

객체지향프로젝트

< Final Report >

소프트웨어공학과

3조 Mandu

나희지 변동건 심지영 하수연

Final Report

< 목차 >

1. NAME OF GAME
2. 동기
3. 특징
4. 기능
5. Function Specification
6. Requirement List
7. Usecase
8. Usecase Diagram
9. Sequence Diagram
10. User Interface Designs & Screenshot of application demo
11. Class Diagram
12. Screenshot of API Manual
13. 각 팀원 별 Work Distribution 및 진행 결과

< NAME OF GAME >

Mandu

< 동기 >

무슨 프로그램을 만들지 토의를 하다 프로그램의 컨셉을 저희 같은 대학생의 모습이 반영되도록 만들고 싶었습니다. 그래서 저희의 모습을 가장 잘 표현한 사물을 찾다가 만두를 생각하게 되었습니다. 얇은 만두피로는 언제 터질지 모르는 우리들의 정신 상태를 표현하며, 만두소는 대학 생활을 하면서 받게 되는 여러 가지 스트레스를 표현하였습니다. 만두는 만두피에 만두소를 넣어 찌는데 이것도 저희의 머리에 많은 전공 지식을 넣어 공부하며 열을 내는 모습으로 나타낼 수 있다고 생각하였습니다. 만두를 위협하는 젓가락은 그런 정신 상태를 위협하는 요소인 과제, 시험공부 등을 표현해 보았습니다. 만두가 터지지 않기 위해 젓가락을 요리조리 피하는 것처럼 대학생활의 위기들을 극복해 가는 우리들의 모습을 표현하고 싶었습니다.

< 특징 >

최근에 기록을 친구와 공유해서 서로 경쟁하는 게임들이 많은데 기존의 게임들은 대개 단계별로 진행된다. 하지만 이런 게임은 단계마다의 과정과 점수가 무시되고 단계에만 의존하게 됩니다. 단계마다에도 도달한 방법과 시간이 다름에도 불구하고 같은 단계에 머물게 되는 것입니다. 이러한 문제를 없앨 수 있는 우리가 만들고자하는 시간 기록을 재는 이 게임은 친구들과 자신의 기록을 공유하고 경쟁하는 사람들에게 최적화된 게임이다. 이 게임의 스테이지는 단순 난이도 상승을 나타내는 정도입니다.

< 기능 >

1. 게임 원리

- 여러 가지 만두 캐릭터 중 원하는 만두 선택 후 방향키를 이용하여 젓가락을 피하는 게임. 만두가 젓가락에 닿으면 게임 종료. 갈수록 젓가락의 수가 늘어나고 어려운 패턴이 나옴. 몇 가지 아이템 획득가능.

2. 화면 및 사운드

- 2.1 화면

게임 첫 화면에 게임시작, 게임방법, 설정, 종료, 랭킹 총 5개의 매뉴얼 버튼을 생성한 창을 띄우고 배경은 캐릭터 만두에서 착안하여 중국느낌이 나는 배경을 사용.

- 2.2 게임 배경

접시 위에서 만두가 젓가락을 피해 도망다니는 모습을 컨셉으로 하여 만두를 담은 접시를 배경으로 사용.

- 2.3 사운드

게임 전체 배경 음악이 메인화면, 게임진행화면, 만두가 죽을 때 나오는 음악 세 종류. 효과음은 버튼 누를 때 나는 소리, 만두 움직일 때 나는 소리, 젓가락 발사 소리, 간장 발사 소리, 간장 맞을 때 다는 소리, 젓가락 맞을 때 나는 소리.

- 2.4 이펙트

이펙트는 만두가 죽을 때 나오는 이펙트, 간장 발사 시 이펙트, 랜덤 아이템획득 시 아이템 이름 이펙트

3. 설정

- 3.1 기능

메인화면에서 메뉴 선택 시 마우스 동작을 감지하는 기능.
메뉴선택 후 메뉴에 맞는 컴포넌트가 등장하는 기능.
게임 진행 중 시간을 표시하고 기록하는 기능.
효과음과 배경음악을 끄거나 음량을 조절하는 기능.
날아오는 젓가락의 숫자가 많아지게 하는 기능.
젓가락이 캐릭터를 향하여 발사되는 기능.
일정 시간이 되면 복잡한 패턴의 젓가락이 발사되는 기능.
만두가 접시를 벗어나지 못하게 하는 기능.
만두가 젓가락에 닿으면 게임오버 되는 기능.
게임 오버 시 기록 저장여부를 묻는 기능.
기록 저장 시 순위를 보여주는 기능.

- 3.2 게임조작

메인화면에서 메뉴 선택 시 마우스 사용.
캐릭터 선택화면에서 캐릭터 선택을 마우스와 방향키를 사용.
게임 진행 중에는 방향키 사용.

4. 세부사항

- 4.1 캐릭터

만두를 캐릭터로 사용. 다양함을 위해 만두 종류별로 캐릭터 선택가능. (군만두, 찐만두, 물만두)

- 4.2 아이템

랜덤박스를 게임 화면에 등장시켜 획득 시 복불복으로 추가 아이템 증정.
간장 아이템 획득하였을 때 간장이 흘날려 피격 시 만두의 움직임 속도 감소.
거대화 아이템 획득하였을 때 만두의 크기가 커짐.

소형화 아이템 획득하였을 때 만두의 크기가 작아짐.
썰드 아이템 획득하였을 때 젓가락 피격 한 번 방어해줌.

- 4.3 장애물

만두를 향해 날아오는 젓가락, 피격 시 속도를 느려지게 하는 간장, 접시 밖으로 나가지 못하게 하는 벽.

< Function Specification >

1. 저장

- 저장여부에서 YES를 선택하면 닉네임과 점수가 저장되는 기능

2. 만두에 관한 기능

- 만두가 접시 밖을 벗어나지 못하도록 하는 기능
- 만두가 랜덤아이템인 거대화를 먹으면 크기가 커지고 만두의 유효범위도 그에 맞게 커지는 기능
- 만두가 랜덤아이템인 작아지기를 먹으면 크기가 작아지고 만두의 유효범위도 그에 맞게 작아지는 기능
- 만두가 랜덤아이템인 간장을 먹으면 속도가 감소하는 기능
- 만두가 젓가락의 피격범위에 들어왔을 때 만두가 죽어 게임이 종료되는 기능
- 만두가 접시 밖을 벗어나지 못하도록 하는 기능
- 만두가 랜덤아이템인 거대화를 먹으면 크기가 커지고 만두의 유효범위도 그에 맞게 커지는 기능
- 만두가 랜덤아이템인 소형화를 먹으면 크기가 작아지고 만두의 유효범위도 그에 맞게 작아지는 기능
- 만두가 랜덤아이템인 간장을 먹으면 속도가 감소하는 기능

