Team T215: Capstone (Phase 2) Report

QUT | IFB399 | 2025-30-05 | S04 | Assessment 3

Project ID	P202
Project Title	Myra 3D Simulation
Industry Partner	Bio Molecular Systems
Industry Supervisor	Kelly Ensbey
Team members	
n10752251	Bett, Ethan
n11318546	Chambers, Milly
n10753508	Vosper, Samuel
n11374268	Yeh, Chen-Jui (Ray)
Team changes	N/A
Project changes	N/A

Table of Contents

1: Team Chapter

1.1: Introduction

Project Context, Goals, Scope

Bio Molecular Systems (BMS) is a biotechnology company that develops and distributes innovative laboratory equipment world-wide. They aim to optimise the operations of clinical facilities around the globe and are regarded as one of the leading providers when it comes to innovative lab equipment. Operating in the life biotechnology industry sector, BMS has established itself as a critical player in laboratory automation.

BMS are most known for their Myra Liquid Handling System, which is an advanced automatic pipetting robot that is designed to handle liquids accurately and efficiently. Using an integrated camera, Myra simplifies set up and minimises maintenance, further enhancing the efficiency. The significance of automated liquid handling systems like Myra became particularly evident during the COVID-19 pandemic, where these devices played a crucial role in PCR testing facilities worldwide, demonstrating their importance in both research and clinical diagnostics applications.

BMS has developed the BMS Workbench Software, which allows users to effectively connect to and operate the Myra. Through this software, users can design assays with precise configurations and analysis options. Furthermore, the software supports qPCR setup, providing the option to generate reactions from specific inputs, or more advanced operations. This high level of customization and precise control allows users to fully utilise Myra to meet their complex laboratory needs (Bio Molecular Systems, n.d.). The integration between Workbench and the physical device forms the foundation of BMS's product ecosystem, enabling seamless workflow automation for laboratories globally.

The primary business goals driving this project align with BMS's strategic objectives to enhance customer experience, reduce operational costs, and expand market reach. Given Myra's high cost, from around $30,000 USD, it's not feasible for every researcher to have their own unit (Bio Molecular Systems, n.d.). This creates several business challenges that the simulation directly addresses: enabling remote training and support, reducing the need for physical demonstrations, allowing potential customers to evaluate Myra's capabilities before purchase, and providing existing customers with a risk-free environment for workflow optimisation.

The scope of the MyraSim project represents a significant technical evolution spanning multiple development phases. Our team was tasked by BMS to create a fully interactive 3D simulation, building on a prototype developed by a previous capstone team. This previous team had developed OpenTKSim, a console application utilising the OpenTK library – a low level graphics framework that, while powerful, required extensive manual implementation of basic rendering features. The previous team faced significant challenges in implementing the desired functionality.

"It's not a game engine, or a framework, or a full renderer, or a complete audio system by itself.

Instead, it's the low-level foundation on which you can build those kinds of things."

Quote from the OpenTK FAQ page (OpenTK, n.d.)

Phase 1 of our project focused on transitioning this OpenTK-based console application to a more manageable architecture. We successfully converted the simulation to a Windows Presentation Foundation (WPF) application utilising AB3D PowerToys, which provided higher-level rendering capabilities and eliminated much of the manual graphics programming required by OpenTK. This conversion established a solid foundation for feature development while maintaining compatibility with BMS Workbench.

Building upon Phase 1's achievements, the scope of Phase 2 encompassed both technical infrastructure improvements and feature enhancements:

1. Rendering Engine Upgrade: A critical component of Phase 2 was upgrading from AB3D PowerToys to AB3D DX Engine, providing improved performance, better graphics capabilities, and enhanced support for complex 3D operations. This upgrade ensures the simulation can handle increasingly sophisticated visualisation requirements while maintaining the high visual fidelity and smooth performance required to achieve the desired user experience.
2. Movement Accuracy: Another critical priority for Phase 2 was achieving a 1:1 movement replication with the physical device. While Phase 1 established basic functionality, several operations exhibited noticeable differences from the physical Myra – including inconsistent pipette movements, timing discrepancies and occasional freezing. This faithful movement reproduction is essential for the simulation to serve as a reliable training and troubleshooting tool.
3. Enhanced Visualisation Features: Implementation of dynamic model generation for specialised consumables, allowing the simulation to accommodate the diverse equipment used by BMS's global client base without requiring manual model creation for each variant.
4. Advanced Troubleshooting Tools: Development of a comprehensive time slider functionality and detailed operation list, enabling users to navigate through completed operations, investigate issues, and optimise their workflows without consuming physical reagents or occupying the actual device.
5. Improved User Interface: Refinement of the simulation interface to provide intuitive camera controls, speed adjustments, and operation monitoring, ensuring accessibility for users with varying technical expertise.

The technical context of MyraSim positions it as a sophisticated bridge between BMS's hardware and software ecosystems. The application receives inputs from BMS Workbench in the form of movement instructions and device commands, processing these through the Direct X based rendering engine and provides visual feedback that mirrors the physical device's operations.

Project Outcomes and Success

Our project aims to provide an interactive simulation of Myra to support users in applying its functionalities effectively and confidently (Bio Molecular Systems, n.d.). Through Phase 2's development, we have delivered a sophisticated simulation platform that establishes the technical foundation for transforming how BMS and their clients will interact with Myra. The simulation serves as a powerful internal tool for BMS engineers, enabling them to reproduce customer reported issues, test new features, and validate protocol designs without requiring access to physical hardware. This internal capability positions BMS to provide more effective customer support once the simulation is integrated into their production environment.

The technical outcomes achieved in Phase 2 create significant potential for future impact across BMS's user base. Once integrated into Workbench, lab technicians and researchers will be able to preview complex protocols before committing expensive reagents, utilising the time slider and operation list features to analyse specific steps in detail – a capability impossible with the physical device where operations run continuously. Training coordinators will gain the ability to onboard new users without monopolising valuable equipment time, while the variable speed controls and multiple camera angles will provide enhanced visibility into operations that would be obscured on the physical device.

From a strategic perspective, MyraSim represents a critical investment into BMS's future capabilities. The simulation lays the groundwork for enhanced customer engagement through remote product demonstrations, reduced support costs through virtual troubleshooting, and improved customer satisfaction through risk-free protocol validation.

Client satisfaction serves as the ultimate measure of our success. Throughout Phase 2, BMS engineers consistently expressed that our progress exceeded their expectations, particularly praising the simulation's accuracy and visual fidelity. While the simulation awaits integration into Workbench before reaching Myra's end users, it is ready to be used as an internal development and testing tool for BMS. We successfully managed scope changes through close collaboration with the client, deferring features like liquid visualisation and circular vortex mixing motions to ensure the delivery of core functionality within project constraints. This approach, combined with transparent communication and iterative development, resulted in a simulation that not only meets the immediate needs of BMS but also provides a solid foundation for future enhancement by subsequent capstone teams. The project's success lies not just in the delivered features, but in establishing the technical architecture that will support MyraSim's evolution from internal tool to customer facing solution.

1.2: Project Setup

Project Management Approach

In this project, our team had adopted a hybrid-Agile-Scrum project management approach. Given the size and the scope of the capstone project, this approach allowed us to balance structured planning with flexibility to respond to client feedback and technical challenges

We apply Scrum principles to manage our tasks throughout different project phases. This project progress was divided into sprints, which we set to be two weeks of time length. In this way, our sprinting could align with our fortnightly meeting with tutor and clients to prepare future sprints based on the feedback received during the meetings. Thus, allowing us to deliver incremental updates, regularly reflect on progress, and incorporate continuous feedback. The use of sprints provide enough structure to plan while maintaining the agile to adapt.

Yet, our implementation also diverged from pure scrum, making the approach hybrid. For example:

- We did not conduct daily stand-up meeting, since it would be difficult to arrange time for it. Instead, we implemented weekly stand-up meetings and asynchronous updates via discord.
- BMS did not serve as a formal product owner role in normal scrum approach. We gather their feedback and make adjustments to the project and tasks, without let BMS having control of our backlog and tasks. This adjustment would be more suitable for student projects like us
- We adapts our sprint plans to accommodate university deadline and capstone schedules, which introduced elements from traditional waterfall scheduling.

These adaptations and adjustments make our approach better suited to the academic environment, while still maintaining the core values of Agile: collaboration, responsiveness to change, and continuous improvement.

There are several tools we used to enhance our management for our project which include:

- **Jira:** Used for sprint planning, task tracking, and backlog management. User stories are broken down into tasks and subtasks, with estimates and priorities assigned collaboratively. Different tasks have their own group that allow use to quickly assign tasks upon functionalities. The Jira board grand us a highly visualized progress on the project, which we could adjust project planning respectively.
- **Discord:** Discord is our main communication tool throughout the project, which provides real-time collaboration and coordination among team members. It allow us to discuss, ask for help, and provide information with team members which resulting an consistent environment to develop the project.
- **OneDrive:** Used for documentation sharing and collaborative editing for the report and relative files.
- **GitHub:** Managed source code with branching strategies to support parallel development and maintain code integrity. Members creates different branch for various features to allow efficient development of the project.

Team roles were assigned based on each member's strengths, interests, and responsibilities, as detailed in our team agreement (Appendix A). Ethan is the project manager for this project, which is in charge of organizing meetings, maintaining Jira, and ensuring team's progress. Where Matt, Ray and Sam are in charge of different technical aspect of the project. Ray was focused on the 3D model consumables, which helps on the visual representation of models in the simulation. Matt contributed with various tests, analysis, and evaluations of the project, making sure the simulation is performing correct action and outputs.

