Deadlight: Survival After Dark

Deliverable 2: Mid-Project Report

Group 16

Simranjeet Singh, Abdelrahman El Banna, Koroush Emari, Ashraf Esam Mahdi

CS4483B - Game Design

March 25, 2026

1. Level Design & Player Guidance

1.1 Spatial Layout

The game takes place in an abandoned urban town rendered from a top-down perspective. The map is structured as a bounded arena (approximately 48 x 38 units) enclosed by perimeter walls that prevent the player from leaving the play area. This confinement creates tension, as the player cannot simply flee from threats indefinitely.

The town is divided into distinct zones, each with a different strategic purpose:

- **Central Safe Zone:** The player spawns at the center of the map, surrounded by barricades that provide initial cover. This area is the safest starting point and serves as a natural home base.
- **Town Buildings:** Seven houses are scattered around the map at varying distances from center. These act as natural obstacles that break line of sight, create chokepoints, and provide cover during combat encounters.
- **Resource Zones:** During the day phase, health (red) and ammo (yellow) pickups spawn at random positions throughout the map, encouraging exploration. Pickups bob gently to draw the player's eye.
- **Perimeter Danger Zone:** Enemy spawn points ring the outer edges of the map. Enemies spawn at a distance of 12-20 units from the player, giving a brief window to react as they close in.

1.2 Visual Cues and Player Guidance

We use several visual strategies to guide the player without explicit tutorials:

- **Ground Paths:** Dirt/sand-colored paths cross the center of the map in a cross pattern, naturally guiding the player toward the four cardinal directions where resources and buildings are located.
- **Color Coding:** Health pickups are red, ammo pickups are yellow, and enemies have a greenish tint. This consistent color language lets players quickly identify objects at a glance.
- **Environmental Landmarks:** Trees, rocks, and crates are placed to create visual variety and help players orient themselves. Houses serve as major landmarks for navigation.
- **HUD Indicators:** A controls hint at the bottom right of the screen reminds players of available actions. Status messages fade in at screen center to communicate phase changes.
- **Camera Follow:** The camera smoothly follows the player with configurable smoothing, keeping the player centered while providing awareness of approaching threats.

1.3 Pacing and Challenge Balance

The day/night cycle creates a natural pacing rhythm. The 45-second day phase provides low-pressure time to explore and collect resources, while the night phase ramps up with wave-based enemy spawning. The day timer is displayed on screen so the player always knows how much preparation time remains. This alternation between calm exploration and intense survival prevents fatigue and maintains engagement across the full 5-night run.

2. Narrative Design & World-Building

2.1 Story Context

The player is a lone survivor separated from an evacuation convoy in a zombie-overrun town. Radio transmissions from EVAC Command provide the primary narrative thread: a rescue helicopter will arrive at dawn on the fifth day if the player can survive. This simple but compelling premise gives every night survived a sense of progress toward a concrete goal.

2.2 Radio Transmission System

At the start of each day phase, the player receives radio transmissions rendered in green terminal-style text at the bottom of the screen. Each night's transmissions are unique:

Night	Transmission Theme
1	Introduction: EVAC Command establishes contact, explains the 5-night timeline
2	Intel update: infected are evolving, shotgun availability mentioned
3	Deteriorating situation: new enemy types detected, assault rifle available
4	Massive surge warning: worst night yet, flamethrower unlocked
5	Final night: boss entity detected, everything on the line

Additionally, gameplay tips are displayed after the narrative transmissions (e.g., reminders about sprinting, reloading, and using cover). This contextual guidance helps new players without breaking immersion.

2.3 Environmental Storytelling

The abandoned town setting tells its own story through environmental details: boarded-up houses, scattered crates, and overgrown vegetation suggest a community that was hastily evacuated. The day/night lighting transitions reinforce the atmosphere, with warm daytime tones giving way to cold, dark nighttime colors that heighten tension.

