Fowler Style Use Cases

SomethingSomethingTacos©

**Case: Start the Game!** (example scenario)

**Scope:** EscapeesDevoid! Application

**Level:** User Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

* Developer: Wants player to have ease-of-access to play their game action quicker.
* Player: Wants to easily access settings and play game at their disposal.

**Preconditions:** Player has started the application.

**Main Success Scenario:**

1. Move mouse to “Start Game” button
2. Left-click on button, and then check the box for your difficulty setting by clicking on it as well!
3. Game Level will start!

**Extensions:**

2a. Choice of game modes can be EZ-PZ, Normal, and Extreme

**Miscellaneous:** This seems self-explanatory, and is an example.

**Case: Movement**

**Scope:** Escapees Devoid! Application

**Level:** User Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

* Developer: Wants player to have intuitive movement to enjoy game.
* Player: Wants to be able to understand movement easily and the movement be smooth and intuitive.

**Preconditions:** Player has successfully selected a difficulty and started the game.

**Main Success Scenario:**

1. Press a directional input button, W A S D or ↑ ← ↓ → , in those respective orders for Upwards, Leftwards, Downwards, and Rightwards movements.

**Extensions:**

1a. Not pressing any directional input will keep the player stationary, as expected.

**Miscellaneous:** Most games use this format, and it’s not terribly uncommon; buttons CAN be re-mapped by Unity, but these are the defaults. This affects in the XY plane as well, and does not deal with the Z-plane dimension.

**Case: Scoring Points / Shooting / Destroying Enemies**

**Scope:** Devoid! Application

**Level:** User Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

* Developer: Wants player to have a fair chance and fulfillment from their game.
* Player: Wants to have an objective and challenge while feeling fulfilled for completing it and playing the game.

**Preconditions:** Player has started the game successfully, and has a mouse / button mapped to LMClick.

**Main Success Scenario:**

1. Tapping the mapped button for shooting (default: LMClick) will spawn a bullet.
2. The bullet hits an enemy, and raises the score by a fixed number of points (currently 100), and removes the enemy from the scene.

**Extensions:**

2a. The bullet travels and misses an enemy, and then spawns once it’s traveled a set distance to avoid too many artifacts on the screen at once. Additionally, the score is raised by 25 if the enemy is avoided by the ship. Enemies get progressively more difficult to avoid / fight as the score counter goes up, making this challenging.

**Miscellaneous:** Bullet collision is detected and handled by unity, and most of this is done with little to no performance consideration since it’s so trivial. Bullets also continue at the momentum and angle the ship was at during shooting; this means they can travel at angles and speeds that are ahead of the path your ship will take, and allows for some interesting maneuverability and skill-shot tactics.

**Case: Taking Damage**

**Scope:** Escapees Devoid! Application

**Level:** User Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

* Developer: Wants player to have a challenge and experience drawbacks for failing to play the game to standards.
* Player: Wants to play game optimally while still have it being challenging; generally wants to avoid damage, but also wants it to pose a threat.

**Preconditions:** Player has started the game successfully and can move.

**Main Success Scenario:**

1. Player collides with enemy target or enemy’s projectile.
2. Player is knocked back by a set momentum calculation (i.e. however fast the projectile is moving) and the ship’s “integrity” is affected on a scalar model by this as well.

**Extensions:**

1a. The player doesn’t collide with the enemy or projectile and takes no damage.

**Miscellaneous:** This damage model makes it difficult to ascertain what exactly will be the remaining ship’s “integrity” as projectile speed increases with difficulty and time, and makes an air of uncertainty present.

**Case: Completing the Game**

**Scope:** Escapees Devoid! Application

**Level:** User Goal

**Primary Actor:** Player

**Stakeholders and Interests:**

* Developer: Wants player to have some sort of conclusion to game, and replay-ability.
* Player: Wants game to be fun and end/start fairly; usually wants replay-ability as well for more fun.

**Preconditions:** Player has started and played the game successfully.

**Main Success Scenario:**

1. The player’s points value reaches a certain threshold (10000 testing value) due to factors in the “Scoring Points” case.
2. The “Game Over” screen is shown, with your points value and options to return to title or play again.

**Extensions:**

1a. The player’s integrity reaches 0% due to some factor in “Taking Damage” case; the game over screen is shown, along with choices to exit to title or to play again.

**Miscellaneous:** More menus and functionality may be added later, but these are the basics for gameplay replay-ability and navigation.