Interactive Systems Project

Bogdan Andrei, IVA

Îndrumător: Prof. Dorian Gorgan

Universitatea Tehnică din Cluj-Napoca

Table of Contents

1. Project Proposal

1.1 Project Domain

This project is a hybrid game combining a **card-based strategy game** with a **dungeon-crawling adventure**. The player engages in card battles against opponents, and if they lose, they must explore a dangerous dungeon, fighting enemies and collecting loot. The loot is then sold to enhance the player's card deck, allowing them to re-enter the card game with improved chances. The ultimate goal is to **defeat the final card game opponent**, concluding the game.

1.2 Functionalities

- **Card Game Mechanics:** Players engage in turn-based card battles, using strategy to defeat their opponent.
- **Dungeon Exploration:** Losing a card game forces players into a procedurally generated dungeon where they must battle enemies.
- **Loot and Economy System:** Items and gold acquired in the dungeon can be used to improve the player's deck.
- **Game Progression:** Winning the final card battle signifies game completion.
- **AI Opponent System:** Opponents in both the card game and dungeon will have adaptive behaviors based on difficulty.
- **Procedural Generation:** The dungeon layouts will be dynamically generated to provide varied experiences.

1.3 Target Audience

This game is designed for players who enjoy strategy-based card games and dungeon-crawling RPGs. The combination of both mechanics ensures an engaging experience for those who enjoy both luck-based and skill-based challenges.

1.4 Source of Inspiration

- *Risk of Rain 2: rouge like games*

- *Darkest Dungeon* – Dungeon exploration and loot-based progression.

- *Hearthstone* – Strategy-driven card duels.

1.5 Technical Specification

- Game Engine: Unity 3D
- Programming Language: C#
- 3D Models & Assets: Blender, Unity Asset Store
- Physics Engine: Unity's built-in physics system
- Input Methods: Keyboard
- Model: python, Pytorch, Pycharm

1.6 Evaluators

To evaluate this project, I will collaborate with 2 evaluators with technical backgrounds in the field of computers:

Silviu Grumazescu

Oana Morar

2. Solution Analysis and Specification

2.1 Task Description and Analysis

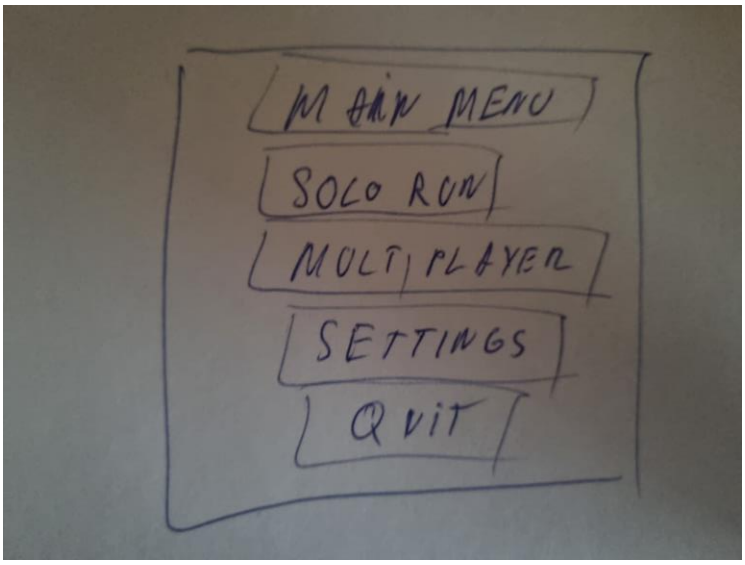
The game consists of two interconnected gameplay phases:

1. **Card Game Phase:**
    - Players construct a deck and play turn-based card battles.
    - Winning grants rewards such as gold, new cards, and progression.
    - Losing forces the player into the dungeon.
2. **Dungeon Phase:**
    - Players explore a procedurally generated dungeon, fight enemies, and collect loot.
    - Defeating dungeon enemies provides gold and equipment.
    - After escaping the dungeon, the player can sell loot and enhance their deck.
3. **Final Objective:**
    - Players must progressively improve their deck to defeat the final card game opponent.

2.2 Low-Fidelity Prototyping

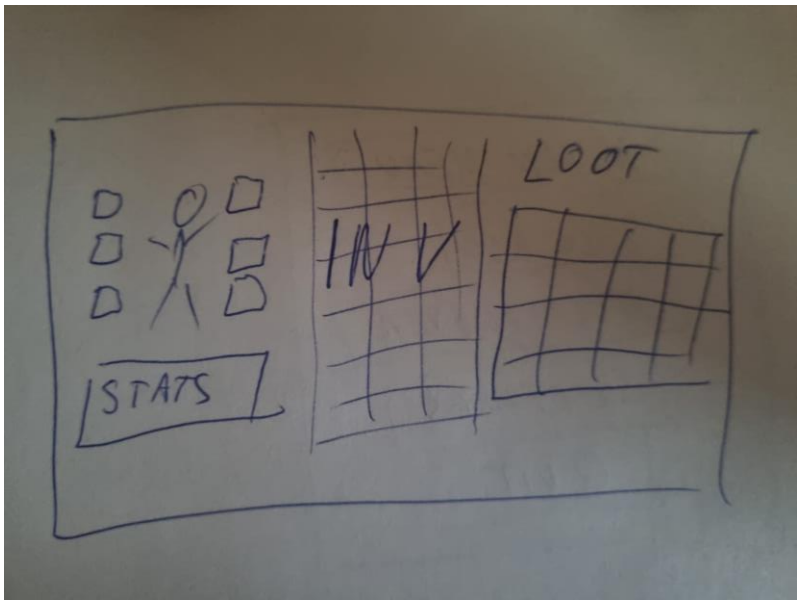
The prototype includes:

- **Menu** : Simple menu where the player selects the game mode and the settings



- **Basic Card Game UI:** A simple interface displaying cards, opponent moves, and the player's hand.

