

E동 5층 자판기 예비 보고서

한국산업기술대학교
2019156023 유광무

① 지원 플랫폼

- I. Windows
- II. OS X
- III. Ubuntu - GUI Mode

② 개발 환경

- I. Python 3.7
- II. OS X 10.14.4
- III. Github - Version Controll

③ 개발 동기

- I. 전산학 기초 수업에서 프로그래밍 과제로 본 프로젝트를 계획
- II. 다른 학생들이 배울 수 있도록 Github에 Open Source 업로드

④ 접근 계획서

- I. 초기 설정
 - 1. 초기 자판기 데이터를 구축
 - 2. 자판기 데이터는 drinkName, drinkPrice, drinkCount 리스트
- II. 현금 투입
 - 1. QInputDialog를 통해 금액 입력
 - 2. 입력받은 값을 coin변수에 저장
 - 2. 투입 잔액을 알려주는 레이블 값을 coin 값으로 변경

III. 현금 반환

1. 투입된 금액(coin)을 0원으로 변경
2. 위의 ④ - III - 2와 동일하게 처리

IV. 음료 주문

1. 주문 버튼 클릭
2. coin값이 drinkPrice 이상이고 drinkCount가 0 초과이면 구매 후 QMessageBox를 통해 구매 완료 안내창 표시
3. 구매 실패 시 QMessageBox를 통해 사유와 함께 구매 실패 안내창 표시

V. 기타

1. 사용자 편의를 위해 PyQt5를 이용하여 GUI 환경으로 개발
2. 종류와 가격을 List로 만들라 지시하여 Dictionary는 이용하지 않음
3. 음료수가 두 개 이상 이더라도 하나의 drinkCount를 공용으로 이용

⑤ 사용 모듈

I. PyQt5

1. GUI 환경의 프로그램 개발
2. Desginer에서 UI를 개발하여 XML 형식의 UI로 추출
3. terminal pyuic5를 통해 XML 형식의 UI를 py로 변환
4. PyQt5 모듈이 없는 경우 pip install PyQt5를 통해 설치 가능

II. json

1. 자판기 데이터 저장 및 불러오기
2. json 포맷으로 파이썬 dictionary 포맷과 일치 -> 호환성

⑥ 사용자 정의 함수

I. Init()

1. 기본적인 프로그램 초기화
2. 음료수 리스트 구축

II. AddDrink(name, price, count)

1. 전역 리스트에 넘어온 파라미터를 추가
2. 추가되는 전역 리스트 다음과 같다
 - drinkName(음료수 이름)
 - drinkPrice(음료수 가격)
 - drinkCount(음료수 갯수)

III. InsertCoin(self)

1. QDialogDialog를 통해 투입할 금액 입력
2. 기본 값은 100원으로 최소 0원 - 최대 10,000원 까지 입력 가능
3. 전역 coin 변수에 입력받은 값을 증가
4. 투입 금액을 알려주는 lblPrice 레이블을 coin 값으로 변경
5. repaint()를 통해 UI를 새로 그림

IV. ReturnCoin(self)

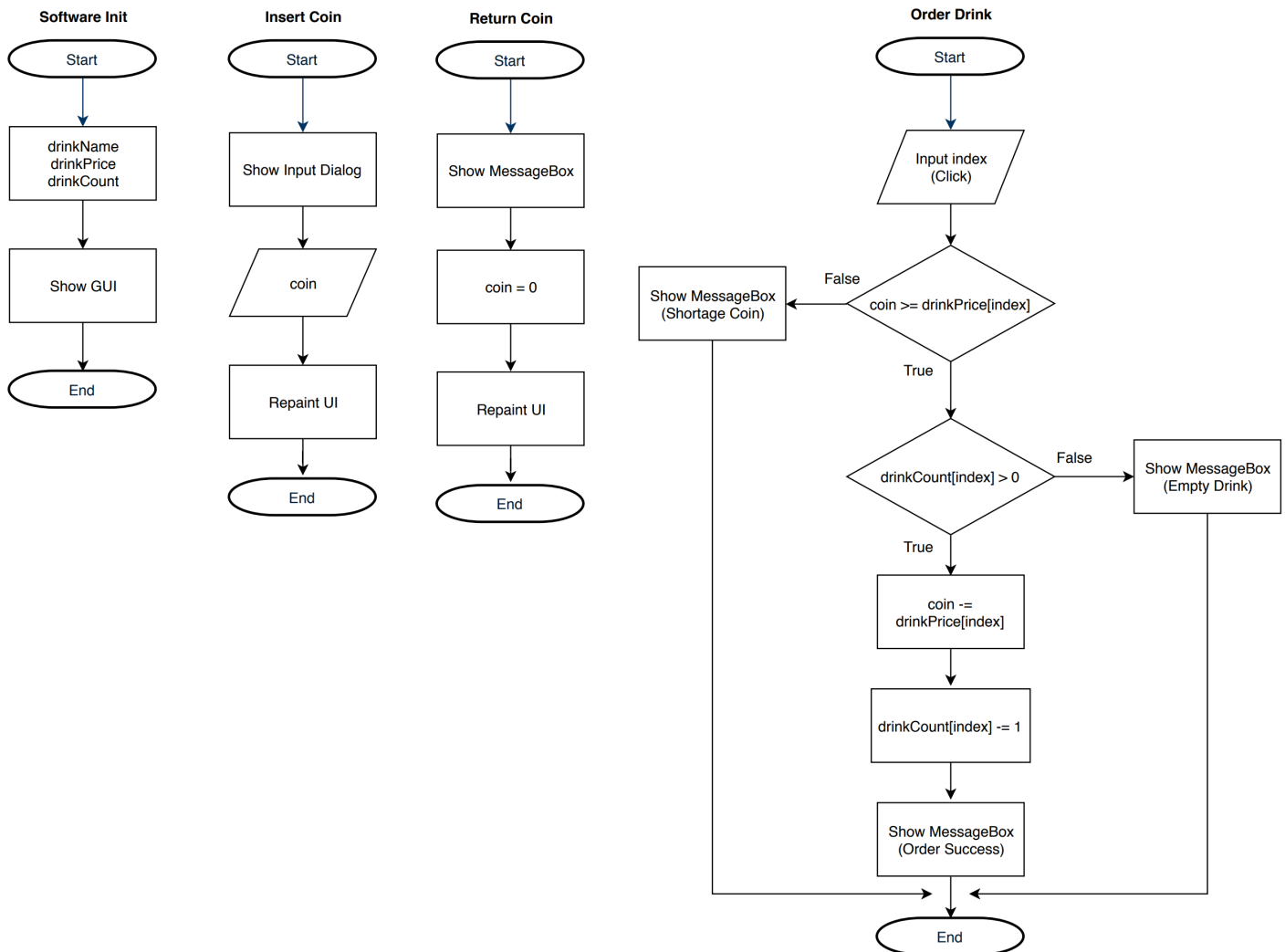
1. QMessageBox를 통해 반환되는 금액 안내
2. 전역 coin 변수의 값을 0으로 초기화
3. 위의 ② - III - 4, 5와 동일하게 처리

