Change Log

Write down what changes we've done so that we can keep track of what has changed throughout the second part of the project.

Change Types
Addition
Deletion
Update

Example:

Change Request Form	
Change Name:	Requirements - Adding New Requirements
Change Request Date:	23/11/24
Requested Change:	Adding the new user and system requirements for the second part of the project.
Priority Details (importance/how long it will take to implement/etc.):	This should be done quickly so it can influence the new architecture and implementation, it should take a few days to implement so should be completed quickly.
Accepted (Yes/No):	Yes

Change Request Form

Change Name:	Risk - Added risk matrix and description
Change Request Date:	25/11/24
Requested Change:	Adding the risk matrix description and metric to the risk register in Risk1
Priority Details (importance/how long it will take to implement/etc.):	This should be done quickly to help plan how to prioritise and mitigate new risks as they are discovered.
Accepted (Yes/No):	Yes

Change Request Form

Change Name:	Risk - Updated risk owners
Change Request Date:	25/11/24
Requested Change:	Updated risk owners to match team members
Priority Details (importance/how long it will take to implement/etc.):	This should be done quickly so that risks have designated owners and can be mitigated effectively
Accepted (Yes/No):	Yes

Change Request Form

Change Name:	Risk - Updated risk types
Change Request Date:	02/12/24
Requested Change:	Updated risk types to be more descriptive
Priority Details (importance/how long it will take to implement/etc.):	This will not take long to implement but should be enacted quickly to help the team better understand and mitigate any potential problems which could arise during the development process.
Accepted (Yes/No):	Yes

Change Request Form

Change Name:	Risk - Refined existing risk descriptions and mitigation strategies
Change Request Date:	02/12/24
Requested Change:	Updated risk impact and mitigation details to better suit team
Priority Details (importance/how long it will take to implement/etc.):	This should be done quickly so that the risk mitigation strategies are optimised for the team to mitigate problems as effectively as possible.
Accepted (Yes/No):	Yes

Change Request Form

Change Name:	Risk - Added new risks and diagram to describe risk management process
Change Request Date:	02/12/24
Requested Change:	Add risk management process diagram and more risks
Priority Details (importance/how long it will take to implement/etc.):	This should be done quickly so that the team is better prepared to mitigate the effects of any potential problems that may arise.
Accepted (Yes/No):	Yes

Change Request Form

Change Name:	Risk - updated risk report formatting for clarity
Change Request Date:	06/12/24
Requested Change:	Change structure of risk register and formatting of parts of report
Priority Details (importance/how long it will take to implement/etc.):	This can be done quickly to improve the clarity of the document.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Testing - Addition of Unit Tests
Change Request Date:	25/11/24
Requested Change:	In order to comply with testing obligations, adding a number of unit tests.
Priority Details (importance/how long it will take to implement/etc.):	Unit tests will make the development of further features better informed and will allow problems to be identified at an earlier stage.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Continuous Integration - Addition of Unit Testing to CI Workflow
Change Request Date:	25/11/24
Requested Change:	Any unit tests for already completed functionality to be added to the GitHub actions
Priority Details (importance/how long it will take to implement/etc.):	This will help prevent problematic code sections from entering the main repository and thus reduce collective debugging effort
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Implementation - Fix Timer
Change Request Date:	25/11/24
Requested Change:	We notice in testing that making the game full screen caused the timer to tick at double speed. We need to fix this as it doesn't make sense
Priority Details (importance/how long it will take to implement/etc.):	Does not impede further development but is an obvious defect from user perspective
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Implementation - Add Leaderboard
Change Request Date:	25/11/24
Requested Change:	Add a completely new feature to the game to display current best scores.
Priority Details (importance/how long it will take to implement/etc.):	This is necessary to comply with requirements however further development does not depend upon this action being completed immediately
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Implementation - Add Achievements
Change Request Date:	25/11/24

Requested Change:	Add a new functionality to the game: Users should be able to perform certain actions in the game and be informed of their success via a popup.
	
	The proposed achievements are as follows
	- Mike Freeman Award
	  - Maintain 80% satisfaction for 3 minutes
	  - Adds 5% satisfaction at the end
	- Bare Minimum
	  - Only 1 of each building
	- Priorities/IsThisAUniversity
	  - Twice the number of study as recreation (and reverse) after 20 buildings
	- Unlucky
	  - Get 3 negative events in one game
	- Lucky
	  - Get 3 positive events in one game
	- Indecisive
	  - Remove 20 buildings
	- Clean Slate
	  - Remove every building after 10 buildings
	- Saviour
	  - Save a burning building
	  - Adds 5% satisfaction at the end
	- Busy Campus

	- Build 40 buildings
	- Master of Change
	      - Under 50% satisfaction for first 3 minutes but then won the game
	- Bankruptcy
	      - Balance goes below 0

