Gameful Employments & Student Studios:
Implementing emerging models for videogame production

White Paper

Author: Shavon White
10 December, 2019
Table of Contents

Introduction .............................................................................................................................................. 3

Building Block Games: A genre on the rise .......................................................................................... 3

Decentralized Game Studios: An emerging model .................................................................................. 4

Background ........................................................................................................................................ 4

Outline .............................................................................................................................................. 5

RESEARCH .......................................................................................................................................... 6

Game Development Environments (GDEs): A pairwise comparison .................................................. 6

METHODS ........................................................................................................................................ 6

RESULTS .......................................................................................................................................... 7

ANALYSIS ......................................................................................................................................... 8

Workplace decentralization: A SWOT analysis ..................................................................................... 8

METHODS ........................................................................................................................................ 8

RESULTS .......................................................................................................................................... 9

ANALYSIS ......................................................................................................................................... 10

CASE STUDY: Making a Building Block Game .................................................................................... 12

OVERVIEW ....................................................................................................................................... 12

RESULTS .......................................................................................................................................... 12

ANALYSIS ......................................................................................................................................... 13

CONCLUSION .................................................................................................................................... 14

Looking towards a Brock game studio ................................................................................................. 14

Action Items ....................................................................................................................................... 14

Final takeaways .................................................................................................................................. 15

A strategic opportunity ....................................................................................................................... 15

Works Cited ....................................................................................................................................... 16

Acknowledgements ............................................................................................................................. 17
Introduction

It is easier than ever for game creators to create short yet provocative games with minimal resources. For a long time, games have been developed almost exclusively through a centralized studio model where all team members work together in the same physical space. As time has passed and technology has improved, however, new advancements in information sharing systems have allowed for the creation of a new kind of production model: decentralized studio production. This production model is defined by the lack of a central workspace and the involvement/input of developers from all over the world. Decentralized studios offer developers many different benefits such as remote collaboration, fast asset acquisition, and work opportunities that may not have been available otherwise. They also offer students and junior designers/developers an opportunity to work on their asset production skills and create usable, portfolio ready content without having to secure a position within a game development company. Decentralized studios can be started by virtually anyone with internet access at anytime, as they effectively remove the need for high-production space and associated costs.

Building Block Games: A genre on the rise

A Building Block Game is a term that represents a quickly produced prototype product that is collaborated on by several parties through remote asset creation. A Building Block Game can be assembled relatively quickly and contain a level of polish equal to the total time spent creating each asset overall. For example, where a regular game jam game might be created within a weekend, and thus have a minimal level of polish, a Building Block Game would have various elements made by various authors over a larger amount of collective work time, leading to a much more polished experience being delivered in a fraction of the time. While several recent AAA titles have received public criticism for severe bugs and other gameplay issues, smaller prototypical games such as Flappy Bird and QWOP have thrived and allowed individual developers to achieve widespread public recognition. As such, small and provocative games of this type are uniquely suited to students entering the game industry and seeking public recognition for their work.

Similarly to the industry disrupting introduction of crowdfunding as an alternative to project funding, advancements made in the field of cloud-based storage technology now make it possible for people to collaborate much easier than it’s been in the past. Through services such as GitHub, Slack, and Microsoft OneDrive, not only can individuals work remotely and upload their content to one central server, they can also access that content from anywhere with an internet connection. This means that
work that previously had to be done in a central office space can be done remotely. While this doesn’t eliminate the cost of hiring a specialized worker, it does drastically broaden the range within which people can collaborate. This has led to many game developers working to produce assets from home while still operating as part of a studio team. These advancements have also led to the creation of massive repositories of asset files and data, and with them the opportunity for file sharing on a massive scale.

Decentralized Game Studios: An emerging model

Defined by their lack of a centralized workspace, decentralized studios are a new model of game creation that allows projects to be worked on by numerous people remotely and asynchronously. Decentralized studios can be set up virtually anywhere with internet access and can forego costly overhead expenses that centralized studios have to work around.

