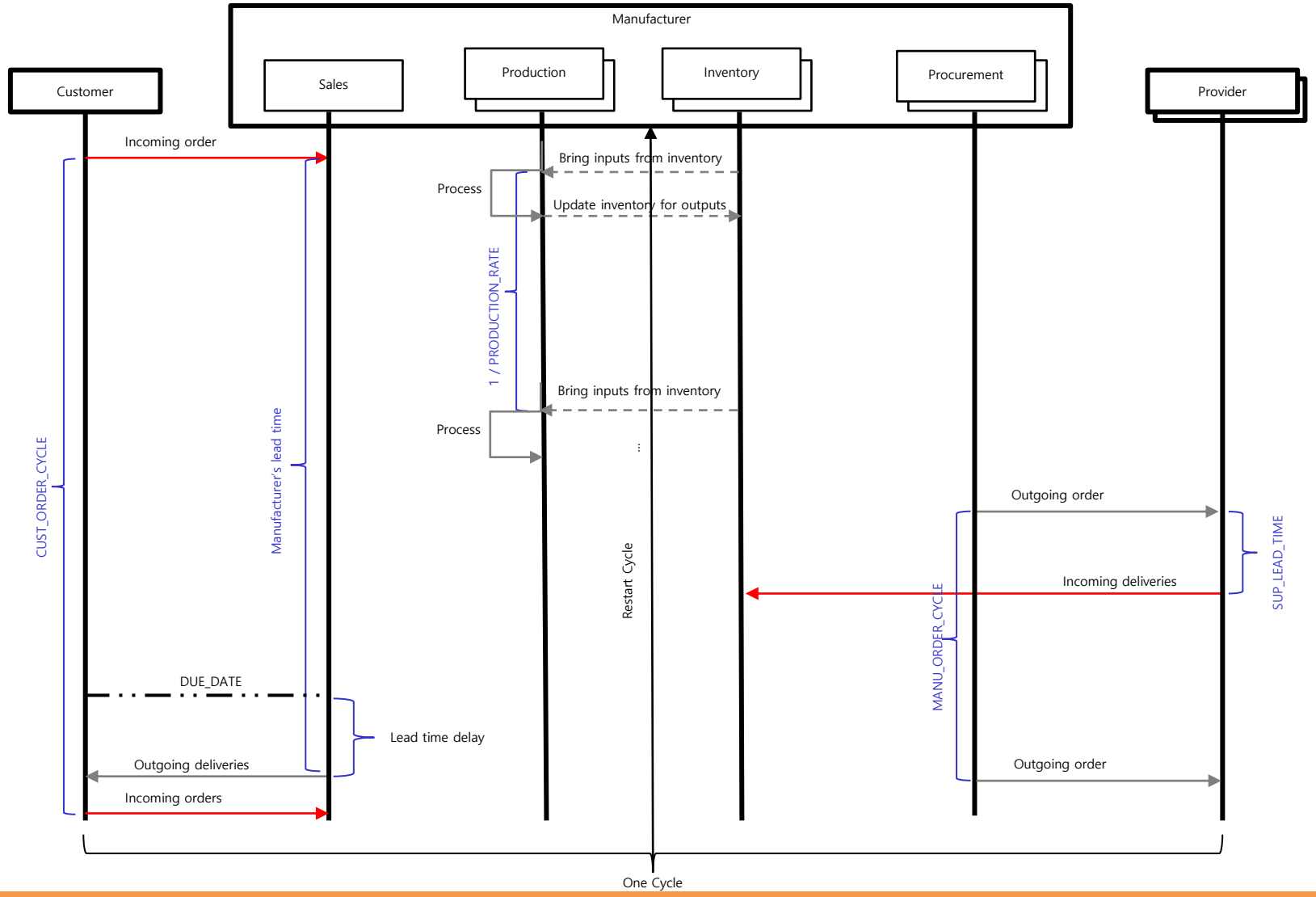
SimPy 기반 재고 운영 관리 시뮬레이터

목 차

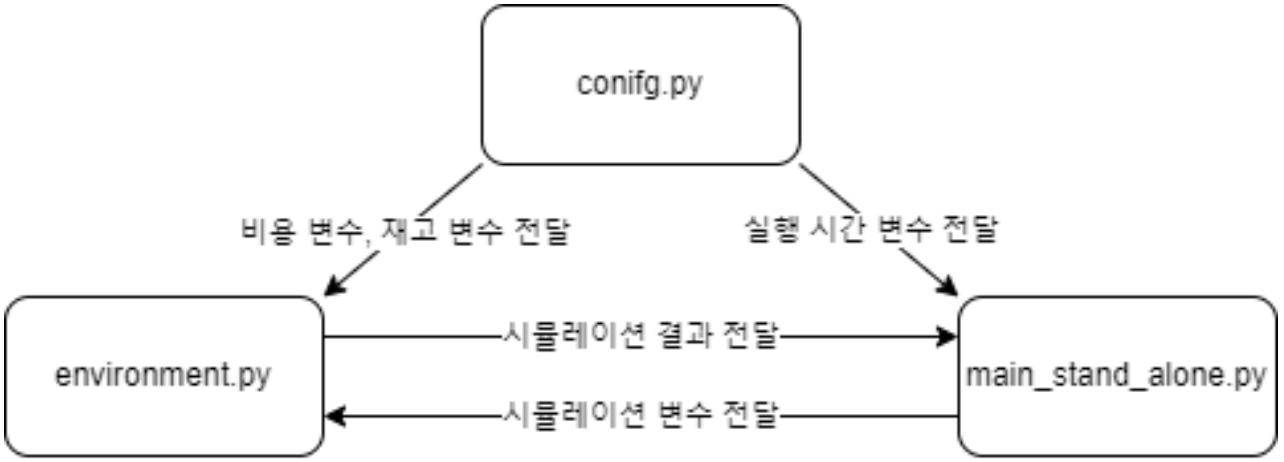
1. SimPy 기반 시뮬레이터 소개

- **SimPy**
  - Python으로 작성된 이벤트 기반 시뮬레이션 라이브러리
  - 복잡한 시스템이나 프로세스를 모델링 및 시뮬레이션
  - 시간의 경과와 다양한 이벤트에 따라 시스템의 동작을 모델링
- **SimPy를 이용한 시뮬레이터 설계**
  - 재고관리 문제는 주문, 배송, 재고 변화 등의 이산적인 사건으로 구성
  - SimPy 시뮬레이션을 활용해 시간을 관리하고 통제하여 시간에 따른 시스템의 변화를 모델링 가능
  - SimPy의 활용으로 다양한 시나리오 조건을 쉽게 구현하고 변경
  - 파이썬 기반 알고리즘으로 다른 파이썬 라이브러리, 도구들과 통합 가능하여 강화학습 알고리즘의 적용 용이

2. Sequence Diagram



3. 주요 파일 설명



- config: 시뮬레이션에 필요한 변수들을 설정하는 파일
- environment: 시뮬레이션의 환경을 설정하고 실행하는 파일
- main_stand_alone: 시뮬레이션의 시간을 진행시키고, 결과를 출력을 하는 파일

4. envinorment.py 클래스 설명-Inventory

- Inventory: 재고의 업데이트와 보유 비용의 계산을 다루는 클래스
  - _cal_holding_cost: 보유 비용을 계산하는 함수
    - 보유 재고*보유기간*단위당 보유비용
  - update_inven_level: 재고를 업데이트하며, 보유 비용을 업데이트하는 함수
    - 현재 보유 재고 + 업데이트 재고
