

Team Komatte Sky Commander Pre-Production

By Elizabeth Wright (Comm Arts) and Shane Simes
(Computer Science) Intro to Game Dev, CMSC 391 / COAR
463, 2024 Fall

DESIGN

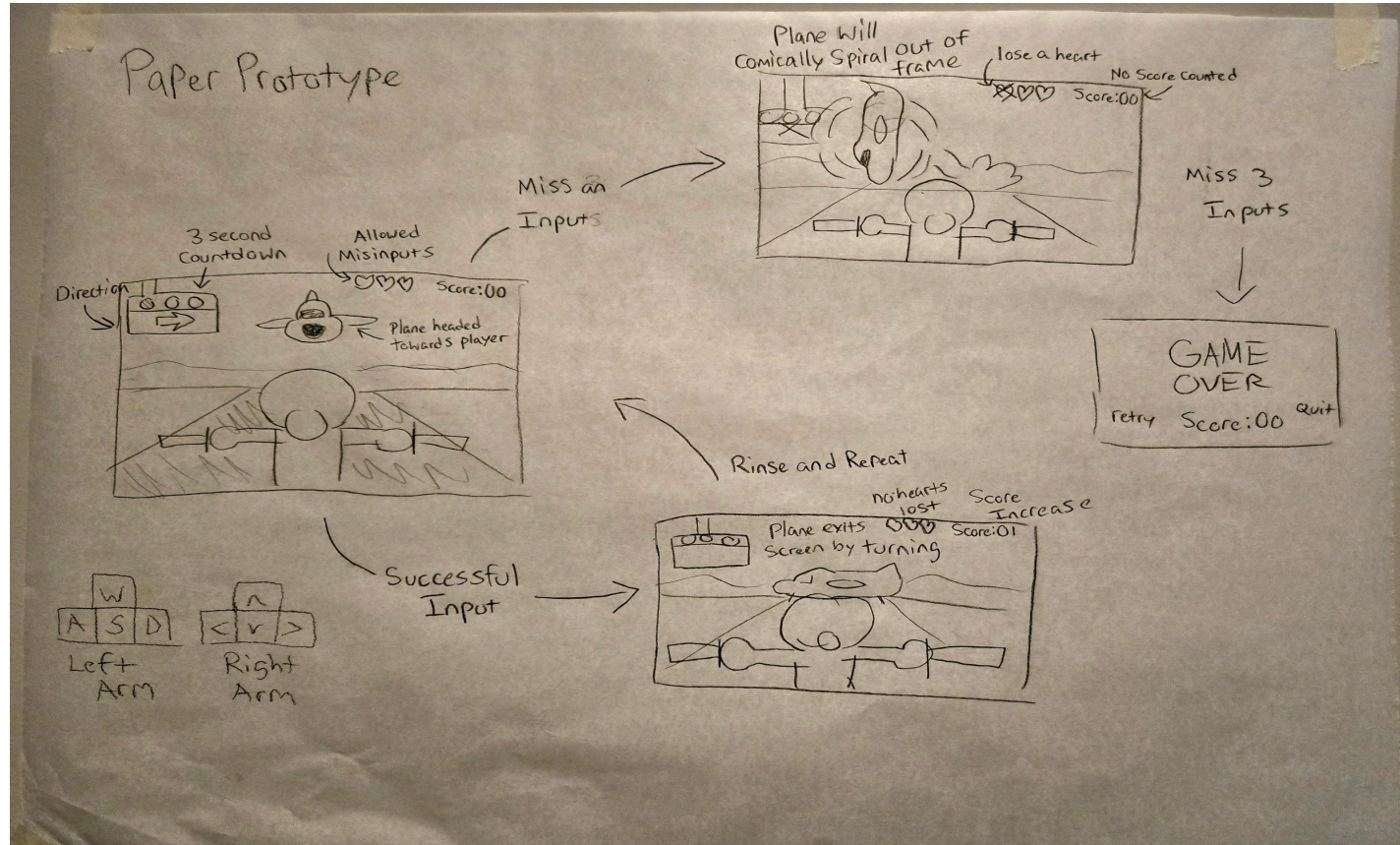
- **HIGH LEVEL CONCEPT**

- Sky Commander
- Single player arcade
- Summary: Sky Commander is an arcade game about a new air traffic controller directing the planes that land with increasing difficulty.
- Unique Selling Point: The controls are unique, requiring both arrow keys and wasd keys or (if possible) motion controls from Switch joycons.

- **DESIGN SPECIFICS**

- PC and Keyboard (maybe switch joycons)
- Audience / Age range: 10+
- Visual Style Similar to Paper Mario / Wario ware (paper puppet).

PAPER PROTOTYPE:



STORYBOARD BRANCH

Fail 3
times



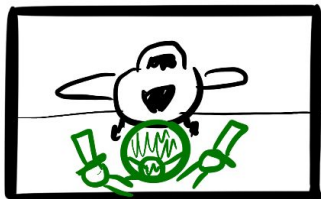
Score 0-10
Bad (charged)



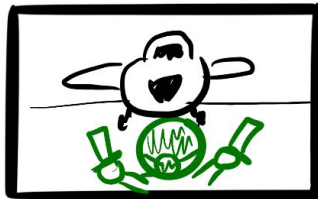
Score 11-20
Normal (sued?)



