



For the purpose of swapping in this assessment, we have acquired all documentation, code and art assets from the other team. We then read through all the documentation and code to get a better understanding of their game and its structure. Additionally, we have created a new GitHub repository to manage version control and any changes to implementation. This ensures that none of our code or images mix; since we should not be using any of our initial game.

When identifying the initial set of “change requests” we compared the requirements document with the current state of the game’s architecture and the features already implemented. This helped us find a set of initial “change requests” which we would begin to act on. To identify possible corrective and perfective changes we play tested the game and looked through the documentation for the code alongside the code itself. Additional changes were suggested when we felt the requirement no longer represented the game we were trying to make.

We will break up the changes being made to the project into perfective, corrective, additive and subtractive changes and assign priorities to each request based on how important the issue being raised is, how easy the change would be to implement, technical and non-technical costs, risks and the project deadline. The priority would be assessed by all members of the team since we are a relatively small group and everybody would be affected by them. Any changes that would mean a change to requirements, architecture, testing or risks would need to be accepted by the group as these changes were more significant.

Most of the documentation inherited from the previous group will remain similar in terms of content. We will attempt to maintain or improve traceability by keeping the style consistent throughout our documentation and similar to the previous groups documentation so we can clearly see how the project has changed and evolved. Changes to documentation will be directly influenced by the proposed changes to the software.

Change Process

1. A change is requested by a team member.
2. The change request is assessed using the following attributes by all members of the team: requirements, difficulty of implementation, technical and non-technical cost, risks and estimated time required to make the change by.
3. Following assessment, the request is accepted and assigned a priority or rejected. If the change means the documentation is no longer relevant then respective document will also be changed to reflect this. The priority system works similarly to the Agile Requirements Change [1]; the highest priority will be implemented first, and whenever a new change is accepted it’s position in the queue is given by its priority. If two tasks have the same priority the request which was accepted first is allocated and implemented first.

In a few cases a higher priority can be ignored if it is dependant on a change which is currently being made. The list is also regularly reviewed to make sure the order of the list reflects the true priority of the task in relation to those around it as priorities can change.

Change justifications

Change Code	Change	Justification	Priority
Perfective Changes			
CP1	Spawning Algorithm (performance and locations)	The spawning algorithm had performance issues which needed to be addressed. We also found the algorithm allowed enemies to be spawned in locations which could not be accessed. The changes required rewriting most of the existing code and so we grouped both changes together. We felt this was a core mechanic to the game and so assigned it a high priority and accepted the proposals.	4
CP2	Higher resolution art assets	It was argued that some of the art assets inherited in the project were inconsistent style and of lower quality. The proposed change was aimed to make the art a unique selling point of the game and to give the art team more freedom, allowing them to use their own style. Additionally the art team found it difficult to build upon the existing set of art and sprites while keeping a consistent style. Arguably this change would help the project attain requirement I1 more convincingly and for this reason was accepted. Since this was a big change and required a lot of work for the art team and some redesigning of the collision code a high priority was given.	5
CP3	Changes to flying movement and obstacle interaction	Changes were proposed to allow flying over obstacles and enemies. There was no change to the architecture however requirements would need to be changed to reflect this change (G7 and C5). The changes were made to eliminate the frustrating elements of the game when the player was trapped and surrounded by enemies effectively mitigating the purpose of the health system (G5). Since the implementation was minimal the change was accepted and given a medium priority.	3
CP4	Improve objective and score readability	The score and objective text was hard to read so this change proposed to change the font to one with a drop shadow. This was accepted and given a medium priority because it would take such little time.	3
CP5	Addition of invincibility frames	The proposed change relates to requirement G5 in the attempt to negate cheap deaths, the health system was not enough. The addition of invincibility frames when damaged would give the player chance to react to being attacked. This change was accepted and given a medium priority because it would not take much time to implement.	3
CP6	Allowing non-circular sprites to be used as projectiles	Required for the implementation of projectiles such as lasers and allowed the game to keep a consistent style as per requirement I1. This was a fairly trivial change so was accepted but given a low priority as this was a very minor change.	2
CP7	Force sprite to face mouse cursor	When play testing the game it was pointed out that the game would feel more natural if the player faced the cursor and therefore faced the direction the player was firing. An explanation of the change can be found in the implementation report. This change was given a medium priority since it was not necessary to complete requirements or variation points	3
Corrective Changes			
CC1	Collision Detection Performance	Some performance issues when playing testing the game were caused by collisions being checked multiple times for just one collision. This was a fairly trivial bug fix to stop the detection method being called multiple times and so was accepted with medium priority	4