Client Relationship and Expectations

Throughout our interactions with our client, Bio Molecular Systems (BMS), we regularly communicated with three key staff members who collectively acted as stakeholders for the project. Among them, Kelly served as our main point of contact and fulfilled a role like a Product Owner in Scrum; providing detailed feedback, clarifying requirements, and helping us prioritize tasks. Communication with Kelly occurred frequently via email and in person, ensuring clarity and alignment at each project stage.

Together with the BMS team, we collaboratively scoped the project, identifying core deliverables that were essential and bonus features that could be included if time allowed. These expectations were discussed and refined during our regular in-person and online meetings. Planning and progress updates were presented in these meetings, often supported by visual demos and walkthroughs of the software, allowing for immediate feedback and discussion.

Through our meetings, our specific features were discussed and confirmed that they met BMS's expectations. This approach allowed us to maintain momentum while ensuring high quality work. Our client consistently expressed satisfaction with our work, often noting that we had met or exceeded expectations.

We also took deliberate steps to build a strong working relationship with BMS. We responded promptly to questions, asked for clarification when needed, and remained open to feedback throughout. This professionalism and transparency helped garner mutual trust and collaboration across the semester.

Team Collaboration

We created a positive project work culture through consistent communication, and encouraged to ask questions, not criticism. This approach garnered significant collaboration throughout the team and gave us confidence in each other's and our own abilities. The team agreement and collaborates were achieved through extensive discussion, ensuring we expressed and understood each other's strengths and weaknesses, allowing us to delegate tasks efficiently and effectively. Similarly, our team roles were also delegated this way. We were all assigned software developer roles on this project, as there were a wide range of features that our client wanted implemented. Ethan was designated project manager as well, as he had prior experience with these roles and we were confident in his abilities to successfully manage our project. Ray focused his efforts on 3D model creation, as he knew the most about that area. Matt targeted his efforts towards back-end functionality, such as connecting the simulation to the DX engine, as he was most knowledgeable about that. Finally, Sam worked on the UI, including the backend functionality associated with it. He was most comfortable with this role as he had prior experience working on different UI designs in other projects. We collectively decided on how much time we would be able to allocate towards this project per week, and how often we would have meetings.

Similarly to last semester, each member's tasks for the week were chosen during our weekly meetings, and through our regular meetings with our client. These tasks were split up according to individual strengths and the availability of each member of that week. This method allowed us to delegate tasks effectively, ensuring the work would get done by the next meeting. However, if a team member was having difficulty completing their task, regular communication was maintained to stay connected and keep on top of the work.

We collectively chose discord as our primary form of communication. This was as it allowed for both voice and text communication, which we used to hold our weekly meetings. Furthermore, the ability to post files, images and pin messages allowed us to keep track of the important links and messages week on week. In terms of file sharing, OneDrive was used to keep all our relevant documentation in one place. This use of file sharing allowed for active and collaborative work on the documents. Github was our chosen code repository, as it was what we all had the most experience with. This let us create our own branches to work on specific features of the project and then merge at a later date to ensure that the overall functionality was still working. Regular commits supplied with the relevant information helped us keep track of the progress of the project, in case anything unexpected were to happen, we would be able to revert back to an earlier, working version.

Consistent, transparent communication was key to our project's success. Through this communication, we were always all on the same page regarding deadlines, expectations and if a team member required extra help to complete their tasks.

Communication Plan

As seen in the communication plan below, at the start of the semester we had planned the best way for our team to communicate with our client, the teaching team, and within our team. We decided that fortnightly meetings with our client, alternating in person and online would be the most effective moving forward with the project. We decided this, partially due to the fact that our client is located in Yatala, and is quite a lengthy drive for each of our team members. We also thought that an online meeting in between our in-person meetings would allow us to keep our client up to date on regular progress, even if said progress wasn't significant. However, it would keep them in the loop and provide consistent communication. Our in-person meetings were aimed at a more formal presentation of our work, as that would provide us more time to show our client significant progress, that we can receive valuable feedback for, and give us the opportunity to have in-depth collaboration. Expanding on this point, when we did have our in-person meetings, we decided early in the project that as we are travelling a fair distance to see the client, we should spend the day there to maximise work on the project and collaborating with the client, all the while gaining valuable experience at working in a real work environment. Our team held a weekly team meeting, to ensure everyone team member is the on same page, while also providing everyone the opportunity to ask for assistance where required. Through this meeting, and regular consistent communication, we collaborated with each other on ideas and the best solutions to solve any problems we may have run into. Furthermore, this constant communication allowed us to be fully prepared for our online meetings with our tutor Ross, every fortnight. In this meeting we showcase and explain our progress on the project and ask any important questions that we may have thought of throughout the duration of the semester.

COMMUNICATION PLAN OVERVIEW				
Communication	Audience	Goals	Schedule	Format
Phase 2 Kick-off	Project Owners (Kelly, Jason and Warren from BMS)	Re-familiarise project owners with our progress, set expectations and priorities.	One off, start of semester. (27/02)	Formal meeting with a presentation prepared.
Meeting with Teaching Staff	Project Team and Teaching Staff (Ross Schamburg)	Present our current progress since the last meeting. Receive feedback and advice on Project Artefacts.	Fortnightly	Formal Online Meeting
Progress Report Presentation	Project Owners (Kelly, Jason and Warren) and possibly other BMS staff	Present our current progress since the last meeting. Receive feedback from BMS	Fortnightly.	Formal meeting with a presentation. Every second meeting will be in person.
Collaborative Work Session	Project Team, certain Project Owners and other staff	Spend the day at BMS working on the project with the	Fortnightly (same day as In-person Progress Report Presentation)	Informal, collaborative.
		help of experienced software engineers.		

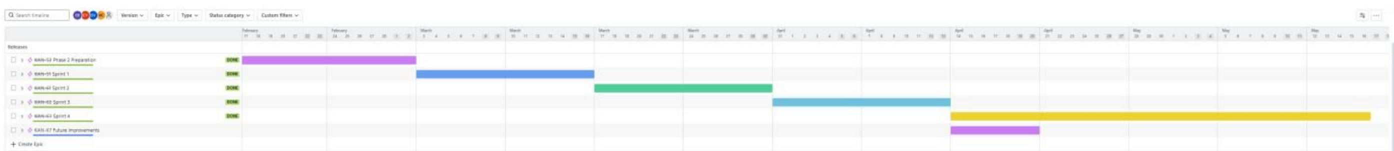
Project Team Meeting	Project Team	Discuss our individual progress and collaborate on project.	Weekly	Informal Online Metting.
Support Sessions	Project Team and certain BMS Staff members depending on topic.	Collaborate and get help / advice from BMS staff. Could be if we are stuck on a problem or just want feedback etc…	Whenever needed.	Informal, either in person or online. Can be real-time or just email correspondence.
Testing Sessions	Project Team and certain BMS Staff members depending on topic.	Get other team members and BMS Staff to test functionality and provide feedback.	Whenever needed. Most likely towards the end of Phase 2.	Informal, either in person or online. Most likely in person if BMS staff is testing.

1.3: Project Plan and Risk

Project Planning and Progress

Our Phase 2 development followed a scrum based agile methodology, with the project structured around a product backlog derived from stakeholder requirements and technical constraints. The product backlog consisted of 11 user stories (See Appendix C), each with clearly defined acceptance criteria, effort estimates and priority.

The user stories were prioritised using a must have / should have framework, with time estimates ranging from 1 to 2 sprints per story. Critical must have features include the core simulator functionality such as Workbench integration, simulation execution and consumable model loading, while the should have features focused on enhanced user experience such as camera controls, playback and navigation features. These were deemed less of a priority as the simulation will eventually be integrated directly into Workbench and will utilise their UI instead.



The project was organised into a structured release plan spanning five distinct sections. See the Jira Page Snapshots (Link in Appendix D) for a detailed breakdown of our Jira Page.

