Changes to the requirements were made once initially when the project was inherited and then as changes were requested we reviewed the requirements to see how they would be met if the change was made. In the cases where the requested change did not impact which requirements were met then the change management process would continue as stated in the change report document. However if the change meant some requirements were no longer met then as a team and also with stakeholders we would discuss whether the requirements were still relevant to the game we were making or whether the requirements were a realistic, accomplishable set of goals given our current resources.

Any changes from Assessment 2 to 3 will be in red. Any requirements removed have been stroked through and will be in red.

- **G1** was *changed* as we felt 8 distinct rounds was a large workload which was not necessary, the criteria for the end product was 8 distinct locations from the campus. We made sure to instead have multiple distinct locations in each round. E.g. different building such as the CS building and Ron Cooke Hub.
- **G4** was *changed* to add that eliminating enemies will also add to the player score and not just completing objectives. This meant players could do the bare minimum to "pass" the game or try to "perfect" the game instead, to make it more enjoyable.
- **G5** was *changed* since we will not be implementing healing as a pickup but instead as the player being healed every round.
- **G7** was *changed* as we felt flying should be used as an escape mechanic and so the player should be able to fly over obstacles which gives the player more freedom and stops the player getting stuck when surrounded by a large amount of enemies.
- **G10** was *changed* due to a resource restriction animating and drawing multiple different weapons for the sprite was unattainable for the current team size and especially since we have decided to change the player sprites to be in 8 directions effectively doubling the amount sprites for the player. To still have some variety we chose to make one weapon ranged and the other a close quarters melee weapon.
- **G11** power-ups were *changed* to be stackable as it felt natural and allowed more gameplay variety.
- **I3** was changed to clarify roughly the type of projection we were trying to mimic with 2d sprites (orthographic)
- **I6** was *added* as we felt damage numbers were necessary to affirm whether a player or enemy was being damaged or not.
- **C5** this *change* was due to the addition of being ability to fly over obstacles both enemies and static obstacles.

To ensure that the right product was being built, we constantly validated our requirements by checking them against the assessment specification, and the requirements below discuss validation through this or through interviews with the project customer.

Risks[1] associated with these requirements are as follows:

- ID1 - our requirements may be more difficult to implement than we had planned or thought. Part of mitigating this is with our validation - to ensure we are doing what we are asked in the scenario and not going above with superfluous and complicated additions.
- ID12 - requirements can (and will) be changed throughout the project so our requirements must be kept simple and straightforward.
- ID13 - requirements may be ambiguous, incomplete or untestable. Again validation of requirements when they are expressed is important here - all requirements must stand up to being tested at regular intervals to ensure they are implemented. Requirements we thought were ambiguous or incomplete can be looked at by cross-referencing it with the scenario document, and using interviews with the customer.

---

[1] These correspond to the risk assessment table, given in *Risk3.pdf*

ID	Requirement	Justification	Test Criteria
Gameplay Systems			
G1	The game is split into ~~at least 8~~ distinct **'rounds'** (levels) where an objective is obtained at the beginning of each. Rounds progress in a linear fashion one after the other. The player-controlled duck character acts within rounds.	**Scenario:** *"At the start of each round of the game, the player needs to acquire an objective"* **Interview:** *[Either continuous or discrete, would depend on narrative context]* **&** *[Okayed this approach]* ~~It was decided to use 8 rounds since tying objectives rounds allowed for an easy to understand game structure~~. Using a linear progression was chosen since it was felt to allow the game a better sense of direction and progress than random rounds. It also allowed the easy ramping up of difficulty as rounds progress and allows for a more structured than disjointed narrative.	**-Rounds follow each other in a consistent order one after the other.**
G2	The game will include at least eight different **'objectives'** (goals), of which there are at least two distinct types. Objectives can be either be failed or succeeded by the player.	**Scenario:** *"Ducks have objectives..."* **&** *"...must support at least eight different objectives"*. **&** *"support at least two different types of objectives"* **&** *"Failing to achieve objectives"* Objectives provide an easy way to both motivate and direct the play, allowing the rest of the game to be based around them. The success or lack thereof provides the driving force for the player actions.	**-There are 8+ objectives. -There are 2+ types of objective -Each objectives has a fail and success state..**
G3	Objectives can be achieved in an approximately 5 minute period. This is not describing an attempt to rush through a level, but normal gameplay.	**Scenario:** *"...using your game for its own promotional activities, e.g., at Open Days, UCAS Days."* **Interview:** *[Okayed this approach.]* This allows achievements to be made in a short time. This assumes that in these situations there will be a short play time, and that **S4** remains. If this audience is to change then making objectives harder to complete would extend a round. Another factor is that this does not impact other audiences as much as it may seem, since longer play times can be achieved by playing several rounds.	**-Each objective is achievable in 5 minutes.**
G4	There is a **'point'** (score) tracking system. Points are given when an objective is completed and may be affected by other factors, such as time and eliminating enemies. Points are cumulative between rounds, but lost in a fail state.	**Scenario:** *"When an objective is completed, the duck is awarded points."* Points provide a simple incentive for re-playability and progress tracking. It also allows indirect competition between different players.	**-Points are given when an objective is completed.**

