skill on a team (e.g. several levels of programmers from junior to senior). Managing this skill/knowledge differential creates another challenge commonly seen in game development (McKenzie et al. 2021).
- F. Individual team members have limited agency w/r to the overall organizational structure, processes, tools, and production pipeline.

Therefore,

Claim 2: *The commonly used model for team-based student developed projects should be changed in order to better support students in learning the skills and knowledge they need.*

To clarify why I believe Claim 2 should be studied more deeply, I will rephrase each lettered item above and articulate them as experiences students could benefit from in terms of their learning goals:

- A. Students should have practice working on something that someone else has started while also being prepared to hand off “incomplete” work for someone else to finish.
- B. Students should have experience working on teams of variable sizes including team sizes changing during the same project.
- C. Students should have practice doing creative work in a context of uncertainty where they do not have the big picture of what they are making, but rather need to understand someone else’s vision and do work that supports that vision.
- D. Students should practice managing communications between different teams and within teams – understanding when and how to share which information in order to better coordinate.

- E. Students should have practice working on teams where members have significantly different levels of skills and experience. Less experienced students should practice learning from those with more experience while those with more experience can practice providing leadership and direction for those with less.
- F. Students should practice adapting to different styles of organizational structure, processes and pipelines. More experienced students could design and determine structures and processes and provide guidance for less experienced students.

I believe that, in the context of current game development education, few team-based studio classes are equipped to provide the kinds of experiences outlined above. Additionally, while soft skills are important in all collaborative game creation contexts, there is nuance in these such that game development in a AAA-like context requires a different kind of soft skill. For instance, being able to communicate well with a small team is not the same as being able to communicate well in a small team and also across small teams. Next I propose the megateam (and variations) as a model to consider.

THE MEGATEAM MODEL

The megateam model is simple: have students work in teams as large as possible. If a capstone class has 70 students, instead of having them break up into smaller teams each working on different games, the entire class works on a single game. A team of 70 is comparable to a medium-sized studio and thus the typical modes of communication and coordination would rapidly cease to be effective (necessitating other solutions), the increased team size allows for specialization, and also makes it harder to maintain a single vision for a game. The megateam model could address items A, C, and D above.

Depending on the time available, a single term or multiple ones, it might not be possible for students to ideate, prototype, and pitch while leaving time to develop, polish, and finish the game. Thus, the instructors might have to provide a vision for the game that will be made. Similarly, course instructors could determine the structure and processes to use – allowing students to practice and learn a certain “way” of making games. Instructors with experience and skills in “running” a game studio might be best suited for this.

While the megateam model might rob students of “high-level” creative autonomy there are still plenty of opportunities for creative and impactful work –especially if the game they work on has significant content requirements (e.g. lots of levels, characters, dialogue, etc.).

The Staggered Megateam

The megateam approach does not address all of the concerns I raised (notably items B, E, and F). However, these could be addressed by changing the wider context of a game program.

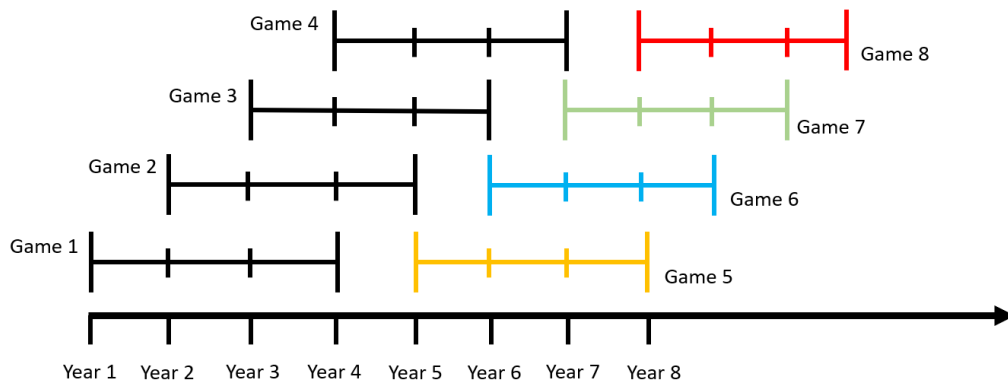


Figure 1: Staggered Megateam Plan with multiple multi-year student game projects in development.

Figure 1, above, shows a multi-year plan in which multiple games are worked on by students in a game program. In Year 4, four games are being developed at the same time (Games 1,2,3, and 4): Game 1 is in the final stages of development, Game 2 still has a year left, Game 3 is in its second year and Game 4 has just started. Rather than work on the same game for four years, students could transfer to a new game every year and participate in a different stage of its development. The game they work on would depend on their progress in the game program. For example, one variation might have 4th year students doing the pre-production work for a game – mostly research, concepting, and creating prototypes. In Year 4, this would mean they would start the development of Game 4. Since it is their last year, they would graduate not seeing the end of the development of the game they began. In this same year, 3rd year students are working on Game 3 (in its 2nd year of development), 2nd year students are working on Game 2 (in its 3rd year), and 1st year students are preparing Game 1 for release – wrapping up a game that began development years before they joined the program they are enrolled in.

