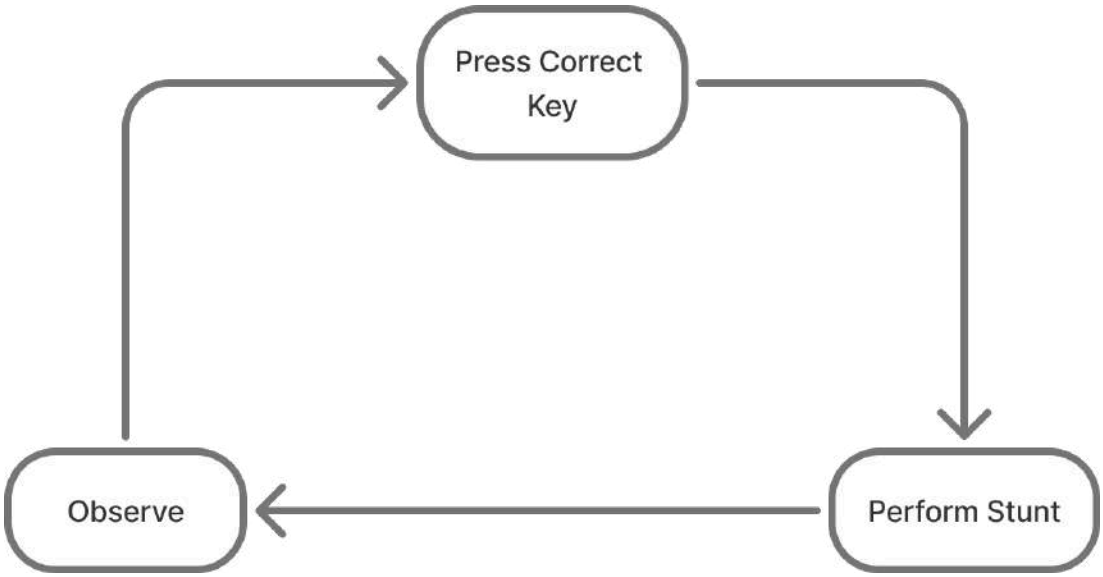
Stunt Club: Design Documentation

Overview

- **Title:** Stunt Club
- **Engine:** Unity
- **Genre:** 3D Auto-Runner / Rhythm-Action / Casual
- **Controls:** 3-Button Keyboard (SPACE, D, S)
- **Platform:** PC (Keyboard)
- **Mode:** Single-Player, Scene-Based
- **Target Audience:** Casual players, action movie fans, players who enjoy flow-state and rhythm games.
- **One line description:** Perform high-octane stunts in a single, unbroken take to become the ultimate action movie hero.
- **Core idea:** Stunt Club is a game about "performing" a pre-scripted action movie. It's a 3D auto-runner where the player isn't just *reacting* to obstacles, but *executing* a pre-choreographed stunt sequence. The "game" is to perfectly nail the timing of the "performance."
- **Description:** Stunt Club is a fast-paced, cinematic, 3-button auto-runner. Players take on the role of a stuntman running through various "movie sets" (a speeding desert train, a chaotic city chase). The character runs automatically, and the player must press the correct key (Jump, Dive, or Slide) in time with on-screen obstacles to successfully perform the stunt. The game's narrative is presented as a movie script, with each level being a "scene" in an action film.

Core Loop



The Process

Ideation

- **Design goal:** To capture the feeling of being an action hero in a single, unbroken "one-take" movie scene. The design prioritizes the **spectacle of performance** and **rhythm-based timing** over complex controls or player agency.

Ideation & USP

- Unique Selling Point (USP): "Rhythm-Action as Performance."
  The core emotion of the game is not survival (like in a traditional endless runner) but perfect execution. The player is mastering a piece of choreography, much like a rhythm game. The "movie set" premise brilliantly contextualizes this.