G5	There is a **'health'** system, represented to the player in hearts. Players lose health from some obstacles ~~and gain it by picking up resources.~~ and gain health back to full health at the end of each round	Provides a challenge to the player as they progress and encounter danger. Insta-kills can feel cheap and annoying, a health system providing a more interesting combat / damage system. Assumes that the combat aspect of the game is not changed by any new requirements. Could be easily mitigated by removing such a system (since removing a system is easier than adding one).	**-Some obstacles damage health.** ~~**-Pickups increases health.**~~ ~~**-Some resources increase health.**~~
G6	The game will feature eight different **'locations'** from around the University of York.	**Scenario:** *"...must take place at the University of York"* **&** *"...must include at least eight locations..."* Provides the backdrop for the game and the context.	**-There are 8+ locations from the UoY.**
G7	There will be present at least five different types of **'obstacle'** for the player to overcome. Obstacles will be objects in a location that impede player progress. At least one obstacle will be generated at random and at least one is tied to an objective. Obstacles physically collide with the player, preventing navigation past them unless the player is flying.	**Scenario:** *"The game must support at least five different types of obstacles..."* **&** *"obstacles in the game to make it challenging"* **&** *"...at least one randomly allocated obstacle..."* **&** *"...at least one objective-specific obstacle..."* Provides the main mean of challenge in the game, directly countering player objective progress. (See **S2**, **S4**, and **S5**) Collision means the player cannot skip obstacles or move past them (which would negate their function).	**-There are 5+ type of obstacle.** **-There is 1+ random obstacle.** **-There is 1+ objective specific obstacle.** **-Obstacles collide with the player.**
G8	A portion of the obstacles will be **'enemies'**, which are aggressive, and actively impede progress. Enemies are capable of decreasing player health and are defeatable. They occasionally drop resources.	**Scenario:** *"e.g., a Guard Swan …"* **Interview:** *[Okayed enemies]* Enemies provide a direct threat to the player. Just passive obstacles would result in a slow pace of game. Enemy use assumes the fast combat remains relevant. Since enemies would be heavy developmentally (movement, combat, health, graphics) a specific feature approval was sought and received.	**-Some enemies decrease health.** **-Enemies are defeatable.** **-Enemies drop resources.**
G9	The player should be able to obtain **'resources'**. Resources aid the objective progress of players, through upgrades or directly to achieve the objective. Some are maintained across rounds.	**Scenario:** *"The ability for the duck to acquire … resources"* **&** *"The objectives should be achievable … by acquiring resources…"* As well as further expanding gameplay it provides a longer term planning aspect, since players can decide on resource use cross-round.	**-(Relevant) resources are maintained between rounds.**