The variation outlined above might seem backwards – the least experienced students (in their 1st year) are working on a game that is almost done. However, it is also the case that they are working on a game with the most scaffolding and support – the big decisions have been made (e.g. art direction, game architecture, core technologies, and key gameplay have been decided and implemented) and the work that needs improving should be easily identifiable. For 1st year students it would require becoming familiar with the vision of the game and aligning their contributions to that vision. On the opposite end, the more experienced students spend a year doing the sort of work that more senior members at a game studio would do (preproduction) including research, prototyping, worldbuilding, and setting the direction for the game that will be built: art, technical architecture, gameplay, etc.

This approach might require important curricular changes: students are taking a project-based class every term if we assume continuous development. There could be “lighter” implementations –perhaps the cycle is only 3 years long, with 1st year students excluded or even a two-year development period. Similarly, there could be years in which students only take one project-based class per academic year instead of two.

Multi-Generational Megateam

Neither of the alternatives above address item E: providing opportunities for students to work with others of different skill and abilities levels. The multi-generational megateam could address that. Here, a game project has a large number of students

(thus, “mega”) but not all the students are in their final year. The idea is that the older (soon to be graduating) students take on leadership roles on the team while the less experienced students function as the “junior colleagues”. The older students would be enrolled in a capstone course while the younger ones are in a pre-capstone course. Both courses would either meet together or could have an overlap in their class time.

Design Class Megateam Post-mortem

In August of 2022 I began teaching semester-long (~16 weeks of class time) graduate game design class in which I implemented a megateam with ~70 students. Due to it being a game design class, I felt it was important for everyone to work as game designers. So, a “no art” rule was implemented and the entire class worked on a boardgame so that programming was not required either. The learning goals were for the students to learn about and practice game design while also practicing the soft-skills necessary and important for the game to be developed on time.

70 students working on a single boardgame is obviously a “too many cooks” situation – most commercial boardgames are designed and developed by a single designer and collaboratively created boardgames rarely have more than a handful of designers involved. Accommodating such a large team of designers required a game type whose scope and scale implied a lot of “content” while allowing for work in specialized areas of game design. Thus, the “spec” for the game the students would have to design was as follows:

- 4-5 player,
- Collaborative,
- Campaign-style boardgame,
- Set in a cyberpunk universe,
- and playable in 30-45 minutes per session.

To heighten the need for large team soft skills I set up an organization with a pre-determined content pipeline, dependencies, and potential (likely) bottlenecks. Navigating the design process would thus require a lot of coordination, communication, and negotiation (i.e. figuring out specific responsibilities, how to manage deliverables, anticipating other’s needs, etc.). For this I outlined nine different teams with brief descriptions of responsibilities, expected team size, and relationships to other teams (see Figure 2 and Table 1). The class then self-selected into teams based on their interests.

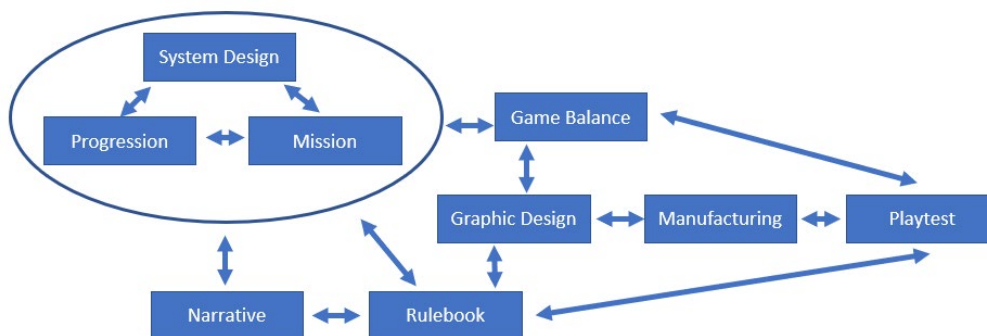


Figure 2: Design class team structure showing significant relations/dependencies.