3. Systems Design & Game Balance

3.1 Resource Systems

The game manages several interconnected resources:

- **Health:** The player starts with 100 HP. Damage from enemies reduces health, and reaching 0 triggers death. Health pickups (red circles) restore 25 HP each. The health bar uses a color gradient from green (full) to red (low) for immediate visual feedback. An invincibility frame of 0.5 seconds prevents chain-stagger deaths.
- **Ammo:** The player starts with a 15-round magazine and 60 reserve rounds. Ammo pickups add 30 rounds to reserves. The reload mechanic (1.2 seconds) creates a risk window where the player is vulnerable, adding tactical depth to combat.
- **Stamina:** Sprinting consumes stamina (shown as a blue bar below health). When depleted, the player cannot sprint until stamina regenerates. This prevents infinite kiting and forces players to manage their escape routes.
- **Points:** Earned from killing enemies (10 per kill) and surviving nights (100 + 50 per night number). Points are spent at the Dawn Phase shop for upgrades, creating a clear risk-reward economy.

3.2 Difficulty Scaling

Difficulty scales across two axes: per-night escalation and player-selected difficulty modes.

Parameter	Night 1	Night 3	Night 5
Enemies per wave	5	8-9	12+
Total waves	4	6	8
Enemy HP	50	70	100
Spawn interval	3.0s	2.0s	1.5s

Three difficulty modes (Easy, Normal, Hard) adjust enemy stats, resource availability, and score multipliers. Hard mode applies a 1.5x score multiplier to the leaderboard, rewarding skilled players while keeping Easy mode accessible for newcomers.

3.3 Combat Feel

We invested significant effort in making combat feel satisfying:

- **Screen shake** on shooting and taking damage provides visceral feedback.
- **Muzzle flash** sprites appear briefly at the fire point.
- **Hit particles** burst from enemies when bullets connect.
- **Death particles** scatter when enemies are killed.
- **Bullet trails** show projectile paths with a fading yellow-orange trail.

- **Damage flash:** A red overlay pulses across the screen when the player takes damage.
- **Enemy health bars** appear above damaged enemies so the player can prioritize targets.

4. Progression Systems & Rewards

4.1 Night-Based Progression

The game spans 5 nights of escalating difficulty. Each survived night advances the player to a Dawn Phase where they can spend points and access newly unlocked weapons. This creates clear, tangible milestones that motivate continued play.

Night	Milestone Unlock
1	Shotgun available in shop (100 points)
2	Assault Rifle available (200 points)
3	Grenade Launcher planned
4	Flamethrower planned
5	Victory - rescue helicopter arrives

4.2 Dawn Phase Shop

After surviving each night, the game pauses and presents a shop interface. Players can spend their earned points on:

- **Health Kit (50 points):** Fully restores health. Essential after tough nights.
- **Ammo Refill (30 points):** Adds 60 reserve rounds. Keeps the player combat-ready.
- **Shotgun (100 points):** Close-range powerhouse with 8 pellets per shot. Available from Night 1.
- **Assault Rifle (200 points):** Automatic fire with 30-round magazine. Unlocks at Night 2.

Shop buttons are grayed out when the player cannot afford an item or has not reached the required night, providing clear feedback about what is available and what to work toward.

4.3 Intrinsic and Extrinsic Motivation

We balance both types of motivation:

- **Extrinsic:** Points, weapon unlocks, score tracking, and the Game Over/Victory stat screens provide concrete rewards. The final score factors in kills, nights survived, and difficulty multiplier.
- **Intrinsic:** The satisfying combat feel (screen shake, particles, responsive controls), the tension of dwindling resources, and the narrative drive to reach rescue create internal motivation. Players want to survive not just for points, but because the game makes survival feel meaningful.