- **Dungeon Exploration Layout:** Top-down or side-scrolling dungeon navigation.
- **Initial AI Behavior:** Basic AI opponents in both the card game and dungeon phase.
- **Inventory System Prototype:** A simple grid-based inventory system for managing loot and resources.

- 

2.3 Game Scenarios

1. **Scenario: Playing the Card Game**
    - The player selects a deck and challenges an opponent.
    - They play strategically, using card abilities to outmaneuver the enemy.
    - If the player loses, they are transported to the dungeon.
2. **Scenario: Exploring the Dungeon**
    - The player moves through procedurally generated dungeon rooms.
    - They encounter enemies and must engage in combat.
    - Loot is collected and sold upon exiting.
3. **Scenario: Upgrading the Deck**
    - After a dungeon run, the player sells items and purchases stronger cards.
    - The improved deck is used for the next card battle.
4. **Scenario: Final Boss Battle**
    - The player challenges the ultimate opponent in the card game.
    - Winning this battle marks the completion of the game.

2.4. Game Mechanics and Technical Implementation

2.4.1 Card Game System

- Turn-based mechanics where players select and play cards.
- Different card types such as attack, defense, and special abilities.
- AI opponent with different strategies depending on difficulty.

2.4.2 Dungeon System

- Procedurally generated dungeons with various enemies using AI models.
- Combat system based on weapon attacks and spells.
- Hidden treasures and secret rooms.

2.4.3 Inventory and Economy System

- Players collect and manage loot.
- Gold and item-based economy for upgrading cards and abilities.

2.5. Development Plan

The development of the game will be divided into the following stages:

1. **Stage I (February - March 2025):** Project proposal, initial prototyping, and game scenario development.
2. **Stage II (March - April 2025):** Game mechanics design, UI development, and AI implementation.
3. **Stage III (April - May 2025):** Advanced AI behavior, dungeon system expansion, and card balancing.
4. **Stage IV (May - June 2025):** Testing, debugging, and polishing.
5. **Final Stage (June 2025):** Playtesting, final documentation, and game launch.

2.6 Report of evaluators

Q1: What type of additional content or features would you like to see in future updates or expansions?

Oana Morar: I find it interesting that the concept effectively combines strategic deck-building with dungeon exploration, creating a dynamic and complex gameplay loop.

Silviu Grumazescu: An interesting feature would be to let the player use these cards in the dungeon for special powers or buffs.

Q2: How do you find the proposed functionalities? What would you like to see implemented?

Oana Morar: The core functionalities are well-structured and cover essential aspects like combat, exploration, AI, and progression. I can really think of other functionalities, i already find the game captivating and captures the players attention because of the posibilities of approach various strategies.

Silviu Grumazescu: The game concept sounds like a very interesting mix of two genres that can complete each other well if there are mechanics that link the two together (for example the cards will have an effect in the dungeon)

Q3: What is your opinion about the profile of the final users?

Oana Morar: The target audience—players who enjoy strategy-based card games and dungeon-crawling RPGs—is well-defined.

Silviu Grumazescu: The features are overall exciting, card games usually lacking interesting ways of acquiring new cards, this concept bringing a new light to the genre

Q4: Any other observations or suggestions?

Oana Morar: Suggestion 1: Allow players to voluntarily enter dungeons for extra rewards, rather than restricting dungeon runs to losses. This would give players more strategic choices and allow risk-taking for greater progression. Suggestion 2: Implement two game modes: an online mode where players compete against opponents of varying skill levels, and an offline mode with AI opponents that match the player's skill level. The offline mode would serve as a training phase, helping beginners become familiar with the game flow and develop strategies before facing real players.

Silviu Grumazescu: I think that the dungeon exploration part should be a permanent option for the player, not only when he loses a card battle

3. Solution Analysis and Specification

3.1 Task Description and Analysis

The proposed game blends two main interactive systems: a card-based duel mechanic and a dungeon exploration mechanic. These components alternate based on the outcome of card battles and together form a unique gameplay loop.

In the first development stage, the primary focus is:

- Designing foundational card battle logic, including UI elements and card deck functionalities.
- Implementing basic player and enemy dungeon mechanics, with combat, loot collection, and procedural map generation.
- Creating a progression system that links dungeon success to improved card game performance.

Player Actions:

- Selecting a deck and playing a strategic turn-based card game.
- Entering the dungeon when defeated, fighting enemies, and collecting items.
- Returning to the card game with upgrades after selling dungeon loot.
- Building a deck strong enough to defeat the final card game opponent.