| Team Name | Core Responsibilities | Interactions with Other Teams |
|--|---|--|
| System Design (4-6 members) | Game’s systems and mechanics – defining what they are and how they’re supposed to work. The most significant portion of this team’s work will happen early in the production process as “big picture” ideas are fleshed out, tested, and iterated on. Over the course of production this team’s work diminishes as the Progression and Mission teams iron out details, the Narrative team develops backstory and lore and the Game Balance team ensures everything fits together well. | Work closely with Progression and Mission teams since their work will depend on the core systems and mechanics. Work with the Narrative team to refine core mechanics and systems that align and reinforce the game’s cyberpunk theme. Work with the Game Balance team, providing them information and materials for them to tweak/modify to improve game balance. |
| Progression (4-5 members) | Overall arc (or arcs) of the gameplay experience as the players play the game’s different missions. This includes answering questions such as how can/should the game’s difficulty change over time, when to introduce new gameplay/mechanics (or not), what new gameplay should be introduced, and how the game evolves over the campaign. | Work closely with the Narrative , Missions , and System Design teams to make sure that, say, gameplay arcs coincide with narrative arcs, that new mechanics can be introduced in missions, and that they align well with the core system. Coordinate with the Game Balance team such that they can tweak and modify aspects to achieve the intended experience. |
| Mission (10-14 members) | Design of the individual missions that comprise the game’s campaign. It is expected that this team will a greater quantity of missions than will be in the final game and that these will be refined and narrowed down. | Work closely with Narrative , Progression , and System Design teams so missions make sense narratively and with the core system. Design new one-shot mechanics. Work with Narrative team for lore and text to be used to describe the missions. |
| Game Balance (5-6 members) | Ensure the game is balanced for fairness, difficulty, smooth progression, and duration. Most work will use spreadsheets and game system modeling software, e.g. Machinations. May not have much to do in the first sprint of the production phase (other than learning some game system modelling tools) but will ramp up after that – requires quick turnaround in order to not bottleneck the production process | Rely on Playtesting to see if the intended balance has been achieved. Receive work from System Design , Mission Design , and Progression teams that is changed, modified, etc. before handing it off to the Graphic Design . |
| Narrative Design (3-5 members) | Worldbuilding, outlining, and writing for the game including flavor text, mission descriptions, and similar. Character design (e.g. back stories) if present in the game are also the responsibility of this team. This also includes broader aspects such as planning narrative arcs, and more. Other than Rulebook team, this team will do the lion’s share of the writing for the game. | Work closely with System Design , Mission , and Progression teams to make sure the game’s mechanics and systems “make sense” with the game’s narrative, world, and cyberpunk thematic genre. Work with the Rulebook team to so style and writing is consistent across game materials. |
| Graphic Design (3-4 members) | Responsible for the barebones, serviceable and functional graphic design of the games’ printed elements. This means simple iconography, use of color, layout (rulebooks, cards, boards, etc.) and typesetting. Limited work in the early stages of production – the other teams will probably use very rough prototypes to do their initial rounds of design –will ramp up significantly once the game’s design starts to get locked in. | Providing game materials to the Manufacturing team so they can print/create them. Get files/information from the Game Balance and Rulebook teams. |
| Manufacturing (6-7 members) | Creation (printing, cutting, sleeving, etc.) of the game’s components (cards, boards, specially marked dice, etc.) as the game is being designed and developed. This team will probably not have much to do in the early stages of the design as other teams will use rough prototypes to do their initial rounds of design –work will ramp up significantly once the game’s design starts to get locked in and it becomes necessary to create multiple copies of different parts of the game for playtests and so on. | Providing game materials to the Playtest team for them to run their playtests. Get files/information from the Graphic Design team. |
| Playtest (18-20 members) | Organizing and conducting all playtests for the game. This includes playtests of sub-sections/systems in the game as well as the “full” game. Playtests will be internal and external though members of this team will often be (internal) players during the playtests. Work will ramp up significantly during production requiring quick turnaround in order to not bottleneck others. | Get materials from Manufacturing in order to run playtests. Get requirements/requests from the Narrative Design , Mission Design , System Design , Game Balance , Progression , Rulebook teams to run playtests to examine specific issues, problems, questions, etc. |
| Rulebook (2-4 members) | Responsible for ensuring players can learn to play the game from only the materials provided in the box. This is accomplished via the game’s rulebook and similar documents (e.g. “Quick Start Rules”, “Mission Book”). Also copyedits materials received to ensure consistency and style. Initial work should consist of determining processes and standard to be used. | Providing game materials to the Graphic Design team so they can layout, typeset them. Get files/information from the Narrative , Mission Design , and System Design teams. Implement feedback from Playtesting . |

Table 1: Megateam sub-teams with core responsibilities and relationships to other sub-teams

The development schedule for the game was organized in three traditional phases: pre-production, production, and polish/release (e.g. Dubois and Weststar 2022).

1. **Pre-Production:** Students did research by playing current commercial collaborative campaign-style boardgames and by doing thematic research on the sci-fi genre of cyberpunk. They then reported via documents and class presentations on their findings. For the games, they shared what they learned in terms of mechanics, gameplay, and systems. For the cyberpunk thematic research, they had to choose a title from a list provided (e.g., movie, book, comic) and submit a short document summarizing the source, highlighting elements they thought interesting to borrow, incorporate, or build upon – e.g., a cool gadget or concept. The pre-production phase concluded with a class discussion where the game’s core design pillars were determined.
2. **Production:** This phase consisted of 4 two-week sprints, with a week off between the 2nd and 3rd sprint. The goal for the production phase was to have a Beta-version of the game (a self-contained playable game with all core mechanics implemented and narrative components complete). Goals were set for each sprint with weekly meetings to track progress and identify problems. At the end of each sprint, progress was evaluated and goals outlined for the next.
3. **Polish/Release:** This phase was reserved for a reduced number of students to make any final changes needed (e.g. final proofread of rulebook) and to prepare any digital files such that they could be distributed (as a print and play game) including uploading to a print-on-demand site for boardgames (e.g. TheGameCrafter.com). This phase also included a public presentation of the game at the end-of-the year expo organized by the game program.