CC2	Collision Detection Accuracy	This change was proposed after finding different resolution sprites were colliding with objects that weren't there. This required adding offsets depending on the size of the sprite. This was a trivial change perhaps more important than the performance since all our new sprites would be a different resolutions to the originals they would be replacing and so was given a high priority. G7 requires this change be complete so obstacles are blocking when they are supposed to be. This change also makes the software much more extensible allowing different types of sprites to be added with relative ease.	5
CC3	Camera behaviour at map borders	This was a trivial fix to stop the camera from moving to a position where it can see past the edge of the map, just drawing the background colour. This change was accepted as our team have had experience with fixing this problem recently and could fix it quickly. This change was given a medium priority since it wasn't a game breaking bug but harmed the quality of graphics in the game.	3
CC4	Fix font for FloatyNumbers	The floating numbers which emitted from the player or enemy when damaged or score was gained were not being displayed correctly and so a new font was needed. This was an extremely short fix and so was accepted and given a medium priority.	4
CC5	Fixing AI pathfinding algorithm	The algorithm being used was a variant of A* search however was not implemented correctly and so enemies would appear to stop at random places around the map. Without the fixes we would no longer be attaining requirement G8 as enemies were no longer impeding progress. The proposed fix would be to reimplement the pathfinding algorithm which wouldn't take much time and so was accepted and given high priority.	5
Additive Changes			
CA1	Addition of FloatyNumbers	This was a change proposed to improve visual feedback. It would require the addition of two new classes a FloatyNumbers class and a FloatyNumbersManager. A discussion of these classes function can be found in the implementation report and the addition to the concrete architecture can be seen in the architecture report. The workload was low for this change and added to the cartoony aesthetics as mentioned in requirement I1. This change was accepted with a low priority as the change was not necessary to meet any requirements.	2
CA2	Addition of swimming sprites and animations	Swimming was not implemented when the project was inherited, but is a criteria which needs to be for the scenario we have been given and so this change was accepted with a high priority.	5
CA3	Addition of a new enemy type	To meet the variation point requiring 8 different obstacles, we proposed a new enemy to get closer to this goal. The other type of obstacles we could introduce are static type obstacles such as buildings and barricades. Since enemies are the most costly to implement since they require some form of A.I. and a lot more art, we thought creating more static obstacles than enemies was the best choice. This change request was accepted with a high priority.	4

CA4	Addition of melee combat	<p>This change was to address the issue of having multiple different types of weapons as a requirement for the game. When the project was inherited the requirement G10 stated multiple weapons would be able to be picked up and used which was not feasible to implement in the time we were allotted. The problem was a lot of art assets would be needed and more than we could make within the deadline set, this meant the proposal would require a change to the requirements. To keep some variety of weapons we proposed two weapons of different types; ranged and melee. Since ranged was already implemented we just needed melee to be implemented. This helped reduce the workload on both the art and programming team and was a compromise to allow the team to focus on reaching the variation points set for this deliverable. This was given a high priority since the workload was still high for the art team.</p>	5
CA5	8 Directional sprites	<p>This change was proposed along with the high resolution art assets to make the game look better. The reasoning was due to aesthetics and the change would mean roughly double the amount of art assets. This change was relatively low priority since it was not a requirement and meant a rather large amount of work for the art team but was still accepted due to the change which forces the player direction to force the mouse instead of the direction of movement. Detail about the implementation can be found in the implementation report along with the change player directional changes.</p>	3
CA6	Powerup Stacking	<p>This change requires a powerup manager class so a change to the architecture is needed however the implementation was fairly simple and the details of the implementation can be found in the implementation report. Overall this was a relatively simple change, the change to the architecture was minimal. Since the change was subtle and didn't address any bugs this proposal had a low priority but was still accepted.</p>	2
CA7	Addition of more objectives	<p>To achieve the variation point of having 2 distinctly different types of objectives also seen in requirement G2. The proposed change adds in a kill X amount of enemies objective type to contrast the pickup flag objective type currently implemented. This was high priority change.</p>	5
CA8	Addition of more levels	<p>Originally to meet the variation point requiring 8 distinct locations from around the campus, the requirement G1 stated we must have 8 distinctly different rounds. We felt making the maps slightly bigger for each round and including multiple buildings in each map was a better way of achieving this variation point. This change resulted in time being saved overall. This was high priority since it was a variation point and must be done but also saved us time.</p>	5
CA9	Addition of a title screen	<p>This proposal was made as a quick way to add polish to the game. This change however does have a side effect of changing the architecture slightly. An addition of the OpeningCrawlScreen class effects the structural architecture and also the behavioural architecture by changing the way the game starts. The specific changes can be seen in the architectural report and further discussion of the start screen can be seen in the implementation report. This was given a low priority.</p>	1

CA10	Addition of boss enemy	This proposal was an extension of the change adding a new enemy. It's purpose is to add both a new obstacle and a new objective to help with achieving the variation points for 8 different obstacles and 8 different objectives, requirements G2 and G7. This change would require somebody allocated to make new art for the boss sprite and also animations as well as a change to the architecture. The programming team will need to create a new AI class. This is a fairly small change no classes need to be removed and the relationships stay the same between all other classes. As this change was required to achieve requirements G2 and G7 a high priority was assigned.	5
CA11	Addition of health bars	The proposal was another way to quickly polish the game, it added more visual feedback that the player is damaging the enemy and lets the player know how strong the enemy is. This was in main due to the speed of the change, whilst keeping consistent with requirement I1. The priority was set as fairly low however; no the changes were being implemented at the time this request was made.	2
Subtractive			
CS1	Removal of JUnit testing	As none of our team have had experience using JUnit for testing and having inherited the project from the previous group which used JUnit testing we had to make a decision whether we carried this forward. After trying to learn how to use JUnit we were having troubles getting the original tests to work. For this reason we decided to remove the JUnit testing and replace it with our own style. The full reasoning for this decision can be seen in the Testing report. This was a major change and so was high priority and one of the first changes made to the software.	6

URLs for modified documentation:

Architecture - <http://www.teampochard.co.uk/Arch3.pdf>

Requirements - <http://www.teampochard.co.uk/Req3.pdf>

Methods and Planning - <http://www.teampochard.co.uk/Plan3.pdf>

Implementation - <http://www.teampochard.co.uk/impl3.pdf>

Testing - <http://www.teampochard.co.uk/game-testing>

Risk assessment - <http://www.teampochard.co.uk/Risk3.pdf>