3. 젓가락에 관한 기능

- 개별적으로 움직이는 기능
- 시간이 지날수록 수가 늘어나는 기능
- 스테이지별로 일정패턴이 반복되는 기능
 - > 만두를 따라다니는 기능
 - > 젓가락에서 젓가락이 나오는 기능
 - > 큰 젓가락이 나오는 기능
 - > 빠른 젓가락이 나오는 기능
 - > 젓가락의 수가 시간이 갈수록 증가하는 기능

< Requirement List >

- 메인화면은 게임시작, 게임방법, 옵션, 랭킹, 게임 종료로 구성되도록 한다.
- 게임에 맞는 배경화면, 배경음악, 효과음을 설정 하도록 한다.
- 게임시작버튼을 누르면 캐릭터를 고르는 화면으로 넘어갈 수 있도록 한다.
- 캐릭터 선택화면에서 Back 버튼을 누르면 메인화면으로 돌아갈 수 있게 한다.
- 캐릭터를 고르면 Start 버튼을 눌러 선택된 캐릭터로 게임이 실행될 수 있도록 한다.
- 캐릭터를 고르지 않았을 시 자동으로 수염만두 캐릭터가 선택되도록 한다.
- 게임방법버튼을 누르면 게임 방법을 설명해주는 화면이 나타나도록 한다.
- 옵션버튼을 누르면 음량 조절이 가능하게 한다
- 배경음악과 효과음의 크기를 따로 설정할 수 있도록 한다.
- 랭킹버튼을 누르면 현재까지 저장된 순위를 확인 할 수 있게 한다.
- 종료버튼을 누르면 게임이 종료되도록 한다.
- 게임화면의 왼쪽에는 플레이화면이 오른쪽에는 게임 진행 정보가 위치하도록 한다.
- 스톱위치가 오른쪽 중앙에 위치하도록 한다.
- 현재 스테이지가 오른쪽 상단에 표시되도록 한다.
- 간장 종지가 오른쪽 하단에 위치하도록 한다.
- 만두를 방향키로 움직일 수 있도록 한다.
- 젓가락이 만두를 향해 날아가도록 한다.
- 젓가락이 임의의 방향에서 날아오게 한다.
- 게임은 총 3단계로 구성된다.
- 일정시간이 지나면 다음 단계로 넘어가도록 한다.
- 단계가 넘어갈 때 마다 랜덤박스를 화면의 임의의 위치에 놓아지도록 한다.
- 랜덤박스 등장 후 6초 지나면 사라지도록 한다.
- 만두 캐릭터가 랜덤박스에 닿으면 랜덤으로 아이 템을 획득하게 한다.
- 랜덤박스를 획득하면 아이템이 자동으로 사용되도록 한다.
- 랜덤박스는 맞으면 느려지는 간장, 한 번 젓가락 을 막아주는 쉴드, 거대화, 소형화로 구성되도록 한다.
- 캐릭터가 젓가락에 닿을 때까지 게임이 종료되지 않도록 한다.
- 캐릭터가 젓가락에 닿으면 게임이 종료되도록 한 다.
- 자신의 점수와 함께 점수 저장 여부를 묻는 창이 뜨도록 한다.

- 저장여부에서 YES버튼을 누르면 닉네임 입력할 수 있도록 한다.
- 닉네임 입력 후 OK버튼을 누르면 자신의 순위를 확인하는 창이 뜨도록 한다.
- OK버튼을 누르면 다시시작과 메인화면 버튼이 있는 랭킹 차트창이 뜨도록 한다.
- NO버튼을 누르면 바로 다시시작과 메인화면 버튼이 있는 랭킹 차트창이 뜨도록 한다.
- 다시시작 버튼을 누르면 캐릭터 선택 화면으로 돌아가게 한다.
- 메인화면 버튼을 누르면 메인화면으로 가도록 한다.

< Usecase >

1. 게임을 실행하면 메인화면이 나타난다.

1-1 게임 시작, 게임 방법, 옵션, 랭킹, 종료버튼이 표시된다.

1-1-1 게임 시작버튼을 누르면 2로 넘어간다.

1-1-2 게임 방법 버튼을 누르면 게임 방법이 설명된 화면이 나타난다.

1-1-2-1 Back 버튼을 누른다.

1-1-3 옵션 버튼을 누르면 음량을 조절할 수 있는 화면이 나타난다.

1-1-3-1 Back 버튼을 누른다.

1-1-4 랭킹 버튼을 누르면 현재까지 저장된 순위를 보여주는 화면이 나타난다.

1-1-4-1 Back 버튼을 누른다.

1-1-5 종료 버튼을 누르면 종료 여부를 한 번 더 묻는 화면이 나타난다.

1-1-5-1 Yes를 누르면 시스템이 종료된다

1-1-5-2 No를 누르면 화면이 닫히고 메인 화면만 남아있다.

1-2 배경음악이 초기화면이 뜰과 동시에 나온다.

2. 게임시작 버튼을 누르면 캐릭터 선택 화면 나온다.

2-1 화면에 3가지 캐릭터인 수염만두, 탄만두, 꽃만두가 나타난다.

2-1-1 좌(←)우(→)방향키 또는 마우스를 이용하여 원하는 캐릭터를 선택한다.

2-2 캐릭터를 선택하지 않으면 기본 캐릭터 수염만두로 선택되어진다.

2-3 Start 버튼 누르면 게임이 시작된다.

2-4 Back 버튼을 누르면 1번으로 되돌아간다.

3. 게임 화면이 표시되고 게임이 진행된다.

3-1 게임 화면

3-1-1 게임 화면에 오른쪽 상단에는 스테이지가 위치한다.

3-1-2 게임 화면에 오른쪽 중앙에는 스태프위치가 위치한다.

3-1-2-1 stage 1 시작 후 10초 뒤 stage 2가 표시되고 stage 2 시작된 후 20초 뒤에 stage 3가 표시된다.

3-1-3 게임 화면에 오른쪽 하단에는 간장종지가 위치한다.

3-2 게임 진행

3-2-1 방향키를 이용하여 상(↑), 하(↓), 좌(←), 우(→)로 만두 캐릭터를 이동시킬 수 있다.

3-2-2 스테이지가 올라갈수록 젓가락의 패턴과 속도가 변화하며 사용자가 느끼는 난이도가 증가한다.

3-2-3 stage 2, 3이 시작될 때 랜덤박스가 접시 위에 등장하며 만두캐릭터가 랜덤박스 에 닿으면 랜덤 아이템이 자동적으로 사용된다. 만약, 닿지 않으면 등장 후 6초 뒤 사라진다.

3-2-3-1 간장이 나오면 간장이 뿌려진다.

3-2-3-1-1 간장에 닿으면 만두의 속도가 게임이 끝날 때까지 느려진다.

3-2-3-2 썰드를 먹으면 젓가락을 1번 방어해준다.

3-2-3-3 거대화를 먹으면 만두의 크기가 4초간 커진다.