For grading and assessment, I used a “Contract Grading” system. Contract grading, broadly, is when the students and instructor agree on what the student will do in order to get a certain grade (Radican 1997). So, a student who is fine with a lower grade can do less work – but to a pre-defined contract. I asked students to indicate what they wanted to work on, including leadership roles in any of the teams, and to what level of commitment. Students could opt to be “Full-time” or “Part-time” setting their time weekly time commitment to ~9 hrs/week or ~4.5 hrs/week. Students happy with a lower grade could decide to be “Part-time”, not choose a leadership position, and also be on the Playtesting or Manufacturing team as these were presented as having opportunities requiring less investment/involvement (e.g. you just have to show up to play the game or print game materials). Over the course of the semester students could change teams (between sprints), leave or assume leadership positions, and adjust their contracts (for more or less work).

I now report on the results of this course design in the form of a post-mortem (Washburn et al. 2016) by outlining those things that went right and wrong. For reasons of space I focus primarily on things related to the rationale for trying a megateam class: i.e. provide students with learning opportunities authentic to the soft skills required for working at a large game studio.

What Went Right

Research during Pre-Production: The pre-production phase was successful in that I correctly anticipated that most students would neither have experience with campaign-style collaborative boardgames of the kind they were tasked with designing nor would they have familiarity with the cyberpunk genre. While no one became an expert in either game style nor genre, the shared experience of conducting this research was eye-opening to many. I realized in hindsight that this experience is uncommon in project-

classes where students develop games for entertainment: students start by brainstorming and pitching an idea for a game but, rarely in my experience, do some research. Here due to the constraints of the spec, students realized that they didn't have significant knowledge or experience to rely on and thus had to "get up to speed".

Collaboratively Generated Design Pillars: Design pillars are commonly used in the game industry – especially on large AAA titles. Rather than having these dictated by myself, they were generated by the class as a whole – students formed small groups during class time and reported back on what they wanted as design pillars. The suggested pillars were then iterated on live in class, consolidated, and refined in a moderated discussion (Figure 3, below). The design pillars proved surprisingly useful for offering critique on the state of the game and to (guide) design discussions and deliberations. Because the pillars were developed by the class as a whole in a process where everyone participated and witnessed how they were refined, there was enough "buy-in" and understanding by the students as to what kind of game they were trying to make. The design pillars also prevented the game from going "off the rails" in pivoting in new design directions based on whims or interests of the moment. In some contexts they were useful in helping settle conflicts – someone arguing for an element had to justify how it supported the design pillars (ideally more than one).



Figure 3: Design Pillars for Megateam Student Game Project.

Clear Delineation of Opinion-Havers (Chickens) and Decision-Makers (Pigs): There is a story used in scrum/agile development to articulate differences between roles. The story is about a chicken and a pig collaborating on a bacon-and-eggs breakfast. Both animals are involved, but only the pig is committed (Blankenship, Bussa, and Millett 2011). We discussed this story in the context of the different teams and talked about how most people would be "chickens" for the other teams and would be "pigs" in their own team. The idea is that teams make decisions for things that are under their purview and that they are responsible for delivering. Others (chickens) can offer input and suggestions – but the decision is not up to them and whatever the team (pigs) decide, they have to accept and work with. For example, if the Narrative team decides that the game's storyline will go in a certain direction – everyone else needs to accept their decision. While everyone can have and share an opinion, only the decision-

makers decide since they are responsible for delivering on those things under their purview.

In my experience student teams rarely operate this way. There are usually either strong-willed individuals to coerce teammates into doing what they want (regardless of the disciplinary area or responsibilities) or everyone bends over backwards to achieving unanimous consensus. This was clearly impossible for a team this large. However, strongarming and/or consensus-drive decision making are achievable at the sub-team level (e.g., the narrative team can use consensus for their own decision-making). The chicken/pigs metaphor allowed teams to take ownership of their “area” in a way that gave them freedom and autonomy in their internal processes and decision-making.

Class Pre-Mortem and Post-Mortem: At the beginning of the semester I explained to the students what the plan was and articulated some of the reasons why it might be a valuable learning experience or how it could go off the rails. As an initial class activity, students brainstormed what success could look like (e.g. “We did not burn out”, “We all came together under one vision”, “We have a functional understanding of the game design process”, etc.) as well as what they thought would be the main barriers that would stand in the way of the success outlined earlier (e.g. “poor time management”, “navigating dependencies”, “resolving disputes”, etc.). I referred to the results of this brainstorming session (a pre-mortem of sorts) throughout the semester and held a class post-mortem at the end of the semester. During the post-mortem I presented both lists (success and barriers). The class post-mortem then centered on discussing whether or not we had seen any of the successes listed and, for the barriers identified, which ones were experienced strongly.

Even though the project ended in a failure of sorts (more on this later), the class identified the following as successes (even if they were achieved at a certain cost/pain): “We learned throughout the process”, “We have a functional understanding of the game design process”, “We all came together under one vision”, and “No jerks”. Of the barriers, students felt strongly that ten of the 17 barriers originally identified became major issues impeding the success of the project: “poor time management”, “unclear goals”, “navigating dependencies”, “uneven distribution of labor”, “the process may not be enjoyable”, “ignoring deadlines and dependencies”, “wasted work/redundancy”, “lack of commitment”, “communication/coordination”, and “resolving disputes”.