      - 한계 재고량 보다 크면 현재 보유 재고량을 한계 재고량으로 설정

4. envinorment.py 클래스 설명-Provider

- Provider: 제조 회사에 자재를 배송하는 클래스
  - deliver_to_manufacturer: 제조사에 자재를 배송하는 함수
    - N일 뒤에 보유 자재에 자재량을 업데이트

4. envinorment.py 클래스 설명-Procurement

- Procurement: 자재를 구매하는 클래스
  - _cal_procurement_cost:
    - 단위당 자재 구매비용 * 주문량 + 준비 비용
  - order_material: 주문량을 입력받는 함수

4. envinorment.py 클래스 설명-Production

- Production: 제품을 생산하는 클래스
  - _cal_processing_cost: 제품을 생산하는데 드는 비용을 계산하는 함수
    - 단위당 제조 비용 * 제조 시간
  - process: 주문량을 입력 받는 함수
    - 자재가 부족하면 종료
    - 24/production rate마다 제품 생산

4. envinorment.py 클래스 설명-Sales

- Sales: 제품을 고객에게 전달하는 클래스
  - _cal_selling_cost: 제품을 생산하는데 드는 비용을 계산하는 함수
    - 단위당 배송 비용 * 배송량 + 준비 비용
  - _cal_penalty_cost: 주문을 충족 시키지 못했을 경우 비용을 부과
    - 부족한 제품 수 * 미달 비용
  - deliver_to_cust: 제품을 고객에게 전달하는 함수
    - 고객의 주문이 충족되었는지 여부 판단, 고객에게 제품을 전달