3-2-3-4 소형화를 먹으면 만두의 크기가 4초간 작아진다.

4. 스테이지와 상관없이 젓가락에 만두가 닿으면 게임이 종료된다.

4-1 점수와 점수의 저장여부를 묻는 창이 뜬다.

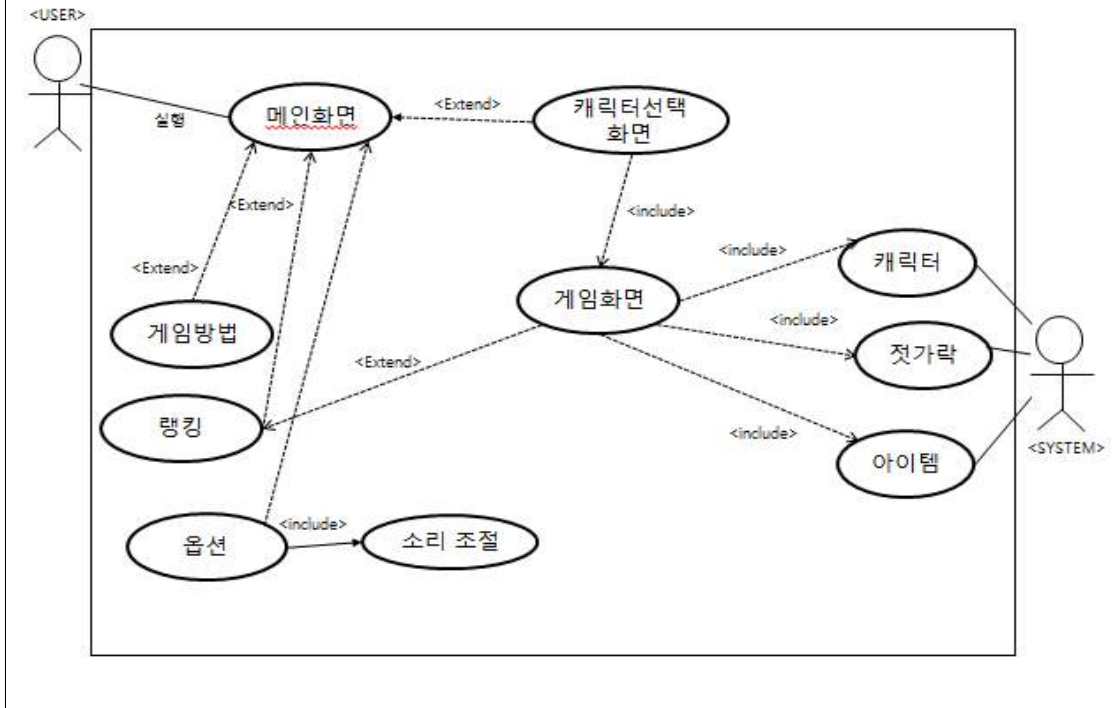
4-1-1 저장여부 YES : 닉네임을 입력 후 ENTER키를 누르면 저장되며 자신의 순위가 나타나고 확인 버튼을 누르면 다시시작버튼과 종료버튼이 뜬다.

4-1-2 저장여부 NO: 다시시작버튼과 메인화면버튼이 뜬다.

4-2 다시시작버튼을 누르면 만두 캐릭터 선택화면으로 돌아간다.

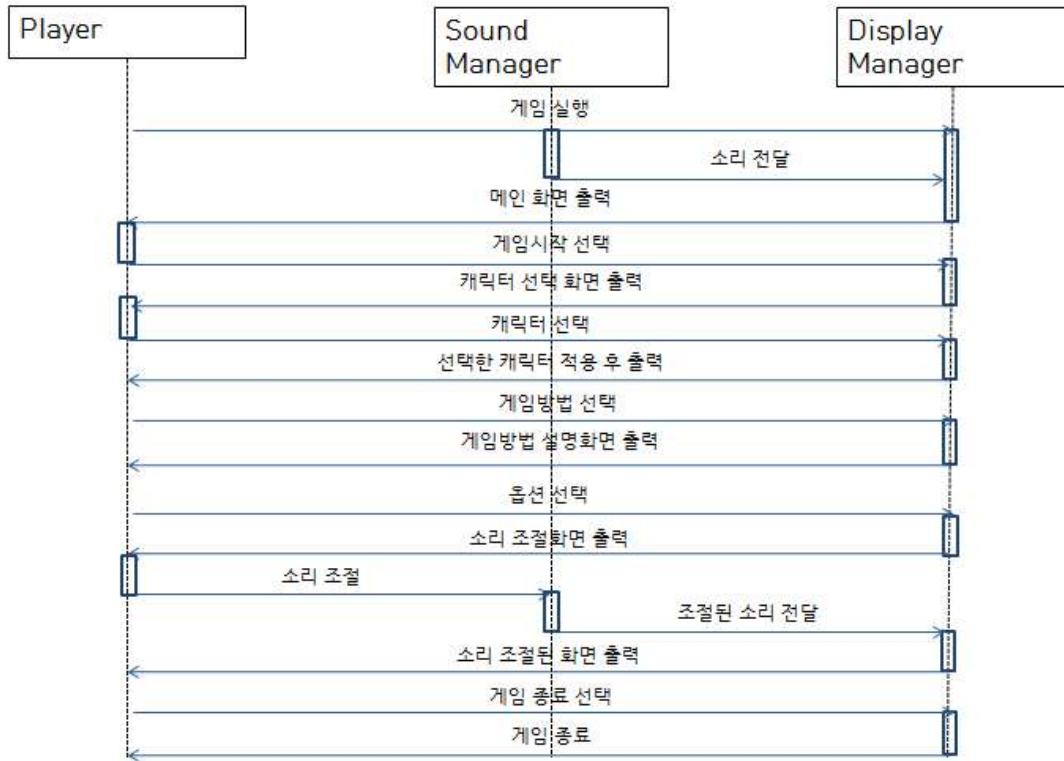
4-3 메인화면버튼을 누르면 1번으로 돌아가게 된다.

