

결과 보고서

20220703 유민하

태피 상점의 큐 시뮬레이션



구현 결과 요약

1.

실행 가능

2.

다중 파일 프로그래밍

3.

입력 처리

4.

시뮬레이션 진행

5.

출력 처리

6.

여러 번 실행 및 분석

7.

창구 개수 실험

8.

전화 서비스

9.

GUI 프로그래밍

*

배속 기능

*

다중 쓰레드



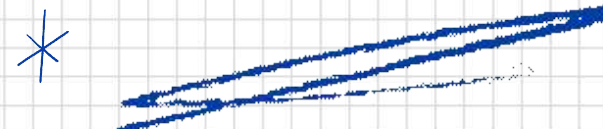


서론

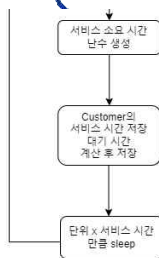
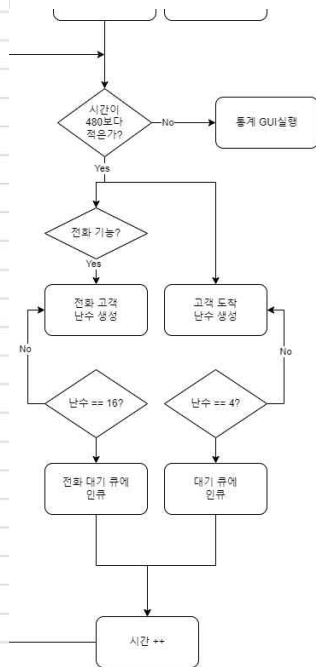
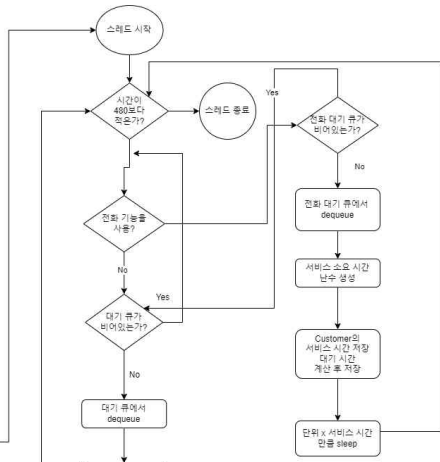
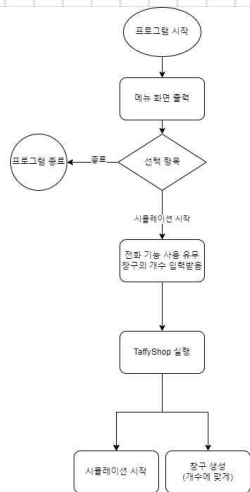
목적

