



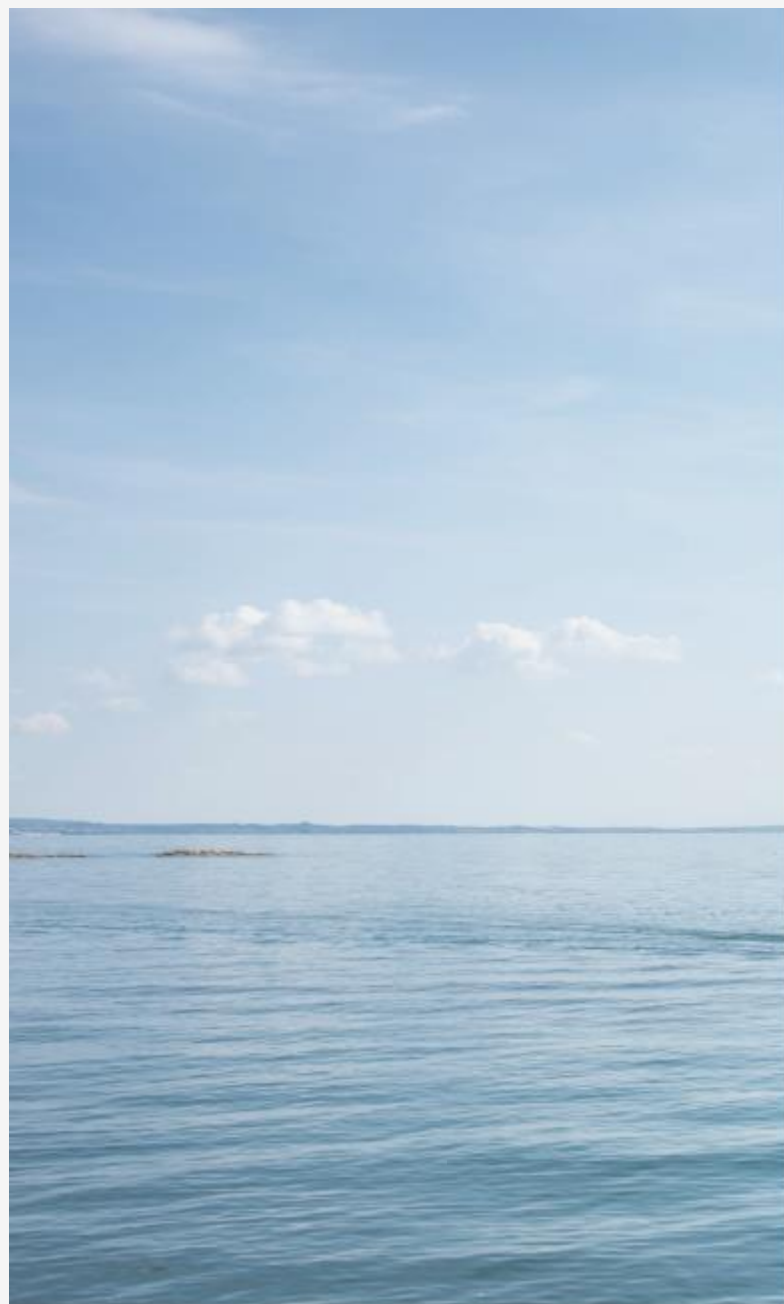
SLG

20200679 안예진

20200886 이재민

+

02



목차

01

시뮬레이션 게임 시연

- 목차와 관련된 세부 내용을
입력해주세요

02

붕어빵 조리 과정

- 목차와 관련된 세부 내용을
입력해주세요

03

손님에게 판매 과정

- 목차와 관련된 세부 내용을
입력해주세요

04

Q&A

- 목차와 관련된 세부 내용을
입력해주세요

제목을 입력해주세요

01

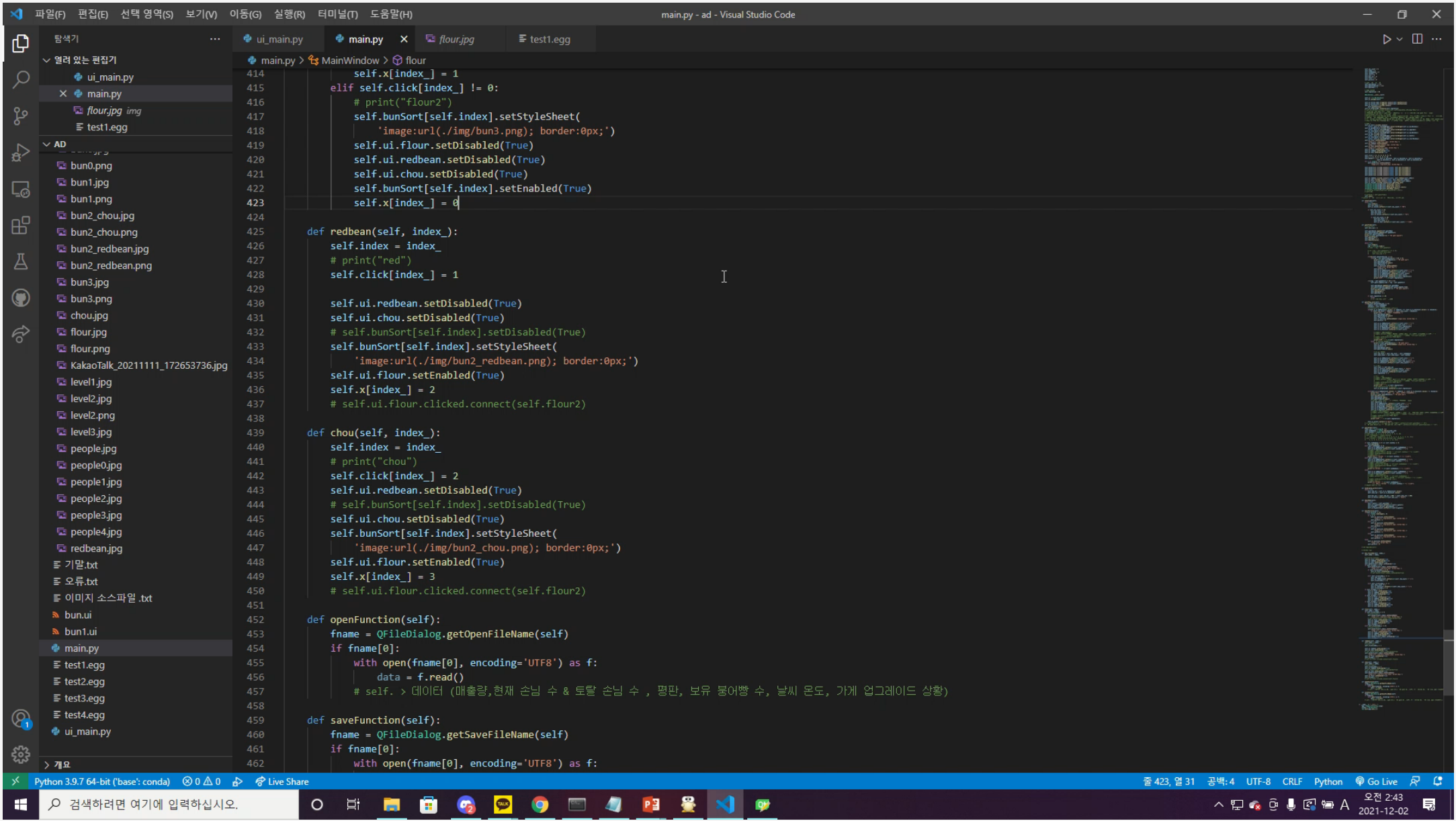
시뮬레이션 게임 시연

목차와 관련된 세부 내용을 입력해주세요

01 제목을 입력해주세요

+

03



+

02

붕어빵 조리 과정

목차와 관련된 세부 내용을 입력해주세요

08

02 제목을 입력해주세요

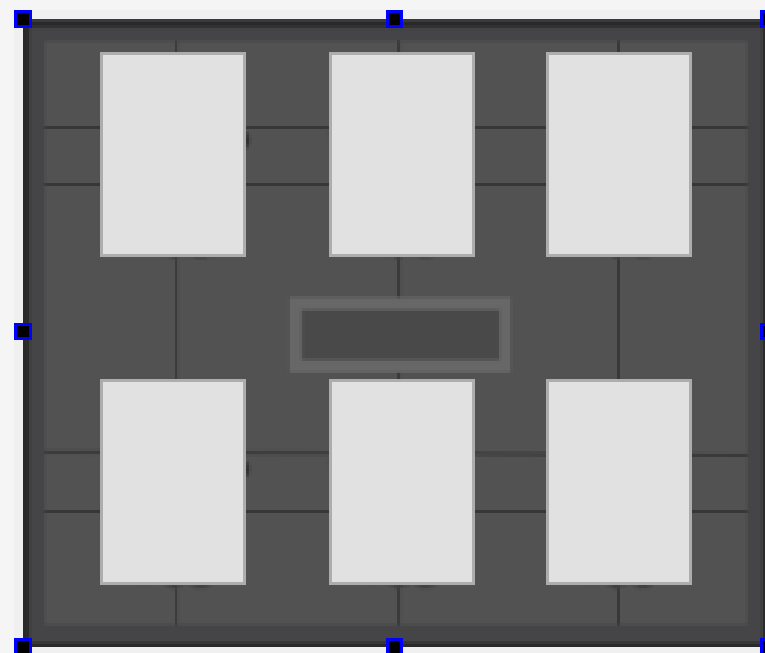
```

self.click = [-1, 0, 0, 0, 0, 0, 0]
self.x = [-1, 0, 0, 0, 0, 0, 0]
self.bunSort = [-1, self.ui.bunLevel, self.ui.bunLevel_2, self.ui.bunLevel_3,
                self.ui.bunLevel_4, self.ui.bunLevel_5, self.ui.bunLevel_6]
for i in range(1, 7):
    self.bunSort[i].setStyleSheet(
        'image:url(./img/bun0.png); border:0px;')

self.bunSort[1].clicked.connect(lambda: self.bun_finished(1))
self.bunSort[2].clicked.connect(lambda: self.bun_finished(2))
self.bunSort[3].clicked.connect(lambda: self.bun_finished(3))
self.bunSort[4].clicked.connect(lambda: self.bun_finished(4))
self.bunSort[5].clicked.connect(lambda: self.bun_finished(5))
self.bunSort[6].clicked.connect(lambda: self.bun_finished(6))

self.ui.redbean.clicked.connect(lambda: self.redbean(self.index))
self.ui.chou.clicked.connect(lambda: self.chou(self.index))
self.ui.flour.clicked.connect(lambda: self.flour(self.index))
self.ui.sellbtn.clicked.connect(self.guestPop_sale)