< Usecase Diagram >

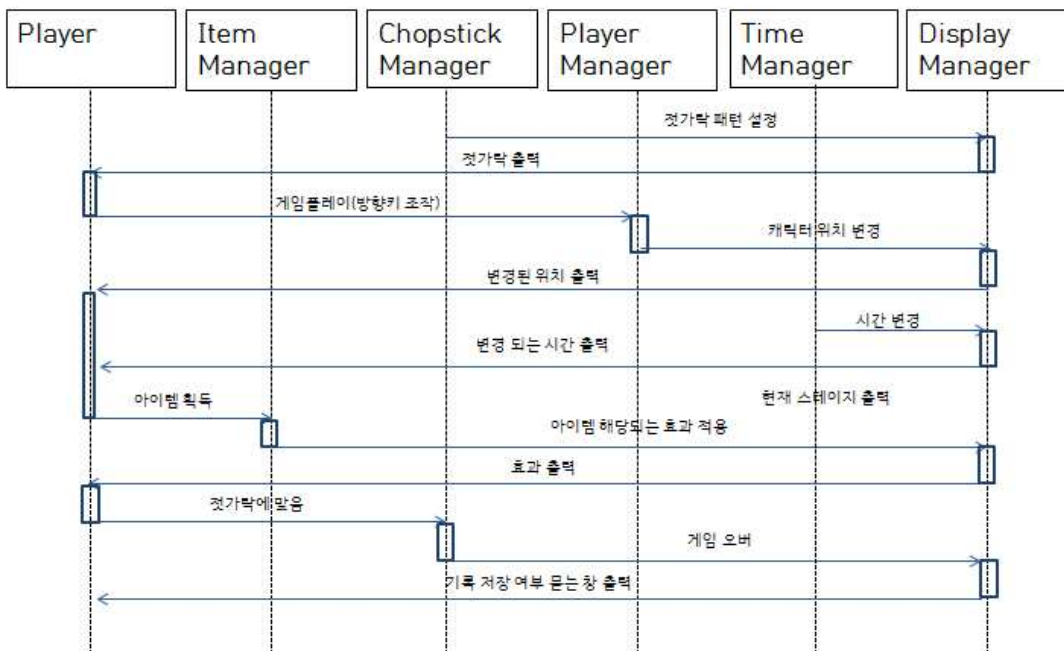


< Sequence Diagram >

- 메인 화면



- 게임 진행 화면



- 랭킹 창

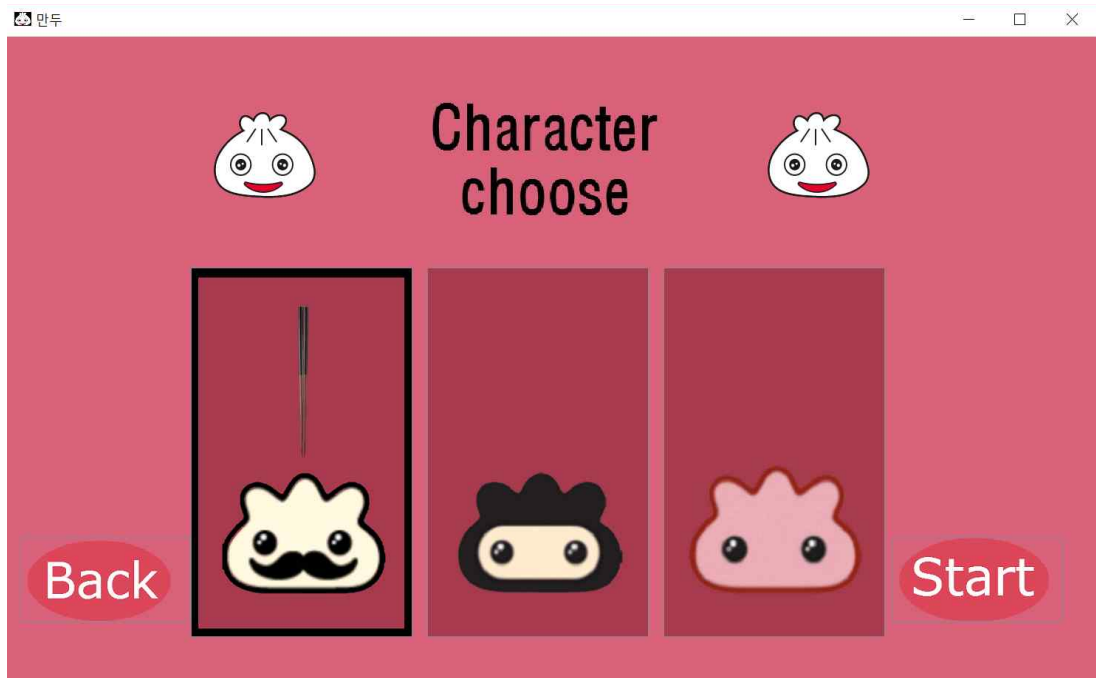


< User Interface Designs & Screenshot of application demo >