I’ve listed this as a success because the class discussion on the topic was reflective and insightful. A key insight for the class was the realization that a group of smart, creative, and motivated people can still fall prey to these problems even with full knowledge and cognizance that they will probably occur. In other words, there was pedagogical value for the students to realize that despite our discussion of problems they identified as potential issues at the beginning of the semester and regular check-ins to discuss them during the semester these problems still happened. The problems happened not because the students were “bad at game design”. They happened because they are hard to address and deal with and because, generally, no singular person is “responsible” for the success of a group this large. In small group dynamics it can be easy to blame a singular bad actor or pin all the issues on a few failures of communication between two people. For a large group things become increasingly more complex, harder to track and manage, even as everyone in the large group sees problems as they occur. While I do not think I was able to offer solutions – greater awareness of the scale of the challenge was eye-opening to many students. In a sense, the insight here is that soft-skills related issues are not “solved” – rather they must always be examined, reflected upon, and continually worked on.

What Went Wrong

Hard to Work “Ahead of Schedule”: Figure 2 depicts the general organization and expected relationships between teams. It outlines which teams interact with each other, specifically with regards to “work product”. I purposefully created an organization with built in bottlenecks: Graphic Design -> Manufacturing -> Playtesting is perhaps the most significant. While a better pipeline could have been designed, I felt that having a lived-experience of being part of a long chain of dependencies would be pedagogically unique and valuable insofar as realizing the importance of both thinking ahead, understanding responsibilities to get things done on time, and so on. As far as bottlenecks go, it was wildly successful – to the project’s detriment but possibly also to the student’s learning benefit.

The main issue was not the bottlenecks themselves, but rather the challenge students had in thinking a few steps ahead, especially for teams earlier in the pipeline. This made it hard to align the team’s goals with those of the sprint. For example, consider a goal for a two-week sprint is to have five missions implemented and playtested in the game. In a “worst-case” scenario, the Playtesting sub-team would have to get a copy of the game with those missions a few days before the end of the sprint so as to have enough time to playtest the game before the end of the sprint. To do so, the manufacturing team would need to get the materials from graphic design a few days before that. Therefore, the graphic design team would need the information some days before that, and so on. So, the team at the beginning of this process would need to have their work done by the end of the 2nd day of the sprint just so that everyone else further down the pipeline could get their work done on time so as to hit the goal for that sprint.

In order for everyone to have something to do all the time, it became necessary to coordinate and look “a few sprints ahead” – so in sprint 2, for example, the Mission sub-team might need to start working on things that would be part of the goal for sprint 4. This kind of planning is not trivial and I, naively, assumed that we would be able to “work it out” along the way. In truth, we should have probably spent an entire week just planning all sprints ahead of time just to see what was feasible in terms of time required to complete tasks, student availability during the sprints, and so on. Problems with planning are commonly reported in industry postmortems with underestimating time to complete tasks being the most common (Politowski et al. 2020), here my offloading of this responsibility to the students was perhaps unfair for them to have to deal with and I should have provided either more support or more scaffolding to help them sort things out.

Dwindling Motivation: Over the course of the semester, as the schedule began to slip, and it became harder to see forward momentum on the project, students began to lose motivation. This decrease in motivation made it, obviously, harder to get things back on track. The final result is that the goals for the 4th sprint were not met, and failed to be achieved even with an additional week. Ultimately the 4th sprint goal was only (partially) met in the post-production phase by a few students explicitly tasked with collecting all the materials. Perhaps most surprisingly, given the high level of excitement at the beginning, the project did not “die”. It simply fizzled out and the grandiose plans of presenting the game at the end-of-year expo were shuttered when an overwhelming majority of the class voted not to present anything at all. As instructor I failed to anticipate the challenges of maintaining motivation throughout all of the sprints and reacted too slowly, and too late, when it started to wane.

Not Enough Scaffolding for Coordination and Communication: I explicitly wanted to highlight the importance and challenge of effectively coordinating and communicating. However, if there is a balance between managing all of that for the students (assuming that it is even possible) and having them come up with their own

processes and methods – I erred on the side of not supporting them enough. The result was that students recognized the challenge and importance in doing this effectively, but that recognition was perhaps too much as it effectively knee-capped the ability for the project to make effective forward progress. There was not enough time to decide on different ways to manage their communication, establish practices, and then modify them based on their success or failures. There are also no obvious nor easy solutions, e.g. large game companies suffer from these problems as much as we did, but I could have relied less on the students to figure this out, and more on setting up clearer structures and processes that I could have followed up on and held students accountable to. In other words, here I should have been more authentic to a large studio experience wherein new hires are onboarded to existing processes and communication methods and channels rather than tasked with figuring these out themselves – especially at the scale of the entire team (e.g. sub-teams could figure out their own processes and communication methods).