```




```

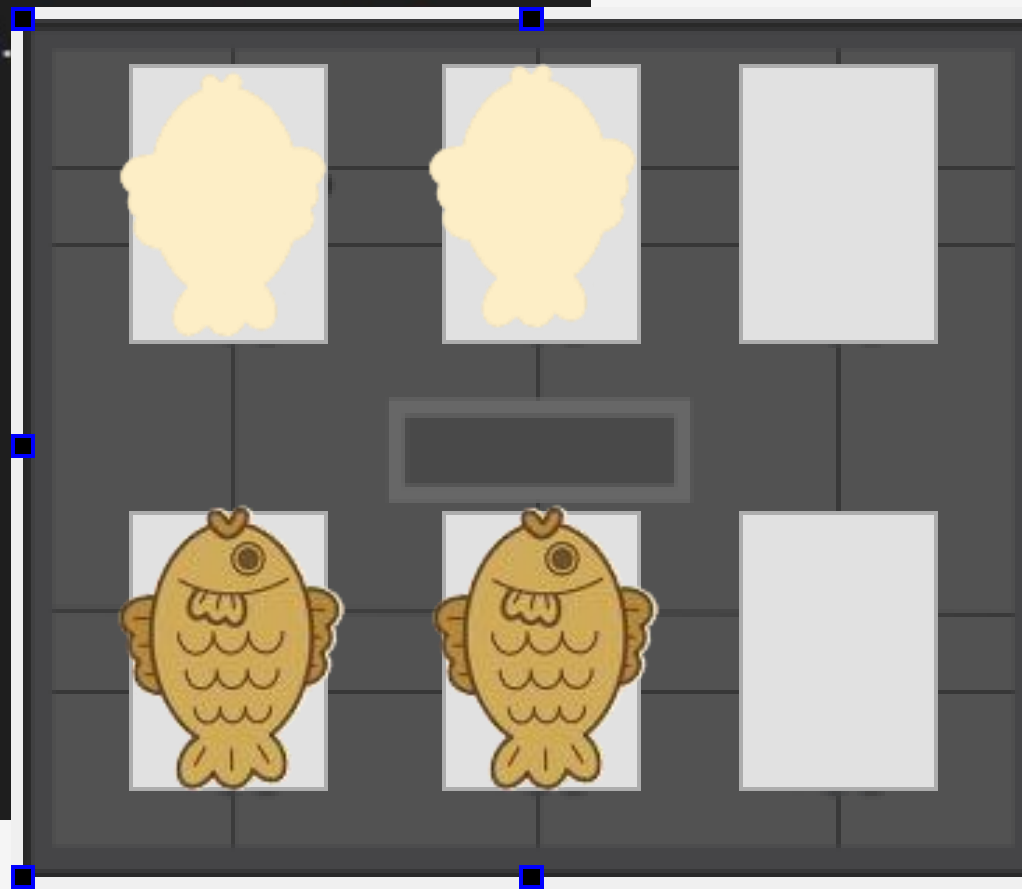
def bun_finished(self, index_):
    self.index = index_

    # print("b finish")
    self.ui.flour.setDisabled(True)
    self.ui.redbean.setDisabled(True)
    self.ui.chou.setDisabled(True)
    self.bunSort[self.index].setEnabled(True)
    if self.x[index_] == 0:
        self.bunSort[self.index].setStyleSheet(
            'image:url(./img/bun0.png); border:0px;')
        if self.click[index_] == 0:
            self.ui.flour.setEnabled(True)
            # self.bunSort[self.index].setDisabled(True)

        elif self.click[index_] == 1:
            self.red_count += 1
            self.ui.redbeanCount.setText(str(self.red_count) + "개")
            self.click[index_] = 0

        elif self.click[index_] == 2:
            self.chou_count += 1
            self.ui.chouCount.setText(str(self.chou_count) + "개")
            self.click[index_] = 0
    if self.x[index_] == 1:
        self.ui.flour.setDisabled(True)
        self.ui.redbean.setEnabled(True)