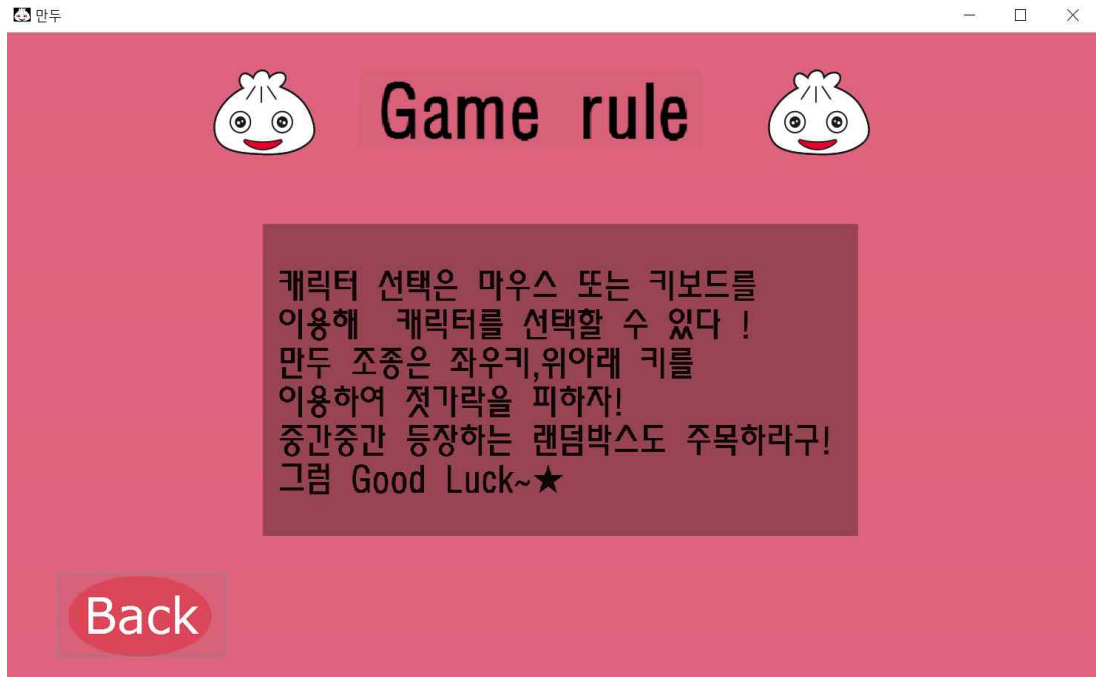
- 메인화면



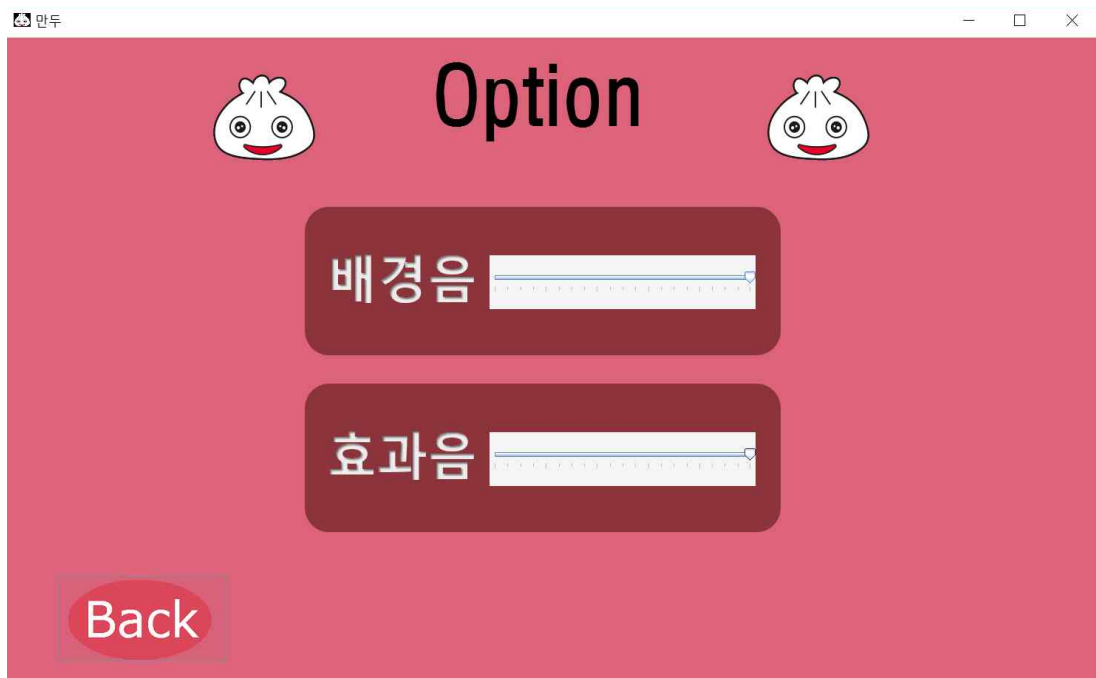
- 캐릭터선택화면



- 게임방법화면



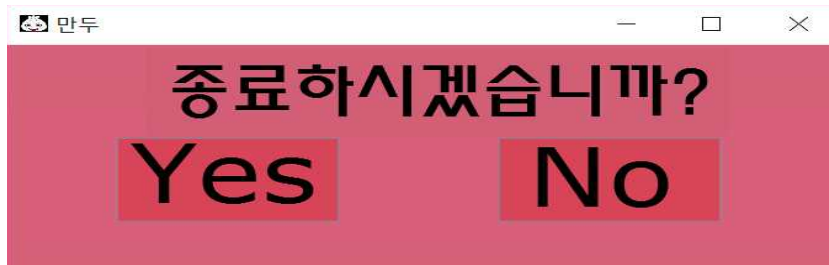
- 옵션화면



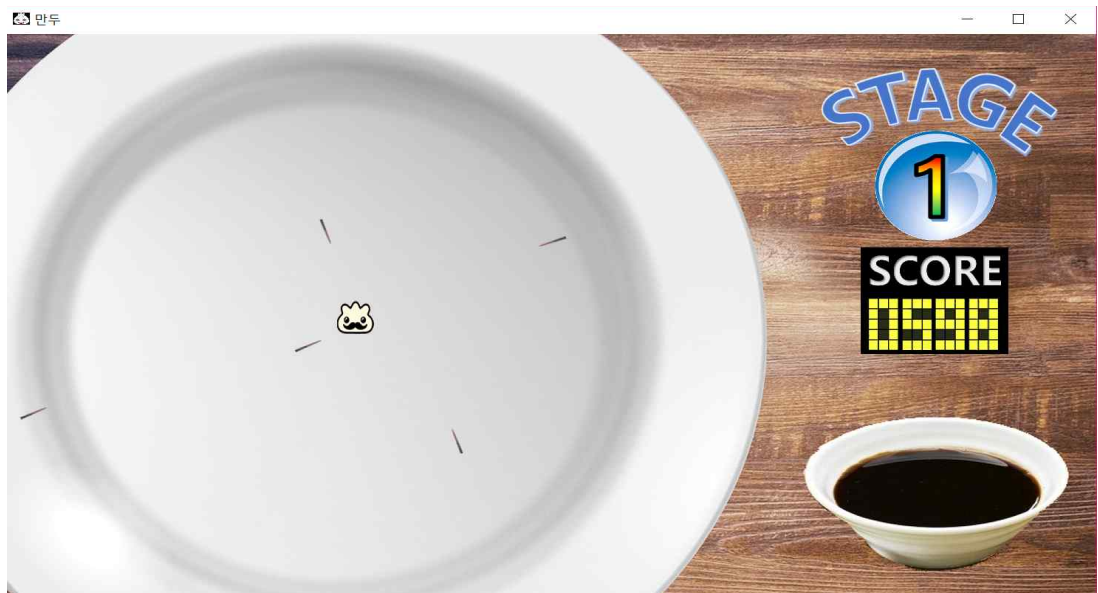