Priority Details (importance/how long it will take to implement/etc.):	Necessary to meet updated requirements
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Continuous Integration - Addition of build output
Change Request Date:	25/11/24
Requested Change:	Produce an output .jar file for every commit to facilitate easy testing for everyone - even if they're not involved in the build process
Priority Details (importance/how long it will take to implement/etc.):	Will be useful further down the line when it comes to seeking wider test feedback. It is possible to produce manual builds though so is not a major priority
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Implementation - Change satisfaction
Change Request Date:	25/11/24
Requested Change:	We agree that the current implementation of satisfaction as a very large number is difficult for users to quantify and understand how their actions are affecting satisfaction. The proposal is to pivot satisfaction to a percentage
Priority Details (importance/how long it will take to implement/etc.):	There is a strength of feeling among involved parties that the satisfaction is not easy to understand and thus should be changed

Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Implementation - Change Building Collision
Change Request Date:	25/11/24
Requested Change:	Update the collision box so that it is possible to place buildings closer together. This is to better represent the fact that buildings have been drawn in perspective and thus should be allowed to have their assets overlap slightly.
Priority Details (importance/how long it will take to implement/etc.):	Purely a cosmetic change but presumably quite simple
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Architecture / Implementation - Change architecture for placed buildings
Change Request Date:	26/11/24
Requested Change:	In the current code, a list of buildings is stored within GameRender alongside various methods to manipulate the list. This is difficult to interpret at a glance and not in keeping with a modular architecture. It also presents a challenge for testing via unit tests.
Priority Details (importance/how long it will take to implement/etc.):	Will need to be updated before the implementation of unit tests (which are in-tern a high priority as above) so should be done soon
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Architecture / Implementation - Change money architecture
Change Request Date:	04/12/24

Requested Change:	The current system of handling money is reliant on a Global Integer variable. This does not fit well into an object-oriented paradigm and introduces additional code duplication for functionality like preventing being overdrawn. For the same reason, it reduces testability. The proposal is to have a global MoneyHandler class that will contain the following methods ● withdraw (**float** amount) ● deposit (**float** amount) ● earn (**BuildingList** buildings) ● earnSemesterly (**BuildingList** buildings) ● getBalance ()
Priority Details (importance/how long it will take to implement/etc.):	This should be done to effectively accommodate development of other functionality such as the Satisfaction
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Implementation / Architecture - Change satisfaction architecture
Change Request Date:	04/12/24
Requested Change:	Similarly to money, the current implementation of satisfaction involves a single, global integer variable. I propose a move to a superior architecture such that SatisfactionHander is stored globally. This will improve flexibility and testability Proposed SatisfactionHander methods: ● getSatisfaction () ● recalculateSatisfaction () ● addBonus (**float** amount) ● applyPenalty (**float** penalty) ● resetSatisfaction ()
Priority Details (importance/how long it will take to implement/etc.):	As was approved above, a change to the outward view of satisfaction is a priority so this change in architecture should be implemented alongside that.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Implementation / Architecture - Add AchievementHander to

	globals
Change Request Date:	04/12/24
Requested Change:	For the sake of consistency with Satisfaction and Money, Achievements could be added to the GlobalsFunctionality
Priority Details (importance/how long it will take to implement/etc.):	Will be a very quick fix but equally of little importance - just for a bit of consistency
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Implementation - Allow Debt
Change Request Date:	06/12/24
Requested Change:	Make it possible for users to go into debt while building their campus in order to increase realism
Priority Details (importance/how long it will take to implement/etc.):	Very tiny tweak required
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Implementation - Add tutorial
Change Request Date:	06/12/24
Requested Change:	It seems like there are a number of different mechanics in the game now which aren't obvious. We should have a simple screen of some description to explain these to users
Priority Details (importance/how long it will take to implement/etc.):	Not part of requirements but will likely increase overall user satisfaction
Accepted (Yes/No):	Yes

Change Request Form

Change Name:	Continuous Integration - Add automatic merge main into active branches
Change Request Date:	06/12/24
Requested Change:	Add a new action to the GitHub Actions such that any changes on the main branch will be merge into other active branches.
Priority Details (importance/how long it will take to implement/etc.):	May improve continuous integration pipeline and improve productivity. On the other hand, current development methodology doesn't involve long-lived branches
Accepted (Yes/No):	Yes

Change Request Form

Change Name:	Implementation - Change building definition format
Change Request Date:	06/12/24
Requested Change:	The way that buildings are currently defined involves a JSON file consisting of several dictionaries. This is then parsed into a number of dictionaries containing building properties. The JSON file is not a valid JSON format and the method of accessing multiple data items from multiple dictionaries is cumbersome.

