

AshTracker: Wildfire Recovery Assistant - Project Report

AshTracker is a wildfire recovery assistant that helps users evaluate burned land plots and recommend suitable recovery actions based on terrain, what was burned, and fire severity. It outputs a recovery map and total cost estimate.

Objectives:

- Automate reforestation and rebuilding suggestions.
- Visualize affected areas in a grid.
- Estimate recovery costs based on terrain and damage type.

Wildfires leave long-term impacts on the environment and communities. AshTracker helps simulate and plan recovery efforts by applying decision logic using data structures and algorithms.

It's novel because it:

- Factors in fire severity and terrain compatibility.
- Suggests cost-effective recovery options.
- Emphasizes rebuilding strategy and budgeting, not just fire tracking.

Requirements:

- Python 3.x
- No external libraries required

To Run:

1. Clone or download the project from GitHub.
2. Open `ashtracker.py` in a Python IDE.
3. Run the script.
4. Enter plot data as prompted.

Main Components:

- Plot class: Stores land data.

- `suggest_recovery()`: Applies decision rules.
- `get_user_plots()`: Collects input from user.
- `display_grid()`: Prints map visualization.
- Cost calculator and print summary.

Functionality:

- Accepts user data for multiple plots.
- Recommends recovery action based on terrain, burn type, and severity.
- Displays results in a grid map.
- Calculates total recovery cost.

Example Test:

Input:

- Plot (0, 0), terrain: hill, burned: pine, severity: high
→ Suggests: `replant_grass`

Output Grid:

```
[RG] [BO]
[RS] [RC]
```

Total Cost Example:

$\$500 + \$700 + \$2000 + \$800 = \textbf{\$4000}$

- User input mimics real-world data.
- Outputs concrete recovery plans.
- Grid gives visual recovery plan layout.
- Total cost estimates help budgeting.
- Shows terrain-specific replanting logic.

Course Concepts Applied:

- **OOP**: `Plot` class and object handling.
- **Functions**: Used for modularity.
- **Dictionaries**: Used for terrain rules and cost mapping.
- **Control Flow**: Decision making with conditionals.

The AshTracker project effectively demonstrates how basic programming concepts can be applied to address real-world environmental challenges such as wildfire recovery. By combining user input with decision-making algorithms based on terrain, burn type, and severity, the tool generates meaningful and context-aware reforestation or rebuilding suggestions. Despite its simplicity, AshTracker models a realistic approach to planning recovery strategies and budgeting costs.

While the current version is text-based and uses simplified rules, it provides a strong foundation for future improvements, such as integrating satellite data, map-based interfaces, or machine learning for smarter suggestions. Overall, this project showcases not only our understanding of core programming principles but also our ability to apply them creatively and meaningfully in a real-world context.