- 랭킹화면



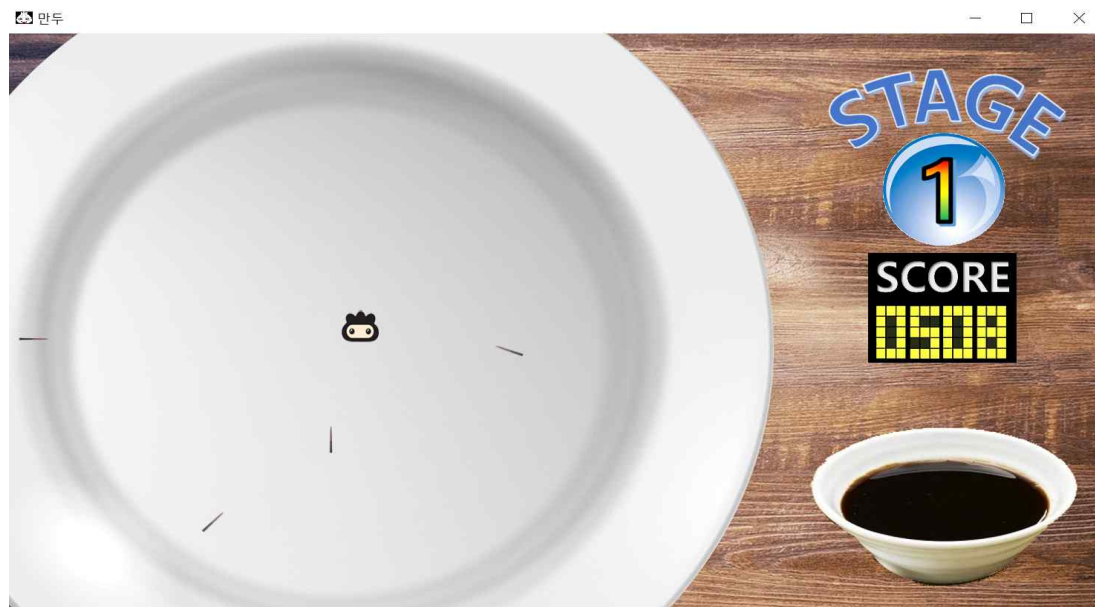
- 게임종료화면



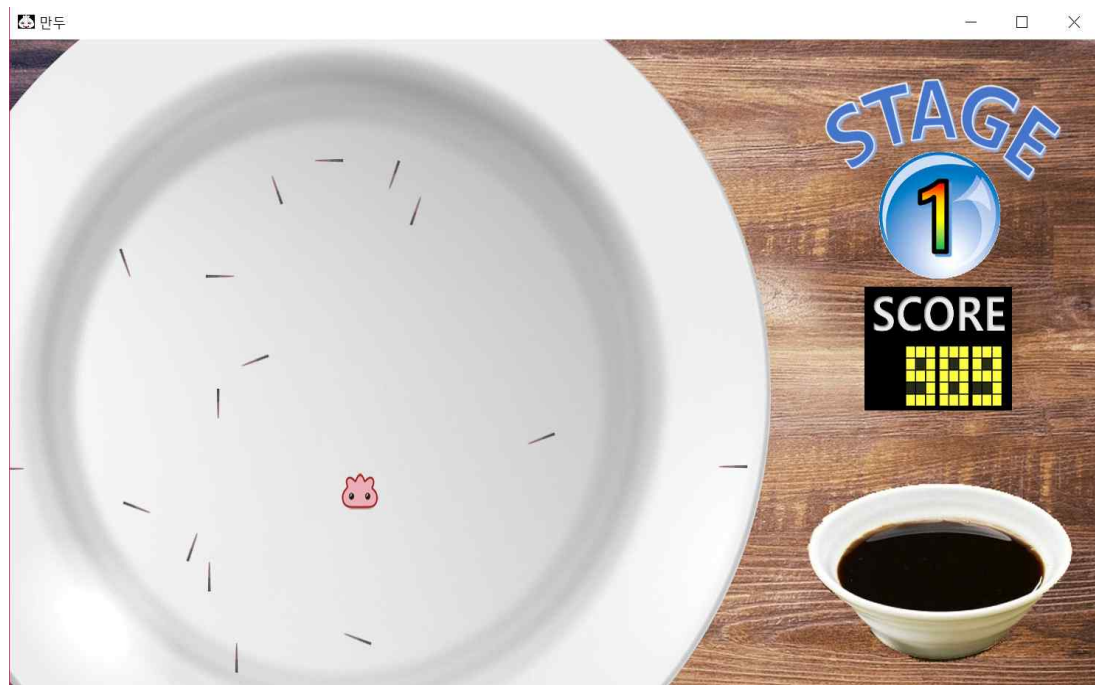
- 게임진행화면(만두 1)



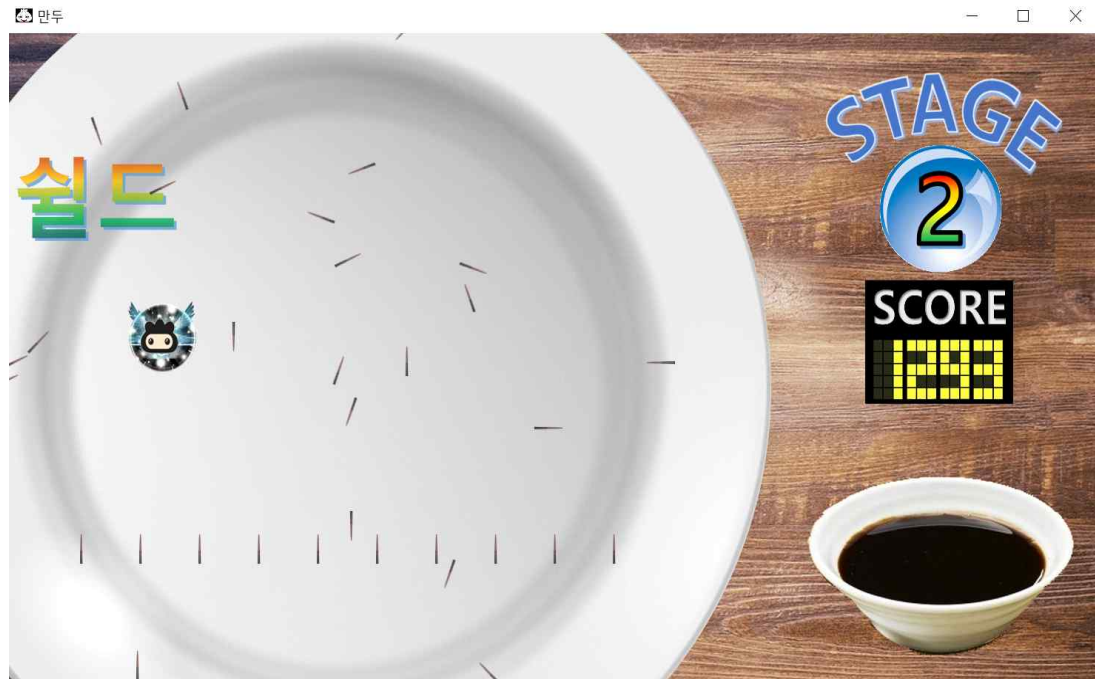
- 게임진행화면(만두 2)



- 게임진행화면(만두 3)



- 아이템(실드)