A solution is to use a similar architecture as is currently used for achievements and events where buildings are defined directly as Java Classes. |
| Priority Details (importance/how long it will take to implement/etc.): | The existing implementation does work will all parts of the code and adding functionality is possible. The refactoring required to change this would be significant and will likely take a reasonable time. It would be so significant that it should ideally be done at a time where there are no divergent branches. For this reason, this is a low priority and may be enacted if there is time. |
| Accepted (Yes/No): | No |

Change Request Form	
Change Name:	Requirements - Update Leaderboard Requirements
Change Request Date:	23/11/24
Requested Change:	Update the existing leaderboard requirements in the original document to fit more in line with the project brief
Priority Details (importance/how long it will take to implement/etc.):	This will be a case of rewording, so will only take a short amount of time, but is important since it is something specified in the brief.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Requirements - Additional Requirements; Already Added Features
Change Request Date:	23/11/24
Requested Change:	Adding in requirements for aspects of the project that have been added but don't have any system or user requirements associated with them (eg. satisfaction)
Priority Details (importance/how long it will take to implement/etc.):	This may take a while to fully sort out but is important to do right away because it might be important when looking at architecture or further implementation for the second part of the project
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Implementation - Add events
Change Request Date:	29/11/24
Requested Change:	Add events into the game
Priority Details (importance/how long it will take to implement/etc.):	This is necessary to comply with requirements however further development does not depend upon this action being completed immediately

Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Requirements: Additional Functional Requirements - Events
Change Request Date:	29/11/24
Requested Change:	Adding in all functional requirements relating to the planned events for the game.
Priority Details (importance/how long it will take to implement/etc.):	This is a necessary part of the requirements as they are part of the project brief. It should take less than a day to write them up as requirements.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Requirements: Additional Functional Requirements - Achievements
Change Request Date:	29/11/24
Requested Change:	Adding in all functional requirements relating to the planned achievements for the game
Priority Details (importance/how long it will take to implement/etc.):	This is a necessary part of the requirements as they are part of the project brief. It should take less than a day to write them up as requirements.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Methods & Planning: New Gantt Charts
Change Request Date:	25/11/24
Requested Change:	Creating new Gantt Charts for the next part of the project.
Priority Details (importance/how long it will take to	This is a high priority so that the project has a clear step-by-step plan and so that all members of the group know what they should be doing week by week.

implement/etc.):	
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Methods & Planning: Team Organisation Method
Change Request Date:	25/11/24
Requested Change:	Changing the way that roles are organised within the project, allocating roles based on familiarity and preference rather than strictly marks-based allocations.
Priority Details (importance/how long it will take to implement/etc.):	This will not take long and is not particularly a high priority, since we don't necessarily need to change the way that Group 2 organised their project. It depends on group preference more than anything.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Methods & Planning: Role Allocation
Change Request Date:	25/11/24
Requested Change:	Changing the names of the old team allocations to fit our team, as well as assigning roles to the new deliverables.
Priority Details (importance/how long it will take to implement/etc.):	This is a high priority task and will only take the span of one conversation to complete, so should be completed as soon as we can.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Methods & Planning - Update User Evaluation Interview Dates
Change Request Date:	9/12/24
Requested Change:	The user evaluations are happening in Friday's practical so the date on the Gantt Chart task will be moved up to reflect this.

Priority Details (importance/how long it will take to implement/etc.):	Not a high priority, but will take about a minute to implement and is not that consequential apart from completeness so we should implement it whenever possible.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Architecture - update previous team's design evolution and system description narratives
Change Request Date:	15/12/24
Requested Change:	The software design process sections in the original architecture document will be reviewed and updated to be more readable and clear.
Priority Details (importance/how long it will take to implement/etc.):	Not high priority, but it will take some time to go over the document.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Architecture - add justification for use of PlantUML
Change Request Date:	15/12/24
Requested Change:	To add a section in the architecture document justifying the use of PlantUML as a tool for modelling class diagrams
Priority Details (importance/how long it will take to implement/etc.):	This is not high priority but will be quick to complete and is a necessary change as it is specifically requested in the assessment paper.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Architecture - link all existing design explanations directly to requirements
Change Request Date:	15/12/24
Requested Change:	To add links to the requirements throughout the architecture document in order to justify each design decision and ensure that all requirements are met.
Priority Details (importance/how long it will take to implement/etc.):	This is high priority as it is important to check that all design decisions are justified and can be closely linked to the requirements document.
Accepted (Yes/No):	Yes

Change Request Form	
Change Name:	Architecture - add sections to describe new functionality
Change Request Date:	16/12/24
Requested Change:	To add new sections that describe the structure and functionality of the new requirements listed in the updated product brief
Priority Details (importance/how long it will take to implement/etc.):	This is a medium priority task and should take some time to complete as a large proportion of the system must be explained in detail. However, it would be useful to complete soon in order to ensure that all requirements have been met and that the application's architecture is as streamlined as possible.
Accepted (Yes/No):	Yes