        self.ui.chou.setEnabled(True)
    if self.x[index_] == 2:
        self.ui.flour.setEnabled(True)
        self.ui.redbean.setDisabled(True)
        self.ui.chou.setDisabled(True)
    if self.x[index_] == 3:
        self.ui.flour.setEnabled(True)
        self.ui.redbean.setDisabled(True)
        self.ui.chou.setDisabled(True)

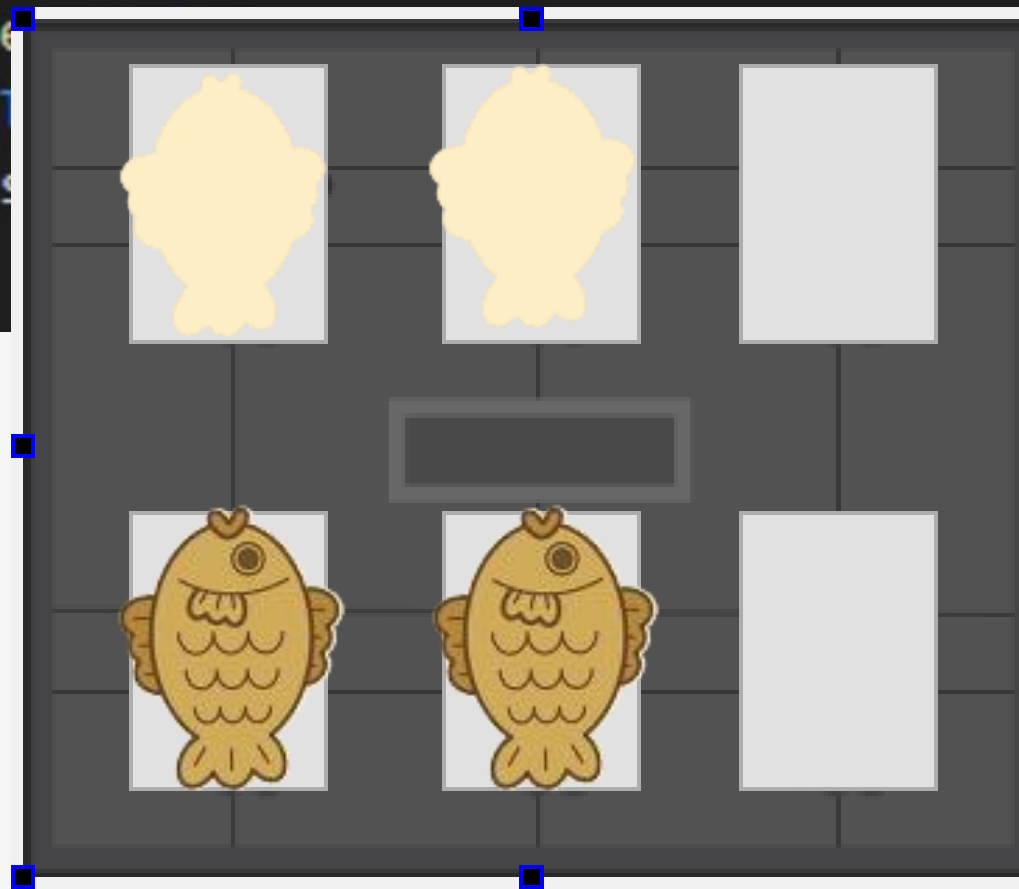
```



```

def flour(self, index_):
    self.index = index_
    if self.click[index_] == 0:
        # print("flour")
        self.bunSort[self.index].setStyleSheet(
            'image:url(./img/bun1.png); border:0px;')
        # self.bunSort[self.index].setDisabled(True)
        self.ui.flour.setDisabled(True)
        self.ui.redbean.setEnabled(True)
        self.ui.chou.setEnabled(True)
        self.x[index_] = 1
    elif self.click[index_] != 0:
        # print("flour2")
        self.bunSort[self.index].setStyleSheet(
            'image:url(./img/bun3.png); border:0px;')
        self.ui.flour.setDisabled(True)
        self.ui.redbean.setDisabled(True)