For students interested in game design, getting experience in a professional work environment can be difficult, especially in the early stages of their academic career. One of the major contributing factors to this is the need for a strong portfolio in order to secure a production position. Although starting a centralized student game studio is one possible solution to this problem, the upfront and overhead costs for running a centralized studio are quite high. Under a decentralized, student-run studio model, students and alumni would be able to develop their skills and portfolios while also creating shareable assets to help their current and future peers. Games released under the studio name, as students come and go over time, would gradually build the brand-reputation of the studio, and of Brock, thus providing compounding benefits to students over time.

In order for this student studio to begin, students would need to operate within one of the many available game development environments (GDEs) in use today. To that end, this paper first looks at four industry standard GDEs that are currently available and compares their efficiency for different tasks. Next, a SWOT analysis is used to compare centralized and decentralized production models. Finally, the results of a case study testing the findings of the GDE and SWOT analyses are shared.

Background

For those not familiar with the current game development landscape, here is a brief overview of the 4 GDEs that are looked at in this paper:

**Dreams** is an application created by MediaMolecule designed to make the game production process as approachable as possible. It allows users to create 3D and 2D graphics as well as general audio design. Any content created in dreams has the
ability to be labelled as “remixable” which allows other users to use said content in their own projects.

**PICO-8** is a fantasy console, meaning it is an emulated version of an retro 8-bit console that never existed. PICO-8 allows developers to make tilesets, characters, music/audio, and levels all within the same program, meaning all aspects of the game’s production can be done without having to change apps or import files.

**Godot** is a free, open source GDE. Godot has many of the same features as the other GDEs. While it is a powerful engine, it is relatively new and therefore lacking some of the polish and quality of life improvements (and community support) of other GDEs such as Unity and Unreal.

**Unity** is one of the most popular GDEs in use today. It is widely seen as the go to GDE. Unity’s asset store is a huge benefit to the platform, allowing for collaboration on a large scale while offering an opportunity for content creators to sell their work (enabling them to use the platforms as a source of income). As with most GDEs, Unity uses code-based programming in the form of C# or UnityScript.

**Outline**

In order to rank the GDEs and extrapolate best practices for a decentralized student game studio, this paper uses a pairwise comparison chart and a SWOT analysis to compare and contrast the functionalities, features, and affordances of different existing solutions to determine which is ideal for the rapid prototyping nature of a student game studio. Additionally, a brief case study shares the results of creating a building block game using the highest-ranked GDE in the pairwise comparison chart and the most effective studio model as determined by the SWOT analysis. Based on the results of this case study, the paper concludes with a list of concrete action items for implementing a student-run game studio at Brock University.

The principal functions compared between the GDEs are:

1. **Accessibility** - how many platforms are supported by the GDE? How many different platforms can the GDE publish to? Are the assets dependent on the application or can they be made and archived separately?
2. **Ease of collaboration** - how many clicks does it take to find, download, and import another developer’s asset into a scene?
3. **Environmental fragmentation** - how many separate applications are required in order to make the different aspects of the game (e.g. Sound, graphics, etc.)?
RESEARCH

Game Development Environments (GDEs): A pairwise comparison

METHODS
To see which GDE is best for prototype design, a pairwise comparison chart shows the number of clicks needed to complete a task. A click is defined as any time the player must press a button of confirmation in order to progress. If there is any required coding, it is considered as less efficient than any program that does not require coding. The chart is also made on the assumption that the user has internet access and can access files located on a shared server. This standardized process of evaluation offers a repeatable method of assessment across development environments.

Official tutorials were used to find the most efficient methods of completing each task.

