Assignment 15.3: Input System

For *Dabbawala Dash* I came up with the following input systems for a digital prototype, keeping the wireframe from Assignment 15.1 in mind. The following list is grouped by purpose, and I've put controllers for a PC (keyboard/mouse) and a typical Gamepad.

Core Riding & Lane Control

The core gameplay of this loop is riding the bike, accelerating, braking, and navigating the streets of Mumbai, by making turns and avoiding spilling your food, requiring the following controls:

Function	Controller PC	Controller Gamepad	Use	Why this input	Alternatives considered
Throttle / accelerate	W	Right Trigger (RT)	Increase scooter speed	W and RT tend to be the standard for most racing games. By maintaining this standard it is easy for new players to learn this function.	I considered the Spacebar for gas but rejected because I think space is better for context actions. I also considered the up arrow, but wanted to keep that for navigation.
Brake	S	Left Trigger (LT)	Reduces speed; smoother braking lowers spill buildup during tight dodges.	Mirrors accelerate placement; LT is the de-facto brake and S comes below the W.	Rejected Space because it didn't make intuitive sense and limits one-handed play (for accessibility.
Navigate Streets (from player viewpoint)	Default arrow keys: ← → ↑ ↓	Left Stick (analog)	Fine, continuous positional steering within the road bounds depending on the direction to be traversed.	Keeps speed (W/S or RT/LT) independent from where you place the scooter. This is from the players viewpoint so easy to navigate and intuitive directions	Discrete lane-swap on A/D or D-pad (kept as a future option for an arcade mode), mouse steering (rejected for consistency).

Note: for the navigation: If the player hits side railings or another car they generate a spill burst, which over time results in a time loss: no death.

Interaction and signaling

Another part of this game is the actual delivery. This involves a brief interaction to deliver the food at the "client". I've also added a secondary "fun" factor into this: the horn. Beeping the horn allows the player to re-create the soundscape of a typical Mumbai street. It can also serve as a traffic aid for NPCs in future iterations (feature TBD).

Function	Controller PC	Controller Gamepad	Use	Why this input	Alternatives considered

Interact/Deliver	Enter	A	Confirm delivery at client, interact with world prompts	Standard "primary action"; easy to discover and doesn't conflict with throttle/brake or navigation.	I considered the "D" for delivery, but that language limits the game, and since the goal is the delivery, the big "space" or "enter" made sense.
Honk	Space	X / Square	This is really for the flavor. Can be later incorporate into light gameplay: quick audible cue that can prompt NPC cars to ease off slightly.	Easy of access of the space and regularly using it is part of the "sonic scape" of Mumbai. Making sure this was a central button/key and easy to use was important.	I considered H (for Honk), but favored spacebar and X/Square because I want it to be something players can quickly go to as part of the "fun" in the game.

NOTE: the horn honk is an optional feature that may not be in the first digital prototype. But it's a neat way to bring the sound of the Mumbai streets to the gameplay while giving the player some agency to participate with no huge impact on the gameplay (unless it is incorporated into the gameplay).

Gameflow and UI

Function	Cont-roller PC	Controller Gamepad	Use	Why this input	Alternatives considered
Pause / Open Menu	Esc or P	Start / Options	Pause, options, restart, quit	Universal convention: easy for playtests	Tab could be used, but I kept that for debugging below
Resume / Back	Esc	B (circle)	Close pause/menu panels without confirming	Matches standard A = confirm / B = back muscle memory from most game controllers. Esc is common on PC applications	I considered back-space for going back after a menu opens, but felt that also is used as delete sometimes.
UI Mouse support	Mouse left-click	N/A	To select the Menu function and settings only		Standard PC UX; helps on laptops especially

Debug & Accessibility

These are specific functions to help with the prototyping and playtesting:

Function	Cont-roller PC	Controller Gamepad	Use	Why this input	Alternatives considered
Restart from checkpoing	R	Y / Triangle	Could have certain checkpoints (e.g. a single delivery) that can be used to quickly check sections and restart	Speeds tuning of the game	Menu-only restart, but thought this works well within the game testing.
Slow-Mo toggle	\ (backslash)	None assigned (only for testing purposes)	Temporarily halves game speed—great for reading		Fast playtest tool; easy to reach, unlikely to conflict.

			collisions and spill tuning.		
Master Mute	M	None	Quick audio mute control during lab playtests		None considered

Future Addons

There are a few other functions I would add on in the future (not in first prototype) that I think could be features of the game:

- Lane signals/indicators: Shift+arrow direction
- Map/route overview (peek) – perhaps a Tab or View/Select function.
- Acknowledge police / pay fine: Confirm with A

At that time would also need to look at potential Gamepad alternatives.

Unity System Inputs

Here is a breakdown of what would need to be in the Unity Input System, broken down into three action maps: Gameplay, UI and Debug:

Gameplay

- Accelerate (Button) — W, RT
- Brake (Button) — S, LT
- Move (Vector2) — Arrow keys (← ↑ → ↓) as a 2D composite, Left Stick (analog)
- *(These need to be clamped within road bounds; railing/car bumps trigger a spill burst, no death).*
- Interact (Button) — Enter / Numpad Enter, A / Cross
- Honk (Button) — Space, X / Square *(optional feature for early prototype)*
- Pause (Button) — Esc / P, Start / Options
- Cancel/Back (Button) — Esc, B / Circle
- Restart (Button) — R, Y / Triangle

UI (menus & overlays)

- Navigate (Vector2) — Arrow keys (2D composite), D-pad, Left Stick
- Submit — Enter / Numpad Enter, A / Cross
- Cancel — Esc / Backspace, B / Circle
- Point — Mouse position
- Click — Mouse left button

Debug (prototype-only)

- SlowMoToggle — \ (backslash)
- MuteToggle — M

Control Schemes

Keyboard&Mouse and Gamepad (auto-switching can perhaps work with PlayerInput).