        self.ui.chou.setDisabled(True)
        self.bunSort[self.index].setDisabled(True)
        self.x[index_] = 0

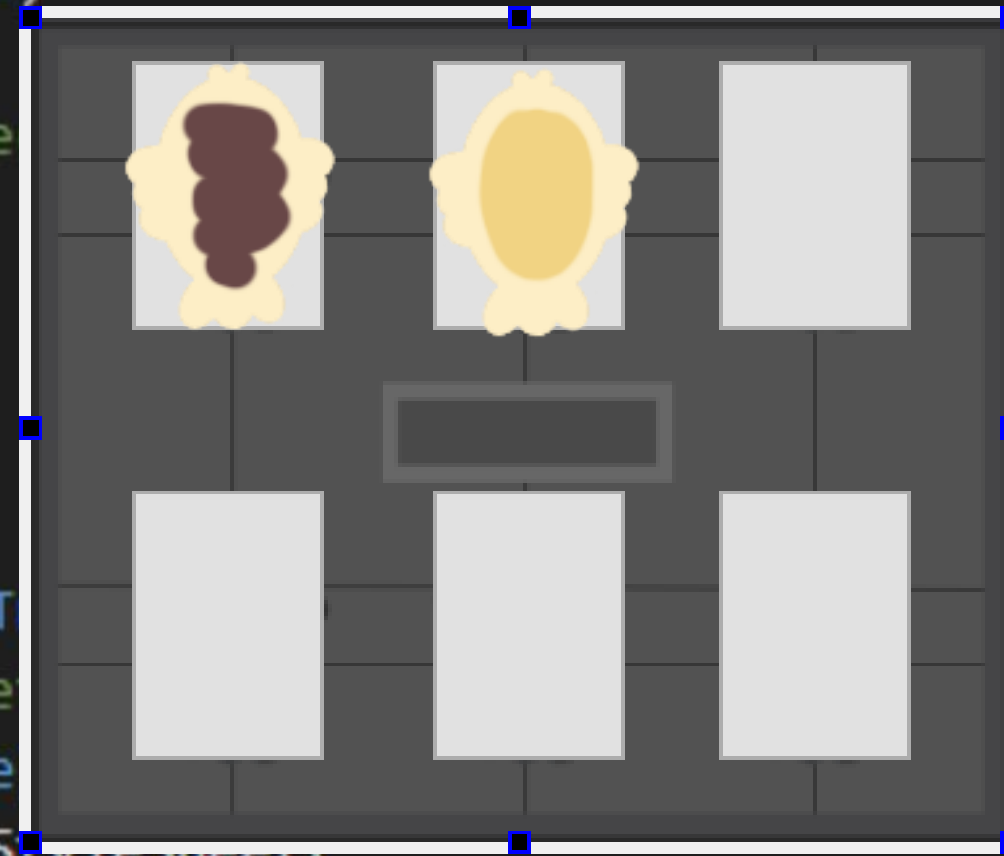
```




```
def redbean(self, index_):
    self.index = index_
    # print("red")
    self.click[index_] = 1

    self.ui.redbean.setDisabled(True)
    self.ui.chou.setDisabled(True)
    # self.bunSort[self.index].setDisabled(True)
    self.bunSort[self.index].setStyleSheet(
        'image:url(./img/bun2_redbean.png); border:0px;')
    self.ui.flour.setEnabled(True)
    self.x[index_] = 2
    # self.ui.flour.clicked.conne

def chou(self, index_):
    self.index = index_
    # print("chou")
    self.click[index_] = 2
    self.ui.redbean.setDisabled(T
    # self.bunSort[self.index].se
    self.ui.chou.setDisabled(True)
    self.bunSort[self.index].setStyleSheet(
        'image:url(./img/bun2_chou.png); border:0px;')
    self.ui.flour.setEnabled(True)
    self.x[index_] = 3
    # self.ui.flour.clicked.connect(self.flour2)
```