Some of the features compared are:
1. How many clicks does it take to start a new scene?
2. How many clicks does it take to find and import another person's uploaded assets?
3. How many clicks does it take to create a first-person character controller?
4. How many clicks does it take to add a song that plays immediately when the scene is loaded?
5. How many clicks does it take to give a character health properties?
6. How many options support the finished product as an exported file?
These each represent an aspect of a minimal viable product for a modern interactive experience.
## RESULTS

### PAIRWISE COMPARISON CHART

**CREATE A FIRST PERSON CONTROLLER**

<table>
<thead>
<tr>
<th></th>
<th>Unity</th>
<th>Godot</th>
<th>PICO-8</th>
<th>Dreams</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)Unity</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(B)Godot</td>
<td>A</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(C)PICO-8</td>
<td>—</td>
<td>—</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(D)Dreams</td>
<td>D</td>
<td>D</td>
<td>—</td>
<td>X</td>
</tr>
</tbody>
</table>

### START A NEW SCENE

<table>
<thead>
<tr>
<th></th>
<th>Unity</th>
<th>Godot</th>
<th>PICO-8</th>
<th>Dreams</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)Unity</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(B)Godot</td>
<td>B</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(C)PICO-8</td>
<td>C</td>
<td>C</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(D)Dreams</td>
<td>D</td>
<td>D/B Tie</td>
<td>C</td>
<td>X</td>
</tr>
</tbody>
</table>

### FIND/IMPORT PUBLIC ASSETS

<table>
<thead>
<tr>
<th></th>
<th>Unity</th>
<th>Godot</th>
<th>PICO-8</th>
<th>Dreams</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)Unity</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(B)Godot</td>
<td>B</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
### ADD A SONG THAT PLAYS ON STARTUP

<table>
<thead>
<tr>
<th></th>
<th>Unity</th>
<th>Godot</th>
<th>PICO-8</th>
<th>Dreams</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C)PICO-8</td>
<td>—</td>
<td>—</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(D)Dreams</td>
<td>D</td>
<td>D</td>
<td>—</td>
<td>X</td>
</tr>
</tbody>
</table>

### GIVE A CHARACTER HEALTH PROPERTIES

<table>
<thead>
<tr>
<th></th>
<th>Unity</th>
<th>Godot</th>
<th>PICO-8</th>
<th>Dreams</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)Unity</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(B)Godot</td>
<td>A</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(C)PICO-8</td>
<td>C</td>
<td>C</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>(D)Dreams</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>X</td>
</tr>
</tbody>
</table>

### SUPPORTED DISTRIBUTION CHANNELS
Dreams appears to be the most efficient GDE, followed by PICO-8, then Unity and Godot respectively. Most of the processes required for a minimum viable interactive product are easiest to set up in Dreams than in the other GDEs. Dreams’s visual logic system allows developers to easily select and place augmenting gadgets onto assets. This completely removes the coding barrier of entry from game development, and in doing so the entire development process is sped up, especially for junior developers with a limited understanding of coding languages. While Dreams has the least amount of options in terms of publishing platforms, the efficiency afforded by its systems make it the best choice for putting together rapid prototypes. Also, due to the early access status of Dreams, its early adopters are very eager to help as well as give feedback, which is critical for aspiring developers.

Workplace decentralization: A SWOT analysis

In order to decide which game studio model would be best to implement with Unity at Brock, a SWOT Analysis on the two different game development models, which are Centralized and Decentralized, will be used.

METHODS

A SWOT analysis is a comparative technique that weighs the strengths, weaknesses, opportunities, and threats of different options against each other in order to identify which is the most effective. This SWOT analysis draws from the author’s experiences working under both centralized and decentralized game production models; these experiences are synthesized and analyzed using the SWOT methodology.
RESULTS

CENTRALIZED

The centralized model is the current model of video game production. It is defined by a central physical workplace for all employees.

STRENGTHS:
- Ensure separation from work space and home space that will minimize the likelihood of distractions
- Workers can be monitored
- Allows for the most efficient transference of ideas (direct communication)
- Questions and concerns are addressed directly at any point
- Will ensure everyone has equal or comparable technology
- A workplace culture is much easier to develop in person versus over the internet

WEAKNESSES:
- Commute times may be an impediment to efficiency, depending on the average employee distance from the work site
- Higher overhead costs (e.g. rent for workspace, food, etc.)
- The potential talent pool is more limited due to the fact that employees must live within a practically traversable distance from the company (unless employees are willing to relocate)

OPPORTUNITIES:
- May be more appealing for grant applications
- No external hosting is required (potentially more secure)

THREATS:
- The studio may become dependent on certain specialists.