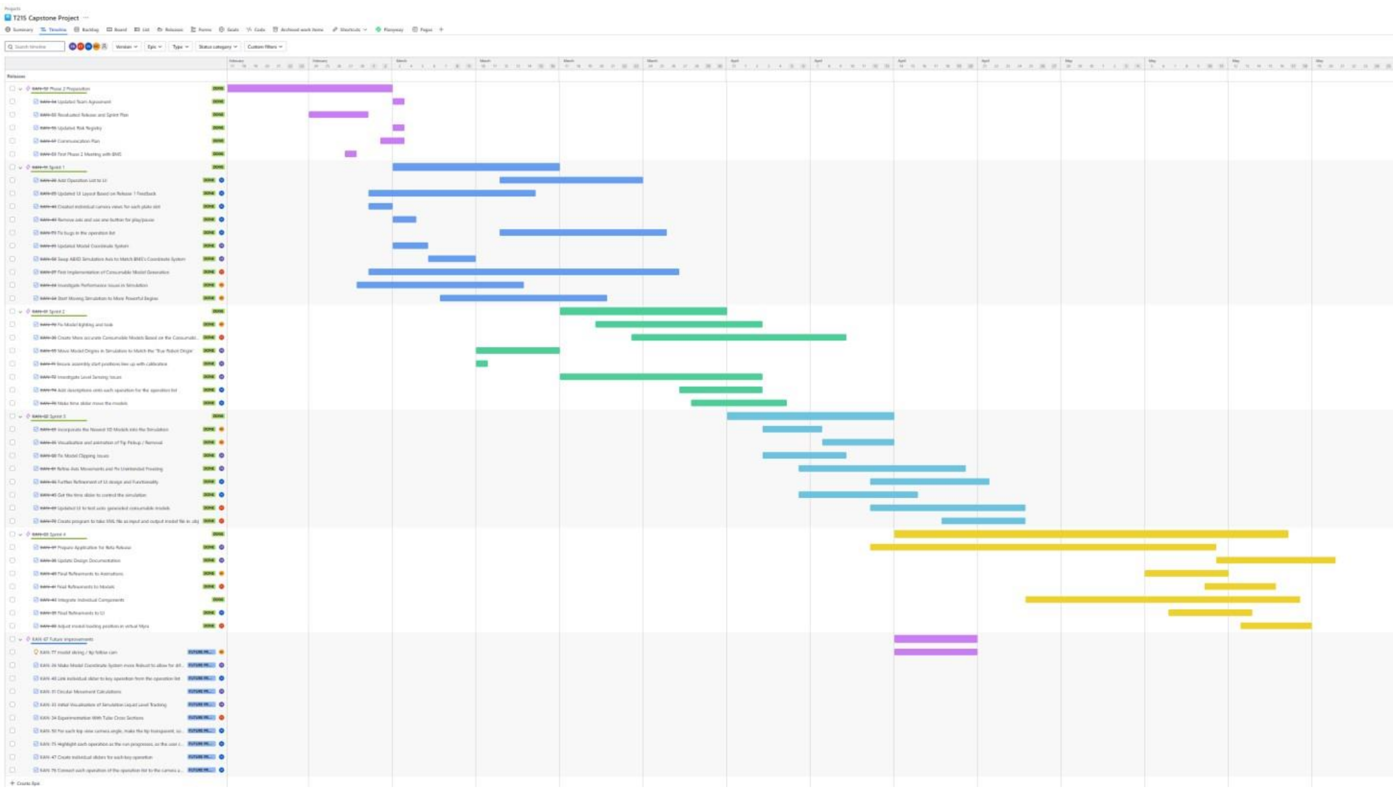
Phase 2 Preparation: This initial phase focused on project setup and stakeholder alignment, including updated team agreements, risk registry updates and communication planning. This initial section helped set us up for success and ensured that we were all on the same page before the Phase 2 development began.

Sprint 1: The first sprint focused on foundational implementations for each of the four major areas, including expanding the UI to include additional functionality, reworking the simulation's coordinate system to match the axis orientation used internally at BMS for Myra's operations, initial implementation of dynamic model generation and the investigation into performance issues and the feasibility of upgrading from AB3D PowerToys to AB3D DX Engine.

Sprint 2: The second sprint expanded on the initial implementations featured in the first sprint by completing the migration to AB3D DX Engine and increasing the visual fidelity of the models, tweaks to the initial implementation of the operation list and time slider, fixing backend issues causing the simulation to not accurately mirror Myra's real movements and expanding the functionality of the dynamic model generation.

Sprint 3: The third sprint consisted of refining and finalising the functionality of the simulation.

Sprint 4: The final sprint involved merging each individual section, adding the finishing touches, and cleaning up the code behind.



Throughout Phase 2, several adaptations were necessary to address technical challenges. When certain tasks proved more complex than initially estimated, tickets were decomposed into smaller more manageable tasks and redistributed across the sprints. Based on stakeholder feedback, some features were reprioritised. Critical features were elevated in priority while some nice to have features were pushed back. A separate Future Improvements Release was created in the Jira Page to track all of these nice to have features that exceeded the current scope. This approach helped us maintain focus and provide BMS with the most value possible, while still keeping track of these ideas for future development.

Risk Management

Risk	Existing controls	Likelihood	Consequence	Risk score	Risk rating	Planned actions	Review date
Corruption or loss of Project Artefacts.	Using a GitHub repository for software and a OneDrive folder for written artefacts.	2	4	8	High	Regularly back up project artefacts to GitHub and OneDrive to minimise potential loss. Keep local copies of artefacts.	3/03/2025
Falling behind on sprint plans and expectations	Maintaining regular communication to ensure no parts of the project are lacking.	2	3	6	High	Keep regular and transparent communication of the work that has been done, and which needs doing. Through this, other team members can help to maintain project deadlines.	3/03/2025

Unforeseen medical emergencies	Constant communication so each team member is aware of any emergencies.	1	4	4	Moderate	Ensure everyone has general knowledge of each others work, so in case one team member is unable to work on the project, it can be split up	3/03/2025

Throughout Phase 2 of this project, we collectively identified risks through forward thinking, of any realistic circumstances that may negatively affect the progress of our work. Initially, we came up with common risks associated with project work, such as the corruption of the project artefact, or falling behind on sprint plans and client expectations. To manage the potential loss of our project artefact, we regularly pushed our progress to our Github repository, and ensured we all had our own working copies, in case there were issues with the repository. As a last resort, we also kept a copy of the project on OneDrive, further adding to the security of our work. To mitigate the risk of falling behind on our work, either through unforeseen medical situations or an increased workload from other units, we maintained regular and transparent communication. As a result of this, we were aware of each other's circumstances, and if we needed to pick up the work that other team members were unable to complete on any given week. Through this approach, we consistently met deadlines with work that exceeded client expectations. We also regularly updated our risk registry when potential risks came to our attention. For example, when the threat of a cyclone beckoned earlier on in the semester, we had to account for any consequences that may result. We decided to maintain regular contact and only continue working on the project when in a safe environment. One of our team members, Sam, lost power for four days. He regularly updated us while this was occurring, however also went to another location where he could continue to work on the project. Through the identification of risks early in the semester and consistent communication throughout, our team was prepared for and overcame any risks and obstacles that came our way, ensuring the client expectations were consistently met.

						between the other team members accordingly.	
Overlapping work tasks	Using project management tool to assign tasks to individuals	2	2	4	Moderate	Maintain the use the management tool, update task status before and after working on the task. Avoiding only verbal notification when working on a task.	3/03/2025
COVID	Online regular communication	2	2	4	Moderate	Continue to work in an online environment, ensuring work and communication is maintained.	3/03/2025
Cyclone	None	4	3	12	Very High	When able, continue to work on the project in a safe, online environment.	3/03/2025
Failing to deliver work that is up to industry partner standard	None	1	3	3	Moderate	Through our regular meetings with our industry partner, ensure their feedback is heard and then worked on for future meetings.	3/03/2025

1.5: Artefact Description

Summary of Work Done in Phase 1

During Phase 1, we successfully migrated the existing OpenTK based simulation to AB3D PowerToys (AB4D, n.d.), establishing a more robust foundation for 3D rendering. To accommodate the new rendering engine, the console application was converted to a WPF application, and BMS's internal assembly was reintegrated. Backend functionality was extended to support key operations such as Level Sensing. This enabled the simulation to complete full runs.

Functionality

MyraSim addresses critical user requirements for BMS Engineers, lab technicians, and researchers who work with the physical liquid handling robot. The simulation functions as a Virtual Device that integrates seamlessly with BMS Workbench, allowing users to setup and execute runs exactly how they would with the physical hardware.

Core User Stories Addressed:

- BMS Engineers can verify simulation accuracy against physical hardware to differentiate between user errors and device faults.
- Lab technicians can visualise Myra's operations at controllable speeds and viewing angles.
- Researchers can troubleshoot experiments virtually before committing expensive reagents.
- Training coordinators have an accessible platform for user training without requiring physical device access.

Key Functionality Delivered:

- Virtual Device Integration: ○ Full compatibility with existing BMS Workbench workflows.
  ○ Simulation receives and processes commands as a physical device would.
- Interactive Visualisation: ○ 3D simulation with multiple camera angles, variable playback speeds, and timeline navigation.
- Accurate 3D Models: ○ Precise representation of Myra's assemblies and consumables using both high-fidelity SolidWorks models and dynamically generated models.
- Operation Tracking: ○ Detailed operation list showing all operations and individual actions preformed during the run.

Architecture

The final system consists of a three-layer architecture that separates the main components of the simulation.

UI Layer:

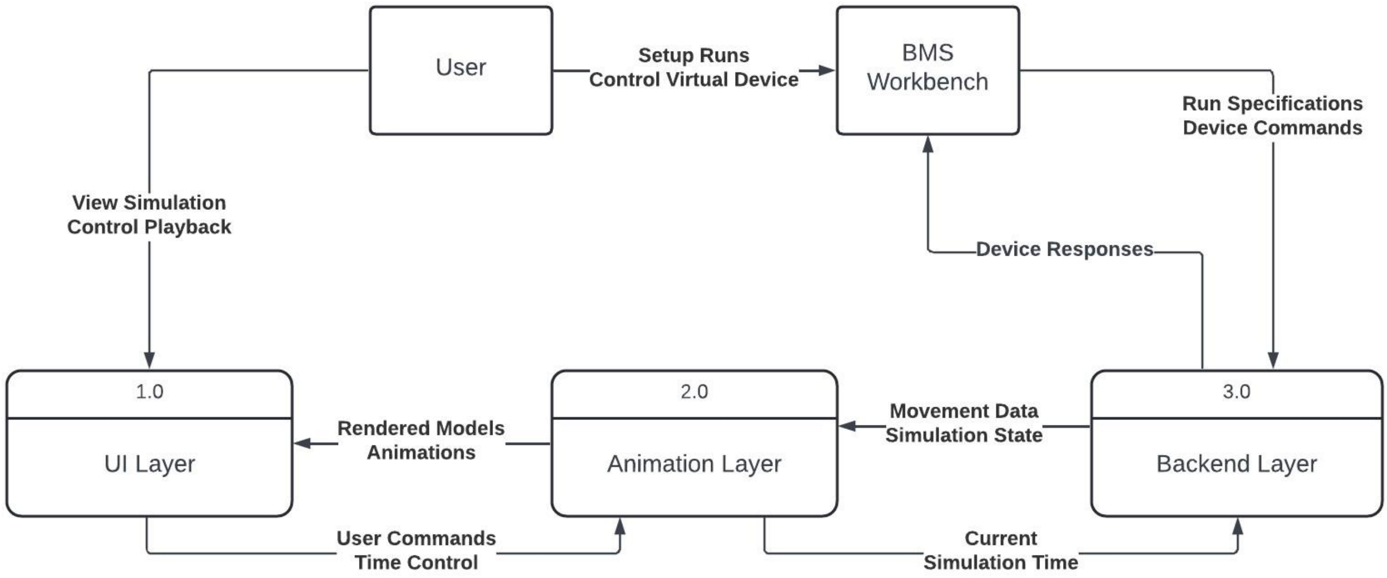
Built using WPF (Windows Presentation Foundation), it handles all user interactions including play/pause controls, simulation speed adjustments, camera manipulation, and timeline navigation.

