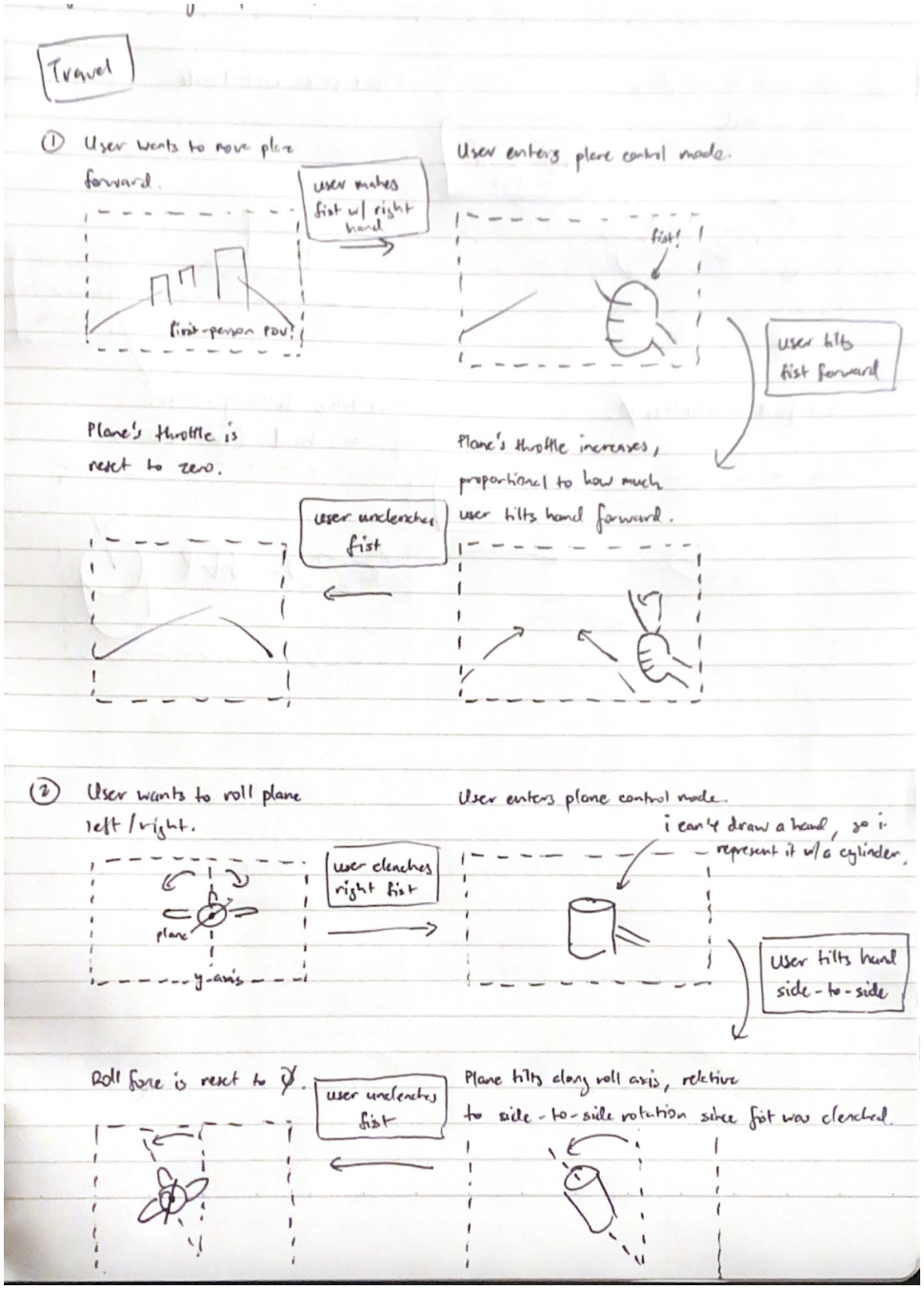
David Cao, Thomas Wang
Project 2 Design Document

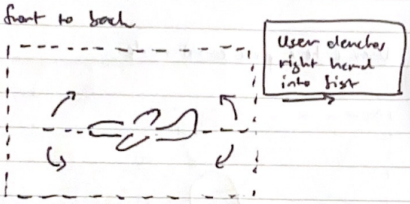
Design Exploration - Interaction Storyboards

Add images of your design explorations below.

