TDD (Technical Design Document)

GeeksInstitute - LaStartupStation

Project Name: Tbourida

Name	Role
Abdellah Aoukrad	Gameplay Programmer/Game Designer/Producer
Zakaria Rezki	Graphic Designer / 3D / 2D
Soufiane Mjadi	Graphic Designer / 3D / 2D
Yassine Ait HMAD	

Version Table

Version No.	Date	Author(s)	Description
1.0	2025-12-18	Yassine Ait Hmad	Draft TDD with same sections as the game idea docs.
2.0	2025-12-23	Abdellah Aoukrad	Final TDD with proper TDD sections.
3.0	2026-01-08	Abdellah Aoukrad	Alpha version TDD.

1. Game Presentation

Tbourida is a 3D rhythm-simulation game featuring a stylized low-poly aesthetic. The technical scope has evolved from a basic prototype to a complex coordination system. The core focus is the "Sorba synchronization," where AI agents follow a player-leader using sophisticated damping and formation logic. By utilizing highly optimized low-poly assets, the project ensures high visual clarity and stable performance, allowing for a high number of on-screen agents (10+ riders plus dense crowds) without performance degradation.

2. Platforms and Hardware Specifications

- **Target Platforms:** PC (Windows 10/11) and Mobile (Android API 26+, iOS 13+).
- **Performance Targets:** The project is strictly optimized for 60 FPS on entry-level mobile devices. This is achieved through aggressive polygon reduction and the use of vertex colors and optimized URP shaders to minimize draw calls.
- **Input Handling:** Implementation via the Unity Input System, supporting WASD/Arrows for steering and Spacebar for the final shooting phase across all platforms.

3. Development Environment

- **Game Engine:** Unity 6.3 6000.3.1f1 LTS.
- **Rendering Pipeline:** Universal Render Pipeline (URP).
- **Version Control:** Unity version control.
- **3D Workflow:** Blender 4.5.5 for modeling and animation, supported by custom automation scripts for asset pipeline efficiency.

4. Software Architecture & Game Systems

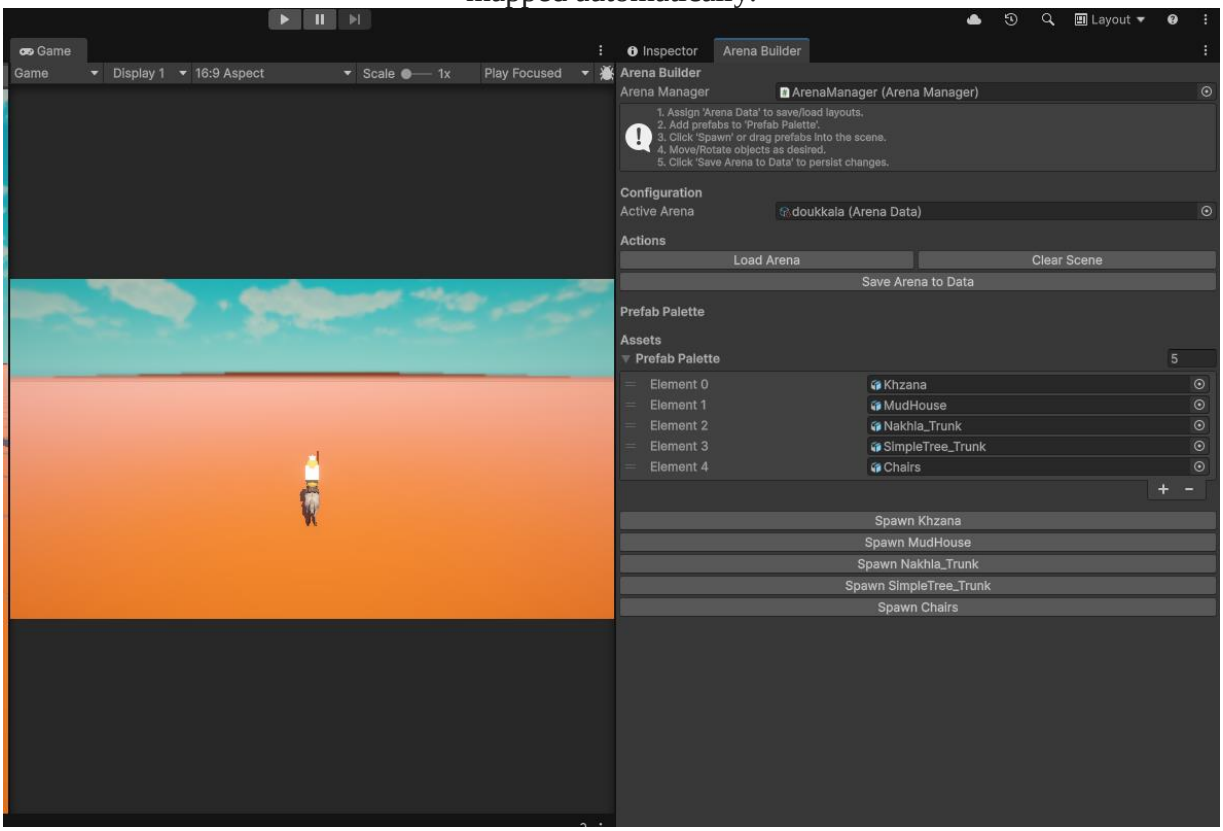
The game follows a decoupled, manager-based architecture coordinated by a central state machine:

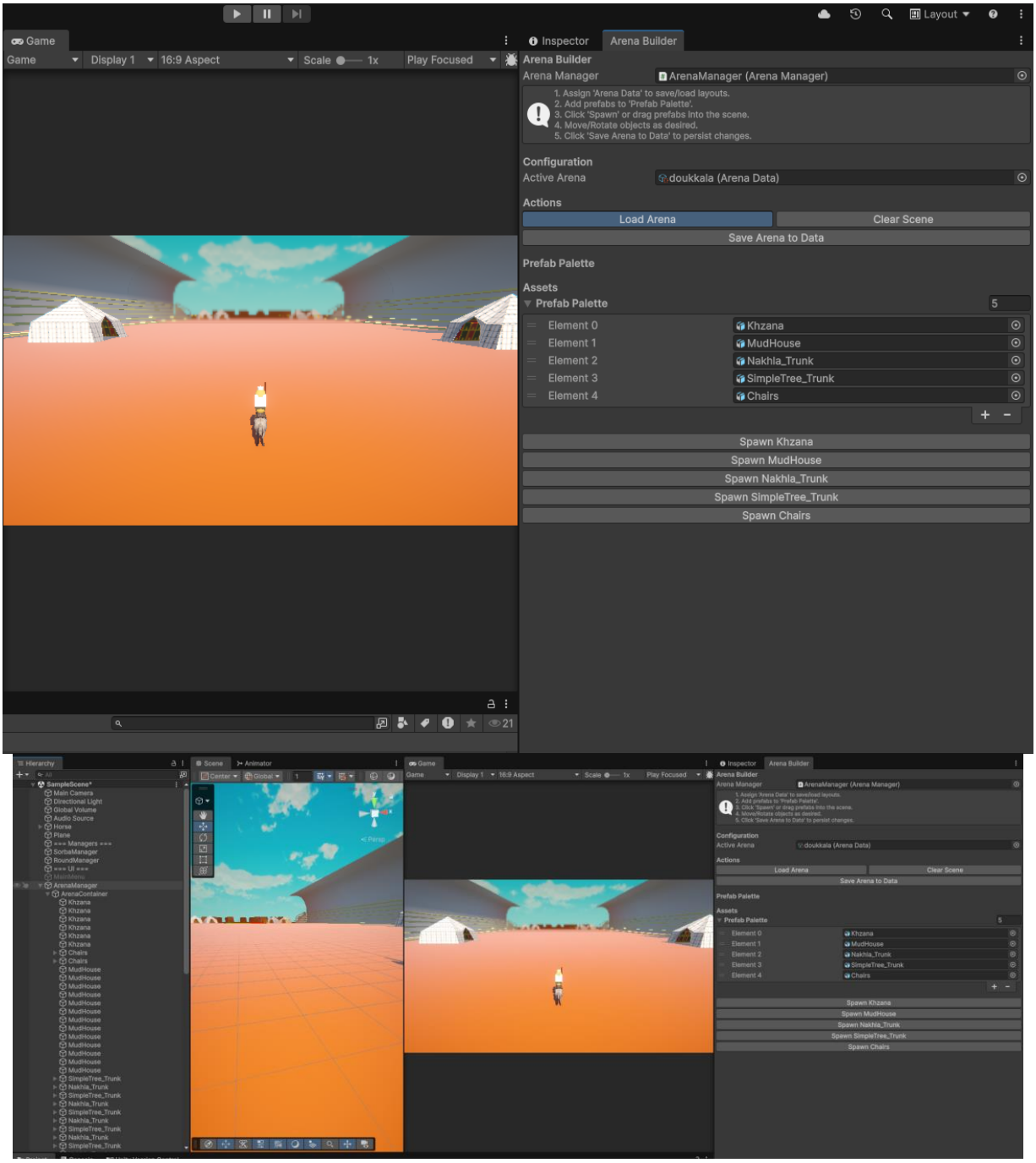
- **GameManager:** Drives the high-level GameState, including the round lifecycle (Menu, Playing, Pause/Resume, Shop, and End-of-round). It handles credit rewards and session transitions.
- **SessionManager:** Manages persistence using JSON and PlayerPrefs. It tracks PlayerData, including credits, owned equipment, shop purchases, and historical round results.
- **MovementEngine:** A dedicated physics-based controller handling horse acceleration, braking, natural velocity decay, and steering clamp logic to prevent unrealistic turns.
- **SorbaManager & AI:** A sophisticated system that auto-finds or spawns agents. It assigns alternating formation offsets and uses smooth damping to ensure AI riders follow the Mokaddem naturally. It captures high-precision timestamps for all riders to compute the synchronization score.
- **ShootingEngine:** Manages the "Baroud" phase. It detects entry into the shooting zone, triggers muzzle flash VFX, and calculates timing accuracy.