System Requirements:

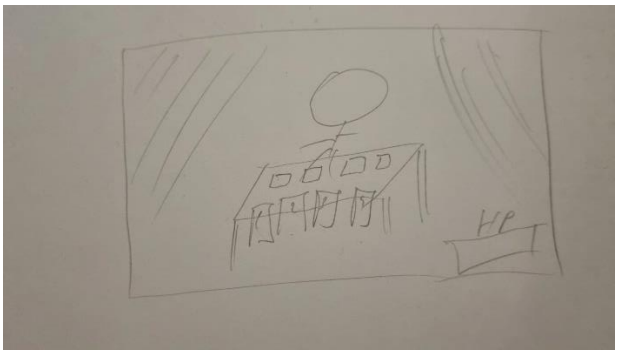
- Card battle framework with AI-controlled opponents.
- Dungeon generation and navigation mechanics.
- Inventory system for loot collection and selling.
- Economic system to allow upgrading cards using in-game currency.

3.2 Low-Fidelity Prototyping

A basic wireframe prototype was designed for both the card battle and dungeon modes:

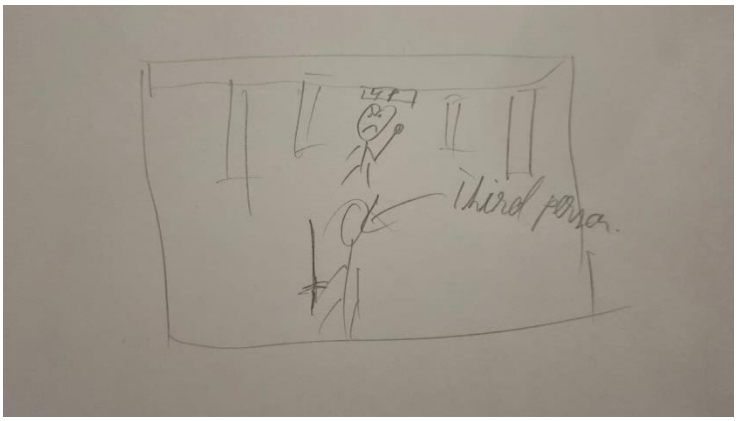
Card Game Interface:

- Displays player and enemy cards.
- Zones for active cards, draw pile, discard pile.
- Health indicators and action point counters.



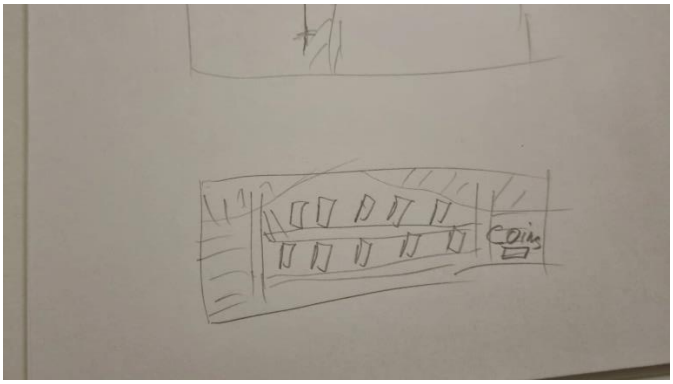
Dungeon Interface:

- Top-down layout for grid-based or side-scrolling dungeon maps.
- HUD showing current health, inventory, and minimap.
- Interaction feedback for combat and loot collection.



Upgrade Interface:

- Shop panel post-dungeon run.
- Upgrade options include new cards, stat boosts, and temporary buffs.



3.3 Game Scenarios

Scenario 1: Card Battle Victory

- The player successfully defeats the opponent in a card battle.
- They are rewarded with currency and new cards.
- They can choose to enhance their deck or face the next opponent.

Scenario 2: Card Battle Loss

- The player loses the card battle and is sent into the dungeon.
- They begin at the dungeon entrance and must navigate through enemies.

Scenario 3: Dungeon Raid

- The player explores rooms filled with enemies, traps, and treasure.
- Items include gold, equipment, and health potions.
- After reaching the exit, loot is tallied and sold.

Scenario 4: Post-Dungeon Upgrade

- Loot is sold, and players use earnings to buy new cards or strengthen existing ones.
- They return to the card game stronger than before.

Scenario 5: Final Card Duel

- The player faces the ultimate opponent.
- Strategic card play and deck enhancements are crucial.
- Victory marks the successful completion of the game.