태피 상점의 운영 시뮬레이션

고객 서비스 및 대기시간을 관리와 최적화하기 위한 프로그램



Flow Chart



시스템 구성

Data

- Customer
- Queue

RunSim

- Counter
- TaffyShop

GUI

- startMenu
- selectCounter
- StaticsGUI
- SimulationGUI



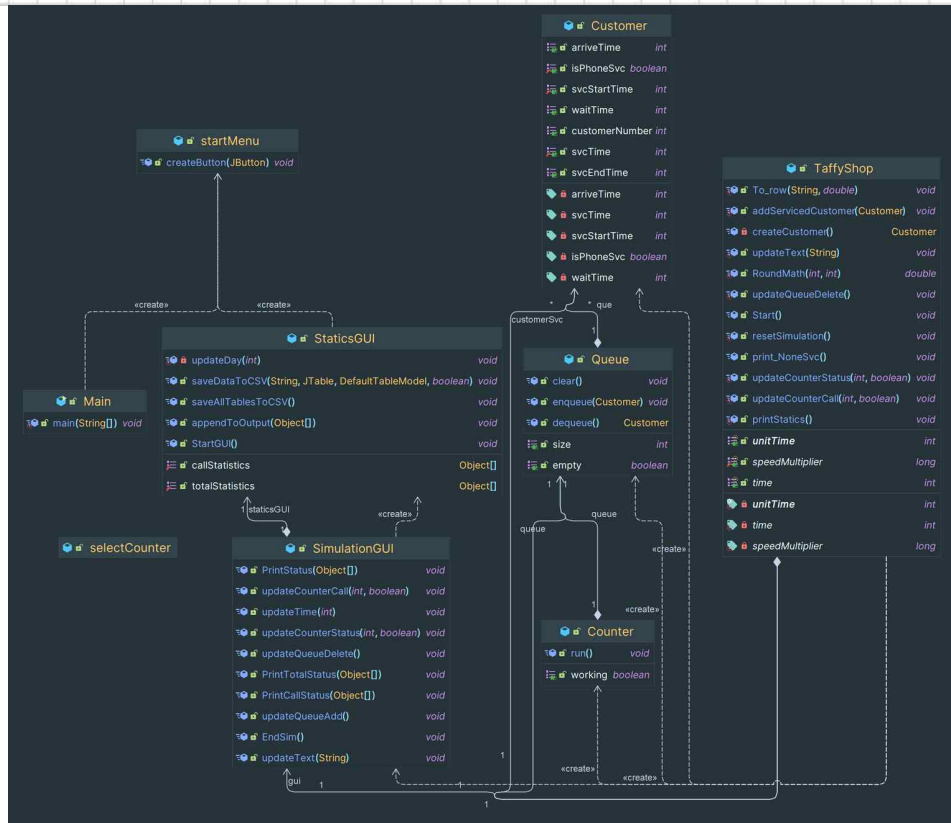


자료구조

```
public class Queue {  
    6 usages  
    private LinkedList<Customer> que;  
    2 usages  
    public Queue() { que = new LinkedList<>(); }  
    2 usages  
    public synchronized void enqueue(Customer data) // 0은 정상적으로 인큐, 1  
    {...}  
    2 usages  
    public void clear() { que.clear(); }  
    2 usages  
    public synchronized Customer dequeue()  
    {...}  
    2 usages  
    public synchronized int getSize() { return que.size(); }  
    6 usages  
    public synchronized boolean isEmpty() { return que.isEmpty(); }  
}
```

```
public class Customer { // 고객 정보 저장  
    3 usages  
    private int arriveTime; // 도착 시간  
    5 usages  
    private int svcStartTime; // 서비스 시작 시간  
    3 usages  
    private int waitTime; // 대기 시간  
    5 usages  
    private int svcTime; // 서비스 소요 시간  
    2 usages  
    private int Num; // 고객 번호  
    3 usages  
    private boolean isPhoneSvc;  
    1 usage  
    private static int customerCount=0; // 고객 수, 고객 번호 부여에 사용  
  
    1 usage  
    public Customer(int arriveTime) {  
        this.Num++; customerCount;  
        this.arriveTime = arriveTime;  
        this.svcStartTime=0;  
        this.waitTime=0;  
        this.svcTime=0;  
        this.isPhoneSvc=false;  
    } // 생성자
```

모듈 설계



주요 알고리즘

```
} else if (!call.isEmpty() && !isWork) // 일을 하지않고 전화 대기 큐에 사람이 있으면 진행
{
    Customer customer = call.dequeue(); // 고객 디큐해오고
    TaffyShop.updateQueueDelete(); // 큐 GUI의 상태 업데이트

    if (customer != null) {
        this.isWork = true;
        TaffyShop.updateText("창구 " + ID + " | 전화 서비스 수행 중");
        ↴
        TaffyShop.updateCounterCall(ID, isWork);
        customer.setSvcStartTime(TaffyShop.getTime()); // 서비스 시작 시간을 설정
        int callSvc = random.nextInt( bound: 10) + 1;
        customer.setSvcTime(callSvc);
    }
}
```



```
if (customer.getSvcEndTime() ≥ 480) { // 전화 서비스 종료 시간이 480분을 넘어가면 480분이 될 때 종료 설정
    int remainTime = 480 - customer.getSvcStartTime();
    if (remainTime > 0) {
        int SvcTime = remainTime - 1;
        customer.setSvcTime(SvcTime);
    } else {
        customer.setSvcTime(0);
        break;
    }
}

try {
    Thread.sleep( millis: unitTime * (callSvc) / TaffyShop.getSpeedMultiplier());
} catch (InterruptedException e) {
    e.printStackTrace();
}

TaffyShop.updateText("창구 " + ID + " | 전화 서비스 종료");
TaffyShop.addServicedCustomer(customer);
this.isWork = false;
try {
    Thread.sleep( millis: unitTime / TaffyShop.getSpeedMultiplier());
} catch (InterruptedException e) {
    e.printStackTrace();
}
}
```

