Change Report

Cohort 3 Group 6 - Carbon Goose

Members: Bailey Horsley, Owen Jones, Rory Ingram, Ken Jacob, Abishek Kingslin Raj, Louis Polwarth, Adam Johnson

Summary

Our team began by familiarising ourselves with the inherited deliverables and code. We focused on ensuring that the project aligned with the updated product brief and addressed the requirements effectively. This phase involved careful analysis of the previous work, identifying key areas for improvement, and implementing changes that aligned with our new methodology and goals.

The initial team utilized Agile with Scrum as their engineering framework. We transitioned to using IBM's Rational Unified Process(RUP). This methodology better suited the fixed and stable requirements of our project and supported a plan-driven development approach. The collaboration tools used for task management, collaboration and code development include Google Drive/Docs, PlantUML, GitHub, IntelliJ IDE, Java 17, LibGDX and Discord. The tools we used remained consistent, but we focused more on RUP-driven modeling with UML diagrams, reflecting the shift in methodology. Key milestones included updating risk mitigation and documentation early, allocating tasks over the break to maintain progress and completing user evaluations and unit tests before revision week.

We updated the architecture by adopting a combination of MVC and ECS approaches to ensure modularity. Refinements were introduced to better meet functional and non-functional requirements, such as improving the satisfaction calculation to consider building proximity, quality, and events. We also incorporated event management and grid-based simulation into the design. Additionally, new components like particle emitters and sound effects were added to enhance user feedback. Our updated diagrams reflect these changes while maintaining the original structure where possible.

The documented User, functional system and the non-functional system requirements were helpful for tracking progress.We added new features and functionalities to address missing elements, ensuring the requirements were comprehensive and aligned with the updated project goals.We employed a colour-coding system to indicate the status of each requirement.

For the risk assessment, our team refined the process by incorporating lessons from the first assessment performed by the initial team. The updated risk assessment focused on addressing implementation-specific risks and risks specific to our team. We modified the priority of risks, including reducing the likelihood of some, such as hardware failure, and increasing the severity of others, such as the impact of bugs on user experience. New risks were also added, including delays in receiving deliverables and challenges with understanding inherited code. Mitigation strategies were updated to reflect these changes, ensuring we could address potential issues effectively. Risks were frequently reassessed, and the risk register was updated regularly to adapt to project needs.

Our team successfully adapted and expanded upon the work of the initial team by building on their foundation while implementing changes to suit our goals. By transitioning to RUP, refining the architecture, and improving risk management, we ensured that the project requirements were addressed efficiently and effectively. Each stage of the process was carefully monitored, enabling steady progress and a structured approach to project development.

Requirements

Old Deliverable URL:

https://carbongoose.vercel.app/Req1.pdf

New Deliverable URL:

https://carbongoose.vercel.app/filesV1/Requirements.pdf

Notable Changes and Justification:

Our group reviewed the previous group's requirements table and identified areas that needed updates to align with the new product brief. The updates included adding a few additional requirements specified in the updated product brief, which address features and functionalities that were not covered in the original table but were important to meeting the updated features of the game. By incorporating these additions we ensured that the requirements were comprehensive, setting a clear foundation for development.

Architecture

Old Deliverable URL

https://carbongoose.vercel.app/Arch1.pdf

New Deliverable URL

https://carbongoose.vercel.app/filesV1/Architecture.pdf

Notable Changes and Justification

There have been two major changes to the architecture documentation that we received from the previous group. Firstly, the algorithm to calculate the satisfaction score was updated to be more realistic to how student satisfaction would be measured in real-life. Previously, the satisfaction score was only impacted by a velocity which updated over time due to building distances (as the crow flies), but was capped by the number of accommodation buildings. We found this extremely confusing in the first iterations of playtesting, and therefore decided to change it to be more intuitive. Additionally, due to the introduction of new variables such as events and more types of buildings, we decided to incorporate that into the calculations:

1. The first change made was for the distances to be calculated using the pathways instead of in a straight line, and this makes much more sense in the context of a university, as people will more likely use roads than walk through fields.
2. The old satisfaction calculation would automatically set the global satisfaction to zero if there was a building category without any buildings in the world (e.g. there are no food buildings). This was changed so that each accommodation has a local satisfaction score, which was made up of multiple parameters such as:
    a. The number of buildings of each category nearby.
    b. The quality of those buildings.
    c. The quality of the accommodation itself.
    d. Any local events to the accommodation which affect satisfaction.
3. We also added a globalSatisfactionModifier to the GameState class to deal with any events that affect the university as a whole.

The algorithm section 6.1 was updated to reflect these changes.