Ineffectual Contract Grading: I have used a different form of contract grading in another class I co-teach and I believe that it can be used to empower and liberate students – encouraging them to assume ownership of their learning experiences in a clear and direct way. I definitely want to use it again in this class. However, I learned that I should have provided much clearer guidelines and expectations for students, especially as the concept of contract grading was novel and confusing for many. The concept of “full-time” and “half-time” was valuable and I would use it again as a metric for effort and commitment, but I would have to provide them with more clearly defined “templates” that better set expectations, minimized feelings of “unfairness” as some students worked significantly more than others for the same grade, and overall recognized the desired intentions to commit. Because I failed to adequately “track” and follow up on each student’s contract (for each sprint), it also became increasingly onerous to hold students accountable for failing to meet their own stated goals.

DISCUSSION

I began this article outlining assumptions about game project-based courses and their relationship to the industry in terms of authenticity. My argument omitted a baseline assumption:

Assumption 0: *It is pedagogically desirable for game students to have experiences authentic to the game industry.*

The game industry has been widely and fairly criticized for a variety of labor-related problems such as exploiting their workers (e.g. crunch), high rates of burn out and turnover, and the like (e.g. Bulut 2020). The reality of working in AAA game production is that it can be as de-humanizing and oppressive as it can be rewarding and fulfilling – and this varies across and within studios and positions. To be fair, there are similar issues and concerns in smaller independent studios (Whitson, Simon, and Parker 2021). So, does a megateam approach simply uncritically ape the worst practices of large profit-driven corporations? I think not. Though it is a danger we should be wary of (Harvey 2019). How do we provide experiences that are authentic to the positive aspects of large-scale game development while alerting students to the negative ones and empowering them to protect themselves and be agents of positive change?

As game educators we have a responsibility to help students with their career goals and aspirations. We also have a responsibility to contribute to both the industry and society as a whole. We are neither beholden to the industry, nor can we ignore it. Ideally, those students of ours who so desire it will be able to succeed in an industry they are excited to participate in while at the same time contributing towards improving it from within. When I speak of helping them succeed I do not mean it in the sense of “preparing cogs

for a machine” where the game industry “knows best”. I mean in the sense of (1) being able to find employment (which is a huge challenge given how competitive the industry is, and how dominating the larger studios are in terms of numbers of positions available compared to small indie studios) (Parker, Cox, and Thompson 2014; Darchen and Tremblay 2015) and then (2) helping the game industry make positive progress. There are plenty of large studios that have positive workplace environments and healthy production practices. We should try to identify these, learn from them, and figure out how to provide learning experiences that are authentic to them as well.

CONCLUSION

I began this paper by challenging our assumptions and notions of what the role of student game projects are in the games educational experience. We use them to provide students with a context for practicing and developing their game development skills. We also expect them to realize that soft-skills, e.g. collaboration, communication, coordination, are fundamental to succeeding in game development. We do this because these experiences are authentic to professional game development. However, there is a mismatch between the kind of professional game development students practice (e.g. small studio, indie) and what most students aspire to and will (likely) end up practicing (i.e. large studio AAA). I have argued that the difference in size and scale of the game development practice requires different skills and abilities – or, more precisely, it’s the same skills and abilities but with important differences. Therefore, we should challenge our notions of the current structure and size of student teams and find models that might be better suited – in the authenticity sense – for our educational goals. I presented one model called the megateam – to simply scale up the size of teams working on a project, and I described a few variations on this idea before presenting a brief post-mortem of one megateam experience.

While the megateam experience was not without flaws, and it failed to deliver a completed game project within the period of time available, it was nonetheless a valuable learning experience for its participants (students and instructor). More broadly, this demonstrates there is potential for experimenting in this space and hopefully this paper serves as a starting point for the kinds of conversations and experiments we should be taking on. There are probably other models we could implement. The game industry is constantly changing, not just in the nature of the games it develops, but also in how they are made. As such, it is our responsibility as educators to ensure that we are always updating, revisiting and reimagining our assumptions – if we are to remain authentic to a certain referent, we must ensure that as that referent changes, so do we.

BIBLIOGRAPHY

- Altizer, Roger, and José P. Zagal. 2014. “Designing Inside the Box or Pitching Practices in Industry and Education.” In *Proceedings of DiGRA 2014*. Snowbird, Utah: DiGRA. http://www.digra.org/wp-content/uploads/digital-library/digra2014_submission_134.pdf.
- Berg Marklund, Björn, Henrik Engström, Marcus Hellkvist, and Per Backlund. 2019. “What Empirically Based Research Tells Us About Game Development.” *The Computer Games Journal* 8 (3): 179–98. <https://doi.org/10.1007/s40869-019-00085-1>.
- Bidarra, Rafael. 2011. “Interdisciplinary Game Projects: Opening the Graphics (Back) Door with the Soft Skills Key.” *Eurographics (Education Papers)*, 9–16. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=992a6c1458c8ab5e117cc17a6c378fc375b00e29>.
- Blankenship, Jerrel, Matthew Bussa, and Scott Millett. 2011. “Managing Agile Projects with Scrum.” In *Pro Agile. NET Development with Scrum*, 13–27. Apress L. P.

