MARMOT.G5

COMP-1869-M01-2025-26 Final Year Group Project

Presentation Speech

Vlad Tomov – 001321349

Vlada Sydorenko - 001268537

Connor Craig – 001308843

Roshni Thapa – 001355831

Eni Haziri - 001303617

WHISPER FIELD

Introduction (Roshni)

MARMOT.G5 is presenting Whisperfield, a mobile farming simulator that blends cozy magic with real-world connection. Built in Unity for iOS and Android, Whisperfield offers a unique witchy farming adventure set in a charming world inspired by Wicca and nature.

Who is Marmot.G5 (Roshni)

Marmot.G5 is a 5-person team, consisting of 2 Programmers Vlad and Connor, responsible for the game systems in Unity including the core gameplay mechanics and real-time Weather API integration; 2 3D artists Vlada and Roshni focusing on building the games low poly world by using tools like Maya and Nomad, defining the witch-inspired aesthetic; and 1 UI/UX artist who designed all the interface elements, creating the visual style for menus and icons using tools like Procreate making sure the users gameplay journey is seamless and engaging during short play sessions.

What is Whisperfield? (Vlada)

Whisperfield is a mobile farming simulator designed to blend cosy farming gameplay with a magical, witch-inspired atmosphere. Players grow and sell crops while managing their own mystical farmland. The game features a charming world influenced by Wicca, nature, and light magical elements, creating a relaxing yet engaging experience. Whisperfield also integrates a real-time Weather API system, making the in-game environment dynamic and more interactive for the player.

Brief Specification (Vlada)

Our project follows all the core requirements of the brief.

It is a mobile management game, the game uses real-world external data through a Weather API, so the in-game weather directly affects how quickly the crops grow, keeping the experience connected to real life. Each play session lasts around five minutes, and the player’s progress continues even when they are offline, which meets the metagame and retention expectations. Our theme is suitable for both kids and older players, and we avoided the restricted settings mentioned in the brief. Instead, we created something unique like a witchy farming concept. Overall, our project meets the brief by combining a clear gameplay loop, real-world data integration and a distinct visual identity. The only requirement that different from the list is the slice type: instead of a vertical slice, ours is horizontal. However, in our previous presentation, we clearly stated that due to technical aspects the game must be horizontal.

Gameplay Loop (Connor)

The gameplay loop of Whisperfield is simple and engaging. Log in, buy some seeds, plant them, water and defend them from pests, and then collect them for some cash and experience points to level up and re-invest it back into growing your farm even more. The game has a feature that saves the player's game progress, so once they launch the game, it will either start a new game or load their previous save file which includes of their placed plants on the land, as well as their progress.

PLAY DEMO VIDEO

Farming System (Vlad)

The game offers a variety of different plants that the player can grow and obtain more by levelling their farmland. There are set farm plots where players can place the seeds on and be ready to defend them from pests, while making sure they have enough water to survive until they have fully grown. Every plant has some sort of different growing time, cash reward, experience reward and such. Depending on how much time the player has to invest in their current play session, they could decide to grow plants which take longer to grow and are harder to defend, but result in more profit.

Farming magic and seasonal cycle (Vlad)

Players can use magical potions that they can throw near their growing plants to give them buffs and effects that help them stay healthy and grow successfully while being under attack. These potions have an AOE (area of effect) which means they target all plants that are close to the potion's explosion radius and give them the full potion's effects.

Gamer persona (Roshni)

Our game appeals to two main gamer personas: Strategists and Zen Gamers.

Strategists enjoy planning, optimizing resources, and working toward long-term goals. They are drawn to systems that reward efficient decision-making and thoughtful progression.

Zen Gamers, on the other hand, seek calm and relaxing experiences that help them unwind. They appreciate simple,

satisfying interactions, gentle visuals, and low-pressure gameplay. By blending depth and relaxation, our game caters to both player types offering engaging management for Strategists and a soothing escape for Zen Gamers.

AIDA (Vlada)

AIDA marketing is a framework that describes the stages a customer goes through before making a purchase: Attention, Interest, Desire, and Action. How does it work for our project?

Attention:
A unique witchy farming concept, magical aesthetic, and real-world weather influence grab attention.

Interest:
Interest comes from the idea that in this game you can grow your own crops in a magical world and be engaged by the provided mechanics and visual aspects.

Desire:
A relaxing atmosphere for Zen players, plus clear progression for Strategists, makes them want to return and expand their farm.

Action:
As an action, we have made a workable game prototype which includes a great art style, a functional grid system, visual effects, an implemented Weather API, and a magical world where the player can do things they expected to do when downloading the app.

Publishing and promotion

To support both mobile and PC players, the game will be released on the Google Play Store for Android devices and on itch.io for desktop users.

Discord
Creating a Discord community where people could share and discuss the game they enjoy, talk about upcoming events, show their progress in the game, simply communicate and make new connections based on shared interests. Conducting votes on what could be added to the game, sharing future game updates, reporting certain technical issues, a community for long-term communication.

itch io

A website about our game has already been created on the platform, where we will be able to show updates for players and developers, collect comments, and therefore expand the audience that would like to test our product, and in the future become active users.