DECENTRALIZED

The decentralized model is an emerging model of video game production. It is defined by the lack of a central physical workplace and a dependency on an internet connection.

STRENGTHS:
- Employees supply their own workstations, minimizing overhead startup costs.
- People can work together no matter the distance, meaning the talent pool is much larger than a centralized studio system.
- There is no need to pay for a physical workspace when everyone can work from home (minimal overhead expenses).
According to research done by Miia Aronen, workers found that being able to self-direct led to them being able to better manage and execute on tasks given to them, versus having scheduled times within which to complete their tasks (Aronen 29).

Workers also found that working at home was less stressful, and allowed for them to complete domestic tasks while taking breaks from work (Aronen 29).

WEAKNESSES:
- Relies heavily on an internet connection, which can completely halt progress if it is down or slow.
- According to Aronen’s research, many workers find the lack of face to face communication negatively effects their interpersonal performance in terms of learning, communication, networking, and sense of belonging. (Aronen 40)
- The lack of a physical work environment means there is a larger need for self-regulation and focus on the part of the group members, as they have to hold themselves accountable for their work being done.

OPPORTUNITIES:
- A lot of services are moving towards cloud-based services, meaning that establishing a decentralized studio will make Brock a pioneer in an emerging market.
- According to Aronen’s study, working remotely gives employees the opportunity to develop their skills in self-management. (Aronen 34)
- Also from Aronen’s study, many workers found that working remotely had the potential to offer a better work/life balance, however this was dependent on the person.

THREATS
- Privacy and data security are at risk if not properly secured.
- Lack of a dedicated workspace means that distractions from family, pets, etc. are likely to hinder concentration. (Aronen 34)

ANALYSIS
The centralized studio model is the current industry standard for a reason. It is the most efficient model of the two, and it has been proven time and time again as a reliable means to produce quality content. As stated in the weaknesses section, the costs of renting a workspace as well as the more limited potential talent pool are the real negatives of this model.

While the decentralized model is extremely cost effective, it is almost entirely reliant on the individual to self-regulate and to ensure their work is completed on time. This can be problematic, especially with the busy schedules of students that often conflict
and are subject to change based on deadlines and extracurricular activities. Beyond this, from a management perspective it is much harder to oversee workers in a decentralized model. However, the fact that many other services are moving towards cloud-based storage points to the future of many digital industries being decentralized.

After assessing the two production models and seeing their strengths and weaknesses, it is clear that a hybrid model is required in order to create the most effective model to be used at Brock University. In this hybrid model, the workers would meet at a centralized point regularly to recap and plan. The majority of the work, however, would be done and compiled remotely by a project manager.
CASE STUDY: Making a Building Block Game

OVERVIEW
To gauge the effectiveness of a decentralized studio model, a game prototype was created with help from other developers remotely and asynchronously. As a building block game, it was created with the goal of being easily accessible and made with various assets that would have taken much longer to produce had they been done during the production time of the prototype. The prototype, named “Super Hero Concept WIP” is a beat-em-up that tasks players with defeating criminals that become increasingly powerful in a semi-destructible arena. Work on the prototype began in July 2019 and concluded in the middle of August. Although 5 people expressed interest in helping, only 2 additional developers contributed assets towards the prototype.

RESULTS
The prototype for the game was finished within 2 months and functions as an effective demonstration of a building block game. There were several public assets and mechanics that were used in order to speed up the production process. The game was liked by several community members and has received positive and constructive feedback.

The player charges their ability meter before engaging enemies. Civilians can be seen running from the enemies. Elements such as the skyline, car, and clouds were made by other developers and are available for public use.
After defeating a number of enemies, the player levels up and has their stats increased.

An enemy being punched into one of the destructible walls of the level.