Animation Layer:

Serves as the visualisation engine, managing 3D models and translating movement data into real-time animations. The MyraAnimations component handles model loading, position updates, and event-driven animations.

Backend Layer:

Contains the core simulation logic and device emulation capabilities. The Virtual Myra component emulates physical hardware including axis controllers, collision detection, and liquid handling operations while managing communication with BMS Workbench through established device protocols.



Data Flow:

Users can interact with the simulation by direct UI manipulation or through BMS Workbench. The backend processes run specifications and generates time-stamped movement data based on information sent by Workbench. The Animation Layer uses this data to update the model positions in real-time.

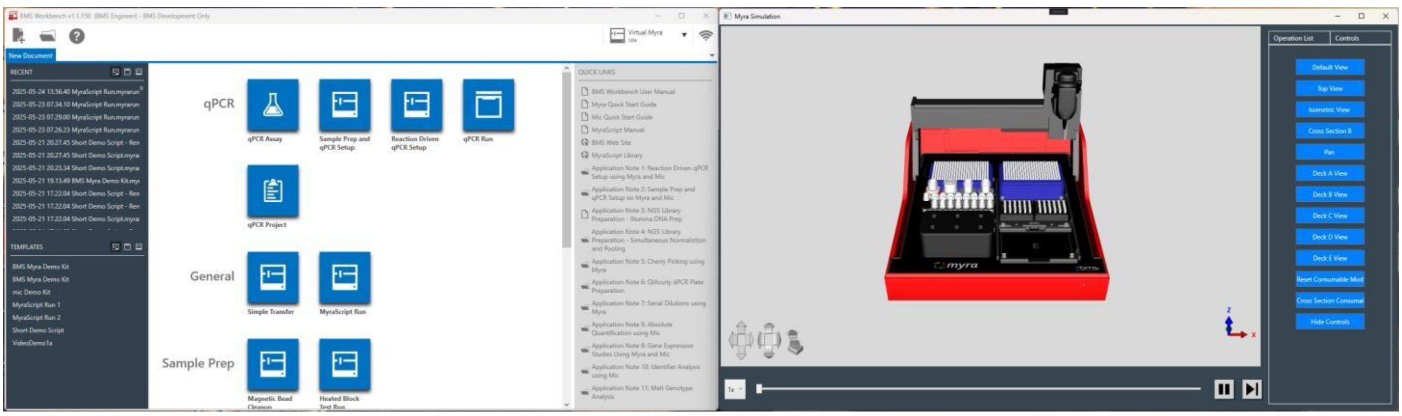
Technical Description

Virtual Device Implementation:

The simulation's core innovation lies in its Virtual Device implementation. The VirtualMyra class emulates the physical robot by utilising the same communication system used by the physical device. The virtual device is initialised in the same way with its own ID and individual controllers for each axis. A NamedPipeSocket is used to send and receive commands from Workbench.

User Interface:

The MainWindow is the entry point for the application. It initialises the assembly models and the virtual device itself. The WPF based user interface provides users with an interactive view of the simulation. The UI design was chosen to resemble the BMS Workbench Software and has been kept as minimal as possible as it will be integrated directly into Workbench in the future.



Users can interact with the simulation in several ways as mentioned above. The size of the slider is based on the amount of time the simulation has currently run for. As many commands sent from Workbench are dictated by the responses sent back from the device, you cannot skip forward to future operations. However, you can change the speed of the simulation to quickly emulate the entire run, then go back and view any operation at a normal speed.

The Operation List tracks the execution of each individual process executed by the device. The system listens for events fired whenever Workbench sends commands to the virtual device. An OperationNode is created whenever an operation is started. The node is either added to the OperationTree as a root Node (if no parent exists) or as a child of the current operation. This creates a hierarchical structure that showcases the relationship between operations.

```
LiquidTransferOperation(Transfer 232.94 µL of Water into C1)
  GetNewTipOperation
  SingleLiquidTransferOperation(Transfer 38.82 µL from A1: Water into C1)
    MoveToWellProcess
      RetractZProcess
      MoveXYProcess
    PrepareForAspirateProcess
    AspirateProcess
    MoveToWellProcess
    DispenseProcess
  SingleLiquidTransferOperation(Transfer 38.82 µL from A1: Water into C1)
  SingleLiquidTransferOperation(Transfer 38.82 µL from A1: Water into C1)
  SingleLiquidTransferOperation(Transfer 38.82 µL from A1: Water into C1)
  SingleLiquidTransferOperation(Transfer 38.82 µL from A1: Water into C1)
  SingleLiquidTransferOperation(Transfer 38.82 µL from A1: Water into C1)
  EjectTipOperation
LiquidTransferOperation(Transfer 323.52 µL of Master Mix into C1)
```

For example, Myra executes a Liquid Transfer Operation where 232.94uL of water is transferred from Well A1 to Well C1. The pipette tip can only hold 50uL of liquid, so the Liquid Transfer Operation is split into multiple Single Liquid Transfer Operations. Each of these operations contain multiple movements such as moving to the starting well, aspirating the liquid, moving to the target well and dispensing.

Movement Calculation:

The Axis Controller class is used to process commands sent from Workbench into individual movements for each axis. Commands / transactions are sent through the Named Pipe. These packets are decoded and turned into usable parameters based on the type of command.

Each movement command is split into segments with specific parameters like acceleration, velocity and jerk. This is used to calculate the position of the axis for each time step.

```
// Method to start a trajectory-based move based on the provided transaction.
1 reference
private void StartTrajectoryMove(ISlaveTransaction txn)
{
    _stopped = false; // Reset the stopped flag.
    txn.StreamCommandPayload(1); // Stream the command payload.

    // Read the trajectory move parameters from the transaction payload.
    var segCount = txn.ReadByte(); // Number of trajectory segments.
    var initialPosition = txn.ReadBEInteger(); // Initial position.
    var checkPointPosition = txn.ReadBEInteger(); // Checkpoint position.
    var levelSenseBaselineDelay = txn.ReadByte() * 10; // Level sensing baseline delay.
    var trajectoryTimeout = txn.ReadBEInteger(); // Trajectory timeout.

    // List to hold the trajectory segments.
    List<TrajectorySegment> segments = new List<TrajectorySegment>();
    var currentPos = initialPosition; // Current position in the trajectory.
    var currentVel = 0; // Current velocity.

    // Iterate over the number of segments to read and create trajectory segments.
    for (var i = 0; i <= segCount - 1; i++)
    {
        var jerk = txn.ReadBEInteger(); // Jerk value for the segment.
        var finalAcc = txn.ReadBEInteger(); // Final acceleration value.
        var finalVel = txn.ReadBEInteger(); // Final velocity value.
        var posOrDelay = txn.ReadBEInteger(); // Position or delay value.
        var startAcc = txn.ReadBEInteger(); // Starting acceleration value.
        var power = txn.ReadBEShort(); // Power value for the segment.
```

Liquid Handling Simulation:

An event is fired whenever the device's position is changed. This event is handled by the LiquidHandlingSimulation class and is used to react and update the simulation environment based on the device's movements.

Level Sensing is performed by slowly lowering the pipette tip while dispensing air and monitoring the pressure. When the tip reaches the liquid, the air is reflected off the surface, causing a pressure spike. This position is recorded to keep track of the volume of liquid in each tube.

Instead of simulating air pressure, the theoretical liquid level is calculated using a unique Volume Displacement Map for each type of tube. The pipette tip is lowered until that level is reached and a collision is recorded.

