

Product Evaluation

Throughout the development of our product, we made sure to evaluate the decisions which we were making. Our key findings coincided with our three submission deadlines:

Minimum Viable Product

When developing our MVP, we had various people who could critique our design and development choices: our clients, our mentor and each other. This evaluation took place as part of our regular in-person meetings which took place either in MVB or in meeting rooms in the Queen's building. As we were all present during these meetings, we decided that the best way to document the observations made was by directly creating tasks on Jira which corresponded to the new ideas that could later be assigned when they were used.

Here our evaluation shows that we needed to make our drone move faster and were asked to add a feature to turn the camera from the left slider, which differed from our initial choice to add a new slider for rotations. It was also requested that we add a minimap to the game. Both of these features were added successfully before the next release. There were other aspects of our app which were evaluated here, such as the initial physics engine for movements and the mock buildings which we used for size comparisons, but no issues were made so we didn't need to make any refinements here.

Beta Release

At this stage of development, we couldn't test with the intended end-users, as our clients' workshops were infrequent and did not line up with our schedules. However, our app will eventually be released onto the Google Play Store. As a result, we needed to make sure our app fits the expectations of a wider audience than simply the attendees of our clients' workshops. So we decided to branch out and perform observational evaluation with fellow students. This evaluation was performed face-to-face as well, so we decided to continue using Jira to track any tasks that arose from this broader feedback.

Here we were able to test some of the advanced gameplay mechanics that we had implemented. Evaluation here found that players often had trouble hitting the loops correctly, so decided to alter how their hitboxes were made. We also had feedback on how the new collision system lined up with the models we created.

Final Release

We proceeded with this stage of evaluation again through observations, but this time we were able to discuss with some intended end-users. We recorded their comments and split it into two main categories: gameplay and visuals (see **Sample Material**).

Our evaluation showed us here that users were not too happy with the UI. They felt it was too basic, especially with the menus being the default style found in many Android apps. They also had some trouble with the customisation, as it wasn't clear what effects the sensors would have on the gameplay. We feel that it is feasible to solve these issues. Given more time, we could design more interesting menus and it would be simple to add an explanation to the sensors.

Evaluation here also found that users were frustrated with our gameplay. They found that the controls seemed too sensitive, especially when turning. They also were irritated by our collisions halting the drone completely when colliding with a building. The former of these issues can be fixed by simply lowering the values associated with our slider, but the latter issue is much less feasible. This would require a major redesign to our physics engine which is not possible in the time that we have left.

Sample Material

Gameplay

Critical comment(s)	Possible solution(s)
"Turning is too sensitive." "It is hard to hover in place."	The slider to control both rotations and height could be made larger, so that more precision can be gained when it is in use. This will allow for slower turning speeds to be used, and should make it easier to find the correct position to hover the drone.
"The loops turn red too fast." "The loops are too far apart, they don't show up on the minimap until you are close." "The timer ran out too fast."	These are level design issues that we can solve by simply changing the level files.
"I keep getting stuck on the building, I want to slide across the walls instead."	This is a much harder issue to solve as it would require changing a large part of how our physics engine calculates collisions.
"The camera clips into objects"	This would require us implementing camera culling, which we did not manage to achieve when we previously attempted it.

UI

Critical comment(s)	Possible solution(s)
"There is a lack of interesting UI" "It's just basic Android" "It is hard to know when sensors are equipped"	Custom UI could be designed to make the menus more interesting.
"I'm not told how the sensors affect the gameplay"	We simply need to add a sentence to each sensor to explain what it does to the game.
"The controls are unconventional, they're not like in other games"	This was an intentional design choice as it was meant to mimic the controls of the drones used in our clients' workshops, so we do not feel the need to change this.