

## **EE/CprE/SE 491 WEEKLY REPORT 1**

**January 24th - February 5th**

**Group number: sddec23-13**

**Project title: Casters and Coders**

**Client &/Advisor: Mat Wymore**

### **Team Members/Role:**

- Brennan Seymour
- Branden Butler
- Theng Wei Lwe
- Wenqin Wu (Cody)
- Edward Dao
- Max Bromet

### ○ **Weekly Summary**

During the first week as a team, we got to know each other and started brainstorming ideas for our project. We ended the week with a good idea of the technologies we intend to use and have started getting everyone setup with the development tools.

- **Past week accomplishments** *(Please describe/summarize as to what was done, by whom, when and, collectively as a group. This should be about a paragraph or two in length. Bulleted points are acceptable as well. Please keep only your technical details related to your project. Figures, schematics, flow diagrams, pseudocode, and project related results are acceptable, but please ensure that they are legible (clear enough to read) and to provide an explanation. If researching a topic, please add a few details about what was learned and how it is relevant to the project. If two or more people worked on a single task, be sure to distinguish how each member contributed to the task. Specific details relating to the assistance provided to other members may be included here. **Do not include classwork, such as individual reflection assignments, and group meetings as part of your duties.**)*

- **Branden Butler:** This first week, we were still brainstorming ideas. I mainly looked at various technologies for embedding scripting languages within the .NET environment Unity uses. Our other engine choice was Godot, but it has language bindings for pretty much everything, so there was only a need to look at .NET limitations for Unity. I've got a few potential options, including Python (via IronPython), Lua, Javascript (via bindings to V8), and Kotlin (via IKVM). Due to the ease of embedding, I think Python and Lua are the top choices at the moment.

• **Wenqin Wu:** For the first week, I did some research on the possibilities to approach up our project (someone from my work), and they said there is certainly a way to do it but the team needs to know the game engine they are working on very well, and one tool we can use is called (IronPython), so it would be writing interfacing between python and C#.

For example: the person in the game would write python code, then IronPython read that into game engine then read the state of the engine to know what to do in Unity

Also as we planned in the meeting with Matt, I designed a questionnaire survey for players to see what they are expecting from the game. I included the programming language they are going to learn, the game style they want, the combat they want, and even the art style they want.

link:

<https://docs.google.com/forms/d/184CRBXuY-DOJU23nyKGt729NGmGmS9qmu58TWWYL05s/edit>

Hopefully we can have this questionnaire sent out and we can have a solid start on this project.

• **Edward Dao:** For the first week, after learning about the general idea of the project and what we plan on using. I did some secondary research on other games that are similar to our idea. What I found interesting from one of the games, called codeCombat. It has a visual on the left and on the right an ide for the user to type out their code. The basis of the games is to have users complete a puzzle with the use of basic python, javascript, and more depending on what the user wishes to learn to complete the game. To help the user there is a recommendation at the bottom of some basic code they could use to implement into their own code if they are confused. Lastly, to get a better understanding of the technology we are using. I downloaded the app we are using and started a tutorial recommended to me to get a better understanding of what we plan on using.

• **Theng Wei Lwe:** For the first week, I did some research on similar code gaming concepts on the Internet. I also went ahead and installed Unity as it's the engine we plan to use on developing the video game. I also researched basic coding tutorials such as W3Schools and Codecademy to get an idea on how to approach teaching beginners how to code.

• **Brennan Seymour:** I did some brainstorming about ways to integrate coding cohesively into a compelling game. It's been done before, but not often to my standards. Though notably more abstract than our goals, TIS-100 is a relevant game about interprocess communication. I also looked around for ways to incorporate Python scripting into a game engine other than IronPython, our main candidate. The two other good options I can identify are to write our own basic interpreter (forfeiting advanced language constructs like OOP), or to pivot to the Godot engine, which has bindings to a far wider array of languages, opening up our options for tools. I went through the official Godot tutorial, something that takes around an hour, and have decided that I like the engine. Though it would take some getting used to. I also set up a fresh Unity project with an appropriate gitignore.