TikTok

At the moment, it is the most popular and the most effective platform for promoting any product, including games. We can encourage users to record videos with the hashtag #whisperfield, where they could show and share their emotions about the game, or launch a trend with the slogan “Farming? But make it witchy.” All that is needed for success is consistency and for the videos to start gaining reach.

SCRUM (Trello Board) (Roshni)

Our Team adopted the Agile SCRUM method to help with the production of our game. It helped in encouraging communication and dividing workload as well. We used sprints planning and Trello Board to keep track of our progress. It was helpful because each member of the team knew what was allocated and what was expected of them in each sprint meeting. Despite having all of these we still had lots of problems with the tasks like technical difficulties with the repo or with the mechanics for the game and other external factors as well.

Post Mortem

Future Ideas and Project Vision (Eni)

Many planned features were set aside due to time constraints, but they form an exciting vision for Whisperfield's future.

A key idea is to introduce dynamic random events, such as hostile mobs spawning to attack crops using different tactics from ranged assaults to throwing dangerous potions.

To counter these threats, players could craft and throw defensive potions, using them to protect plants from debuffs, pests, and attacks.

We would also love to expand player creativity with a custom grid-placement system, allowing farmers to design and build their own unique farmland layouts.

Finally, a deeper magical crafting system would let players combine harvested ingredients to conjure various potions, charms, and items, unlocking further gameplay and content.

Together, these features would add rich layers of strategy, customization, and lively interaction to the world of Whisperfield.

Individual Assessment:

Vlad

As the lead programmer for Whisperfield, my job was to handle all the complicated/heavy tasks such as coding the necessary systems that communicate with each other to create a smooth game loop. I was also handling majority of the Unity and git problems for the group, helping the team around with any branch merging issues, etc.

As for what I learned from working on this project is that everything takes a lot of time, even if it's the smallest thing. Especially, if there are many features and expectations from what the final version of the project should look like. I was working on the project almost every day (if not every day), and the amount of complexities the game starts to be in after all the systems start to work together makes it very easy to get lost. Fortunately, coding and maintaining clean code helped me to always stay on track and not get lost in thousands of lines of code. This project dramatically improved my programming skills as I spent way too many hours into coding, trial and erroring different features in the game and having fun with them until I get the right result I am looking for.

Vlada

In this project I worked as one of the two 3D designers. My main responsibility was creating low-poly assets for the crops, specifically the vegetables that the player grows. Since crops are one of the core elements of the game, my focus was on making sure the models matched the style of mobile games while still maintaining the witchy mood of Whisperfield rather than looking like a standard farm-management game.

I worked primarily in Maya using box modelling, which was the most efficient method for producing simple low-poly shapes while keeping enough control to adjust face counts and achieve an intentionally “angular but detailed” look. I also used Adobe Substance Painter to texture the models. The colour palette had to stay bright, but still reflect the magical atmosphere of the game. The main tones were cool toned purples and blues, so even bright elements needed a subtle cold undertone.

From a technical perspective, all models were tested inside Unity and displayed correctly. However, I encountered an issue with Maya’s Embedded Media option, which only worked inconsistently. Because of that, most textures had to be re-assigned manually inside Unity after import. This added extra steps to the workflow, but all assets were still fully functional and worked properly in the engine.

Working with the second designer was smooth — we coordinated our tasks, divided assets efficiently, and regularly checked each other’s progress so we didn’t overlap or waste time. Collaboration helped us maintain a unified style and stay productive despite the tight schedule.

At the start of the project, I was ambitious and aimed to create 30 models for 10 vegetables (three growth stages each), plus extra decorative props. As development progressed, I realised that finding references for “witch-themed vegetables” took more time than planned, because not every vegetable naturally fits the magical aesthetic. As a result, the final outcome was 8 vegetables and 24 models, along with additional assets such as a witch’s cauldron, stone paths, and decorative pumpkins. I also arranged the final scene layout, and my teammate expanded it with more environmental details.

Working in a team was challenging at times, especially since everyone had other coursework and personal responsibilities in life. Even so, we managed to complete the project successfully and our game meets the requirements of the brief.

Eni

As the UI/UX Designer I focused on establishing the visual identity. My first task was developing the overall theme and colour palette of the game. To achieve overall consistency in aesthetics, I hand-drew all UI assets using Procreate and polished them in Adobe Photoshop, where I adjusted details such as colours and sizing. producing these assets was the part I found easiest due to it being one of my main strengths.

Once the visuals were complete, I imported my assets and converted them into 2D UI sprites and then implemented them into Unity’s UI system. This involved arranging layouts and making sure that each spite matched and fit correctly in the designated places. I also wrote a few scripts in C# to make the UI more interactive, which included a scroll feature for the loading screens background, a rotation animation for the loading screen buffer, and a jiggle animation used for elements like the bell and potions. These features added movement and personality to the design.