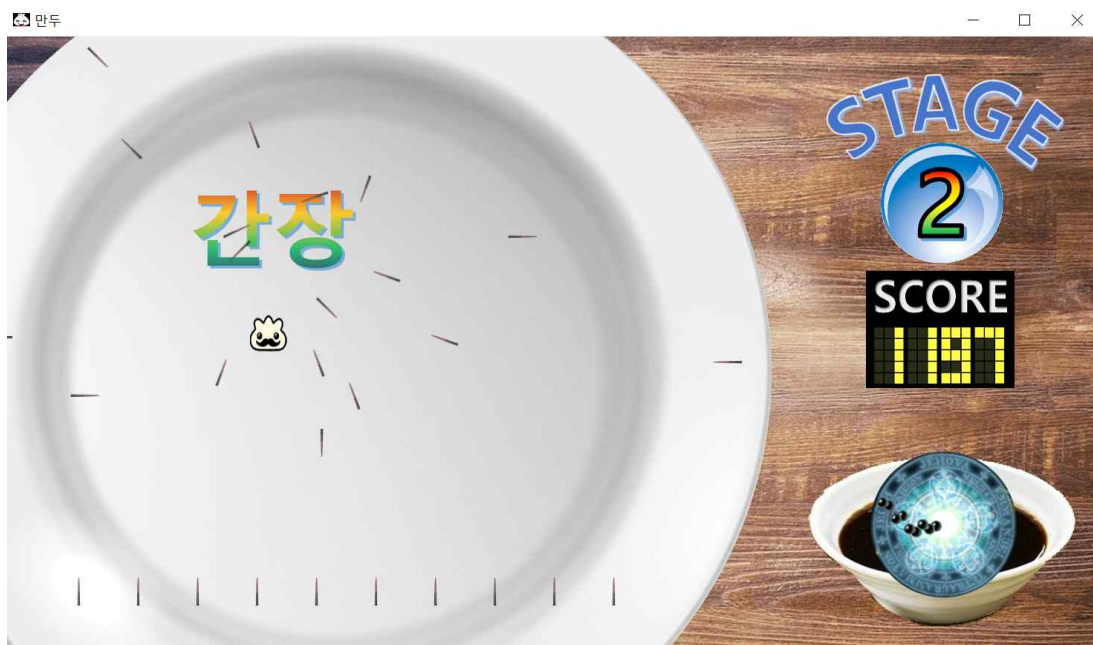
- 아이템(거대화)



- 아이템(소형화)



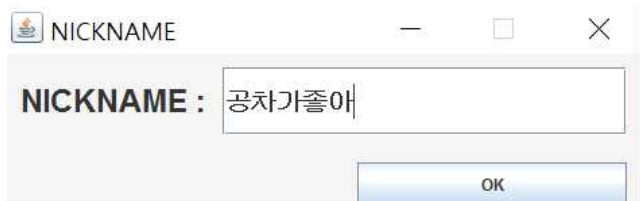
- 아이템(간장)



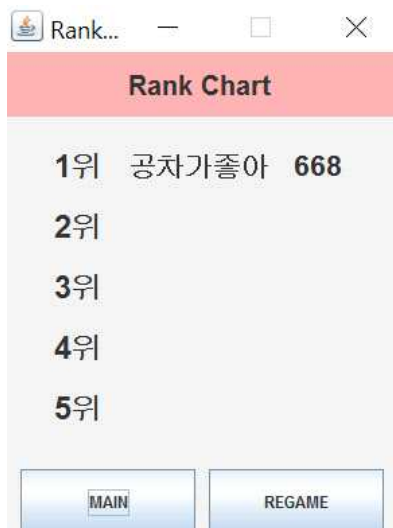
- 게임오버화면



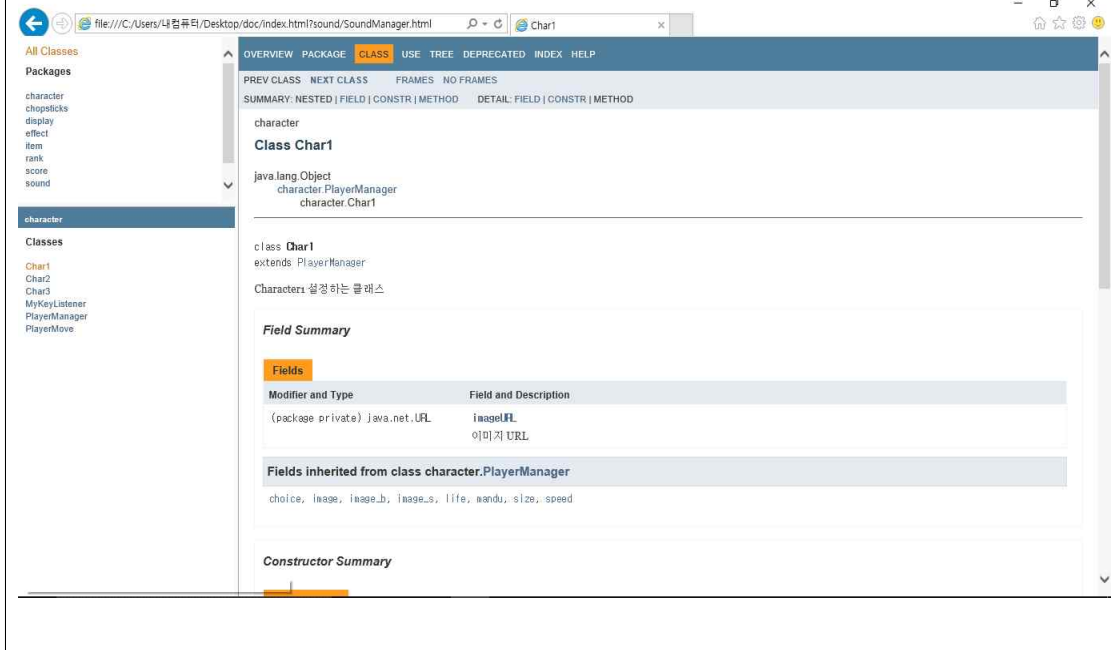
- 닉네임입력화면



- 랭크차트화면



< Screenshot of API Manual >



< 각 팀원 별 Work Distribution 및 진행 결과 >

코드 전체 라인 수 : 3571

나희지 (770) :

저는 display manager를 담당하여 메인화면, 게임방법, 옵션, 랭킹, 캐릭터 선택, 게임 스테이지 배경 GUI를 구현했습니다. JFrame을 상속받는 display Manager 클래스를 만들어서 Main 클래스 및 Howto, Option, Character_Choose, Game_stage 클래스가 display Manager 클래스를 상속하게 합니다. Main화면에서 게임방법버튼을 누르면 Howto클래스를 불러와서 게임 방법 창으로 넘어오고 Exit클래스는 새로운 창을 띄우므로 JFrame을 상속합니다. Exit버튼을 누르면 Exit클래스가 호출되어 종료여부를 한 번 더 묻는 창을 띄워서 No버튼을 눌렀을 경우 Main창이 남아있게 하고 Yes일 경우 시스템 자체가 종료되게 하였다. Ranking 클래스에서는 이 때까지 저장된 상위 5명의 닉네임과 점수를 보여주는 클래스인데 이 닉네임과 점수는 Ranksave(nick).txt과 Ranksave(score).txt에 저장되어 있습니다. 두 파일에 저장된 닉네임과 점수를 얻어오기 위해선 rank manager의 Rank클래스를 상속받아 OpenFileDialog()이라는 메소드를 사용하여 닉네임과 점수를 얻어옵니다. 대신 프레임을 만들기 위해서 Ranking클래스에 JFrame객체를 불러왔습니다. 그래서 Main의 랭킹버튼을 누르면 파일에 저장되어 있는 순위와 점수를 불러옵니다. Character_choose 클래스 내부에 있는 keyListener클래스와 MouseListener클래스를 이용하여 좌우 방향키

와 마우스로 해당 캐릭터 객체를 호출시켜서 Game_stage에 가져다 놓았습니다. 그리고 캐릭터를 아무것도 선택하지 않고 start버튼을 누를 시에는 자동으로 1번 캐릭터가 호출되는 요구사항을 구현하였습니다. Main에 있는 JButton을 통해 이동한 화면들은 MouseListener클래스를 이용해 만든 Back버튼을 통해 Main화면으로 돌아갈 수 있습니다. Option클래스와 Game_stage클래스에서는 GUI만을 구현하였습니다. 각 요구사항 기능 구현 정도는 100%이고 부족한 부분은 찾지 못했습니다.