Whenever a collision occurs, the current move is cut short, and a response is sent back to workbench.

```
// If Myra is above a Tube and has a tip
else if (_currentTip != null && tubeDef != null)
{
    // Calculate tip orifice height
    var tipOrificeHeight = currentPos.Z - _currentTipParameters.TipLength;
    var newTipOrificeHeight = newPos.Z - _currentTipParameters.TipLength;

    // Preform level sensing
    if ((e.LevelSensingActive && _liquidVolumeMap.ContainsKey(well)) && e.LevelSensed != true)
    {
        // Determine liquid height in tube
        var liquidLevel = tubeDef.LevelMap.LookUpDisplacement(TubeVolume(well)) + TubeBase(well);

        // If tip is above liquid level and new position is below liquid level
        if ((tipOrificeHeight >= liquidLevel && newTipOrificeHeight < liquidLevel))
        {
            // Adjust the encoder position to calculated liquid level
            var robotLiquidPos = _coordinateConverter.RealToRobot(new XYZTriplet(currentPos.XY, liquidLevel + _currentTipParameters.TipLength));
            e.NewEncoderPosition = robotLiquidPos.Z;
            e.NewStepPosition = robotLiquidPos.Z;

            // Set Level Sensed to true
            e.LevelSensed = true;
            _sim.LevelSensePosition = robotLiquidPos.Z;
```

The P axis represents the piston inside of the Pipette head. It is used to aspirate liquid by creating a vacuum in the tip and dispense it by pushing air out. At the end of each move, the amount of liquid in the tip and tube is updated based on the change in the P axis.

```
// Calculate change in position
var change = e.EndPosition - e.StartPosition;

// Calculate volume of liquid in tip based on change in P position
double vol = _tipVolumeMap.LookUpVolumeuL(Math.Abs(change));
```

MyraAnimations:

Myra Animations is used to keep track of and animate the 3D models. The simulation features a precise representation of Myra, including all major assemblies such as the deck, X-Y gantry and the pipette head. The movement of each assembly is animated by updating its position each frame. The movement data generated from the Axis Controller is used to determine the model's position based on the current simulation time.

```
// Find the most recent MoveData entry within the list that
// is less than or equal to the current simulation time.
var mostRecentMoveData = moveDataList
    .Where(moveData => moveData.Time <= _vDevice.simulationTime.SimCurrentTime)
    .OrderByDescending(moveData => moveData.Time)
    .FirstOrDefault();
```

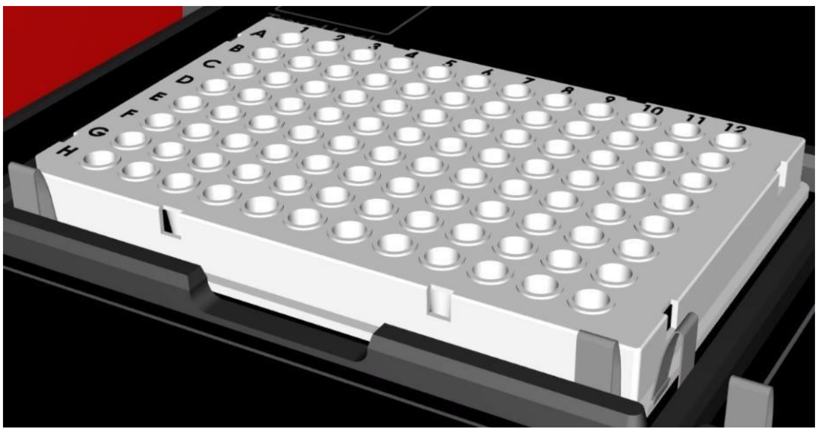
The simulation underwent a significant technical upgrade by migrating from AB3D PowerToys to AB3D DX Engine (AB4D, n.d.). The AB3D DX Engine leverages DirectX technology to deliver improved rendering performance, enabling the simulation to handle complex 3D models while maintaining smooth frame rates. This upgrade was crucial as PowerToys struggled when loading in complex consumable models. This caused the frame rate to drop significantly and resulted in a very unpleasant user experience.

Another major improvement in model fidelity came in the form of replacing AB3D's native OBJ model loader. AB3D's model loader has several limitations, especially when handling separate material files. These .mtl files accompany the model's OBJ file and contain information about the object's material, colour and reflectiveness. These files often failed to load, resulting in missing textures.

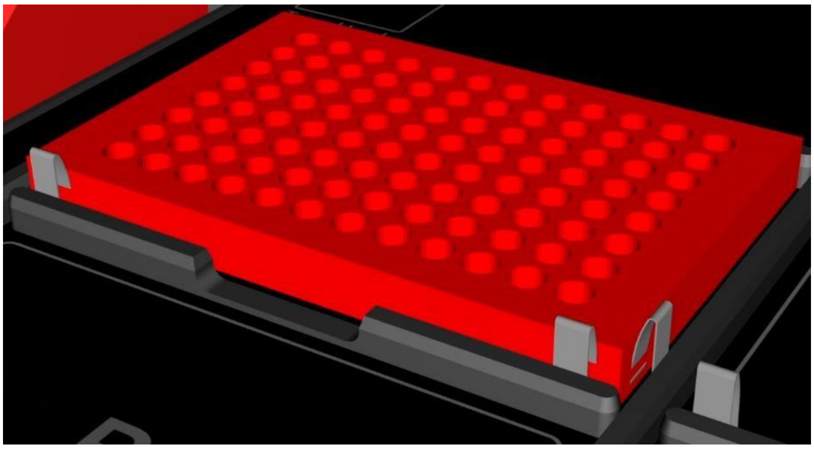
OpenTK **AB3D PowerToys** **AB3D DX Engine**

The Assimp Library offers much greater flexibility when importing models, offering a wider range of file formats and more robust logic for handling materials (Vilmring, n.d.). This was especially valuable as it allowed us to fine-tune the model's material, textures and other visual properties in Blender before importing them into the simulation.

A dual tier approach has been used for the consumable models placed on the deck. Highly detailed models are used for the most common consumables - such as the tip rack, and loading blocks developed by BMS. These models are created by BMS's engineers in SolidWorks and exported into a format usable by the simulation. As a large amount of manual preprocessing is required, there is a limited selection of these high-fidelity models.



Models are generated dynamically for less common plates that do not justify this development effort. These simplistic representations are created using XML files that define critical dimensions such as the width, height and depth of the plate, as well as the location of individual wells. These files are an integral part of Workbench and allow operations to remain flexible and compatible with any plate used by clients. While these generated models lack the physical refinement of the predefined models, they provide sufficient geometric accuracy to visualise Myra's interactions with the plates.



Quality Metrics

Code quality and functionality was continuously tested throughout the development of the simulation. The Test Plan outlines the comprehensive testing framework that was utilised. The tests were performed to validate both the core functionality of the simulation and its integration into the intended workflow. A more detailed version with numbered steps for each test is provided in the Test Plan Document.

Test Name	Brief Description
1. Software Installation and Setup	Verify the simulation and Workbench can be setup and run on a new device.
2. Device Connection	Verify that the simulation can connect to Workbench in the form of a virtual device.
3. Run Execution	Validating the entire workflow by opening Workbench and connecting to the virtual device, setting up either a MyraScript or demo run, completing the pre-run checklist and executing a run.
4. Consumable Model Loading	Ensure all consumables load correctly when pressing the 'load consumables' button as well as when dynamically loaded in when connecting to the device. Test both predefined models and dynamically generated plates show up correctly.
5. Camera Controls	Test all camera manipulation features including: all predefined camera angles, clicking and dragging and AB3D camera movement buttons.
6. Playback Controls	Testing play / pause functionality and speed selection capabilities, ensuring simulation responds correctly to different speed settings.
7. Time Navigation	Validating time slider functionality for scrubbing through simulation timeline and skipping forward to resume the simulation.
8. Operation Monitoring	Confirming the operation list displays all relevant operations and processes in real time. Ensuring the extend button correctly expands the operation tab.

Testing Framework and Results:

The above tests were conducted across several device configurations using a tiered approach to ensure broad compatibility and reliable performance across all potential deployment scenarios.

Test Number	Minimum Spec		Standard Spec		High End Spec	
	Debug	Release	Debug	Release	Debug	Release
1		✓	✓	✓	✓	✓
2		✓	✓	✓	✓	✓
3		✓	✓	✓	✓	✓
4			✓	✓	✓	✓
5			✓	✓	✓	✓
6			✓	✓	✓	✓
7			✓	✓	✓	✓
8			✓	✓	✓	✓

Minimum Specifications: The core functionality tests (Tests 1-3) were successfully completed using the release build in a virtual machine environment. While all critical tests passed, significant performance degradation was observed due to the AB3D DX Engine's reliance on hardware acceleration, which is unavailable in virtual environments. Despite reduced framerates, the simulation maintained full operational capability, validating the robustness of the simulation's core systems under constrained conditions.

Standard Specifications: Comprehensive testing of all 8 test cases was conducted on a standard issue BMS Work Laptop using both debug and release builds. This dual-build approach ensured feature parity between development and production environments while validation performance under typical development conditions. All tests passed, confirming the simulation's functionality in the intended environment.

High-End Specifications: The complete test suite was also conducted on two high-performance desktop PCs. These tests were critical to validate frame rate independence and timing accuracy. Early development versions had Myra's movement tied to frame rate, causing the simulation to run faster than expected at 'real-time' speed on high-performance hardware. High-end testing confirmed that these issues were successfully resolved, with results showing no differences in simulation timing or behaviour between standard and high-end configurations. This validation was essential to ensure consistent real-time simulation performance regardless of hardware capabilities.

Issue Resolution and Quality Improvements:

This iterative testing approach resulted in the identification of several critical issues that were addressed. Performing these tests also resulted in several crashes at first, leading to several optimisations for the release build. These included switching from x86 to x64 architecture for improved stability, transition to a 'portable' release to eliminate .NET runtime dependencies and disabling other build features like Single File Packaging to resolve compatibility issues.

Testing using both builds also highlighted issues with the model loading system, as there were file path inconsistencies between the two versions. This resulted in further development of a more robust model loading system. Several smaller bugs were also identified such as the simulation speed to not decreasing as expected, no play / pause persistence after using the time slider and certain operations not being displayed correctly in the Operation List.

User Feedback Integration:

The simulation underwent continuous evaluation by several BMS Engineers throughout its development lifecycle, with their feedback systematically integrated to enhance both functionality and usability.

Hardware configuration testing proved essential when a BMS Engineer identified the critical frame rate dependency issue. Since development was conducted primarily on high-end desktop PCs, the timing related problem remained undetected until initial user testing on standard hardware. User reports of poor frame rates also directly influenced the decision to upgrade from AB3D PowerToys to AB3D DX Engine, resulting in substantial performance improvements.