- Bouchard, Matt, and Sean Gouglas. 2021. "Computers and Games (INT D 450)." In *Teaching the Game: A Collection of Syllabi for Game Design, Development, and Implementation Vol. 2*, edited by Richard E Ferdig, Emily Baumgartner, and Enrico Gandolfi, 421–35. Pittsburgh, PA: ETC Press.
- Brown, Quincy, Frank Lee, and Suzanne Alejandre. 2009. "Emphasizing Soft Skills and Team Development in an Educational Digital Game Design Course." In *Proceedings of the 4th International Conference on Foundations of Digital Games*, 240–47. FDG '09. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1536513.1536557>.
- Bulut, Ergin. 2020. *A Precarious Game: The Illusion of Dream Jobs in the Video Game Industry*. Cornell University Press. <https://ecommons.cornell.edu/bitstream/handle/1813/103967/9781501746543.pdf?sequence=1>.
- Comber, Oswald, Renate Motschnig, Hubert Mayer, and David Haselberger. 2019. "Engaging Students in Computer Science Education through Game Development with Unity." In *2019 IEEE Global Engineering Education Conference (EDUCON)*, 199–205. <https://doi.org/10.1109/EDUCON.2019.8725135>.
- Darchen, Sebastien, and Diane-Gabrielle Tremblay. 2015. "Policies for Creative Clusters: A Comparison between the Video Game Industries in Melbourne and Montreal." *European Planning Studies* 23 (2): 311–31. <https://doi.org/10.1080/09654313.2013.865712>.
- Decker, Adrienne, Christopher A. Egert, and Andrew Phelps. 2016. "Splat! Er, Shmup? A Postmortem on a Capstone Production Experience." In *2016 IEEE Frontiers in Education Conference (FIE)*, 1–9. <https://doi.org/10.1109/FIE.2016.7757399>.
- Diefenbach, Paul J. 2008. "Teaching Soft-Skills: Digital Game Development in a Multi-Discipline Environment." In *Eurographics (Education Papers)*, 9–13.
- Dubois, Louis-Etienne, and Johanna Weststar. 2022. "Games-as-a-Service: Conflicted Identities on the New Front-Line of Video Game Development." *New Media & Society* 24 (10): 2332–53. <https://doi.org/10.1177/1461444821995815>.
- Garda, Maria B., and Pawel Grabarczyk. 2016. "Is Every Indie Game Independent? Towards the Concept of Independent Game." *Game Studies* 16 (1). <http://gamestudies.org/1601/articles/gardagrabczyk>.
- Harvey, Alison. 2019. "Becoming Gamesworkers: Diversity, Higher Education, and the Future of the Game Industry." *Television & New Media* 20 (8): 756–66. <https://doi.org/10.1177/1527476419851080>.
- Hewner, Michael, and Mark Guzdial. 2010. "What Game Developers Look for in a New Graduate: Interviews and Surveys at One Game Company." In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 275–79. SIGCSE '10. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1734263.1734359>.
- Hill-Whittall, Richard. 2015. *The Indie Game Developer Handbook*. CRC Press.
- Hogue, Andrew, Bill Kapralos, and François Desjardins. 2011. "The Role of Project-based Learning in IT: A Case Study in a Game Development and Entrepreneurship Program." *Interactive Technology and Smart Education* 8 (2): 120–34. <https://doi.org/10.1108/17415651111141830>.
- Jones, Randolph M. 2000. "Design and Implementation of Computer Games: A Capstone Course for Undergraduate Computer Science Education." In , 260–64. ACM.
- Karlsson, Tobias, Jenny Bruska, and Henrik Engström. 2022. "Level Design Processes and Challenges: A Cross Section of Game Development." *Games and Culture* 0 (0): 15554120221139228. <https://doi.org/10.1177/15554120221139229>.
- Katchabaw, Mike, Derek Elliott, and Stephen Danton. 2005. "Neomancer: An Exercise in Interdisciplinary Academic Game Development." In *Proceedings of the*