Additionally, we added our new structural and behavioural diagrams to the end of the document to reflect the changes that we have made to the codebase, as we have added several new components which were necessary to implement the new requirements for assessment two, as well as to polish the look and feel of the game itself. For example, we have added Particle Emitters to increase the amount of user feedback as they play the game, as well as sound effects for the same reason. As much as possible, we have kept the internal game structure and behaviour the same, so as to keep the original diagrams and explanations in the architecture document in-date, and to keep the project reflecting the aims of the original group.

Method Selection and Planning

Old Deliverable URL

https://carbongoose.vercel.app/Plan1.pdf

New Deliverable URL

https://carbongoose.vercel.app/filesV1/Method%20Selection%20and%20Planning.pdf

Notable Changes and Justification

One of the first notable changes that was made to the method selection and planning section where we changed the methodology that we followed for development from scrum methodology to IBM's RUP method as it fit better with how we preferred to develop our game as a group. So we removed the parts referencing this methodology and updated with our new methodology.

Another change we made was to remove the parts of the report that referenced alternative softwares that could be considered as we just continued to use the same strategy from assessment 1 and did not consider any alternatives anymore as they were not required.

We also had a different team organisation strategy from the previous group which meant that this was updated in the report to reflect this change. We changed it as during assessment 1 we used a different strategy which we found worked effectively so we thought we would continue to use this method.

Another significant change made to the method section and planning report was to remove the previous group's Gantt charts are they are no longer applicable to us and replace these with newly updated Gantt chart to reflect this next phase of the project.

Another change is the work breakdown diagram as the content of this diagram is different in assessment 2. So the previous work breakdown diagram was removed and a new one added to reflect this progression in the project.

Finally the project plan evolution part of this report has been updated to reflect this next stage of the project. The old project plan was removed to facilitate this change as they are no longer required for this stage of the project. We then added a week by week plan for our project in this next stage.

Risk Assessment and Mitigation

Old Deliverable URL

https://carbongoose.vercel.app/Risk1.pdf

New Deliverable URL

https://carbongoose.vercel.app/filesV1/Risk%20Assessment%20and%20Mitigation.pdf

The updated risk assessment reflects a shift in priorities based on the challenges faced during Assessment 2. While the initial risk assessment of the selected team covered various risks across project, product, and business areas, our revised risk assessment focused more on implementation-specific issues and the project and business risks relevant to our team.
Some risks from the initial assessment required only minor adjustments for this assessment and were tailored to our team's current needs and situation.

Exclusion of risks from the initial assessment :

1. "Competition between teams"
   Unlike the first phase, where teams had to present their version of the game to compare and compete, this was not directly required in the second assessment

Risk priority and mitigation updates :

1. "A computer from the team breaks/stops working"
   The likelihood of this risk has been reduced from medium to low, as all team members had reliable hardware and no such issues occurred during the development phase.
2. "Time slippage, time is wasted"
   The likelihood of this risk has been reduced from high to medium due to improved task tracking and communication, although it remains a concern due to tight deadlines.
3. "Requirements are not met or with errors"
   The likelihood of this risk has increased from low to medium as challenges in understanding inherited deliverables and unfinished requirements made it more relevant during the development process.
4. "Severity of bugs in the code"
   The severity of this risk has increased from medium to high, as bugs identified during testing and user evaluation had a greater impact on user experience, highlighting the need for more thorough debugging efforts.The mitigation was improved from simply checking the code to increasing testing coverage and conducting thorough quality reviews during user evaluation. Unlike the assessment 1, this assessment involved user evaluation and software testing, which helped better identify and address bugs.
5. "Having engine limitations. For example: a computer cannot run a program because of its system."
   The likelihood of this has been reduced from high to low, as most of the game engines' features proved sufficient for our implementation, with no significant limitations encountered.

Inclusion of risks to the updated assessment :

1. "Delay in receiving editable deliverables from the selected team reducing productivity"

This risk was added due to delay in obtaining necessary files from the selected team. The development process of the game relies on timely access to editable deliverables, and delays in receiving them caused unfinished tasks and affected overall timelines.

2. “Unit testing files incompatible with the code”
   The risk was added because incompatibility between the unit testing files and the game code could lead to significant testing challenges. It could prevent effective testing and lead to missed bugs or issues.
3. “Lack of understanding of code from the selected team by team members causing misinterpretation of code and delayed progress”
   This risk was included as it was discovered that the team faced challenges in understanding code provided by the selected team, leading to misinterpretation and confusion. This could lead to misconfigured game features.
4. “Illogical events causing user confusion”
   This risk was added due to identified events within the game that did not logically follow the product brief or were not clear to the user.