**ANALYSIS**

After working on the prototype for the case study, some issues associated with the decentralized game studio became apparent. Based on the successes and shortcomings of the case study, a decentralized model still needs a centralized accountability structure tied to regularly occurring face-to-face meetings. This will discourage workers from abandoning duties and ensure that the team is as efficient as possible.
CONCLUSION

Looking towards a Brock game studio

The video game development landscape is changing, bringing with it emerging opportunities for building content, studio recognition, and rapid iteration upon promising game concepts. Where it was once near impossible to complete a game on one’s own, it is now possible to create, access, and share assets and full game mechanics and systems via cloud storage. This is especially useful for fast paced game development scenarios such as game jam events, as well as rapid prototyping of ideas in general. Using the environments and studio models assessed in this paper, teams can quickly transform an agreed-upon concept into an attention-getting video game prototype. With the rise in popularity of casual games that have a minimal number of gameplay systems, being able to access and remix other people’s assets is a massive time saver and efficiency boost.

After comparing the different GDEs, it became apparent that the GDEs designed with all the necessary production tools included are more efficient at completing tasks. The workflows within these applications are intended to be as self-enclosed as possible, whereas a GDE like Unity relies on the majority of the assets to be imported from third-party applications. This is very helpful for rapid prototyping.

Action Items

To capitalize on these emerging trends in video game development, a student-led Brock game studio should:

- Require students interested in joining the program to sign a content release form that would allow other students to use their work freely. A creative-commons license should be explored when developing this form.

- Build an internally-hosted repository for re-usable, licensed assets. The repository should have a clear directory structure, access structure, and backup method.

- Share the repository with all the students so they can view and upload, but not edit or delete assets files. This will ensure that there are no problems with missing content that would otherwise break dependent projects.

- Implement a standardized method for teams to upload their assets to the repository. This will be required in order for teams to release their game using the studio name and resources.
• Create a formal discussion channel where members can post questions, tips, requests, and general discussion. This will help to foster the development of the community and allow for coordination between members. A platform such as Slack, Discord or Reddit would work well, however for school related purposes, a private chat system would work best in order to avoid third-party spamming.

• Create a studio name and agreement terms that teams sign when joining the studio. Develop standardized brand standards (such as a logo) to which teams must conform when releasing their game.

By following these steps, a student-led building block studio will be able to create and share content on a larger scale, which will drastically reduce prototyping and general project production time. Setting up a studio with this model is great for fostering new talent in undergraduate students who have an interest in the different facets of game design. It is a great way for both students and alumni to get real-world experience working on creative interactive products in a team setting, while also helping their peers in ways they may not even consider at the time. This studio will allow Brock University students to build name recognition in the game development space while providing a sustainable bridge from university to industry that is designed to grow over time.

Final takeaways

• Asset sharing and curation are integral to the building block game student studio model.
• Dreams for PS4 is the most efficient GDE available for rapid prototyping and building block game development.
• Students looking to grow their portfolio can do so rapidly with this studio model.
• This studio model could work with any GDE and any number of GDEs simultaneously; however, choosing one to focus on one means more assets will be available for that specific repository. For the purposes of starting a single studio it is best to focus on one at first.

A strategic opportunity

If this studio model were to be implemented at Brock, production efficiency on projects would increase over time as more assets are added to the repository. This would lead to higher quality end-products for student portfolios and a faster production rate for projects overall. Implementing a student-led studio with decentralized elements is not only feasible, but also beneficial to both current and future students looking to work in the game design or interactive application design industry. It also allows students to acquire industry experience after graduating, which will help to increase talent retention within the Niagara region.
Works Cited

Acknowledgements

I would like to acknowledge the contributions of several people who helped to make this project happen. Firstly, I'd like to thank Dr. Alex Christie for sponsoring me and helping to direct this white paper. Without him and his positive reinforcement, this would not be possible. I would also like to thank Andrew Roth for his continued interest and help with this project from planning through execution. Additionally, I would also like to thank Scott Newman, who replied to one of my YouTube posts asking for help. He contributed a substantial amount to the project’s animation and gameplay mechanics at many key points and was a crucial member of the process overall. I’d also like to thank Reddit user Popcorninmyummytummy for contributing to the sound design and production aspects of the project. Without his help, the prototype would sound a lot worse! The prototype featured in this paper can be found on the Dreams platform as ‘Super Hero Concept WIP (Destructible Walls) V0.2”.