• **Max Bromet:** This week, I mostly did brainstorming on the different ways that we could structure the game around learning to code while keeping it fun to play. An idea that I am a fan of is making unique spells by writing code, and letting the player use said spells to address various puzzles/challenges that they encounter as they play. I wasn't the one to come up with this idea, but I really liked the idea of having boss battles that can be won by writing code that accomplishes a really specific task, like a centipede that can be defeated through writing sorting algorithms. Beyond brainstorming, I also installed and familiarized

myself with Unity, as it looks like it's going to be the engine that we're going to be using moving forward.

○ **Pending issues**

- Team Member 2: Wenqin Wu
  - We better need multiple consulting resources if possible.
  - Need to research if IronPython can really work with Unity.
- Team Member 3:
- ...

○ **Individual contributions** *(Creating this section is optional, but it is **Required to include the "Hours Worked for the Week" and their "Total Cumulative Hours" for the project for each member somewhere relevant in your report. Your individual weekly hours should be at a minimum of 6-8 hours for this course. So please manage your time well. Also, ensure that individual contributions support your claim to the weekly hours. Be honest with the reports.***)

<u>NAME</u>	<u>Individual Contributions</u>	<u>Hours this week</u>	<u>HOURS cumulative</u>
Branden Butler	Research into embedded scripting technologies and engine capabilities	3	3
Wenqin Wu	<ul style="list-style-type: none"> <li>● Consulting someone about the game engine and technology(IronPython) we are using,</li> <li>● Environment set up.</li> <li>● Designed a questionnaire for users so that the team can have a clear target on the project.</li> </ul>	3	3
Edward Dao	<ul style="list-style-type: none"> <li>● Secondary research</li> <li>● Downloading and setting up environment</li> <li>● Starting a tutorial</li> </ul>	4	4
Theng Wei Lwe	Installed Unity and performed research for video game ideas.	2	2
Brennan Seymour	<ul style="list-style-type: none"> <li>● Installed Unity and Godot</li> <li>● Did some research &amp; brainstorming</li> <li>● Set up a fresh Unity project in the repo</li> </ul>	4	4
Max Bromet	<ul style="list-style-type: none"> <li>● Installed Unity and set up an environment.</li> </ul>	3	3

	<ul style="list-style-type: none"> <li>• Helped brainstorm game ideas</li> <li>• Researched similar game ideas</li> </ul>		
--	---	--	--

○ **Comments and extended discussion** (Optional)

*Wenqin Wu; Feel free to add or modify the questionnaire I made.*

○ **Plans for the upcoming week** (Please describe duties for the upcoming week for each member. What is(are) the task(s)?, Who will contribute to it? Be as concise as possible.)

- Branden Butler: Experiment with IronPython and Lua integration in Unity, check limitations of IronPython 3.4. Experiment with IKVM to see if Kotlin is a viable alternative or not. Continue researching other language options
- Wenqin Wu: send out the survey, collect the data and discuss the result with the group. Keep researching the game engine based on what has been told.
- Edward Dao: Help with anything or complete task given and complete tutorials
- Theng Wei Lwe: Go through Unity tutorials and help out with outstanding tasks.
- Brennan Seymour: Attempt to interface between python and a Unity dotnet script.

○ **Summary of weekly advisor meeting**

Our first meeting with Mat went over the expectations of the project and what the project should look like. We discussed possible types of games we could make as well as what programming languages we could possibly use. Finally, we discussed potential research the group could perform, such as surveys or speaking to faculty.

**Grading criteria**

Each weekly report is worth 10 points. Scores will be awarded as follows:

- **8 – 10:** Progress for your project seems to be suitable. Documentation and hours reported by team members are adequate.
- **6 – 8:** There is scope of improvement both in your report and your project progress. Can consult with instructor/TA after class for further inputs.
- **< 6:** Please talk to instructors/TA after class hours about any difficulties that you/your team is facing.

Each weekly report should be unique in that they have a unique set of supporting details

for your contributions. So please do not just copy your reports from the previous week. In addition, please avoid any personal pronouns (he, she, I, you). Try to keep your reports as neat as possible.