V. OrderDrink(self, index)

1. clicckd.connect에서 호출되어 일반적으로 파라미터 넘길 수 없음
2. 그렇기에 lambda를 이용하여 음료수 index를 넘김
3. 전역 coin 변수가 drinkPrice[index]의 값 보다 크다면
drinkCount[index]와 비교
4. drinkCount[index] 값이 0 초과이면 -1 감소시키고 전역 coin 변수 값을
drinkPrice[index]만큼 감소
5. QMessageBox를 통해 구매 완료 안내를 제공

⑦ 순서도

- I. GUI 환경으로 절차적으로 진행되지 않기에 각 기능에 맞춰 순서도를 나눔
- II. 기능별 순서도를 나누어 설명을 작성하려 하였지만, 하나의 순서도 사진을 원하여 하나의 사진으로 대체



⑧ 소스코드

I. 자판기 음료수 초기 설정

```
def Init():
    AddDrink("아이스스", 600, 5)
    AddDrink("레몬워터", 1500, 5)
    AddDrink("옥수수 수염차", 1300, 5)
    AddDrink("초가를 우유차", 1300, 5)
    AddDrink("트레비", 1000, 5)
    AddDrink("밀키스", 800, 5)
    AddDrink("펍시", 800, 5)
    AddDrink("햇식스", 1000, 5)
    AddDrink("사이다", 1000, 5)
    AddDrink("코코 리치", 1000, 5)
    AddDrink("립톤", 1000, 5)
    AddDrink("스파클링 사과", 1000, 5)
    AddDrink("스파클링 포도", 1000, 5)
    AddDrink("가나", 600, 5)
    AddDrink("레쓰비", 600, 5)
    AddDrink("칸타타", 1000, 5)
    AddDrink("레쓰비 카페타임", 1000, 5)
    AddDrink("게토레이", 800, 5)
    AddDrink("코코 포도", 800, 5)
    AddDrink("잔치집 식혜", 800, 5)
```

```
coin = 0
drinkName = []
drinkPrice = []
drinkCount = []

def AddDrink(name, price, count):
    global drinkName, drinkPrice, drinkCount

    drinkName.append(name)
    drinkPrice.append(price)
    drinkCount.append(count)
```

II. 현금 투입

```
def InsertCoin(self):
    global coin

    i, okPressed = QInputDialog.getInt(Dialog, "현금 투입", "투입할 금액을 입력해주세요\n(최대 투입 금액: 10,000원)", 100, 0, 19000, 1)
    coin += i

    ui.lblPrice.setText(str(coin) + "원")
    Dialog.repaint()
```

III. 음료 주문

```
def OrderDrink(self, index):
    global coin, drinkName, drinkPrice, drinkCount

    if coin >= drinkPrice[index]:
        if drinkCount[index] > 0:
            coin -= drinkPrice[index]
            drinkCount[index] -= 1

            QMessageBox.about(Dialog, "음료수 구매 성공", drinkName[index] + "(을)를 "
                + str(drinkPrice[index]) + "원에 구매하였습니다.\n남은 갯수: " + str(drinkCount[index]) + "개")
            ui.lblPrice.setText(str(coin) + "원")
            Dialog.repaint()
        else:
            QMessageBox.about(Dialog, "음료수 구매 실패", drinkName[index] + " 음료수의 갯수가 없습니다")
    else:
        QMessageBox.about(Dialog, "음료수 구매 실패", "금액이 부족합니다")
```

IV. 잔돈 반환

```
def ReturnCoin(self):  
    global coin  
  
    QMessageBox.about(Dialog, "현금 반환", "현금을 " + str(coin) + "원을 반환하였습니다")  
  
    coin = 0  
    ui.lblPrice.setText(str(coin) + "원")  
    Dialog.repaint()
```

⑨ 실제 프로그램