핵심 코드 설명 (쓰레드)

“

```
import threading
```

```
threading.Thread(target=self.totaltime).start()  
threading.Thread(target=self.guestTime).start()  
threading.Thread(target=self.bunDisplay_profit).start()  
threading.Thread(target=self.GuestNum).start()
```

상단 문장에 대한 간략한 설명을 입력해주세요

+

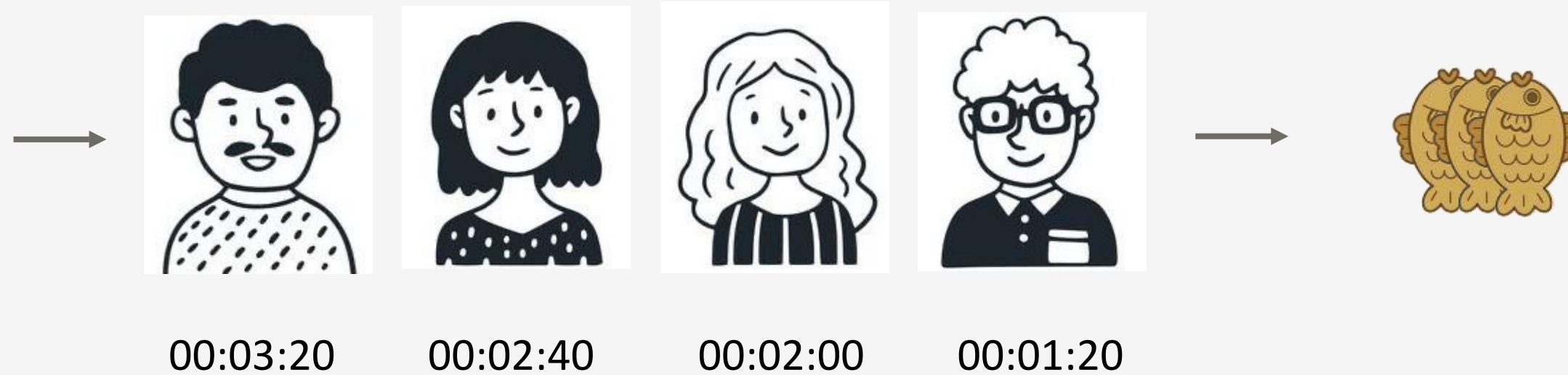


03

손님 판매 과정



손님 입장 및 퇴장 기능



- 손님 Queue와 입장 시간이 기록된 Queue 필요
- 손님 입장 시 손님 번호와 입장을 각각의 Queue에 append
- 손님이 퇴장할 경우 두 개의 Queue를 함께 Pop

손님 입장 및 퇴장 기능



- 평판 하락
 - 판매 봉어빵이 보유 봉어빵의 개수보다 많을 경우
 - 구매 수량과 판매 봉어빵이 일치하지 않을 경우

손님 대기 관련 변수

self.guestQueue = [-1]

self.guestTimeLine = [-1]

self.guestNum = 1

while True:

time.sleep(1)

log = int(time.time())

if(len(self.guestTimeLine) >= 2):

if((log - self.guestTimeLine[1]) % 40 == 0):

print(str(self.guestQueue[1]) + "번 손님 pop")

self.guestQueue.pop(1)

self.guestTimeLine.pop(1)

self.reputation -= 2

self.GuestPicture()

self.ui.G_redBeanCount.setText(str(self.init) + "개")

self.ui.G_chouCount.setText(str(self.init) + "개")

self.ui.S_redbeanCount.setValue(int(self.init))

self.ui.S_chouCount.setValue(int(self.init))

self.ui.progressBar.setValue(int(self.reputation))

print("평판 : " + str(self.reputation))

반복문을 통해 1초마다 함수가 실행되도록 함
log : 진행 상황 기록

손님이 Queue에 한 명 이상 적재되었을 경우

손님 입장 시간부터 40초마다 손님 원소 pop
손님 큐(guestQueue) 와
입장 시간 큐 (guestTimeLine)의
첫 번째 원소를 함께 pop

판매 버튼에 count한 붕어빵 개수
손님이 구매하고자 하는 붕어빵 개수
모두 초기화

```
if((log - self.gameStart) % 60 == 0):  
    self.guest_append_time = log - self.gameStart  
    self.guestQueue.append(self.guestNum)  
    self.guestTimeLine.append(self.guest_append_time)  
    print(str(self.guestQueue[-1]) + "번 손님 append")  
    self.GuestPicture()  
    self.guestOrder()  
    self.guestNum += 1  
  
if self.reputation <= 10:  
    break
```

게임 시작부터 1분마다 실행

손님 입장 시간 기록 후
손님 원소와 손님 입장 시간 원소를
Queue에 각각 append

손님 이미지와 주문 내용 함수 호출 및 손님 번호 count

평판이 10 이하일 경우 게임 종료

“ 주요 코드 : 스레드

```
threading.Thread(target=self.totaltime).start()  
threading.Thread(target=self.guestTime).start()  
threading.Thread(target=self.bunDisplay_profit).start()  
threading.Thread(target=self.GuestNum).start()
```

- 시간을 재는 함수와 게임 실행 화면이 동시에 동작하도록 스레드 구문 사용
 - 각 함수의 실행이 병렬로 처리되도록 함



발표를 들어주셔서
감사합니다 :))