---

4. Game Design

4.1 3D Object Scene and Visual Elements

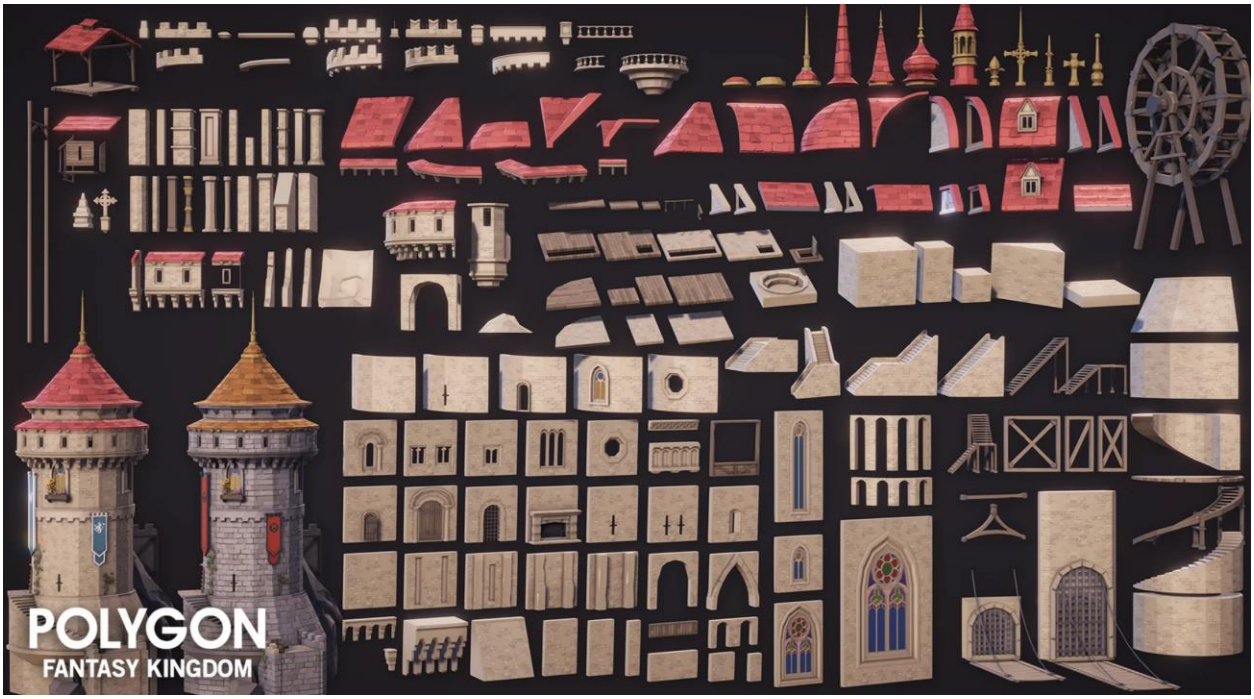
Card Game Scene:

- 2D interface with animated card flips, effects for attacks, and damage visualization.
- Backgrounds may represent the theme of the battle (forest, dungeon, castle, etc).

Dungeon Scene:

- Procedural tiles for walls, floors, and doors.
- Interactive elements: treasure chests, switches, traps, enemies.
- Visual indicators for player health and stamina.

Dynamic Objects:

- Enemies patrol or react to player presence.
- Animated effects for card abilities, attacks, and loot pickups.
- 

4.2 Game Strategy

Objectives:

- Win card duels.
- Gather resources in dungeons.
- Use rewards to upgrade your deck.
- Defeat the final boss.

Player Progression:

- Performance in dungeon affects card strength.
- Better decks lead to easier victories in future card games.
- More powerful enemies are introduced as the player progresses.

Victory Conditions:

- Each dungeon completion improves the player's arsenal.
- Final victory requires a fully developed and balanced deck.

4.3 Interaction Techniques

- **Card Game:** Drag and drop to play cards, click to activate abilities.
- **Dungeon:** Arrow keys or WASD for movement, mouse for interacting with items.
- **Combat:** Buttons for attack/defend during dungeon fights.
- **Inventory:** Drag items, click to use or sell.

4.4 AI and Difficulty Scaling

- Enemy AI adapts to player behavior over time.
- In card battles, AI uses predefined strategies that scale in complexity.
- In dungeons, enemy AI follows patrol paths, attacks when in range, and uses randomized tactics based on the player's approach.

Raportul evaluatorilor:

What do you think about the proposed tasks? Are the interaction modes clear enough?

Morar Oana: The proposed tasks are well-aligned with the game's dual-structure—combining card battles and dungeon exploration. Objectives are clear and contribute directly to player progression. Interaction modes are also clearly defined and appropriate for the respective scenes (e.g., drag-and-drop in the card game, WASD for dungeon movement). No ambiguities were found in control mechanisms.

Popescu Ionut: The proposed tasks are well-structured and effectively balance both card-based strategy and dungeon exploration, forming a strong gameplay loop. The interaction techniques are mostly clear—drag-and-drop for cards, keyboard movement for dungeons, and mouse-based interactions for combat and inventory are standard and user-friendly. These controls should be easy for players to understand and master, supporting smooth transitions between game modes.

What do you think about the prototyping of the game? Are there any elements that are extra, or missing?

Morar Oana: The prototyping includes core elements needed to support both the strategic card battles and exploration within dungeons. A few possible suggestions include:

A tutorial or onboarding process specifically designed to guide players through the combined mechanics of card games and dungeon exploration.

Additional details on how dungeon rewards translate into deck upgrades could further improve clarity.

Popescu Ionut:

The low-fidelity prototypes cover all essential areas, providing a good foundation for UI and scene layout. However, the dungeon interface could benefit from clearer visual feedback mockups, such as damage indicators or enemy alerts.

What is your opinion about the game scenarios? What can be improved?

