When we first started to think about what kind of game we wanted, we found that we had lots of ideas, and we needed to simplify our choices. The games that we thought would be viable turned out to have many features, a lot of which were not actually necessary. This caused us to turn our focus onto games that were simpler to produce but could be evolved to make a more entertaining product.

After sharing ideas for simpler games as a group, we settled on a game modelled after the popular flash game “copter”, which we believed would be much more viable to produce, as well as being a sort of “blank slate”, upon which that we could incorporate many new features and twists that would make the game much more enjoyable and interesting to play.

Once we settled on a game, we needed to think about the features we wanted to incorporate and negotiate a balance between what was achievable in the time frame we had, and what the use cases would desire.

We already had many ideas about what we could add into the game and what we would like to do, but we were unsure which ones to go ahead with. By pausing and thinking about the use cases in relation to the ideas we already had for the game, we were able to obtain a much clearer understanding of what features that we had thought of would be most suitable for the game. Not only did this help us organise our ideas and have a much more focused approach to building our game, but it also helped us to prioritise our workload, as we could see from understanding the use cases which features would be most important to the users. For example, when writing the use case for a player navigating the spaceship around the map to avoid obstacles, we had questions about how to implement certain mechanics of the game. “How are we going to cause the spaceship to fall?”, “How are we going to procedurally generate the map?”. Asking these questions and looking at the game from the perspective of the player brought the most pressing matters into focus and allowed us to tailor the gameplay experience specifically for the player.

We also looked at different types of players. For example, we originally had a tutorial level which would play every time the game was run. While this was great for new players as it would teach them the controls in a consequence-free setting, for experienced players it would be very tiring and possibly deter them from starting a new game if they had to go through the tutorial each time. Based on that, we implemented an option to skip the tutorial so that experienced players could get right to the gameplay.

Overall, looking at the game through the lens of different types of users allowed us to uncover new features that were required, as well as ascertain which of our ideas would provide the most improvement to the experience of a certain user without hindering the experiences of other users, and therefore, has been crucial in the development of our game.

300 word reflection on what we learned writing user stories and use cases

Insights from writing the user stories:

* Became aware that the ideas we came up with when first discussing our game, actually meant a lot of features, not all of them necessary, we are giving ourselves too much work
* More awareness around how much effort is required for developing our game
* Clearer understanding and insight into how our game will work
* Helped us to prioritise our workload

Insights from writing the use cases:

Ideas about how to implement each feature, how much time and effort it could take, and which bits of implementing it could be challenging. For example, when writing the Use Case for a player navigating the spaceship around the map to avoid obstacles, we had questions about how to implement certain mechanics of the game. “How are we going to cause the spaceship to fall?”, “How are we going to procedurally generate the map?”.