4. envinorment.py 클래스 설명-Customer

- Customer: 고객이 제품을 주문하는 클래스
  - order_product: config.py에서 주문량을 가져와서 저장하는 함수

4. envinorment.py 클래스 설명-Customer

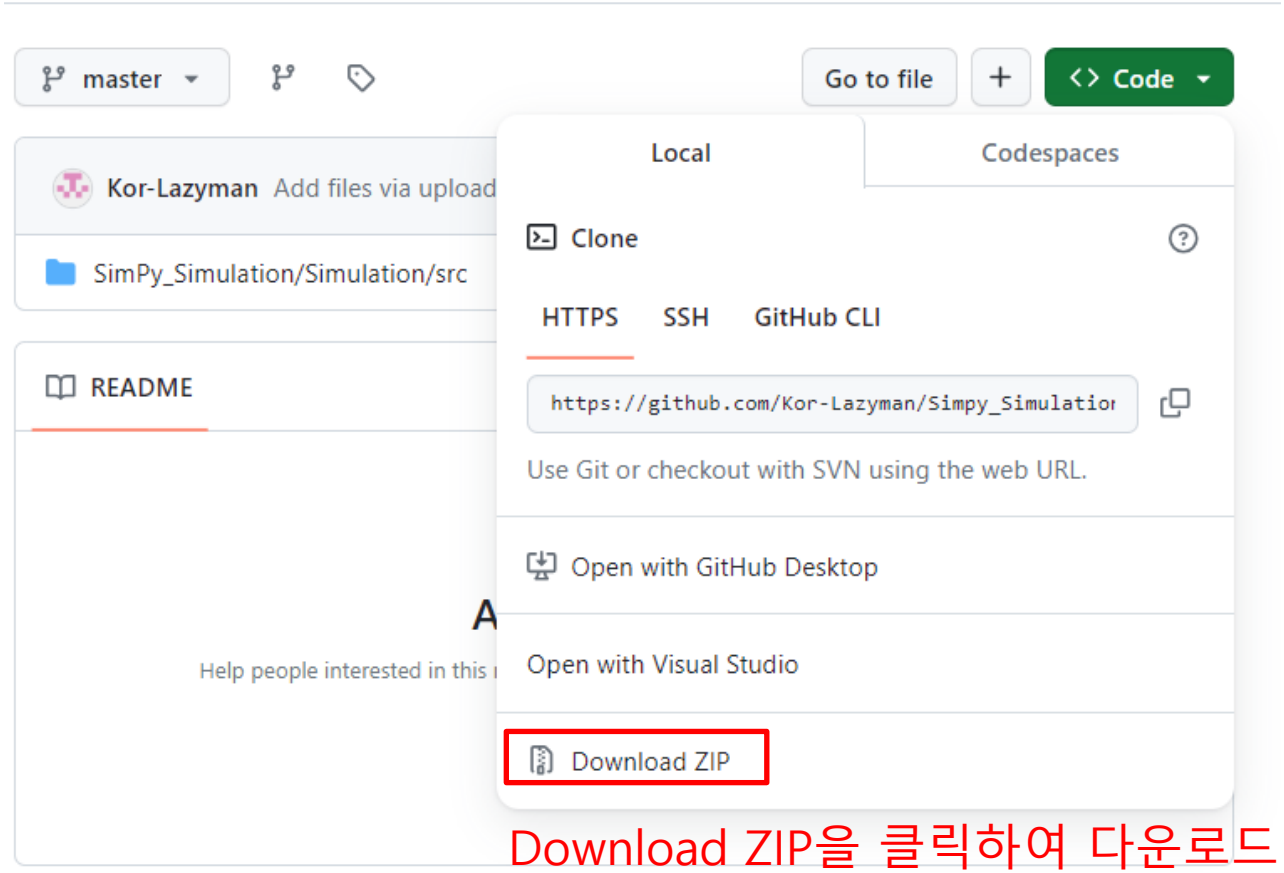
- Customer: 고객이 제품을 주문하는 클래스
  - order_product: config.py에서 주문량을 가져와서 저장하는 함수

4. envinorment.py 클래스 설명-Customer

- Customer: 고객이 제품을 주문하는 클래스
  - order_product: config.py에서 주문량을 가져와서 저장하는 함수

5. 설치 방법 및 실행법

1. Github에서 시뮬레이션 코드 다운로드



Download ZIP을 클릭하여 다운로드

5. 설치 방법 및 실행법

2. SimPy 설치: Anaconda 프롬프트에서 pip install simpy를 입력

```
(base) C:\Users\User>pip install simpy
Collecting simpy
  Obtaining dependency information for simpy from https://files.pythonhosted.org/packages/48/72/920ed1224c94a8a5a69e6c12
75ac7fe4eb911ba8feffddf469f1629d47f3/simpy-4.1.1-py3-none-any.whl.metadata
  Downloading simpy-4.1.1-py3-none-any.whl.metadata (6.1 kB)
Downloading simpy-4.1.1-py3-none-any.whl (27 kB)
Installing collected packages: simpy
Successfully installed simpy-4.1.1
```

5. 설치 방법 및 실행법

3. 위치 이동: Anaconda 프롬프트에서 cd ["시뮬레이션 파일 경로"] 입력

```
(base) C:\Users\User>cd C:\Users\User\Desktop\SimPy_Simulation\Simulation\src

(base) C:\Users\User\Desktop\SimPy_Simulation\Simulation\src>
```

4. 파일 실행: Anaconda 프롬프트에서 python "main_stand_alone.py" 입력

```
(base) C:\Users\User\Desktop\SimPy_Simulation\Simulation\src>python "main_ stand_alone.py"
```