The UI was also streamlined based on stakeholder feedback emphasizing minimal interface requirements, as the simulation will be directly integrated into Workbench in the future. The operation list information density was also optimised by trimming operation details deemed unessential by BMS Engineers.

Project Success and Future Development:

While this project represents a work in progress that will continue to evolve, it has achieved significant success and exceeded BMS's expectations. The result is a fully functional simulation platform that BMS Engineers can immediately utilise for training, demonstration and development purposes. This project has also established a robust technical foundation that provides an excellent starting point for future expansion by subsequent Capstone teams and integration into BMS Workbench by their engineering staff, ensuring that the simulation will continue to grow and adapt to meet BMS's evolving needs in laboratory automation and training.

2: Individual Chapter: Ethan Bett (n10752251)

2.2: Project Setup

Project Management Approach

My primary contribution to project management was creating and organising our Phase 2 Jira page. I created individual sprints that aligned with our timeline and populated each sprint with initial tickets that outlined the main objectives for each group member. The initial tickets were then expanded on by each individual by adding more detail and splitting them up into individual tasks based on their own inherent knowledge of that section.

Client Relationship and Expectations

I helped establish and maintain a professional relationship with BMS throughout the Project. At the beginning of Phase 2, I set up a meeting to refresh them on our Phase 1 progress, address concerns about project scope and align on the project's direction. Open and honest communication in this meeting helped set client expectations and allowed us to focus on the most important aspects of the simulation. I maintained regular communication with the BMS team outside of our fortnightly meetings to update them on our progress and ask for technical clarifications. I would often send the engineers images of how the models looked in the simulation. This was essential as there were some issues when exporting them from SolidWorks, resulting in them looking different to how they appear in the final SolidWorks render. This continuous communication helped establish trust and transparency about the projects progress throughout phase 2.

Team Collaboration

My approach to team collaboration was to focus on creating an inclusive environment where we all felt supported and not afraid to ask for help. I actively participated in creating our team agreement, ensuring that everyone had input and we were all comfortable with the established expectations.

I maintained consistent communication through our Discord channel, providing regular updates about my individual progress and actively asked group members for feedback on my work. This open communication style encouraged similar transparency and help maintain our cohesion as a team. Beyond my own tasks, I regularly checked in with team members to ask how they are going and offer my assistance when needed. When technical challenges arose, I arranged one on one calls with team members to collaborate directly and solve problems together.

Communication Plan

During the initial planning phase, I collaborated with the team to create a communication plan that established our routinely communications with BMS, our tutor and each other. During our initial Phase 2 meeting, I proactively communicated our intended meeting schedule to BMS, informing them of our desire to hold fortnightly meetings, alternating between in person and online. Communicating this in advance simplified the process of organising subsequent meetings, allowing them to anticipate and plan for our regular check ins.

I also helped organise weekly team meetings either the day before or day of our meetings with the client / tutor. These sessions helped us coordinate our presentations, address any concerns we have, and ensure we are all on the same page for the meetings.

2.3: Project Plan and Risk

Project Planning and Progress

I took responsibility for maintaining the Jira page we used to plan and keep organised throughout the project. I continuously refined and created additional tickets as the requirements became clearer. This iterative approach allowed me to break down complex tasks into smaller, more manageable components.

When technical challenges arose or when individual tasks proved too large for an individual sprint, I collaborated closely with group members to reassess priorities and shift tickets to different sprints to ensure we maintained a realistic timeline with achievable goals.

Risk Management

I collaborated with the rest of the team to create the Risk Management document. We updated this document throughout the project as new risks were raised. The biggest unforeseen risk was the Cyclone that hit Brisbane early in the second Phase. We kept in communication with BMS as we had to delay our meeting with them due to Yatala loosing power. Due to our initial planning, regular backups of the project were already made in the event that something unexpected (like the cyclone) caused data loss.

As the project manager I tried by best to negate any severe problems by encouraging constant communication between the four of us. This ensured that we were all on the same page and able to address and issues as soon as possible before they became major obstacles.

2.4: Project Experience

Reduction in Project Scope:

Situation:

After establishing the foundation during Phase 1 and gaining a comprehensive understanding of the underlying systems, it became clear that we were facing an incredibly ambitious task. MyraSim encompasses a wide range of desired functionality, and it became clear that we would not be able to implement everything by the end of Phase 2.

Task:

Early in Phase 2, I needed to facilitate a difficult conversation to address the project scope. I was concerned about disappointing BMS as I was worried that our inability to deliver all the desired features would be perceived as a failure.

Action:

Rather than attempting to conceal our concerns or making unrealistic promises about deliverables, I decided to approach the conversation with complete transparency and honesty. During our first formal meeting of Phase 2, I initiated a conversation about project scope and asked about what features should be prioritised as we were realistically not capable of implementing everything within the desired timeframe.

Result:

The outcome of this transparent approach exceeded my expectations entirely. BMS were very understanding and expressed that they had never expected us to deliver every conceivable feature by the end of Phase 2: as the plan was always to pass it on to another Capstone Team. They appreciated our honesty and agreed with us saying that they would rather us build a solid foundation for the next team over a rushed implementation of every desired feature.

Learning:

This experience taught me valuable lesson when it comes to professional communication: transparency and honesty is always better than overpromising and under delivering. I had initially assumed that admitting our constraints would disappoint them, but instead I learnt that experienced industry professionals actually value realistic assessments and honest communication above ambitious promises.

Overcoming the Fear of Asking Stupid Questions:

Situation:

Working on a highly technical project meant that encountering technical roadblocks was inevitable. During development, I found myself struggling to understand a particular function within the LiquidHandlingSimulation Class, which is critical to the simulations core functionality. Despite spending a considerable amount of time attempting to decipher the code's logic, I remained confused about its implementation and purpose.

Task:

I needed to gain clarification on this function as its unexpected behaviour was leading to issues and had halted my progress. I was hesitant to seek help as I was worried my question would seem stupid to the engineers at BMS and was concerned that asking it would reflect poorly on my abilities.

Action:

Despite my reluctance at first, I decided to reach out to one of BMS's engineers and ask for guidance. I approached the conversation by sharing my current understanding of the function, acknowledging my confusion and asking for clarification on the specific section that was unclear to me.

Result:

Not only did they provide clear insight into specific function that I was struggling with, but the conversation also naturally evolved into a broader discussion about the entire class and the difference in implementation compared to the physical robot. This conversation provided me with a much deeper understanding of how the physical robot operates provided me with knowledge that I would have otherwise never known.

Learning:

I learned that what I perceived as a potentially 'stupid' question was actually a valuable opportunity for learning and collaboration. I learned that industry professionals (at least the ones at BMS) appreciate genuine curiosity and engagement with their work, and collaboration leads to a much greater learning experience when compared to struggling alone.

Consequences of Rushed Implementations:

Situation:

The night before a client meeting, I was working on implementing the Assimp Library to import models, however I was unable to get them to load properly due to multithreaded concurrency issues, where two separate threads were trying to access the variable. This resulted in runtime exceptions causing the program to crash immediately.

Task:

I needed to fix this issue ASAP to get the new models working for the demonstration the next day. The core problem involved synchronising access to the shared variable with proper thread safety to avoid race conditions.

Action:

Given the time constraints, I implemented a suboptimal workaround to resolve the immediate issue rather than taking the time to implement a proper solution. I told myself that I would return to this section soon to fix it and I failed to document exactly what I changed to get it functional.

Result:

The immediate result was successful, the project was no longer crashing, and we were able to show BMS the latest version of the software. However, this short-term success resulted in long term headaches. Later, towards the end of Phase 2, when adding the final features, this poor implementation began to have unintended consequences. When attempting to address the problem, I realised that I had completely forgotten the specific changes I made months earlier. I was forced to spend considerable time comparing the current code with earlier versions, to reverse-engineer my own work to understand what I had changed.

Learning:

This experience was very frustrating and taught me a crucial lesson about maintaining code quality even when under a time crunch. I learned that quick fixes with the intention of cleaning up later rarely get properly addressed and fixing the inevitable issues becomes more time consuming that implementing a proper solution from the start.

3: Individual Chapter: Milly Chambers (n11318546)

3.2: Project Setup

My Contributions:

1. Investigating performance issues
   - Analysed framerate drops and concluded that performance issues were related to engine limitations
2. DirectX Migration and Lighting Fixes
   - Led the transition from the legacy engine to the DirectX based one, ensuring compatibility with newer 3D models.
   - Adjusted shader parameters and lighting to fix bright, over-reflective appearance of models.
3. Model and animation integration
   - Collaborated with Client to ensure that models worked with the simulation optimally.
   - Experimented with camera functionality to see what was possible within the engine.

3.3: Project Plan and Risk

Project Management Approach

- Weeks 1-2:
  - Investigate Performance Issues ○ Start moving to a more powerful engine
- Weeks 3-4: ○ Fix model Lighting and Look
- Weeks 5-6: ○ Incorporate the latest models into the simulation
- Weks 7-8:
  - Final Refinements to models and animation

Risk Management

- Risk: Corruption or loss of Project Artefacts. ○ Mitigation: using a GitHub repository for software and a OneDrive folder for written
- Risk: Falling behind on sprint plans and expectations ○ Mitigation: Maintaining regular communication to ensure no parts of the project are lacking.
- Risk: Overlapping work tasks ○ Mitigation: Using project management tool to assign tasks to individuals
- Risk: Failing to deliver work that is up to industry partner standard ○ Mitigation: regular meetings with our industry partner, ensure their feedback is heard and then worked on for future meetings.
- Risk: Unforeseen medical emergencies ○ Mitigation: Constant communication so each team member is aware of any emergencies.