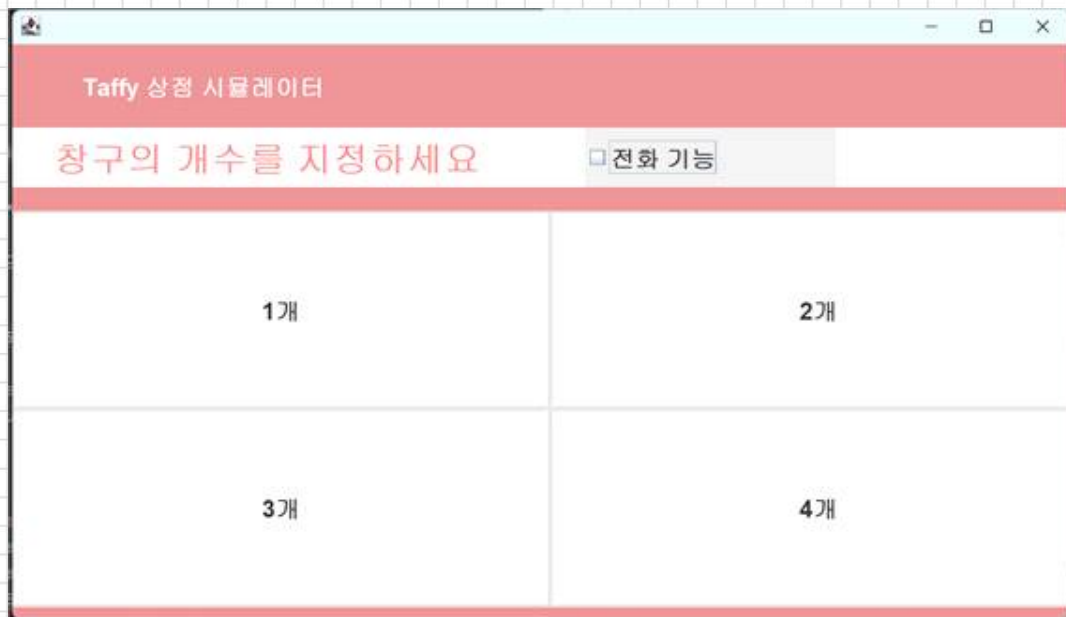
실행화면



메뉴 화면



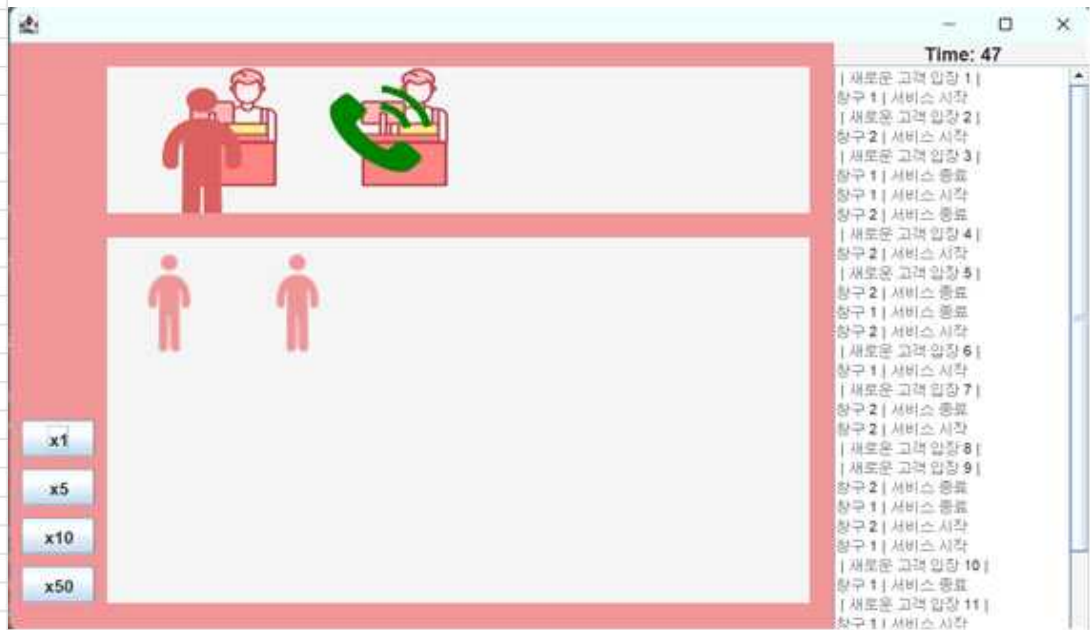
실행화면



- 창구 선택
- 전화 기능 사용
유무 선택 화면



실행화면



시뮬레이션 시작

- 전화 기능 사용 체크

- 창구 2개 선택
- 배속 기능이 있는 버튼
(1~50배속)

: 1= 실제 시간의 100 배속

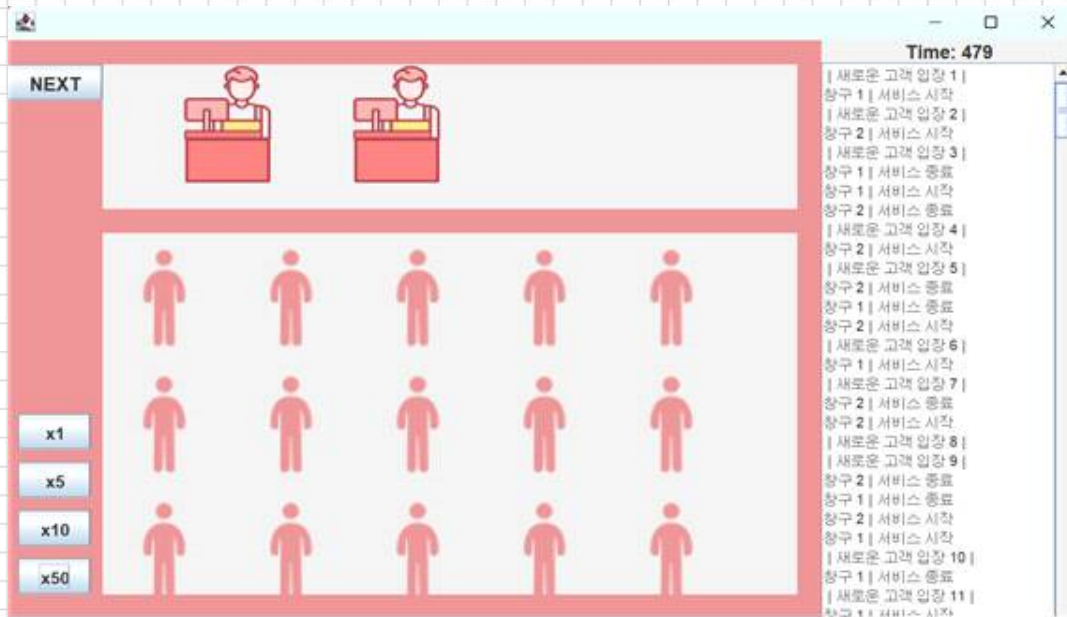
상황에 맞는 GUI 표현

창구

대기 큐 표현



실행화면



시뮬레이션 종료

- 종료되면 Next 버튼 활성화
: 통계 자료 화면으로 이동
- 대기 큐에 많은 손님이 남아있는 모습



실행화면

고객 개인 통계				
고객 번호	도착 시간	시작 시간	대기 시간	서비스 시간
1	4	4	0	8
2	5	5	0	9
4	14	14	0	2
3	7	9	2	8
5	16	18	2	9
7	26	27	1	6
6	25	25	0	9
9	33	34	1	7
8	32	34	2	10
10	40	42	2	6
11	42	46	4	3
12	43	50	7	7
14	44	58	14	5
15	46	60	14	5
17	52	66	14	3
16	51	64	13	7
19	55	74	19	2
20	56	77	21	4
21	58	79	21	10
22	59	84	25	5
23	60	88	28	5

