

Development Log: Electronic Device Manufacturing Factory Simulation

Project Overview: A simulation of an electronic device manufacturing facility using Python and SimPy, designed to analyze production efficiency, workstation performance, and system bottlenecks.

Version 1.0: Initial Implementation

Date: February 13th, 2025

Changes: Initial version of the code

Core Structure:

- Created Workstation, Product, and Factory classes.
- Defined WorkstationStats, SupplierStats, and FactoryStats dataclasses for tracking metrics.
- Implemented sequential workflow with 6 workstations.

Key Features:

- Workstations process products with normal-distributed delays.
- Material resupply system with 3 automatic devices.
- Workstation failures and repairs (exponential repair time).
- 5% product rejection rate at final quality check.

Issues:

- Accidents caused unhandled `simpy.Interrupt` errors.
- Simulation crashed after accidents, leading to incomplete data.

Version 1.1: Accident Handling Fix

Date: February 18th, 2025

Changes: Changes to the logic of the accident method and the bottleneck delay

Accident Logic:

- Added `self.accident_occurred` flag to gracefully stop processes.
- Modified `generate_products()` and `accident()` methods to check the flag.
- Used `simpy.Interrupt` handling in `Product.process()` to log accidents without crashing.

Version 1.2: Final version of the code

Date: February 19th, 2025

Changes:

Refined Interrupt Logic:

- Wrapped `generate_products()` in a try-except block to handle interrupts.
- Ensured all processes check the `accident_occurred` flag before proceeding.
- Removed redundant `env.exit()` calls and replaced with flag-based termination.

Tools & Dependencies

Python Libraries:

`simpy` (discrete-event simulation).

`random` (stochastic modeling).

Development Environment:

PyCharm IDE.

Python 3.13.