

DISPATCH: Episode 1 — Pivot

Program Documentation (Functions and Story Explanation)

This documentation explains how each function in the program works and how it connects to the story of DISPATCH: Episode 1. The explanation will focus on both the technicality of the code and the narrative purpose of each scene, based on my own understanding of how the program runs.

1. General Program Flow

The program is a text-based visual novel game written in C++. The story follows Robert, a superhero whose choices slowly shape his identity, the game runs in a linear order but player decisions inside each scene create different outcomes.

Each major part of the story is written as its own function. These functions are called one by one in the Main function, which keeps the story organized and easy to follow.

2. Global Variables

The program uses global variables to allow decisions made in one scene to affect later scenes.

- `isMerciful` – Tracks whether Robert acts with mercy.
- `isRomanticTensionActive` – Tracks romantic connection with Blonde Blazer.
- `streetFightSuccess` – Records the outcome of the alley fight.
- `Blazer_Impression_Score` – Measures Blonde Blazer's perception of Robert.
- `Robert_Mercy_Rating` – Numerical value representing moral choices.
- `combatStyle` – Describes how Robert fights in the final battle.
- `publicReputation` – Represents how the public views Robert.

These variables are read later in the **epilogue** to determine the ending.

2. Main Function

The `main()` function acts as the backbone of the program. It controls the overall flow of the story.

How it works:

- Displays the game logo
- Shows a short introduction message
- Calls each scene function in the correct order
- Ends the program after the epilogue

From a story perspective, it also represents the full journey of Episode 1, from the opening moment to the final outcome.

3. Game Logo

This function prints an ASCII type logo of the game using multiple `cout` statements in order for it to be printed.

Story purpose:

- Gives the player a clear introduction before the story begins, as the logo meant to give a retro-style aesthetic

Technical role:

- It contains no conditions or inputs and only outputs text to the console

4. Scene Interrogation — Apartment Rooftop

Story Context

Robert interrogates the Soothing Goon while holding him over a rooftop, this will be the first moral test in the game and establishes Robert's direction.

Decision Paths

Choice 1: Pull him back

- Robert shows mercy
- `isMerciful = true`
- Mercy rating increases
- Reputation becomes Compassionate

Choice 2: Let him drop

- Robert acts ruthlessly
- Mercy rating decreases
- Reputation becomes Ruthless

Invalid Choice

- Represents hesitation
- Goon falls anyway
- Smaller penalty to mercy rating
- Reputation becomes *Unpredictable*

Technical Role

Uses `if–else if–else` statements to:

- Handle valid and invalid input
- Assign story consequences
- Store long-term moral impact

The choice influences later dialogue and the final epilogue.

5. Street Fight — Alleyway Confrontation

Story Context

Robert is ambushed by a thug. This scene tests instinct and reaction speed, not morality.

Decision Paths

Choice 1: Right Hand Punch

- Attack fails
- `streetFightSuccess = false`

Choice 2: Left Hand Punch

- Attack succeeds
- `streetFightSuccess = true`

Invalid Choice

- Represents hesitation
- Robert gets hit
- Failure is recorded

Technical Role

Stores the result in `streetFightSuccess`, which later affects how the streets remember Robert in the epilogue.

6. Scene Bar Flambae — Superhero Bar

Story Context

Robert was seen confronted by Flambae, he must act quickly before things get heated.

Decision Paths

Choice 1: Throw Alcohol

- Fire spreads violently
- High-risk decision
- `Blazer_Impression_Score` increases significantly

Choice 2: Throw Water

- Fire is extinguished
- Safer choice
- Smaller impression gain

Invalid Choice

- Program defaults to water
- Represents instinctive recovery
- Blonde Blazer notices composure

Technical Role

Demonstrates fallback logic, where invalid input is safely redirected into a reasonable narrative outcome.

7. Scene Billboard — Billboard Above the City

Story Context

Robert and Blonde Blazer share a quiet moment above the city as Blonde Blazer decides to get close with Robert, the player chooses whether to pursue a romantic connection or maintain emotional distance.

Decision Paths

Choice 1: Kiss Her

- Romance begins
- Impression score increases
- `isRomanticTensionActive = true`

Choice 2: Let the Moment Pass

- Emotional restraint

- No romance

Invalid Choice

- Awkward silence
- Slight penalty to impression
- Relationship weakens

Technical Role

Uses conditionals to track emotional decisions that affect relationship-based dialogue later.

8. Scene Combat Toxic — Final Encounter

Story Context

Robert faces Toxic in the climax of Episode 1, this scene reflects how Robert handles the situation.

Decision Paths

Choice 1: Punt Him Away

- Creative combat style
- Shows control and adaptability

Choice 2: Stomp Him Down

- Brutal approach
- Ends the fight decisively

Invalid Choice

- Hesitation
- Fight becomes dangerous
- Combat style marked as *Reckless*

Technical Role

Sets `combatStyle`, which directly affects the final summary.

9. Epilogue — Ending:

The epilogue summarizes who Robert has become based on all previous decisions.

How It Works

- Reads all global variables
- Uses multiple `if–else` statements
- Prints a personalized ending

Decision Influence

- Mercy affects leadership style
- Combat success affects street reputation
- Romance affects emotional legacy
- Impression score affects public opinion

This function ties the entire episode together.

10. Conclusion

Overall, the program successfully combines storytelling and programming logic. Each function represents a scene, while global variables allow decisions to carry weight throughout the game. Which is why each result is a clear example of how functions, conditionals, and variables can be used to create an interactive narrative.