

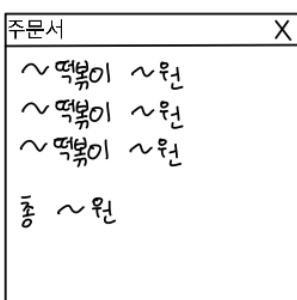
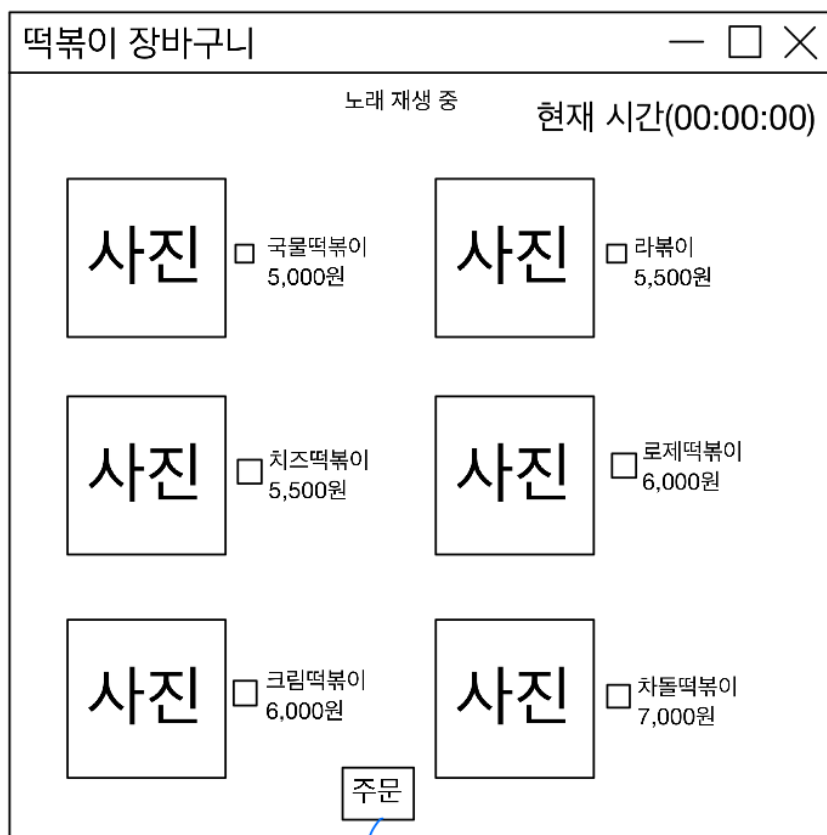
0. 떡볶이 장바구니 만들기 (작성자: 2071201 황지연)

1. 프로젝트 개요

☐ 소프트웨어 개발 목표

배달어플처럼 먹고싶은 떡볶이 종류를 장바구니에 담은 후 주문하기 버튼을 누르면 해당 매장에 주문서가 가도록 만들고 싶습니다. 또한 떡볶이를 고르는 동안 노래가 나와 심심하지 않도록 만들 예정입니다. 디자인보다는 기능에 집중하여 실제 서비스에서도 구현될 수 있으면 좋겠다는 생각으로 만들고 싶습니다.

☐ 초기 설계 화면



□ 유사 소프트웨어 분석

배달의 민족 어플과 요기요 어플을 분석했을 때 각각의 메뉴에 사진들이 있진 않고 메뉴이름과 가격만 있었습니다. 하지만 이런 어플들에는 리뷰들이 있어서 소비자들이 참고할 수 있는 사항이 있었습니다.

□ 프로젝트 차별성

저는 음식 사진들과 이름, 가격이 동시에 뜨게 만들었으며 떡볶이 장바구니인 만큼 떡볶이 노래가 나오도록 차별을 두었습니다. 또한 프로그램 자체에 현재 시간이 나오도록 만들었습니다. 메시지 다이얼로그를 이용하여 주문서를 만들었습니다.