Morar Oana: The game scenarios support a structured progression—from resource gathering and card dueling to upgrading decks and defeating the final boss. Potential improvements:

Providing clearer risk-reward cues, such as optional, higher-risk paths in dungeons with corresponding rewards.

Expanding on the dynamic relationship between dungeon performance and card strength to make the progression system even more engaging.

Popescu Ionut:

The scenarios create a solid narrative and gameplay progression. They guide the player logically through defeat, recovery, and eventual victory. However, they could benefit from more branching paths or consequences based on choices, such as optional side quests or different dungeon exits.

Technological and implementation considerations

Unity is the primary game development platform used for this project, offering an integrated environment ideal for building both 2D and 3D interactive systems. Its robust toolset enables the seamless creation of diverse game mechanics, including the card-based strategy elements and the procedurally generated dungeon system featured in the game. Unity's built-in physics engine and asset management streamline the implementation of dynamic gameplay and visual effects, while its support for C# allows for modular, scalable code architecture. Additionally, Unity's compatibility with tools like Blender and its extensive asset store accelerate the development of custom 3D models and interactive environments. Overall, Unity's versatility and efficiency make it a suitable choice for developing the complex, hybrid gameplay experience envisioned in this project.

Key reasons for choosing Unity include:

- **Cross-platform capabilities** – Easily deploy the game across multiple platforms with minimal configuration changes.

- **Enhanced rendering pipeline** – Delivers high visual quality for immersive environments and animated effects.
- **DOTS (Data-Oriented Technology Stack)** – Improves performance, especially for managing procedural dungeons and AI behavior.
- **UI Toolkit and Visual Scripting** – Speeds up interface development and allows rapid prototyping without deep coding.
- **Robust asset integration** – Seamlessly works with tools like Blender and supports the Unity Asset Store for quick asset sourcing.
- **Strong community and documentation** – Ensures access to tutorials, troubleshooting guides, and third-party support.
- **Scalability and modular architecture** – Supports structured, maintainable code ideal for a complex hybrid game system.

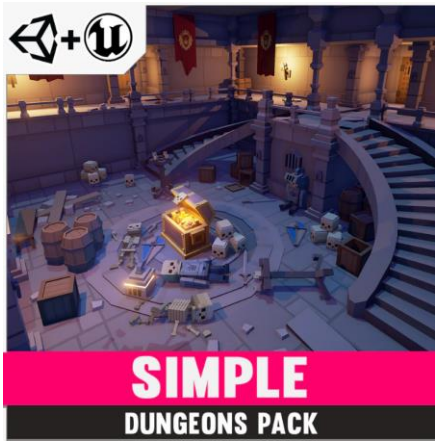
3D Objects imported from the Unity Asset Store Synty:

For playable characters icons: The POLYGON Fantasy Character Pack – Mini includes a diverse set of low-poly, stylized fantasy characters such as knights, mages, goblins, and villagers, all rigged and optimized.

For playable characters: A collection of low-poly, stylized fantasy NPCs and heroes, ideal for populating towns, quests, and story-driven scenes in Unity or Unreal projects.

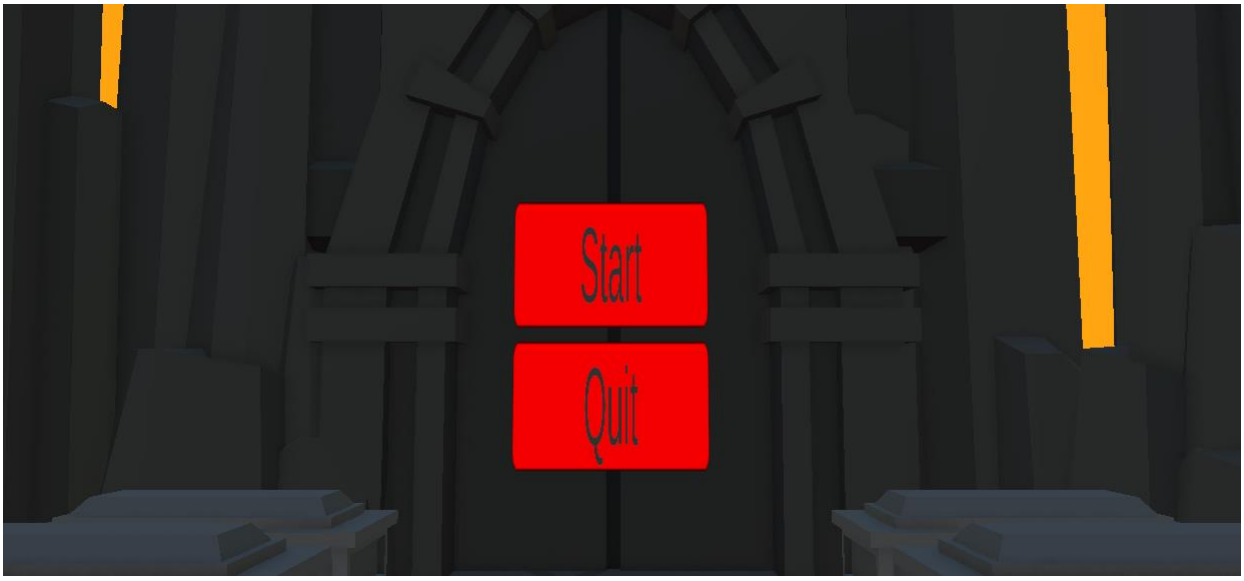
For enemies: A diverse set of detailed fantasy enemies and bosses, including ogres, golems, and dark mages, perfect for creating high-impact combat encounters.