3.4: Project Experience

DirectX Migration

- Situation: As Consumable models began to be implemented, performance tanked even on high-end computers.
- Task: Migrate the Program to the high-performance DirectX based engine.
- Action: researched Ab4D's DXengine through their provided examples on GitHub and applied the relevant parts of the code to our program.
- Result: Program was successfully converted to high performance engine with performance improving tenfold (Based on FPS count).

- Learning: Selecting relevant parts of code from examples and learning how to apply that to our program.

Client Scope Creep

- Situation: Client expectations and requirements were given at the beginning of the Capstone project, but as we progressed into phase two it was clear that not all client requirements could be delivered in time.
- Task: Communicate with Client and negotiate deliverables
- Action: During a Client meeting we assessed what we have done so far, and what more we could feasibly do with our remaining time.
- Result: It was agreed that the fluid visualisation could be removed from the project scope and given to the next Capstone group.
- Learning: Communicating clearly to the client about expectations is incredibly important to ensure a quality final result.

Model Issues

- Situation: The given models for our simulation needed to be edited to better fit our simulation, as initially they were quite rough and needed more detail to make the simulation as good as it could be.
- Task: communicate with Client on what is needed, and what our simulation is currently able to handle and achieve models that were a good balance between detail and simulation performance.
- Action: over a couple weeks we worked together with the client to modify the models to achieve what we desired.
- Result: New models were created for us that matched our requirements.
- Learning: Communicating clearly to the client about our needs to create a quality product proved critical.

4: Individual Chapter: Samuel Vosper (n10753508)

4.2: Project Setup

Project Management Approach

I participated in the set-up of project management through creating tickets on Jira of all of the tasks that I need to have completed. These tickets were regularly updated and created to maintain effective project management. As I had the role of software developer on the team, this was very important to have done, as it not only kept track of my work, but was also visible to my team members, so they were also aware of my progress.

Client Relationship and Expectations

As we didn't have a specific client liaison role on our team, we all shared the responsibility of communication with the client. Through this communication, I managed our clients expectations through frequent discussions on what tasks they wanted completed, and what was realistically viable. Some of the features they requested I either did not have the time for, or believed it was out of my scope of practice.

Team Collaboration

I contributed to the team's collaboration by always remaining active in our teams messaging platform, in case team members had questions or thoughts. I tried to always maintain a positive and friendly attitude, to foster more collaboration on the project, and in turn exceed client expectations. While I didn't have much involvement in the team agreement, I ensured to always participate in team meetings and provide help if any issues within the team or project were to arise.

Communication Plan

In terms of my team members, I consistently provided my thoughts and feedback throughout our discussions, ensuring to be friendly and supportive. With our tutor and client, I provided updates on the progress of my work, and asked questions, to make sure I received valuable feedback from both. To set the meetings up with our client, we first discussed the availability of our team members and the client, and then emailed our client confirming the date of our next meeting. That way, it ensured professional communication throughout the semester. Likewise, during our meetings I made sure to do my best to speak professionally, to not only give a good look for myself and my team, but to also garner experience working in a professional environment.

4.3: Project Plan and Risk

Project Planning and Progress

To plan out my specific tasks, I discussed with our client what tasks are highest priority. I then created tickets for these tasks on Jira and personally decided the order to complete the highest priority tasks. When I began working on a specific task, I would update the status of the ticket related to that task to *In Progress*, and once completed I would make it as *Done*. Each meeting with the client I would then show them the progress I had made, and ask for feedback and comments to further improve it to their liking. If something needed to change, I would then begin working on that task, updating my Jira tickets as I went along. Through this strategy, my specific tasks were coordinated to meet the clients expectations, and my progress was transparent.

Risk Management

While none of our roles related to risk management specifically, we all shared the responsibility of managing said risk. At the beginning of the semester, we all sat down and brainstormed any form of risk that may affect our project, and the ability to meet deadlines. As any new risks came to our attention, such as the possibility of a cyclone, we added the risk to our risk management table, and made certain precautions in case any of these risks were to occur. While no major risk did occur throughout the duration of this project, my team and I ensured to maintain regular communication each week, to keep ahead of our work in the case where something were to happen.

4.4: Project Experience

One challenging situation that occurred was when I was creating the operation list of the simulation, where I had to print out all the relevant operations and processes that were sent down the pipeline. This was challenging as firstly, I haven't had much experience with binding data in XAML. I also had to create a function to iterate through all of the information from the pipeline, and organise it in a tree like structure, all the while trimming the information to only portray certain text that was necessary. Through extensive research and assistance from one of our client's employees, I was able to overcome this problem and learn how to both bind XAML elements to backend functions. I have also learnt how to create a tree structure using information received from a pipeline, while editing this information accordingly.

Another challenge was one I faced at the end of the project. The last feature I had wanted to add to the simulation was the ability to use a 'mini' time slider for each key operation of the run. However, to do this was more challenging than I thought. As no duration was attached to each operation, it was difficult to determine the start and stop times for each. I ended up running out of time implementing this feature, however it has still been a valuable lesson. I learnt to never underestimate tasks, as they may be more difficult than initially anticipated. Looking back, a possible way for me to achieve this task would be to set the start and end times of each operation as they come through the pipeline, then use this information in the time slider to accurately view each operation separately.

Getting the slider to accurately control the simulation proved to be quite a challenge, as I also hadn't had much experience with issues such as this before. I wanted the slider to control the simulation time, which wasn't very difficult to implement, however ensuring it moved the models as I intended was where the problems began. Initially, I was able to get the slider to move the simulation back once you released control of the slider, however that wasn't good enough. The simulation should change as the slider is being dragged, so the user can tell where they want to drag it to. After a long time of researching possible methods, and how to actively update the UI through backend manipulation, I learnt how to successfully implement this function.

5: Individual Chapter: Chen-Jui Yeh (Ray) (n11374268)

5.2: Project Setup

- Create model generation functions
- Connect generated models into virtual Myra
- Adjusting model loading methods

In phase 2, most of my work is related to consumable model generation and integration into the virtual Myra simulation. This included creating model generation functions, rendering generated models into simulation views, and adjust model loading process to support both predefined models and dynamically generated models.

5.3: Project Plan and Risk

As a core developer responsible for implementing consumable model generation and integration into the virtual Myra simulation, my individual planning focused on researching tools, developing model generator logic, and secure integration compatibility. In Phase 2, I set up my tasks and goals based on sprint plans established during the meetings, with updates are record via Jira and communication often on Discord. Where I usually divide my allocated job into smaller subtasks for better management and. Including testing external libraries, evaluating Boolean support, and update model loading logics.

In terms of risk management, I focused on identifying and mitigating risks related to my work like tool compatibility, technical limitations, and data integrity. Reliance on Blender scripting was the first early risk found, which was quickly marked and mitigated due to deployment conflicts. Helix Toolkit was supposed to be the mitigation, yet the downside of lacking Boolean operation support is huge, and I have to seek for alternative risk mitigation. Eventually, I pivoted to AB3D, a library already used in the project, which allowed seamless model generation and integration. Another significant risk involved parsing and using the XML files provided by BMS. If the XML structure was inconsistent, it would increase the difficulty for model generation for the simulation. Where I respond to this risk by developing fallback mechanisms that if certain XML is incorrect of missing, the system will log the issue for future fix. I also incorporated input validation steps to minimize unexpected behaviour.

By participating in both planning and risk reviews, I make sure my technical contributions aligned with broader team objectives while remaining adaptable to evolving requirements. This experience strengthened my ability of planning to a technical project within a structured timeline and anticipate risks during early development phases, skills that I will carry forward to future collaborative software developments.

5.4: Project Experience

Ab3D geometry operation adaptation

After discussing with out clients about consumable requirements during the early planning stage of Phase2, it became clear that it would be necessary for the simulation to support dynamic model generation. This was especially important if the predefined models were unavailable, and the simulation is also should be capable of creating models from most of the XML configuration files with specific structures. My tasks was to research and implement the dynamic model generation, ensuring compatibility with our C#-based simulation. In the beginning, I had investigated multiple tools and libraries across multiple programming languages. My first attempt involved using Blender's scripting capabilities to fit our requirements, which worked technically, yet it has a major limitation of the need to install Blender in user end. This was conflicted with our deployment goals and client expectations.

Following that, I decided to have a look on Helix Toolkit, a C# library know for its ease of use in 3D rendering. Although Helix made it simple to visualize basic models, yet it does not support Boolean operation which is essential requirement for accurately modelling features like well holes on the tube plate. This drawback prompted me to revisit our existing tools. Eventually, I realized that AB3D, the rendering library already integrated into our simulation, was capable of much mor that we had initially used it for. AB3D provide build-in support for complex model generation, including Boolean operations. Thus, it's used for our model generation functions and had provide solid output throughout our test. This solution is fully integrated into our project without any additional dependencies required.

This experience had given me a lesson about thoroughly understand the capabilities of the existing tolls before turning to external options. Although Blender and Helix Toolkit offered powerful features, yet they did not align with our expectation and functionality needs. In hindsight, staring with a deeper exploration of AB3D could have saved lots of time and effort of mine. Nonetheless, the process significantly strengthen my understanding and ability to deal with 3D models, as well as the capability to evaluate and align tools with real-world project constraints

Simulation coordinate

During Phase 2 of the project, one of the key challenges I cased was learning and adapting to the simulation's 3D coordinate system to make sure that generated consumable models where placed properly on simulation. Since the simulation was already structed with specific layout and design for components like plate, pipe, and Myra itself, It is important that the model generated would align correctly in both position and scale.

