

Use cases

Use cases for developers (content creation)

Use case 1: Creating *and editing towers*

Actor: Game Developer

The goal: To create or edit a tower for use in the game

Prerequisites: The Tower Editor interface is open

The main scenario:

- The developer chooses the option "Create a new tower" or "Edit an existing tower".
- Enters the parameters of the tower: price, type of damage, rate of fire.
- Determines the available improvements for the tower by creating a chain or tree of improvements.
- Sets visual parameters: icons, animations, effects.
- Saves the changes.
- The tower becomes available for use on maps.

Alternative scenarios:

- The developer can remove or modify an existing tower upgrade.
- If the parameter is invalid (for example, a negative price), the system notifies the developer of the error.

Use case 2: Creating and editing enemy types

Actor: Game Developer

Objective: To create or edit the type of enemies to use in the game

Prerequisites: The Enemy editor interface is open

The main scenario:

- The developer chooses the option "Create a new enemy" or "Edit an existing enemy".
- Introduces enemy characteristics: life points, speed, defense.
- Adjusts the strengths and weaknesses of the enemy in relation to certain types of damage.
- Determines the loot that the enemy leaves when destroyed.
- Adds visual elements: enemy model, animations.
- Saves the changes.
- The enemy becomes available for use on maps.

Alternative scenarios:

- If no visual element is added, the system applies the standard model.

Use case 3: Creating a global technology tree

Actor: Game Developer

Goal: To create or edit a global technology tree

Prerequisites: The interface of the technology tree editor is open

The main scenario:

- The developer selects the option "Create a new technology tree" or "Edit an existing one".
- Defines the available tower upgrades and their dependencies.
- Sets requirements for unlocking upgrades (for example, the number of points or level progress).
- Adds descriptions of improvements and related effects.
- Saves the tree.
- The technology tree becomes available for use in the game.

Alternative scenarios:

- If the configuration of the improvements is invalid (for example, a cyclic dependency), the system notifies the developer of the error.

Use case 4: Creating and editing a map

Actor: Game Developer

Purpose: To create or edit a map

Prerequisites: The map editor interface is open

- **The main stream:**
- The developer chooses the option "Create a new map" or "Edit an existing map".
- Determines the location of the paths for the movement of enemies.
- Sets the zones where the player can build towers.
- Adjusts the objectives of the card: the amount of health that needs to be protected.
- Creates scenarios for the appearance of waves of enemies (including conditions for the player to manually launch waves).
- Defines the conditions for winning and losing (for example, destroying all enemies or protecting a target).
- Saves the map.
- The card becomes available for the player to select.

Alternative scenarios:

The developer can set up an endless game mode without a clear winning condition.

If the map violates the game rules (for example, there is no available path for enemies), the system reports an error.

Use cases for players (gameplay)

Use case 5: Choosing a card to play

Actor: Player

Goal: Select a card and start the game

Prerequisites: The player is in the main menu

The main scenario:

- The player enters the "Map Selection" menu.
- Selects one of the available cards.
- Presses the "Start the game" button.
- The game loads and the player lands on the selected map.

Alternative scenarios:

- If the card is blocked (for example, due to an incomplete technology tree), the system informs the player that the card is unavailable.

Use case 6: Gameplay on the map

Actor: Player

Objective: To play on the map, build and control towers

Prerequisites: The map is loaded

- The main scenario:
- The player sees the enemies moving around the map.
- Builds towers in accessible areas.
- Manages towers (selects targets, sets priorities).
- Launches a wave of enemies manually (if provided for by the rules of the map).
- Improves towers as resources accumulate.
- It waits for the completion of all waves or the fulfillment of the winning conditions.
- Receives a reward for successfully completing the map.
- Alternative scenarios:
- The player can temporarily disable the towers or redirect them to certain targets.
- If resources run out, the player cannot build or upgrade towers.

Use case 7: Saving a replay of the game

Actor: Player

The goal: To save a replay of the completed game

Prerequisites: The game is over

The main scenario:

- After completing the game, the player sees the option "Save replay".
- Presses the button to save the replay.
- The system saves the game data for replay.

Alternative scenarios:

- If an error occurred while saving, the system notifies the player and offers to try again.

Use case 8: Viewing a saved replay

Actor: Player

Goal: To view a replay of a previously completed game

Prerequisites: Saved replay is available

The main scenario:

- The player opens the "Replays" menu.
- Selects a previously saved replay.
- Starts replay playback.
- He watches the passage of the game in automatic mode.

Alternative scenarios:

- If the replay file is corrupted, the system notifies the player and prompts them to select another replay.
- These Use cases describe the main interactions for both developers who create content for the game and for players who are directly involved in the gameplay.