- **CameraEngine:** Features a dynamic chase camera with speed-based FOV zoom, a subtle "braking dip" for weight feedback, and a side-view transition during the shooting climax.

5. Custom Tools Developed

- **ArenaBuilderWindow:** A custom Unity Editor tool allowing developers to spawn environment prefabs from a palette and save track layouts into ArenaData ScriptableObjects.
- **Blender Export Script:** A Python-based automation tool that exports Blender collections as individual FBXs directly into Unity folders, ensuring baked textures and materials are correctly mapped automatically.





6. Graphic Rendering & Optimization

- **Asset Optimization:** All models are optimized for low-poly counts (e.g., house < 100 triangles).
- **Crowd System:** A fully rebuilt stadium model featuring optimized "billboard" and low-poly crowd models with alpha/transparent textures to simulate thousands of spectators with minimal CPU/GPU impact.
- **Materials:** Utilization of custom interior/exterior textures for the Khzana (tent) and mud houses. Trees and Nakhla (palm) foliage have been optimized into lightweight assets.

7. Menus & HUD

- **UI Technology:** Migrated to Unity UI Toolkit (UIDocument) for modern, responsive layouts.
- **HUD Elements:** Real-time Sorba Alignment Indicator, Speedometer, and Charge Progress Bar.
- **In-World UI:** Start, Optimal, and Finish lines are rendered as 3D meshes to avoid UI overlay clutter.

8. Physics & Gameplay Logic

- **The Drift:** A Perlin Noise function generates lateral forces that the player must actively counter via steering.
- **Scoring Math:** The synchronization score is calculated as the standard deviation of all shot timestamps within the Sorba. AI agents fire with a randomized delay (10ms–50ms) to simulate human variation.

9. Audio System

- **Dynamic Audio:** Hoofbeat loops are mapped to current velocity via the SoundManager.
- **Triggers:** Impactful "Baroud" gunshot samples and crowd "Zaghroutas" (ululations) are triggered based on the final performance rating.