2. 소프트웨어 기능 분석

□ 소프트웨어 시나리오(500자)

인터넷 쇼핑을 할 때 흔히 사용하는 장바구니라는 개념을 도입하여 장바구니 창을 직접 만들어 보겠습니다. 처음 화면에는 이미지 처리 기능을 이용해 여러 개의 떡볶이 사진들을 띄웁니다. 동시에 이벤트 처리 기능을 이용해 각각의 떡볶이마다 이름, 가격을 적은 후 체크박스를 만듭니다. 체크 박스는 하나를 선택하거나 여러개를 동시에 선택할 수 있습니다. 마지막에 주문이라는 버튼도 생성하여 주문을 누르면 어떤 종류이고 각각의 가격은 얼마이며 총 가격은 얼마인지 다이얼로그기능을 통해 메시지로 보여줍니다. 스레드의 기능을 사용해 프로그램이 현재 시작된 시간부터 디지털 시계로 시간을 보여줍니다. 사운드 처리 기능을 이용해 프로그램이 시작되면 떡볶이 노래가 나오며 마우스가 프로그램을 벗어나면 노래가 일시 중단되고 다시 마우스가 프로그램으로 올라오면 노래가 계속 나옵니다.

(1) 기본 설계

□ 핵심적 기능과 흐름

- 이미지 6개(떡볶이 사진) 구현
- 체크박스 6개 구현
- 이름, 가격 레이블 6개 구현
- 스레드를 이용한 현재 시각 구현
- 사운드 구현을 이용해 떡볶이송 구현
- 떡볶이송 반복 재생
- 주문 버튼 이벤트를 이용하여 메시지 다이얼로그로 새로운 창 구현
- 새로운 창에서 주문했던 떡볶이 이름, 가격과 총 합산 가격 구현
- 다시 선택하기 기능 구현

□ 화면의 설계 및 구성



□ 소프트웨어 상세 기능 기술 및 우선순위

가장 첫 화면에 바로 떡볶이 장바구니가 나타납니다.
떡볶이 종류를 선택하는 체크박스가 가장 중요합니다.
버튼을 누르기 전까지 노래가 재생됩니다.
그동안 현재 시각의 초가 계속 흘러야 합니다.
버튼을 눌렀을 때 새로운 창으로 메시지 창이 잘 나타내야 합니다.
메시지 창에 원하는 대로 올바르게 가격과 이름이 나와야 합니다.

□ 소프트웨어 화면 구성

떡볶이 장바구니

떡볶이송 연주 계속


현재시간: 14:15:35

	<input type="checkbox"/> 국물떡볶이 5,000원		<input type="checkbox"/> 라볶이 5,500원
	<input type="checkbox"/> 치즈떡볶이 5,500원		<input type="checkbox"/> 로제떡볶이 6,000원
	<input type="checkbox"/> 크림떡볶이 6,000원		<input type="checkbox"/> 차돌떡볶이 7,000원

주문

MessageDialog

주문서

 국물떡볶이 : 5,000원
크림떡볶이 : 6,000원
차돌떡볶이 : 7,000원

총 합계: 18000

OK

order

sum

□ 기초 클래스 설계

Project
+ panel
+ playAudio(String pathName)
+ Project()

MyPanel
+ img1, img2, img3, img4, img5, img6, chk1, chk2, chk3, chk4, chk5, chk6, btn
+ void paintComponent(Graphics g)
+ MyPanel()

cTread
+ JLabel / int hour, min, second
+ void run()
+ cTread(JLabel tlabel)

□ 소프트웨어 개발 순서

- 떡볶이 장바구니 창 구현
- 이미지 1개 구현
- 체크박스 1개 구현
- 레이블 1개 구현
- 이미지, 체크박스, 레이블 위치 맞추기
- 나머지 이미지, 체크박스, 레이블 구현
- 주문 버튼 만들기
- 떡볶이 송으로 사운드 구현
- 스레드로 현재 시각 구현
- 주문 버튼 눌렀을 시 다이얼로그를 이용해 새창 구현
- 체크박스에 이벤트를 작성하여 다이얼로그 메시지창에 이름과 가격 구현