Core of the Game

The game's 3-button mechanic distills complex action sequences into a simple, accessible, rhythm-based challenge. The fun emerges from this refined loop:

1. **Anticipation:** The player sees an upcoming obstacle (e.g., a low-hanging truck).
2. **Information:** The game provides a clear, diegetic cue (a "SLIDE" sign) *before* the obstacle.
3. **Execution:** The player's "performance" is judged on their ability to press 'S' at the precise moment.
4. **Reward:** Success results in a cool, cinematic animation, maintaining the "flow." Failure results in a clumsy (but non-fatal) stumble, "ruining the take" and breaking the flow.

Player Behavior	**Underlying Emotion**
Pressing the key at the perfect moment	Mastery, Flow, "Feeling Cool"
Pressing the key too early/late	Frustration, "Broke the flow"
Seeing the "Script" before a level	Anticipation, Narrative Context
Reaching the end of the scene	Spectacle, Accomplishment

In-game View & Design Challenge

Design Challenge: Cognitive Load

The central challenge was: How do you make a 3-button auto-runner engaging while ensuring the player is never confused?

Key questions we asked:

- Should the player also control steering?
  - **Decision:** No. An auto-run ("on-rails") system keeps the player's cognitive load focused 100% on **timing**, which is the core mechanic.
- How do we teach the player *which* of the 3 buttons to press for each obstacle?

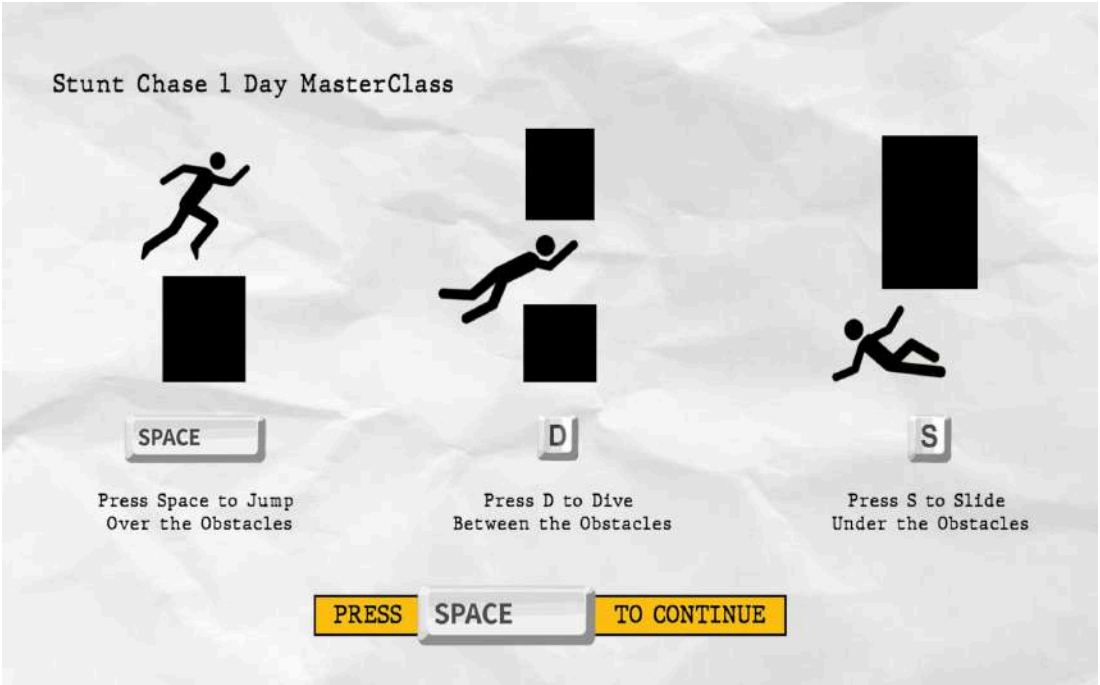
Iteration & Cognitive Load

Iteration 1 - (Rejected Concept): Explicit "Feed-Forward" Cues

- **Concept:** The game places explicit "JUMP," "DIVE," and "SLIDE" signs in the world *before* each obstacle.
- **Outcome (Rejected):** This approach was ultimately discarded. While it made the game more accessible, it removed the core "stunt performer" fantasy. It turned the game into a simple rhythm-matching exercise ("Simon Says") rather than a test of skill and intuition. The player was just *following directions* instead of *making decisions*.