For map design: A modular low-poly dungeon environment pack featuring rooms, stairs, props, and lighting setups suitable for RPG exploration and procedural generation.



Implementation of the game in Unity technology

The main menu was implemented in Unity as a fully integrated **3D world-space interface**, positioned within the game environment to enhance immersion from the moment the game launches. The menu features two primary interactive buttons—**Start** and **Quit**—designed as 3D objects rather than traditional 2D UI elements. These buttons are placed on a stylized dungeon gate, reinforcing the game's dark fantasy theme. Lighting and shadows are handled by Unity's **real-time rendering engine**, allowing the buttons to react naturally to the surrounding environment. The buttons are controlled using **colliders and C# scripts**, enabling interaction via mouse clicks or raycasting. This design not only serves as a functional user interface but also contributes to the visual storytelling of the game by blending seamlessly into the scene.



The intro scene is implemented in Unity as a **cinematic 3D environment** designed to establish the game's dark and mysterious tone from the outset. It features a large, stylized dungeon entrance with detailed architectural elements and dramatic vertical lighting strips on both sides. These glowing light sources are created using **emissive materials and real-time lighting**, casting warm highlights that contrast with the otherwise cold, stone structure. The scene uses Unity's **Standard Shader or URP (Universal Render Pipeline)** for physically-based rendering, ensuring realistic light behavior and shadow projection. Camera angles and lighting are carefully positioned to draw the player's attention toward the central gate, setting up the transition into the main menu or gameplay. This scene functions both as a visual hook and a loading environment, immersing players immediately and smoothly transitioning into the game's interface.

An animated NPC is used in the options menu to enhance visual engagement and thematic consistency within the game. The character—a stylized blacksmith holding a hammer—is placed next to an anvil, suggesting a workshop setting that aligns with a customization or settings context. This NPC is implemented using a **skinned mesh renderer** and an **Animator component** in Unity, enabling smooth skeletal animations such as idle movements, hammering, or reacting to user interactions. The glowing vertical lights in the background continue the game's lava-lit dungeon theme, and dynamic lighting enhances the character's visibility and realism. The hammer and anvil not only serve as environmental props but also visually reinforce the purpose of the options menu—as a place to "reforge" or adjust game elements. This interactive scene design transforms a standard menu into an immersive space, using **character animation and spatial UI** to elevate the player experience.

The first boss room was implemented in Unity as a standalone scene that serves as the entry point for the initial card battle. The environment is constructed using **modular 3D assets** imported from Blender and Unity Asset Store packages, enabling efficient layout and reusability. Static objects like furniture, props, and walls are marked as **static for light baking**, improving performance and visual quality through precomputed global illumination. The room uses **baked**

and mixed lighting, with green-tinted directional lights and point lights to establish mood and highlight key interaction zones, such as the boss table and cauldron.

The boss character is rigged and animated using Unity's **Animator Controller** with blend trees and trigger-based transitions, allowing for idle behavior and custom animations (e.g., reacting to card plays or winning/losing the round). The camera is configured using a **Cinemachine virtual camera**, ensuring smooth transitions and framing during the boss introduction and card battle phases. The scene also includes invisible **colliders and event triggers** to detect when the player enters the room, which then triggers a scripted sequence that initializes the card battle system via a custom C# manager class.

Card battle UI elements are rendered in **screen space overlay**, while the 3D background remains active to provide continuity between interaction and setting. Scene management is handled through Unity's **SceneManager API**, allowing the card battle system to run as an additive overlay or seamlessly transition from exploration to combat. This setup ensures scalability for adding future boss encounters while maintaining modular and clean code architecture.

Evaluation:

Q1: What is your opinion about the object scene? Are there any objects missing or that you would like to see included?

Morar Oana: The object scene is comprehensive and consistent with the game's theme. The use of modular assets and stylized models from the Unity Asset Store supports visual coherence.

However, the implementation could benefit from additional interactive props or environmental storytelling elements in the dungeon to enhance depth.

Popescu Ionut: What is your opinion about the object scene? Are there any objects missing or that you would like to see included?

Q2: Does the 3D implementation of UI elements (e.g., main menu, NPCs) enhance player immersion compared to traditional 2D menus?

Morar Oana: Yes, the use of 3D elements adds a lot of immersion by placing UI in the game world itself, rather than layering it over gameplay. This helps maintain narrative flow and visual consistency. It also gives players a stronger sense of presence in the game, especially when NPCs or environments react to interactions, creating a more engaging and less detached interface experience.

Popescu Ionut: Yes, the 3D implementation significantly enhances immersion. Embedding the main menu and option interfaces directly into the environment makes the interface feel like a natural part of the game world. Animated NPCs and environmental lighting contribute to storytelling and atmosphere, making even menu navigation feel engaging and in-universe.

Q3: Are the lighting, animation, and scene transitions implemented in Unity contributing to a cohesive and immersive gameplay experience?