전체 통계	
통계 항목	값
평균 대기 시간	24.646
평균 서비스 시간	4.402
평균 전화 서비스 시간	4.950
총 서비스 시간	361.000
총 전화 서비스 시간	99.000
서비스받은 총 고객 수	82.000
전화서비스 총 고객 수	20.000

전화 서비스 통계				
고객 번호	발신 시간	수신 시간	대기 시간	서비스 시간
13	43	45	2	9
18	54	55	1	5
28	70	72	2	1
26	68	70	2	8
32	81	82	1	1
36	99	100	1	4
44	123	129	6	5
48	131	135	4	2
52	153	155	2	7
54	159	163	4	3

통계 화면

- 일반 서비스 고객 정보
- 전체 통계
- 전화 서비스 고객 정보

: 전화 서비스 활성화 시에만 데이터가 작성된다.



실행화면

callStatics.csv

statics.csv

totalStatics.csv

1	통계 항목	값	
2	평균 대기 시간	24.646	
3	평균 서비스 시간	4.402	
4	평균 전화 서비스 시간	4.95	
5	총 서비스 시간	361	
6	총 전화 서비스 시간	99	
7	서비스받은 총 고객 수	82	
8	전화서비스 총 고객 수	20	
9	서비스를 받지 못한 고객	62	
10	통계 항목	값	
11	평균 대기 시간	8.256	
12	평균 서비스 시간	4.248	
13	평균 전화 서비스 시간	5.043	
14	총 서비스 시간	548	
15	총 전화 서비스 시간	116	
16	서비스받은 총 고객 수	129	
17	전화서비스 총 고객 수	23	
18	통계 항목	값	
19	평균 대기 시간	1.565	
20	평균 서비스 시간	4.256	
21	평균 전화 서비스 시간	5.152	
22	총 서비스 시간	715	
23	총 전화 서비스 시간	170	
24	서비스받은 총 고객 수	168	
25	전화서비스 총 고객 수	33	

출력

- 3가지 테이블이 따로 csv파일로 작성된다.
- 여러 번 시뮬레이션이 동작했을 때 아래에 표를 추가하는 방식으로 구현함



실험 결과

통계 항목	값
평균 대기 시간	24.646
평균 서비스 시간	4.402
평균 전화 서비스 시간	4.95
총 서비스 시간	361
총 전화 서비스 시간	99
서비스받은 총 고객 수	82
전화서비스 총 고객 수	20
서비스를 받지 못한 고객	62

통계 항목	값
평균 대기 시간	8.256
평균 서비스 시간	4.248
평균 전화 서비스 시간	5.043
총 서비스 시간	548
총 전화 서비스 시간	116
서비스받은 총 고객 수	129
전화서비스 총 고객 수	23

실험 결과

통계 항목	값
평균 대기 시간	1.565
평균 서비스 시간	4.256
평균 전화 서비스 시간	5.152
총 서비스 시간	715
총 전화 서비스 시간	170
서비스받은 총 고객 수	168
전화서비스 총 고객 수	33

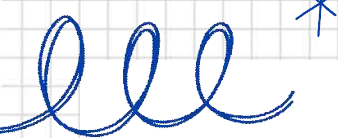
통계 항목	값
평균 대기 시간	121.738
평균 서비스 시간	2.115
평균 전화 서비스 시간	5.69
총 서비스 시간	129
총 전화 서비스 시간	165
서비스받은 총 고객 수	61
전화서비스 총 고객 수	29
서비스를 받지 못한 고객	95

결론

태피 상점의 운영 시뮬레이션 개발

창구의 개수와 전화 기능의 유무에 따라 차이를 볼 수 있었음

차후 사용자의 운영 상점에 맞는 조건을 설정한 시뮬레이션으로 상점 운영에 도움을 줄 수 있을 것



느낀점

시뮬레이션, GUI 프로그램 개발은 처음이라 어려웠음

GUI 프로그래밍에 흥미를 느낌