Iteration 2 - (Finalized Concept): Intuitive Obstacle-Based Decisions

- **Concept:** All explicit signs are removed. The player must intuitively decide the correct action (Jump, Dive, or Slide) based *purely* on the shape, height, and placement of the obstacle.
- **Outcome:** This is the key design insight. The game trusts the player to "read" the environment like a professional stunt artist.
    - A low barrier or gap. The player must *intuit* a **JUMP**.
    - A high-hanging obstacle (like a laundry line or the underside of a truck). The player must *intuit* a **SLIDE**.
    - A mid-height opening (like barrels or a window). The player must *intuit* a **DIVE**.
- This design moves the **decision-making** (what button to press?) *into* the high-pressure moment, forcing the player to rely on instinct.



Cognitive Load Study:

- With this design, the primary cognitive load is a high-speed, demanding blend of **perceptual problem-solving** (instantly identifying the obstacle's "type") and **reaction time** (executing the correct action in the tiny timing window).
- The game is no longer "read-then-perform." It is a continuous, high-speed loop of **"observe, analyze, decide, execute."**
- By refusing to "tell" the player the answer, the design creates a much steeper challenge but also a far greater sense of mastery. Success feels "earned," as the player is not just following instructions but demonstrating true environmental awareness.

Design Challenge: Cognitive Load

The design brilliantly balances attention. The player "reads" the action cue (the sign) and then "performs" the action on the obstacle. This "read-then-perform" loop is the entire game.

Screen Layout & Presentation

(This section parallels the "Split-Screen" analysis in the template, focusing on the core presentation choice.)

The "Movie Set" Frame

The most important design decision was not the screen layout, but the **narrative frame**. The game is presented as a movie shoot.

- **Side-by-Side (Rejected):** N/A
- **Top-Bottom (Rejected):** N/A
- **Final Decision - The "Single Take" Frame:** The game uses a single, unbroken camera perspective (per scene) that follows the player.

Final Decision - The "Movie Set" Frame

Why this frame is critical:

1. **Narrative Context:** It gives *purpose* to the auto-running. The player isn't just running; they are "following the script" and "hitting their marks."
2. **Level Structure:** It neatly packages levels into "Scenes" that can be selected.
3. **Goal & Motivation:** The goal isn't an arbitrary "high score" but to "Complete the Scene" and get the "Director's" approval.
4. **Contextualizes "On-Rails":** The "on-rails" mechanic feels like a *choice* (the player is following a camera dolly on a set) rather than a *limitation*.

Key Insight
"When the game is framed as a performance, the on-rails mechanics and simple controls feel empowering, not restrictive. The player is the action hero."

Finalized Gameplay

Overview
The final design is a hybrid auto-runner/rhythm-game. The player's journey through the "scene" is a "note highway" of Jumps, Dives, and Slides. The game's diegetic cues (the signs) are the "notes," and the obstacles are the "timing windows."

Controls & Mechanics

End Criteria

- **Success:** Player reaches the end of the scripted sequence (e.g., jumping off the train before it explodes).
- **Failure:** The game is highly permissive. Failing a stunt causes a stumble and a break in the "flow," but does not cause a 'Game Over'. The goal is a perfect "take," so these stumbles are minor "failures."

Game Elements (Stunts)

- **JUMP (SPACE):** Used to go **over** obstacles.
  - *Examples:* Low mats, gaps between train cars, onto platforms.
- **DIVE (D):** Used to go **through** mid-height obstacles.
  - *Examples:* Barrels, windows/gaps, vegetable carts.
- **SLIDE (S):** Used to go **under** high obstacles.
  - *Examples:* Barriers, laundry lines, under trucks.

Gamepad / Control Schema

Final Mapping Schema

The controls are mapped mnemonically and spatially.