Morar Oana: The current lighting and animations are visually effective and create the right mood for each scene, especially the intro and boss rooms. Scene transitions are functional, but could benefit from animation blending or short cinematic cuts to smoothen the shift between gameplay phases. Still, Unity's lighting system and real-time feedback are used well to maintain a cohesive visual tone.

Popescu Ionut: Absolutely. Real-time lighting and emissive materials set the mood for each scene, from the eerie intro environment to the tense boss room. Character animations and well-positioned cameras add life to NPCs and create dramatic transitions between gameplay phases. Unity's tools help tie these elements together into a consistent and immersive visual presentation.

5. Technological and Implementation Considerations (Continued)

Expansion of the Game Implementation in Unity Technology

During Stage IV, the focus shifted toward completing the feature set and refining the game using Unity. Emphasis was placed on enhancing the gameplay mechanics, expanding game content, improving user interface interactions, and preparing the system for formal evaluation. The game was developed entirely in Unity using C# as the primary scripting language. Integration of Unity Asset Store content and custom Blender models enhanced visual fidelity.

5.1 Card Game System Enhancements

a) New Card Types and Effects

To increase game complexity and depth, several new card types were introduced:

- **Basic Cards**: Simple damage, defense, or healing cards that form the foundational deck elements.
- **Gem-Enhanced Cards**: Cards infused with collectible gems from dungeons, boosting their abilities. For example, a Fire Gem adds burn effects, while a Frost Gem adds freeze effects.
- **Status Cards**: Cards that apply ongoing effects such as Burn (damage over time), Freeze (skip enemy turn), and Poison (progressive health reduction).
- **Area-of-Effect (AoE) Cards**: Cards that damage all enemy cards in play. Useful for crowd control during multi-enemy encounters.
- **Buff and Debuff Cards**: Cards that temporarily increase a player's card damage, reduce opponent attack power, or enhance defense.

Each card includes:

- **Tooltips**: Contextual descriptions and stats visible on hover.
- **Animations**: Particle effects, glow effects on rare cards.
- **Audio Cues**: Sound effects during casting, triggering, or resolving effects.

- **Metadata**: Including rarity (Common, Rare, Epic, Legendary), energy cost, type (Attack, Defense, Support), and upgrade level.

b) Improved AI Opponent Logic

The AI logic for enemy players in the card game was upgraded with the following capabilities:

- **Deck Analysis**: The AI adjusts its strategy based on the cards in the player's deck.
- **Bluffing and Delayed Actions**: AI may hold powerful cards for surprise rounds.
- **Difficulty Tiers**: Higher levels introduce predictive counterplay and combo prioritization.
- **Adaptive Learning**: AI tracks the player's most-used strategies and responds more effectively in repeated matches.

5.2 Dungeon Phase Expansion

a) New Enemies and Boss Fights

- **Tiered Enemy System**: Enemies are categorized (Grunts, Elites, Bosses), each with custom animations, sounds, and stats.
- **Mini-boss Encounters**: Rare rooms containing elite enemies with unique loot drops.
- **Final Boss Room**: Procedurally generated arena with cinematic intro, phase-based mechanics (e.g., invulnerability, summons).
- **Enemy AI Logic**: Uses Unity AI and state machines with transitions (Idle → Alert → Attack → Die).

b) Procedural Environment Upgrades

- **Room Templates**: Libraries of prefabs with random decorations.
- **Environmental Hazards**: Lava tiles, swinging blades, and floor traps randomly instantiated.
- **Dynamic Lighting**: Flickering torches, particle-based fog, ambient dungeon hum.
- **Secret Rooms**: Accessed via wall switches or destructible walls.

5.3 Loot and Resource System Improvements

a) Item Variety and Economy

- **Loot Categories**:
    - Weapons: Increase damage dealt in dungeon.

    - Spells: Grant temporary or permanent buffs to card mechanics.
    - Armor: Reduce damage taken or add resistance (e.g., poison, fire).
    - Components: Used for crafting or trading.
- **Drop System**:
    - Controlled via weighted RNG and modified by player's dungeon progression level.

b) Deck Upgrade System

- **Card Infusion System**: Dungeon items can be embedded into cards for passive upgrades (e.g., +1 attack, +10% crit chance).
- **Shop Enhancements**:
    - Buy/sell interface with filters and search.
    - Confirmation dialogs and preview system.

5.4 User Interface (UI) and User Experience (UX) Improvements

a) Interface Redesign

- **Card Game Screen**:
    - Responsive layout adapting to resolution.
    - Floating notifications for card effects.
- **Dungeon HUD**:
    - Timers for status effects and cooldowns.
    - Health and mana represented with stylized bars.
- **Menu and Settings**:
    - Language toggle.
    - Keybindings editor.
    - Volume sliders per audio category.

b) Accessibility Features

- Adjustable text size and contrast mode.
- Icon tooltips for visually impaired users.

5.5 Technical and Backend Work

- **Prefab System**: Modular prefabs for scalable object instantiation.
- **Audio Manager**: Layered audio system for ambient, combat, interaction.
- **Post-Processing**: Vignette, motion blur, and depth-of-field for immersion.
- **Performance Optimization**: Batching, occlusion culling, and GPU instancing.
- **Bug Tracker**: Internal JSON-based issue logger for QA testing.