G10	~~The player will be able to acquire ‘**weapons**’ in the game, which allow the player to damage enemies. Weapons beyond the default should be dropped by enemies and done so in a set order.~~ The player will spawn with a ranged weapon and a melee weapon.	Gives the player a method to deal with enemies, providing the main player interaction besides movement. due to **I1** including weapons does not conflict with the requirements of **S4**. ~~Linear progression of weapons allows a smooth ramp up of player ability as the game progresses, and provides an incentive to not avoid enemies.~~	**-The player has a ranged weapon and a melee weapon.** **-Weapons damage enemies.**
G11	The player should be able to obtain at least three distinct **‘powers’** (abilities). Powers provide the player character with new abilities, which aid their progress. Powers are obtainable from pick-ups which are dropped by enemies when they are defeated. Power ups are stackable.	**Scenario:** *“The ability for the duck to acquire special powers...”* **&** *“The objectives should be achievable … by acquiring special powers…”* **&** *“...at least three different Duck Special Powers”* **&** *“... can be acquired as a round of the game progresses.”* Provides a way to change up gameplay and keep it interesting.	**-There are 3+ powers.** **-Pickups exist for powers.** **-Powerups stack if differently typed**
G12	A fail state is reached when an objective has been failed or if the player runs out of health.	**Scenario:** *“...a game ends”* Due to the relatively short round playtimes and the want for a challenge, having a complete reset on objective failure is not overly hard, but does provide an incentive for re-playability. This has the risk of being potentially annoying for a more casual audience, but making the difficulty ramp up slowly will allow early rounds to be easy enough.	**-Game fails when an objective fails.** **-Game fails when player dies.**
G13	The game has a win state obtainable when the final round has been completed.	**Scenario:** *“...a game ends”* Since the round ordering is linear rather than random it is natural to have the win state be at the end of the round.	**-Game wins when final round is finished.**
Interface / Visuals			
I1	The game will use a cartoony / arcady design style.	This style is simple to work with and create images for, as well as allowing combat and enemy mechanics without conflicting with **S4**.	**-Consistent graphical style.**
I2	The obtained points will be displayed at all times.	**Scenario:** *"The GUI must always show the points that the duck has acquired"*	**-Points always on-screen.**
I3	The game will use a flat looking background, but characters and obstacles will appear from a side perspective. The 2d style will try to mimic orthographic projection.	This was decided to be the preferable choice since it allows graphics to be produced in a more systematic way, as well as allowing map design to be done in a more discrete manner.	**-Consistent graphics implementation.**

I4	~~There will be a minimap which will show the location of the duck character and its location in the round. It will use a fog system (unveiled during exploration)~~	~~**Scenario:** *"The GUI must always show … the location of the duck"* **Interview:** *[Could show location within the wider world or just an idea of where you are in the round/land.]*~~ ~~It was decided to use a minimap since this fit with the perspective choice (by convention), and provided an easy way to display the map as a whole without providing too much information.~~	~~**-Duck position displayed constantly.**~~ ~~**-Map updates on exploration.**~~
I5	The current objective will be displayed at all times on screen.	This ensures that the player is aware of what the need to do and how to progress, keeping them informed.	**-Objective always on-screen.**
I6	When a player or enemy is damaged or score is gained this will be displayed as a number on screen.	This ensures the player has visual confirmation that they are succeeding or failing at damaging enemies and stops confusion	**-On screen damage and score**
Control / Movement			
C1	The game will allow the player to move the duck through the use of the keyboard.	**Scenario:** *"The ability for the duck to move throughout the...University."* The keyboard was selected since it is a standard input device most conventionally used for movement.	**-WASD/Arrows move the duck up/right/ down/left.**
C2	There are three distinct movement modes; waddling (slow), swimming (moderate), and flying (fast).	**Scenario:** *"Ducks innately have the ability to waddle (fairly slowly), swim (fairly briskly), and fly (quickly)."*	**-Waddle,swim, and fly modes exist.**
C3	Waddling is the ground movement type allowing movement across solid ground.	Assumed to be standard movement type from the name and the fact default movement should be slowest unless introducing a slowdown mechanic/	**-Waddle is slow.** **-Waddle is default movement.**
C4	Swimming is the water movement type, which automatically happens when moving onto water in a map.	Since swimming requires water and would eliminate the ability to walk, this movement type is merely used as an alternate default movement. This requirement means some maps must be designed for water, but this should not take too much development time.	**-Swim is medium speed.** **-Swim is contextual.**
C5	Flying will be a player activated movement mode, where the player can move quickly for a short period of time. If currently flying over an obstacle the player will continue to fly until they can land.	If constantly available there would be no reason to use any other movement. Allowing it to be activated instead of just contextual allows movement and gameplay to be more interesting and interactive for the player.	**-Flying is faster.** **-Flying can be activated.** **-Flying has a cooldown period.**