5. Testing & Iteration

5.1 Iterative Development Process

Our development followed an iterative approach where each feature was implemented, tested, and refined before moving to the next:

Issue Found	Solution Applied
Player rotated toward mouse (looked unnatural with pixel art)	Removed body rotation; bullets fire toward mouse cursor independently
Enemies required NavMesh (complex setup, failed at runtime)	Created SimpleEnemyAI with direct Rigidbody2D movement
Unity tags not defined (Enemy/Player tags caused crashes)	Switched to GameObject.Find() and component-based detection
Bullets passed through enemies (trigger vs collision mismatch)	Added both OnTriggerEnter2D and OnCollisionEnter2D handlers
No visual feedback on hits (combat felt hollow)	Added screen shake, muzzle flash, hit particles, damage overlay
Placeholder circle sprites (game looked unfinished)	Integrated Top-Down 2D RPG asset pack with proper sprites
No game flow (just endless spawning)	Built GameFlowController with Day/Night/Dawn phases and menu system

5.2 Playtesting Observations

Internal team testing over multiple sessions revealed several balance issues that were addressed:

- Day phase initially set to 180 seconds felt too long for demo purposes. Shortened to 45 seconds to maintain pacing while still allowing meaningful exploration.
- Enemy health of 50 HP made Night 1 too easy. Scaling now adds 10 HP per wave within each night, plus base increases per night number.
- Ammo scarcity was a frequent complaint. Added more ammo pickups during day (4 per day phase) and made the ammo shop item affordable at 30 points.
- Players were confused about when night would start. Added a visible day countdown timer that shows remaining seconds.

6. Technical Notes

6.1 Project Architecture

The project uses a modular architecture with clear separation of concerns:

Directory	Contents	Key Scripts
Scripts/Core	Game management, camera, effects	GameManager, GameFlowController, WaveSpawner, DayNightCycle
Scripts/Player	Player mechanics	PlayerController, PlayerShooting, PlayerHealth, Bullet
Scripts/Enemy	Enemy behavior	SimpleEnemyAI, EnemyHealth, EnemyAI
Scripts/Systems	Game systems	PointsSystem, ResourceManager, ProgressionManager
Scripts/UI	User interface	LiveHUD, GameUI, ShopUI
Scripts/Narrative	Story systems	NarrativeManager, DialogueUI, EnvironmentalLore
Scripts/Level	Level elements	LevelManager, MapZone, SpawnPoint, Obstacle
Scripts/Data	ScriptableObjects	WeaponData, EnemyData, NightConfig, DifficultySettings

6.2 Key Unity Components

- **ScriptableObjects:** WeaponData, EnemyData, NightConfig, and DifficultySettings are all ScriptableObjects, enabling data-driven design where game parameters can be tuned without code changes.
- **Singleton Managers:** GameManager, PointsSystem, ResourceManager, WaveSpawner, and GameEffects use the singleton pattern for global access while maintaining clean initialization order.
- **Custom Editor Tools:** DeadlightSetupWizard and LevelEditorTools provide one-click scene setup and rapid level element placement. The Deadlight menu in Unity's toolbar gives quick access to all tools.
- **Event-Driven Architecture:** Systems communicate through C# events (OnGameStateChanged, OnHealthChanged, OnAmmoChanged, etc.) rather than direct references, keeping components decoupled.
- **Programmatic Scene Setup:** TestSceneSetup builds the entire playable scene at runtime, loading sprites from Resources and constructing all game objects, UI, and managers programmatically.

6.3 Asset Integration

We integrated the free "Top-Down 2D RPG Assets Pack" from the Unity Asset Store, which provides character sprites (with 4-directional walk animations), NPC sprites used for enemies, environmental objects (houses, trees, rocks, crates), and ground tiles. The PlayerAnimator component switches between directional sprites based on the player's movement velocity, creating smooth 4-frame walk animations.

6.4 Team Contributions

Member	Primary Responsibility
Simranjeet Singh	Player systems, core architecture, integration, project setup
Abdelrahman El Banna	Enemy AI systems, wave spawning, difficulty balancing
Koroush Emari	Level design, environment layout, map zones
Ashraf Esam Mahdi	UI systems, audio, narrative/dialogue implementation