- DiGRA 2005 Conference*. Vancouver, Canada: DiGRA. <http://www.digra.org/wp-content/uploads/digital-library/06275.08442.pdf>.
- Keogh, Brendan. 2023. *The Videogame Industry Does Not Exist*. Cambridge, MA: MIT Press.
- Kessler, Robert, Mark van Langeveld, and Roger Altizer. 2009. "Entertainment Arts and Engineering (or How to Fast Track a New Interdisciplinary Program)." *ACM SIGCSE Bulletin - SIGCSE '09* 41 (1): 539–534.
- Kristiadi, Dedy Prasetya, Ferry Sudarto, Dian Sugiarto, Richard Sambera, Harco Leslie Hendric Spits Warnars, and Kiyota Hashimoto. 2019. "Game Development with Scrum Methodology." In *2019 International Congress on Applied Information Technology (AIT)*, 1–6. <https://doi.org/10.1109/AIT49014.2019.9144963>.
- Lemarchand, Richard. 2021. *A Playful Production Process: For Game Designers (and Everyone)*. Cambridge, MA: MIT Press.
- McGill, Monica. 2008. "Critical Skills for Game Developers: An Analysis of Skills Sought by Industry." In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, 89–96. Future Play '08. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1496984.1497000>.
- McGill, Monica M. 2009. "Defining the Expectation Gap: A Comparison of Industry Needs and Existing Game Development Curriculum." In *Proceedings of the 4th International Conference on Foundations of Digital Games*, 129–36. FDG '09. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1536513.1536542>.
- McKenzie, Tim, Miguel Morales-Trujillo, Stephan Lukosch, and Simon Hoermann. 2021. "Is Agile Not Agile Enough? A Study on How Agile Is Applied and Misapplied in the Video Game Development Industry." In *2021 IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE)*, 94–105. <https://doi.org/10.1109/ICSSP-ICGSE52873.2021.00019>.
- Parker, Rachel, Stephen Cox, and Paul Thompson. 2014. "How Technological Change Affects Power Relations in Global Markets: Remote Developers in the Console and Mobile Games Industry." *Environment and Planning A: Economy and Space* 46 (1): 168–85. <https://doi.org/10.1068/a45663>.
- Parmentier, Guy, and Thierry Picq. 2016. "Managing Creative Teams in Small Ambidextrous Organizations: The Case of Videogames." *International Journal of Arts Management* 19 (1): 16–30. <http://www.jstor.org/stable/44989675>.
- Petrillo, Fábio, Marcelo Pimenta, Francisco Trindade, and Carlos Dietrich. 2009. "What Went Wrong? A Survey of Problems in Game Development." *Computers in Entertainment* 7 (1): 13–22.
- Phelps, Andrew, Christopher Egert, and Aaron Cloutier. 2021. "Magic Production Studio (IGME-580)." In *Teaching the Game: A Collection of Syllabi for Game Design, Development, and Implementation Vol. 2*, edited by Richard E Ferdig, Emily Baumgartner, and Enrico Gandolfi, 764–77. Pittsburgh, PA: ETC Press.
- Politowski, Cristiano, Fabio Petrillo, Gabriel Cavalheiro Ullmann, Josias de Andrade Werly, and Yann-Gaël Guéhéneuc. 2020. "Dataset of Video Game Development Problems." In *Proceedings of the 17th International Conference on Mining Software Repositories*, 553–57. MSR '20. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3379597.3387486>.
- Radican, Lynda S. 1997. "Contract Grades: An Agreement between Students and Their Teachers." *Alternatives to Grading Student Writing*, 285–90.

- Redfield, Carol. 2021. "Game Development (CS4335;CS6335)." In *Teaching the Game: A Collection of Syllabi for Game Design, Development, and Implementation Vol. 2*, edited by Richard E Ferdig, Emily Baumgartner, and Enrico Gandolfi, 479–507. Pittsburgh, PA: ETC Press.
- Shaffer, David Williamson, and Mitchel Resnick. 1999. "'Thick' Authenticity: New Media and Authentic Learning." *Journal of Interactive Learning Research* 10 (2): 195–215.
- Shahani, Noor Un Nisa. 2020. "HR: The Study into Recruitment and Selection in the Growing Game Development Industry: Importance of Soft Skills." *SSRN*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3902525.
- Suvajdz, Marko, Gabriel Coleman, and Jordan Smith. 2021. "Game Design and Production (DIG 4527)." In *Teaching the Game: A Collection of Syllabi for Game Design, Development, and Implementation Vol. 2*, edited by Richard E Ferdig, Emily Baumgartner, and Enrico Gandolfi, 798–811. Pittsburgh, PA: ETC Press.
- Trier, Kristin Kiri, and Theresa Treffers. 2021. "Agile Project Management in Creative Industries: A Systematic Literature Review and Future Research Directions." In *2021 IEEE Technology & Engineering Management Conference - Europe (TEMSCON-EUR)*, 1–8. <https://doi.org/10.1109/TEMSCON-EUR52034.2021.9488611>.
- Washburn, Michael, Pavithra Sathiyarayanan, Meiyappan Nagappan, Thomas Zimmermann, and Christian Bird. 2016. "What Went Right and What Went Wrong: An Analysis of 155 Postmortems from Game Development." In *Proceedings of the 38th International Conference on Software Engineering Companion*, 280–89. ICSE '16. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2889160.2889253>.
- Whitson, Jennifer R. 2020. "What Can We Learn From Studio Studies Ethnographies?: A 'Messy' Account of Game Development Materiality, Learning, and Expertise." *Games and Culture* 15 (3): 266–88. <https://doi.org/10.1177/1555412018783320>.
- Whitson, Jennifer R., Bart Simon, and Felan Parker. 2021. "The Missing Producer: Rethinking Indie Cultural Production in Terms of Entrepreneurship, Relational Labour, and Sustainability." *European Journal of Cultural Studies* 24 (2): 606–27. <https://doi.org/10.1177/1367549418810082>.
- Zagal, José P., and John Sharp. 2011. "A Survey of Final Project Courses in Game Programs: Considerations for Teaching Capstone." In *Proceedings of DiGRA 2011*. Hilversum, The Netherlands: Digital Games Research Association. <http://www.digra.org/wp-content/uploads/digital-library/11307.14290.pdf>.