6. Game Evaluation

6.1 Functional Correctness Evaluation through Cognitive Walkthrough

6.1.1 Scenarios

Scenario 1: Card Battle
In this scenario, the player engages in a tactical turn-based battle using their deck:

- **T1:** Select a card from the current hand
- **T2:** Activate a special ability targeting the enemy
- **T3:** End the turn and observe the outcome

Scenario 2: Dungeon Exploration
This scenario simulates navigating a procedurally generated dungeon:

- **T1:** Move through rooms using WASD controls
- **T2:** Avoid traps (e.g., blades, lava, gas)
- **T3:** Collect loot items scattered around the map

6.1.2 Scenario Evaluation

Scenario 1: Card Battle

1. Will the user understand what task to perform?
   Yes, the screen clearly displays the player's hand and the enemy, prompting an action.
2. Will the user recognize the correct interaction techniques?
   Yes, cards can be selected via mouse click or number key shortcuts.

3. Is appropriate feedback provided?
   Yes, animations, sound effects, and health bars respond to each move.
4. Is there a risk of errors?
   Minimal, invalid actions are automatically disabled.
5. Will users understand the system's feedback?
   Yes, card effects and enemy reactions are immediately visible.

Scenario 2: Dungeon Exploration

1. Will the user know what they are supposed to do?
   Yes, objectives are clearly indicated in the HUD, and enemy presence guides progression.
2. Will the user identify the correct controls?
   Yes, standard WASD movement and mouse interactions are intuitive.
3. Is visual/auditory feedback present?
   Yes, traps trigger animations, loot emits effects, and ambient sounds enhance context.
4. Is there a risk of incorrect interaction?
   Low, but user inattention may cause trap collisions.
5. Is the system feedback easy to interpret?
   Yes, status updates and interactions are clearly conveyed.

6.1.3 Conclusions and Recommendations

The cognitive walkthrough shows that the system is functional and intuitive. However, adding a quick-access guide for controls and game rules in the pause menu is recommended to support new players.

6.2 Usability Evaluation

The usability evaluation involved user observations, task performance tracking, and post-session feedback.

Participants: Two users with gaming experience participated in the session.

Tasks performed:

- Navigating a dungeon and avoiding traps.
- Selecting and playing cards during battles.
- Collecting loot and observing inventory changes.

Findings:

- Interface elements were clear and intuitive.
- Players appreciated immediate visual and sound feedback.
- Some users suggested adding a help overlay or tutorial on first play.

Recommendations:

- Add onboarding tips or tooltips for new players.
- Include visual hints for card synergy or effects.

6.3 Heuristic Evaluation

Heuristic	**Observations**
Visibility of system status	Information on HP, mana, and loot is always visible on the HUD.
Match between system and real world	Card mechanics and dungeon controls follow fantasy-world logic.
User control and freedom	Players can exit battles or dungeons at any time.
Consistency and standards	Consistent use of icons, colors, and visual language.
Error prevention	Invalid options are disabled, actions are confirmed.
Recognition rather than recall	Information is presented contextually with tooltips and labels.
Aesthetic and minimalist design	Clean UI, complementary color palette, minimal clutter.
Help users recognize, diagnose, and recover from errors	Clear messages accompany failed actions.

7. Development of the Improved Game Version

To enhance the game, the following changes were made:

- Added animations
- Introduced a list of all interaction techniques, accessible from both the main menu and the pause menu
- Other minor UI improvements (e.g., text size adjustments) and parameter tweaks
- Damage output balancing adjustments

8. Conclusions and Critical Assessments of the Game's Development, and Future Development

The game successfully combines card-based strategy with dungeon exploration, creating a cohesive and engaging gameplay loop. Core features such as the inventory, card upgrades, and dungeon system were implemented effectively, supported by a consistent visual style and intuitive controls.

Evaluation feedback confirmed the game's structure is clear and enjoyable, with suggestions focused on improving onboarding and giving players more freedom during progression. Overall, the project meets its initial goals and offers a strong foundation for future improvements.

8.1Future Development

Future development should focus on expanding the depth and replayability of the game. Key directions include:

- **Enhanced Replayability**: Introduce random events, alternate storylines, and multiple endings based on player choices.
- **Voluntary Dungeon Access**: Allow players to enter dungeons by choice, not only after defeat, to provide more agency and strategic options.
- **Multiplayer Mode**: Develop competitive and cooperative online modes where players can duel or explore dungeons together.
- **Advanced AI Behaviors**: Continue refining enemy and boss AI for more challenging encounters that adapt dynamically to player strategies.
- **Expanded Tutorial and Accessibility**: Implement an interactive tutorial and accessibility options (e.g., adjustable contrast, narration, key remapping) to support a wider range of players.

- **Additional Content**: Add new cards, enemies, dungeon biomes, and boss encounters to keep the experience fresh.

The foundation created in this version provides a scalable base for long-term updates and potential commercial release. Feedback-driven iterations will continue to guide improvements and ensure the game evolves to meet player expectations.

Source Files: https://github.com/TheWolfOfFarron/InteractiveSistemsProject.git