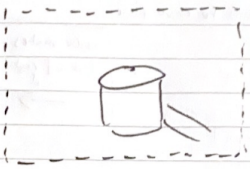
Each design exploration needs to be in the form of a storyboard that depicts the full interaction
(you can think of it as a before state, an action or set of actions, and an after state).
An example of a simple interaction storyboard is here. You can sketch on paper or use tools like Milanote.



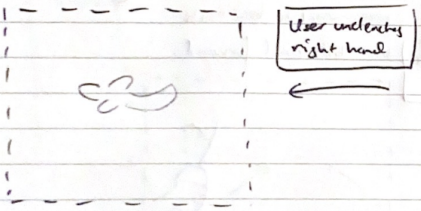
③ User wants to tilt plane front to back



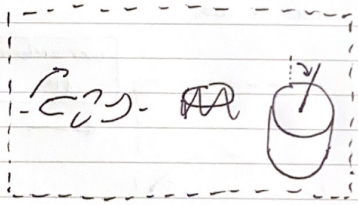
Plane enters control mode.



Tilt force is reset to ϕ.

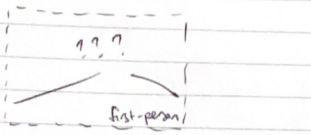


Plane tilts, following rotation of user's hand (front/back)

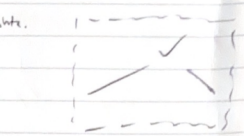


Wayfinding

① User is lost & wants to know the direction of the next checkpoint.

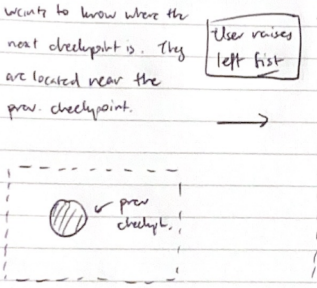


3D minimap appears at the bottom of the user's fov, showing the relative directions of the next 2 checkpoints.

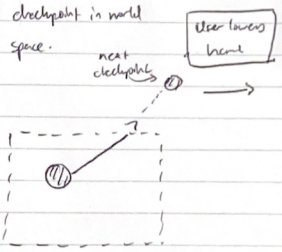


The minimap disappears.

② User is still lost & wants to know where the next checkpoint is. They are located near the prev. checkpoint.

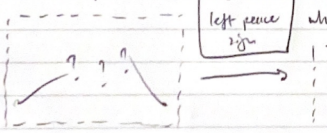


A line is drawn from the prev to next checkpoint in world space.



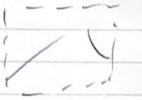
The line disappears.

③ User is lost & wants the general direction of the next checkpoint.



At the corner of their vision, a triangle is displayed pointing towards which direction the checkpoint is in.

The HUD disappears.



Heuristic Evaluation Guide

Perform a heuristic evaluation of the system you implemented.

***Violation**: How does your system violate the heuristic? Be specific.*
***Severity (high, medium, low)**: How much does this violation impact the user experience?*
***Recommendation**: How can you fix this issue? If any of your design choices work better, mention it.*

Heuristic	Violation	Severity	Recommendation
1 **Visibility** *of* **System Status** **Designs should *keep users informed* about what is going on, through appropriate, timely feedback.**  Interactive mall maps have to show people where they currently are, to help them understand where to go next.	Although there is immediate feedback between control and action (e.g. giving a right thumbs up moves the plane forward), the system doesn't give any textual feedback about the state of the gesture recognition system.	Medium	The system could have a text display that shows what gestures it thinks each hand is making, as well as the confidence level of the system in each gesture.
2 **Match between System and the Real World** **The design should speak the users' language. Use words, phrases, and concepts *familiar to the user,* rather than internal jargon.**  Users can quickly understand which stovetop control maps to each heating element.	The plane controls somewhat mirror how real-life plane controls work: there is a throttle control which adds thrust force to the plane, as well as pitch and turning controls which control the plane in a realistic way.	Low	An alternative FPV-style control system could provide increased maneuverability, where the orientation of the plane follows the orientation of one of the hands.
3 **User Control** *and* **Freedom** **Users often perform actions by mistake. They *need a clearly marked "emergency exit"* to leave the unwanted action.**  Just like physical spaces, digital spaces need quick "emergency" exits too.	In our system, crashing into a building or the ground is very forgiving: the user is respawned at the last checkpoint *and* they're pointed in the direction of the next checkpoint. As a result, the user can "emergency exit" by simply crashing.	Low	An explicit control for resetting to the last checkpoint could be added to make this process of "emergency exiting" easier and more fluid.