Domain	Input	Cognitive Framing
High Action	SPACE (Jump)	The largest, most common "action" button. Spatially "up."
Mid Action	D (Dive)	A forward, aggressive action. Player "Dives" *through*.
Low Action	S (Slide)	"S" for Slide. A strong mnemonic. Player "Slides" *under*.

HUD & Feedback System

Design Problem

How to provide clear, instant, non-intrusive feedback for a high-speed game.

System Solution - Diegetic Cues

The HUD is minimal (e.g., "Press M for Main Menu"). The *real* feedback system is **diegetic**, built directly into the game world.

- **Early Approach (Rejected):** A traditional HUD with flashing icons (e.g., "Press SPACE!").
- **Issue:** This breaks immersion ("un-cinematic") and draws the player's eye away from the action.
- **Final Implementation:**
  - **Diegetic "Feed-Forward":** Yellow road signs appear in the world.
  - **Attention economy:** The sign is placed far enough ahead to be read, but close enough to the obstacle to be relevant. This "read-ahead" mechanic is the core loop.

Interaction Flow

1. Player auto-runs through the "set."
2. Player's eye spots a yellow sign (e.g., "DIVE") in the distance.
3. Player "prepares" to press 'D'.
4. Player sees the corresponding obstacle (e.g., barrels) come into view.
5. Player presses 'D' at the correct time to execute the cinematic stunt.
6. The loop repeats.

Stunt Taxonomy & Spatial Mapping

Key	Action	Spatial Logic	Example Obstacle	Psychological Intention
SPACE	Jump	**Over** (Vertical)	Low wall, gap between cars	Agility, Leaping
D	Dive	**Through** (Horizontal)	Barrels, window, cart	Daring, "Action Hero" move
S	Slide	**Under** (Low)	Barrier, truck, laundry line	Evasion, "Slick" move

HUD Visualization

- The HUD *is* the world.
- All information is "diegetic," meaning it exists within the game's fictional space.
- This adheres to minimal-intrusion UI principles, maximizing immersion and "flow."

Psychological & Design Frameworks

Principle	Application in Stunt Club
Flow Theory (Csíkszentmihályi)	The 3-button, fixed-timing challenge creates a perfect "flow channel." The difficulty is static, and player mastery comes from learning the "rhythm" of the scene.
Diegetic Design	All critical UI (Jump/Dive/Slide cues) is placed *inside* the game world as signs, reducing cognitive load and improving immersion.
Feed-Forward	The game *always* tells the player what to do *before* they need to do it (via the signs). The challenge is execution, not puzzle-solving.
Kuleshov Effect (Cinematic)	The "performance" is created by editing: (Action 1: Press 'D') + (Action 2: Character dives) = (Player "feels" they *did* the dive).