To address the issue, I spent time analysing the coordinate system in the simulation and also how AB3D handle model size, transformation, and translation. I conducted tests with both the simulation and AB3D models to find the correct logic on coordinate system, where turns out the Y axis and Z axis are change respectively. This mismatch caused early test models to be rotated or misaligned. By adjusting the logic and value passed and comparing result visually in the simulation, I was able to correctly understand the coordinate system and make model generator produce correct results.

This experience improved my spatial reasoning and taught me the importance of validating assumptions when working with third-party coordinate systems. It also reinforced the value of hands-on testing when documentation is limited or unavailable.

Appendices

Appendix A: Final Team Agreement

Name	Roles	Responsibilities	Knowledge and Skills	Artefacts
Ethan	Project Manager	• Sprint Planning • Organising Team Meetings • Progress Reporting	• Teamwork • Agile Framework • Familiarity with Jira	Phase 2 Jira Page
	Software Development	• Develop and test additional functionality for MyraSim	• C# Programming • Object Oriented Programming • Good Software Practices	Developed and expanded major backend functionality: -Axis Controller -Liquidhandling Simulation -MyraAnimations Helped integrate team's individual components by expanding backend to accommodate the respective functionality.
	Technical Writer	• Write MyraSim User Guide • Write Software Architecture Document	• Writing experience • Innate understanding of MyraSim code	Setup and Execution Guide, Architecture Overview, Model Specifications. Contributed to the script for our demo video, Test Plan and User Story Documents.
Matt	Software Development	• Develop and test additional functionality for MyraSim	C# programming Object Orientated Programming 3D model manipulation	Started GitHub Repository to ensure team could effectively collaborate on code and ensure we were protected against data loss through source control.
	Software testing	• Extensively test both my and others implemented features to ensure they work in most use cases	Extensive Knowledge into how the program is intended to run.	Ensured all functionality made was fully functional before publication to GitHub page
Ray	Software development	• Develop and test additional	C# Programming NET structure.	Developed all model generation related files
		functionality for MyraSim	3D model understanding	and adjust model loading files.

	3D Model creation	• Develop and test 3D model generating program.	C# Programming, 3D model knowledge, Blender, Helix3D library.	- XMLDefinitions folder - ModelGnerate.cs - Update file in Fronted folder - MyraAnimation.cs - Help others with model issue if there are bugs. Test visualization and loading of generated model.
Sam	Software Development	• Develop functionality to work with the Myra UI	• C# Programming • Object Oriented Programming	Developed all the required backend functions for all the major UI components
	UI Development	• Develop and improve current Myra UI	• XAML Programming • Regular Communication with Industry Partner	Developed the main features of the UI, including the layout, operation list, certain camera views, time slider and other buttons
	Tester	• Functional Testing • Acceptance Testing	• Prior experience with manual testing methods	Ensured the functionality of the UI works as intended, including from a user's perspective
	Technical Writer	• Writing test plan • Demo video script • Various parts of the report	• Writing	Wrote the test plan to ensure the current functionality of the software works in different environments, and in future iterations of these project. Contributed to the script for our demo video.

Appendix B: Team Contract

Generative AI Disclosure Statement

We do not intend to use GenAI for this project.

We will update this statement if we decide to utilise AI in the future.

Social Contract

We are excited for this opportunity to work on a big project and understand the importance of teamwork. We are all committed to work together and play off each other's strengths. We all know how life can get in the way sometimes and acknowledge that flexibility and understanding are key to maintaining a strong team. We will support each other through challenges and celebrate our success together.

- We are committed to spend roughly 10 hrs a week on this project
- Open and honest communication
- Making sure we are all on the same page
- Life happens - make sure we help one another when something unexpected happens
- Reply to messages within 48 hours
- Ensure all work is completed efficiently and on time where possible
- Plan to have weekly meetings outside of tutor and industry partner meetings.

Teamwork Checklist

- I will contribute and share my ideas equally with other team members
- will listen to and value the ideas of other team members
- I will be open to new ideas and to different ways of working
- I will encourage other team members
- I will give feedback in the form of constructive criticism
- I will bring a positive attitude to teamwork in this project
- I will complete tasks assigned within the group on time
- I will attend all team meetings decided on by the group
- I will do my share of the work associated with the team project
- My contributions will equal (or exceed) others in quality Names and signatures:

Ethan Bett - EB

Milly Chambers – MC

Ray Yeh – RY

Samuel Vosper – SV

Date: 22/05/2025

Appendix C: User Stories

ID	User story	Acceptance Criteria	Estimate time	Priority

1	As a user, I want to open the simulator software, so that I can begin a simulation session.	- Application launches properly - No startup errors or crashes occurs - A visible UI indication the simulation is ready for use	2 sprints	MUST
2	As a user, I want to move the camera and switch between different angles, so that I can view the simulation from various perspectives.	- User can rotate, pan, and zoom the camera. - Users can select at least 3 predefined angles. - All camera transition is smooth and responsive.	1 Sprint	SHOULD
3	As a user, I want to connect the simulator to Workbench as virtual device, so that I can simulate the physical connection process.	- The simulation appears in Workbench as an available device. - Workbench successfully sends command to the simulator. - Connection is persistent throughout the run.	1 Sprint	MUST
4	As a user, I want to load consumable models into the simulator, so that I can visualise the real-life usage scenarios of those specific plates.	- All major consumables (eg: tubes, plates) load without error. - Consumables are visually positioned correctly on the platform. - Loading a model does not crash or stall the application.	2 Sprint	MUST
5	As a user, I want to start a run using the virtual device, so that I can simulate Myra's execution of a run from start to finish.	- Simulation begins upon command with appropriate animation and state updates. - All consumables used in the run are correctly loaded in the simulation. - Progress is visibly shown (eg: progress bar, operation lists)	1 Sprint	MUST
6	As a user, I want to run different types of Runs (Myra Demo, Simple Transfer, MyraScript), so that I can validate a variety of workflows.	- Simulator works for different types of runs (Myra Demo, Simple Transfer, MyraScript). - Able to switch between runs and reset the simulation state without closing the program.	2 Sprint	MUST
7	As a user, I want to simulate an entire run without errors, so that I can mimic the physical device's execution of operations.	- Simulation run from start to finish without crashing - No warnings or errors occur in console/log. - Output matches expected scripts.	2 Sprint	MUST
8	As a user, I want to pause, rewind, speed up, or slow down the simulation, so that I can analyse specific moments and device behaviour.	- Playback can be paused/resumed mid-run. - User can change playback speed (eg., 0.5x, 1x, 2x) - Rewind or skip-back functions do not break the simulation state.	1 Sprint	SHOULD
9	As a user, I want to skip to different operations within a simulation, so that I can	- A timeline or step-based UI allows jumping to operations - Simulator update visuals and stat accordingly	1 Sprint	SHOULD
	quickly navigate to relevant segments.	- Skipping does not cause UI or functionality errors.		

10	As a BMS staff member, I want to follow the user guides with minimal assistance, so that I can validate ease of use and documentation clarity.	- User's can complete installation and runs a demo without outside assistance. - Guide instructions match UI and terminology exactly.	1 Sprint	MUST
11	As a BMS staff member, I want to install the simulator and the compatible Workbench version, so that I can confirm setup compatibility.	- Installation completes with no errors on BMS's standard Work Laptop. - Required dependencies are included or documented. - Simulation runs without performance issues when using this standard hardware.	1 Sprint	MUST

Appendix D: Project Artefact References

- Link to Source Code, Built Version and BMS Workbench Installer
- IFB399 - T215 Project Artefact Backup
- Setup and Execution Process: "Submission Artefact - Team 215\Setup and Execution Process.pdf"
- Architecture Overview: "Submission Artefact - Team 215\Docs\Architecture Overview.pdf"
- Model Specifications: "Submission Artefact - Team 215\Docs\Model Specifications.pdf"
- Test Plan: "Submission Artefact - Team 215\Docs\T215 Simulation Test Plan.pdf"
- Jira Page Snapshots: "Submission Artefact - Team 215\Project\Jira Page Snapshots.pdf"
- Side by Side Comparison Video: "Submission Artefact - Team 215\Demo\Side by Side Demo.mp4"
- Short Demo Video: "Submission Artefact - Team 215\Demo\Short MyraSim Demo.mkv"
- Demo Video for BMS: "Submission Artefact - Team 215\Demo\Demo Video for BMS (With Voice Over).mp4"
- T215 Phase 1 Page
- Phase 1 Demonstration Videos
- Phase 1 Source Code

References

AB4D. (n.d.). *Ab3d.DXEngine: DirectX 11 3D rendering engine for WPF and WinForms*. Retrieved from AB4D: https://www.ab4d.com/DXEngine.aspx

AB4D. (n.d.). *Ab3d.PowerToys: The ultimate 3D graphics toolkit for WPF and WinForms*. Retrieved from AB4D: https://www.ab4d.com/powertoys.aspx

Bio Molecular Systems. (n.d.). *About Us*. Retrieved from Bio Molecular Systems: https://biomolecularsystems.com/about-us-bio-molecular-systems/

Bio Molecular Systems. (n.d.). *Home Page*. Retrieved from Bio Molecular Systems: https://biomolecularsystems.com

Bio Molecular Systems. (n.d.). *Workbench Software – Integrating Mic, Myra & Myra+*. Retrieved from Bio Molecular Systems: https://biomolecularsystems.com/myra-liquid-handling-system/software/ OpenTK. (n.d.).

Basics & Requirements. Retrieved from OpenTK: https://opentk.net/faq.html

Vilmring, K. K. (n.d.). *The Open Asset Importer Library*. Retrieved from Assimp: https://github.com/assimp/assimp