4 Consistency *and* Standards **Users should not have to wonder whether different words, situations, or actions mean the same thing. *Follow platform conventions.***  Check-in counters are usually located at the front of hotels, which meets expectations.	The physical controls for this plane control don't really mirror those in the real world. Drones are normally controlled with controllers, and figuring out how to orient the plane isn't entirely intuitive.	Medium	The FPV control method from Section 2 could be implemented, which provides a more intuitive way of orienting the plane in a virtual space (at the expense of less realism with regards to the physics of plane kinematics).
5 Error Prevention **Good error messages are important, but the best designs carefully *prevent problems* from occurring in the first place.**  Guard rails on curvy mountain roads prevent drivers from falling off cliffs.	The wayfinding techniques do a passable job of preventing users from getting completely lost; the minimap shows users the direction of the next checkpoint, and the line between checkpoints is a useful guide. However, the visibility of the minimap is somewhat poor, and it's easy for the minimap to be lost.	Medium	An additional wayfinding technique can be added which points an arrow in the direction of the nearest checkpoint. This arrow would function similarly to waypoint markers in FPS games.
6 Recognition Rather Than Recall ***Minimize the user's memory load* by making elements, actions, and options visible. Avoid making users remember information.**  People are likely to correctly answer "Is Lisbon the capital of Portugal?".	The UI doesn't show any of the potential interactions the user can make with the system. It's assumed the user already has prior knowledge about how to operate the system, and as such discoverability is relatively poor.	High	The recommendation here is similar to that in the Help and Documentation section; explicit text could be added which displays the controls for the plane to the user.
7 Flexibility *and* Efficiency of Use **Shortcuts — hidden from novice users — may *speed up the interaction* for the expert user.**  Regular routes are listed on maps, but locals with more knowledge of the area can take shortcuts.	Because of how the gesture control system works, it's very difficult for an advanced user to do advanced maneuvers with the plane by performing multiple turning controls at once; only one gesture per hand is recognized at a time, so the plane can only pitch, then only turn, etc.	High	The gesture system could potentially be modified to detect multiple gestures at a time, or the FPV control method from Section 2 could be implemented.

8 Aesthetic *and* Minimalist Design **Interfaces should not contain information which is irrelevant. Every extra unit of information in an interface *competes* with the relevant units of information.**  A minimalist three-legged stool is still a place to sit.	Our UI has two elements: the text display in the center which shows the checkpoints and time remaining, as well as the minimap at the bottom showing the direction of next checkpoints. However, the minimap can be hard to see in certain plane orientations, and sometimes parts of the world will obscure the minimap	Low	The minimap could be moved closer to the center and could be made to always draw over other elements of the screen.
9 Recognize, Diagnose, *and* Recover from Errors **Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.**  Wrong-way signs on the road remind drivers that they are heading in the wrong direction.	The error state of this system is that the user gets lost; if a user gets lost, there's no explicit mechanism for getting the user back on track other than crashing or trying to use our wayfinding aids and having the user reorient themselves.	Medium	An explicit control could be added which reorients the plane towards the next checkpoint, allowing the user to gracefully recover from this error state and continue onwards.
10 Help *and* Documentation **It's best if the design *doesn't need* any additional explanation. However, it may be necessary to provide documentation to help users complete their tasks.**  Information kiosks at airports are easily recognizable and solve customers' problems in context and immediately.	Our design doesn't provide any explicit documentation for how to actually control the plane—no text is displayed when the state of the system changes, and no help text pops up with suggestions for actions or controls for the system.	High	A display at the bottom of the user's field of vision could display controls for how to interact with the plane (i.e. text would show which gestures correspond to what movements); this display could be persistent, or could disappear on first interaction.