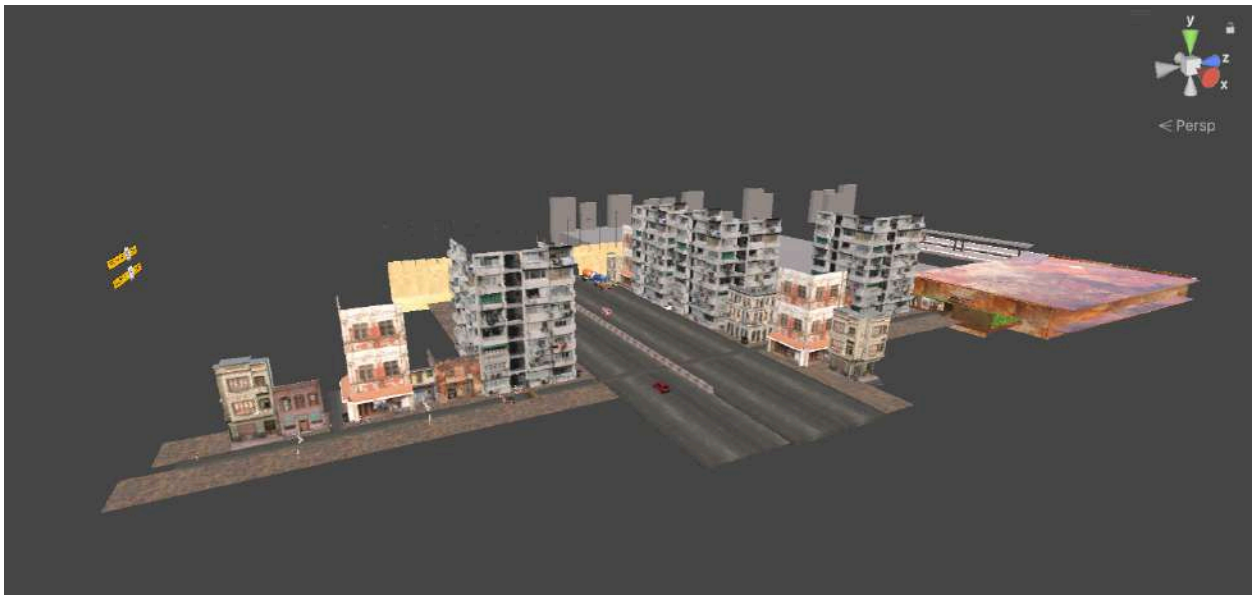
Key Insight

The joy of Stunt Club is not in challenge, but in **spectacle**. It's a "power fantasy" game that makes the player *feel* like a skilled action star by simplifying the complex choreography of a stunt into a single, satisfying button press.

Visuals

Visual Design Goals

- Cinematic, low-poly, and highly readable. The player must be able to parse obstacles at high speed.



Visual Pillar: "The Action Movie Set"

- **Reference & Inspiration:** The levels are direct homages to action movie set-pieces.
  - **"Desert":** A "Run on top of a train" scene (e.g., *Mission: Impossible*, *Uncharted*).
  - **"City":** A "Chase through a crowded street" scene (e.g., *Bourne* series or a Bollywood action film, given the "Chennai Beach" sign).

- **3D World:** Stylized, low-poly environments that serve as "sets."
  - **Tutorial:** A gym, for "training".
  - **Desert:** A speeding train in a vast, arid landscape.
  - **City:** A dense, urban street with Indian-inspired (Tamil) posters and signs.

- **Lighting: Cinematic & Dramatic**
  - **"Desert":** Saturated golden-hour/sunset lighting to create a high-stakes, dramatic feel.
  - **"City":** Harsh, bright daylight to create a sense of exposure and a chaotic, grounded-in-reality chase.

- **Animation Strategy: Exaggerated, Readable, Fun**
  - All animations are "over-the-top" to emphasize the "stunt" concept.
  - **Success States:** The jumps, dives, and slides are all overly-dramatic and "cool."
  - **Fail-States:** Failing a stunt results in a "ragdoll" stumble. This adds a layer of physical comedy and is a low-penalty-feedback mechanism.

- **UI/UX Complement**
  - **Diegetic UI:** The "Stunt Chase... MasterClass" and "Script" screens are the only non-diegetic UI, serving as menus.
  - **In-Game UI:** All in-game UI is diegetic.

Sound

We want the sound to be cinematic, layered, and impactful, reinforcing the "movie set" fantasy.

Layers

1. **BGM (Cinematic Loop):**
   - A driving, high-energy orchestral or electronic track that builds tension, appropriate for an action scene.
2. **SFX (Stunts & Environment):**
   - **Player:** Loud footsteps, a "whoosh" for dives/slides, grunts.
   - **Fail-State:** A comical "thud," "oof," or "ragdoll" sound.
   - **Environment:** The roar and clatter of the train, the street-level ambience of the city.

3. **Narrative SFX:**
    - Crucial sounds that tell the story, such as the massive train explosion or a director's "Action!" and "Cut!"

Why this works

- **Cross-modal congruence:** A loud "whoosh" sound paired with a "Dive" animation makes the action feel *more* impactful and satisfying.
- The sound design completes the fantasy that the visual design and gameplay mechanics establish.

THANK YOU