During the project, I would Regularly test the game to ensure that my UI elements functioned correctly and didn't cause any bugs. If I did see an issue in the repo or a bug show up, I would communicate that to the team and ask if anyone knew anything about it before trying to resolve it and potentially causing a bigger bug.

In conclusion, this project had strengthened my skills surrounding UI/UX design and gave me a better insight into C# scripting. I also have a deep understanding of how important communication is and the importance of regular testing.

Roshni

As one of the 3D artists in the group, my primary focus was on creating low-poly environmental assets for the game. Throughout the production, I tried out a new 3D tool called Nomad Sculpt which I found to be easier to use when creating dynamic shapes and exporting it to game engines. I also established our games public presence by creating and publishing the official game website on itch.io.

A key lesson that was learnt was in production planning, our initial game asset ideas were optimistic as we underestimated the full process from sketching the design, creating and exporting to the game engine. But having 2 3D modellers working on different things at the same time for the game helped speed up the process on creating 3D assets.

Connor

This project was my first real experience with Unity and C#, so it was a major learning curve at the beginning of the project. It's also why most of the core programming workload stayed with Vlad, and I either helped him with it, or programmed systems that were simpler, and more to my skill level. Over the past three months, though, I've become a lot more comfortable with the engine, and I'm now more confident working on smaller systems and features on my own.

A portion of my time was spent working on our plant-growth VFX, and the rain effect for when the weather turned to rain. I used Unity's particle system to create the visual effects and messed around with the editor, like changing the scaling to make the rain look more realistic and natural.

I also handled the SFX and BGM across the project using AudioSources and simple edits to the scripts to trigger events and manage the playback.

One of the biggest tasks I took on was implementing the WeatherAPI system using OpenWeatherMap. At first, the idea of working with an API seemed daunting, but once I started working on it and searching up YouTube tutorials for it, it really wasn't as hard as I thought. This gave me a big confidence boost on the programming side and is a part of the game I'm proud of, even if it's a small part of it.

I also worked on the early stages of the inventory system. I built the basic structure and initial functions before Vlad took over and fully implemented it. Even though my part was small, it helped me get experience with handling data.

Group Reflection (Eni)

The positive areas of our group are that each member of the team has a specific area they specialise on which is great to help assign tasks and have a good workflow, we maintained a steady progress on the development of the game as well as keeping the concept of lowpoly and the witchy aesthetic theme.

The main negative aspect is that our group lacks proper communication which has given us a difficult time in several situations while developing the game.

WHITE BOX / BLACK BOX Testing

1 White Box testing – 3d Environment Assets (Roshni)

The 3D environmental assets produced by me have passed validation with clean geometry, containing no non-manifold edges and correctly oriented normals. Polygon counts were strictly limited across all models to meet performance targets. To further optimize memory usage, a stylized palette of plain colors was implemented in place of traditional textures.

2. White Box Testing – SpawnGridObject Function Statement Coverage (**Vlad**)

As for the white box testing, I have covered the first element which is statement coverage. I applied this to one of the functions in the grid system which is what the game uses to spawn grid objects and place them onto the farm plots/cells. I changed the code of the function to print all statements that are being executed from within the function and the output was 14/15 statements, which means 93% of the code was triggered. The reason it wasn't 15/15 was because of the way the function was coded to return true (if everything worked correctly, and the object was spawned) or false (if something failed which skips the rest of the code and returns out of the function). Since the function executes everything correctly, the code skips that "return false" statement which is why it is not included in the total statements covered calculation.

3. White Box Testing – 3D Assets (Vlada)

All of 3D models have clean and correct geometry with no broken or problematic polygons. Textures display properly with no flickering or visual artefacts. The models behave as expected when scaled up or down. Pivot points are placed correctly, so all objects rotate and position accurately. Overall, the models meet the required standards and function reliably.

4. White Box Testing – SpawnGridObject Function Branch Coverage (**Vlad**)

The next element of the white box testing is branch coverage. This testing detects how many branches from within the code have been executed. For the branch coverage test, I use the same function where it spawns the grid object onto the farm plots/cells. However, this time I implemented a number counter feature for every branch in the code (if/else). There are 3 branches in the function, meaning 6 total possible outcomes (true/false), for each branch only 2. At the end of the function, it printed that 3/6 (50%) branches were executed, meaning the branches were all true and all data was validly passed through the function.

5. White Box Testing – LoadGridCells Loop Testing (**Vlad**)

Finishing the white box testing with the last element, which is loop testing and for this example, I am testing it on a for loop that loads all saved plants from a previous player play session. In this case, I am testing to see if the for loop works by printing lines in the console for each loaded plant from the save file.

6. White box testing: UI Assets (Eni)

For The UI, I carried out white box testing by making sure that each asset was integrated well and worked with every other asset in the build. This consisted of me checking each anchor, layers, scales and each asset import to make sure that the asset was displayed how it was intended to be and that it was consistent across the entire UI. I constantly checked each UI sate that made sure that the game flowed correctly regarding the UI. I also checked colour palettes and themes of each UI to maintain the aesthetic.