3. 상세 클래스 설계

Project
+ 주문버튼: JButton + 체크박스(1~6): JCheckBox + 오디오 파일: Clip + 노래재생글자, 시각: JLabel
+ 레이블 위치, 프레임, 스레드, 오디오 초기 설정: Project()

MyPanel
+ 떡볶이사진(1~6): ImageIcon + 체크박스 6개 실제 구현: JCheckBox + 버튼 실제 구현: JButton
+ 체크박스, 버튼 위치 추가: MyPanel() + 메시지다이얼로그 내용추가: void actionPerformed(ActionEvent e) + 떡볶이 사진 화면에 그리기: void paintComponent(Graphics g)

4. 화면설계와 동작흐름 설계

떡볶이 장바구니

떡볶이송 연주 계속

현재시간: 22:40:50



☒ 국물떡볶이 5,000원



☐ 라볶이 5,500원



☒ 치즈떡볶이 5,500원



☐ 로제떡볶이 6,000원



☐ 크림떡볶이 6,000원



☒ 차돌떡볶이 7,000원

주문

먼저 먹고 싶은 떡볶이들을 체크한 후 주문하기 버튼을 누릅니다.

주문서

i

국물떡볶이 : 5,000원
치즈떡볶이 : 5,500원
차돌떡볶이 : 7,000원

총 합 계: 17500

OK

그러면 주문서가 화면에 뜨면서 선택한 떡볶이와 가격, 총 가격이 나옵니다. 이 후 OK버튼 누르면 초기화면으로 돌아가 다시 선택할 수 있습니다.

5. 소프트웨어 사용설명서

요즘 코로나 19로 인하여 밖에 나가서 사먹기 보다는 배달음식을 시켜 집에서 먹는 추세입니다. 그래서 이 프로그램은 배달어플에서 사용할 수 있는 떡볶이 장바구니입니다. 먹고 싶은 떡볶이의 체크박스를 선택하고 주문하기 버튼을 누르면 떡볶이 종류와 가격에 해당하는 주문서가 나오도록 만들었습니다. 실제로 매장에서 사용시에는 주문서가 해당 매장에 전송되게 하면 됩니다.

6. 프로젝트 이후 추가할 내용

마우스가 체크박스과 주문버튼 위에 올라와 있을 때에도 노래가 흘러나오도록 구현해야 합니다. 노래 소리가 매우 커서 프로그램 안에서 소리 조절할 수 있도록 만들어야 합니다. 진짜 장바구니처럼 떡볶이 종류뿐만 아니라 재료를 추가할 수 있는 메뉴와 결제 방법도 있어야 합니다.

7. 개발 후기

파일처리 기능대신 메시지 다이얼로그 기능을 사용하여 떡볶이 이름과 가격이 나오는 주문서를 만든 것이 떡볶이 장바구니라는 취지에 잘 맞았던 것 같습니다. 다른 배달 어플들을 살펴봤는데 가게의 메뉴들과 연관된 노래가 배경음악으로 들리는 어플은 없었기 때문에 이 부분에서 차별성을 두었습니다.

이미지와 체크박스 구현을 했는데 화면에 뜨지 않아서 애먹었는데 패널 위가 아닌 배치관리자를 사용해서 추가하려고 해서 그랬던 것이여서 패널 위로 추가해서 해결했습니다. exe파일을 만들려고 JSmooth라는 어플을 사용하는 데 어플 실행이 안되서 알아보니 환경변수때문이라는 것을 알게 되었습니다. 인터넷 검색을 통하여 해결했습니다. 하지만 만든 exe파일을 실행시켜보니 이미지와 오디오가 나오지 않았습니다. 이 파일들이 상대적 위치가 아닌 절대적 위치로 적어야 한다고 해서 절대적 위치로 적으니 실행이 잘 되었습니다.

저는 프로젝트를 3주 정도 만들었습니다. 너무 어려워서 포기하고 싶을 때도 많았지만 하다가 안되면 다음날에도 계속 시도해서 결국 완성했습니다. 처음이라 완성도가 높지는 않지만 스스로 뿌듯함을 많이 느꼈고 포기하지 않고 계속 시도하면 결국 할 수 있다는 것을 깨달았습니다.