

Project Title:**Factory Simulation Game: A Data Structure-Driven Approach****Objective:**

The objective of this project is to create a small factory simulation game using **Pygame**, integrating various data structures such as **Stacks**, **Queues**, and **Trees** to simulate production lines, resource management, and factory operations. The game will challenge players to efficiently manage factory operations, optimizing production cycles using interactive game mechanics. The project will demonstrate practical applications of the data structures learned in the course.

Description:

The factory simulation game will allow the player to manage a factory's production line, where raw materials are transformed into products and output through different stages. The game will have the following components:

1. Factory Production Line Simulation:

- The factory will consist of a series of stations where products are processed. Each station will perform a specific function, such as assembly, packaging, or quality check.
- Players will need to manage the flow of materials and products through these stations, ensuring that the factory is efficient.

2. Data Structures Implemented:

- **Queues:** A queue will manage the materials and products that need to be processed. Products will enter the queue when raw materials arrive, and when a product completes its stage, it will be dequeued and sent to the next station.
- **Stacks:** A stack will be used to store items temporarily at different stages of the production line, allowing for easy retrieval and undoing of certain steps, such as rolling back an error in the production process.
- **Trees:** A binary tree will be used to represent different production stations in the factory. Each node will represent a stage in production, and the tree structure will allow the factory to expand with different production stages and complexity.

3. Factory Operations:

- The player can adjust production settings, such as worker speed, production quotas, or raw material input.
- The game will simulate a series of production cycles, where the player must ensure all materials are processed efficiently.

4. User Interaction:

- The player will interact with the factory through a graphical interface using **Pygame**.
- Players will drag and drop materials into stations, monitor queues of materials and products, and adjust factory settings.
- The goal is to optimize production, reduce idle times, and maximize product output.

5. Functions and Classes:

- **Factory Class:** Handles the overall operation of the factory and controls the simulation.
- **ProductionLine Class:** Manages the queue and stack of materials and products.
- **Station Class:** Each station in the production line is represented by a node in a binary tree.
- **Queue Class:** Manages the material flow.
- **Stack Class:** Temporary storage for product parts.
- **BinaryTree Class:** Represents the production stations and their relationships.

Evaluation and Testing:

To evaluate the success of the project, the following criteria will be used:

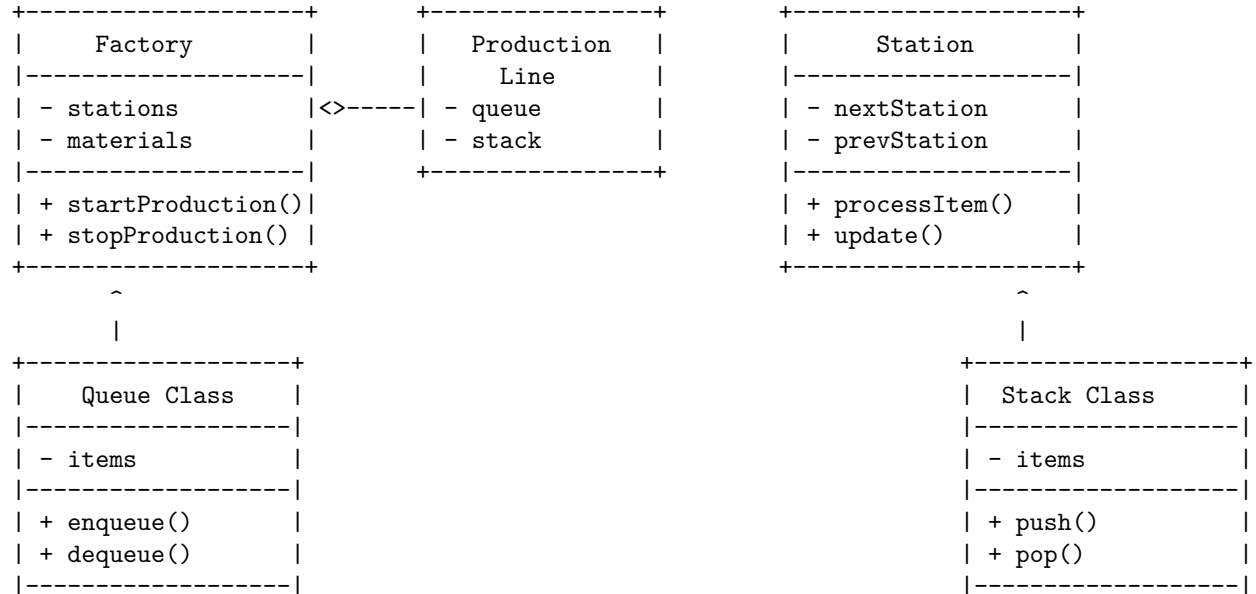
1. **Correctness of Data Structures:** Ensure that the **Queue**, **Stack**, and **Binary Tree** are functioning as intended for managing materials and production flow.
2. **User Interaction:** Check that the player can effectively manage the production line and interact with the game.
3. **Performance:** Test the game for smooth performance, even as the complexity of the factory increases.
4. **Edge Cases:** Handle edge cases, such as full queues, stacks, or when production is paused.

Testing will include:

- Unit testing for each data structure.
- Integration testing for the overall factory operations.
- Manual testing by running the game and identifying any issues in production flow.
-

Diagram:

UML Class Diagram:



Timeline:

Week	Task	Details
Week 1	Research and Design	Design the structure of the game and the data structures used.
Week 2-4	Implement Factory, ProductionLine, and Queue	Set up basic game framework using Pygame and create the Queue structure.
Week 4-7	Implement Game Mechanics	Integrate user interaction and game logic. Implement material processing and production cycles.
Week 7	Testing and Debugging	Test game flow, data structure functionality, and fix any bugs.
Week 8	Finalizing Features and Optimization	Optimize game performance and improve user interface.
Week 9	Final Report and Presentation Preparation	Prepare the final report and presentation for Week 10.