Score 21-∞
Pretty Good
(Not your fault?)



Keep
Succeeding



Game continues and increases
in difficulty or 3 missed Input

All Game Over
Screens allow
For Retry

Visual Assets List

- ***Visual Assets Needed:***
 - Background
 - Player Character Assets
 - Animations of: Plane landing, Plane crashing (failing), Timer countdown, and background distractions.
 - Game Over Screens (depending on score)
 - (Possibly) a screen (like a comic panel) of introducing the premise and main character.
 - UI Assets (Most likely from online or Unity Store)
- ***Audio Assets Needed:***
- Airport Runway Ambiance.
- Background Music
- Plane landing and crashing sound effects (cartoonish)
- More cartoon sound effects that I cannot think of right now

Programming Assets List

Resources

- **Unity Documentation for Input System:** Official documentation on Unity's new Input System for managing device input, including Joy-Cons.
- **Third-Party Libraries or SDKs:** Use libraries like *Rewired* or *InControl* for easier Joy-Con support.
- **YouTube Tutorials:** Specific tutorials for integrating Joy-Con controllers with Unity, including handling motion controls.
- **Joy-Con Specific Libraries:** Libraries like *Joy-Con-Unity* or *JoyConLib* can help in accessing Joy-Con specific features like motion sensors.

Types of Code to Implement (More Detail In Notes)

1. **Player Input Code:** Code to handle the WASD and arrow key inputs for controlling the air traffic lights.
2. **Airplane Approach Code:** Script to control the airplane's movement and scaling effect, making it appear to approach the player.
3. **Collision Detection:** Code to detect if the airplane reaches the player (for game over logic or score deduction).
4. **UI/Score Management:** Code for tracking and displaying the player's score, attempts, or any countdowns.
5. **Game Logic Code:** Code to manage the flow of the game, such as restarting on game over, keeping track of score, and handling difficulty increase.
6. **Animation Control:** Code to manage animations for various assets, ensuring smooth visual transitions (e.g., for traffic light movements and airplane approach).

Joy Con Implementation

Types of Code to Implement

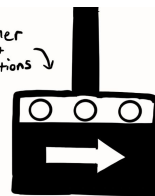
- **Joy-Con Input Handling:** Code to detect and manage inputs from Joy-Cons, including buttons, thumbsticks, and motion sensors.
- **Motion Control:** Code to read the accelerometer and gyroscope data from the Joy-Cons for motion-based controls, such as tilting to move the player or interacting with the game.
- **Button Mapping:** Code to map specific Joy-Con buttons to in-game actions, like using the Joy-Con buttons for specific controls or interactions (e.g., menu navigation, gameplay actions).
- **Joy-Con Pairing:** Code to handle the process of detecting and pairing the Joy-Con controllers with the game, ensuring proper device recognition on macOS (or other platforms).

Asset Concepts:

Filler Assets 2

Distractions:
(Will move across the Screen in BG)

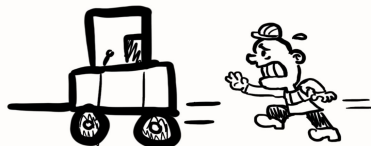
Timer
+
Directions



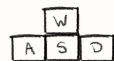
Possible directions:



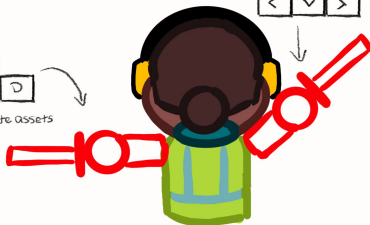
STOP



Hero/Player



Arms are Separate Assets



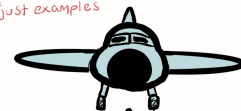
Filler Asset 1

Silhouette



* Threshold numbers are just examples

Low Difficulty 0-10



Normal Difficulty 11-20



Med Difficulty 21-30



High Difficulty 31-∞



CALENDAR & ROLES

NOVEMBER	2024		Elizabeth		Shane	
Sunday	Monday	Tuesday	Wenesday	Thursday	Friday	Saturday
				storyboard pitch		
27	28	29	30	31	1	2
				Art pre product		
3	4	5	6	7	8	9
		MVP Assets	MVP Assets	MVP Assets		
			Initial code setup	Basic Controls	Basic Controls	Core gameplay loop
10	11	12	13	14	15	16
		Animations	Animations	Animations		
		Core gameplay loop	Implement score tracking		Failure conditions	core loop and scoring testing
17	18	19	20	21	22	23
	Menu Assets	Menu Assets				
	UI implementation	UI implementation	Final testing	Final testing		
24	25	26	27	28	29	30
		UI / SFX	UI / SFX	UI / SFX		
1	2	3	4	5	6	7

REFERENCES

- ***Risks:***
 - There's only two people working on this project
 - Figuring out input combos and scoring
 - Increasing the difficulty
- ***Resources Available:***
 - SFX Assets from Unity Store
 - (Any youtube tutorials for code)