변동건 (1629) :

Chopstick Manager, Sound Manager, Timer, Effect Manager, item의 간장, display의 옵션을 맡았으며 100% 구현을 완료하였다. Effect Manager는 계획에 없었지만 프로젝트 진행 중 추가하게 되었다.

Chopstick Manager는 젓가락의 출력을 담당하는 최상위 클래스이다. Chopstick Manager를 상속받아 젓가락의 방향과 방향에 따른 이미지를 불러와 게임프레임에 출력하고 캐릭터가 젓가락에 닿을시 피격처리를 담당하는 Shooting클래스를 구현하였다. 또한 Shooting클래스를 이용하여 젓가락이 일정한 패턴으로 발사되게 Shooting클래스의 객체를 생성 시켜주는 5개의 Pattern클래스들을 구현하며 Pattern클래스들을 통합하여 관리하는 Patterns클래스를 구현하였다.

Pattern1은 젓가락이 임의의 방향에서 만두를 향해 날아가도록 하는 클래스이며 Pattern2는 6방향에서 젓가락이 날아가도록 한다. Pattern3는 Pattern1에 유도 기능을 추가하였고 Pattern4는 하단에서 Pattern5는 측면에서 젓가락이 1행으로 날아가도록 하는 클래스이다. 이 클래스들은 게임진행 중에 Patterns클래스에 의해 반복호출되며 게임화면에 출력된다.

젓가락에 캐릭터가 닿을 경우 캐릭터의 생명이 1줄어들고 실드가 없는 경우 생명이 0이 되며 모든 쓰레드를 중지시키고 종료화면을 불러온다.

Sound Manager는 사운드를 통합 관리하는 클래스이다. 각종 배경음과 효과음의 출력을 담당하며 다른 클래스에서 Sound Manager객체를 만드는 것으로 음악을 실행시킬 수 있도록 하였다. 배경음과 효과음의 음량정보를 외부 클래스에서 접근할수 있도록 set()메소드와 get()메소드를 구현하여 Display의 Option에서 출력되는 .wav파일의 음량을 조절할 수 있게 하였다.

Timer클래스는 객체를 생성하는 순간부터의 시간을 측정하는 역할을 하며 이를 이용하여 점수를 계산하고 게임화면에 출력하도록 하위클래스들을 구현한다. Score클래스가 시간을 점수로 변환하고 ScorePane클래스에서 Score클래스를 상속받아 현재 스테이지와 점수를 게임화면의 오른쪽에 출력한다. 스테이지는 점수는 1초당 100점씩 올라가도록 하였고 스테이지는 점수가 1000점일 때, 3000점일 때 올라가도록 하였다. 또 getScore()메소드를 구현하여 Rank Manager에 점수를 넘겨줄 수 있도록 하였다. 캐릭터가 젓가락에 닿아 죽었을 시에 점수가 오르는 것이 멈추며 점수를 Rank Manager에 넘겨준다.

Effect Manager는 게임 진행 중에 발생하는 이펙트들의 출력을 담당한다. Effect Manager를 상속받는 Effect클래스들이 간장이 날아갈 때의 이펙트나 만두가 젓가락에 피격 될 때의 이펙트 등, 각각의 이펙트들을 출력하도록 구현하였다.

간장은 랜덤박스에서 간장이 나왔을 경우 간장 이펙트와 간장이라는 텍스트가 화면에 출력되는 이펙트를 호출하며 간장종지에서 간장이 발사되며 캐릭터가 간장에 닿을시

Player Manager를 호출하여 캐릭터가 느려지도록 구현하였다.

옵션은 메인 화면에서 옵션버튼을 클릭하였을 때 옵션창이 나오며 2개의 슬라이더를 이용하여 배경음과 효과음의 크기를 각각 조절할 수 있도록 하였다.

심지영 (534) :

Player Manager 클래스를 전체적으로 맡아 캐릭터에 적용되는 전체적인 효과를 구현하였다. 캐릭터 선택 시 setCharacter 메소드에서 캐릭터 생성자를 불러 캐릭터가 각각 이미지만 다르고 기본적으로는 같은 기능을 하도록 구현하였고 MyKeyListener 클래스와 PlayerMove로 캐릭터가 키 입력을 받아 상하좌우로 움직이는 것과 boolean변수를 up, down, left, right를 이용하여 대각선으로도 움직이게 가능하도록 구현하였다. 캐릭터의 크기, 속도, 생명을 변경할 수 있도록 메소드 구현하였다.

Item Manager클래스에서는 간장을 제외한 랜덤박스와 거대화, 소형화, 실드를 맡아 스테이지가 바뀔 때 화면에 랜덤박스 이미지를 임의의 위치에 출력하여 6초간 이를 유지하였다가 캐릭터와 닿을 시에는 랜덤으로 해당 효과가 구현되도록 하였고 닿지 않으면 사라지도록 하였다. 거대화와 소형화는 Player Manager의 메소드를 불러와 캐릭터의 크기를 변화시키고 이를 약 4초간 유지하도록 하는 기능을 구현하였다. 실드는 실드 이미지를 캐릭터의 뒤쪽에 출력하고 life를 +1하며 화살을 한 번 맞을 때까지 이미지를 유지하고 만약 실드가 있는 상태에서 한번 더 실드를 얻어도 생명이 더 추가되는 것이 아닌 그 상태를 유지 하도록 구현하였다.

각 기능의 구현정도는 Player Manager와 MyKeyListener 클래스는 모든 기능들이 잘 구현되어 100%이고 PlayerMove 클래스에서 속도 조절이 잘 되지 않아 80%이다. Item과 관련된 클래스들은 랜덤 박스 출력과 각 아이템의 기능은 각각 잘 적용되지만 이따금씩 아이템을 먹지 않아도 아이템 효과가 실행되는 에러가 발생하여 80%이다.

하수연(638) :

저는 Rank Manager부분과 Display manager의 Over class만을 맡아 진행하였습니다.

Rank Manager의 구성은 Rank class, showNS_end class이고 Rank class의 메소드는 Input(int score, String nickname), OpenFile(), output()이 있고 showNS_end class의 메소드는 showNS_end() 생성자만 있습니다. Over class의 메소드는 Over(int score), enterNickname()이 있습니다. Over class의 YES, NO버튼 showNS_end class의 MAIN, REGAME 버튼이 클릭될 때 각각 다른 기능 등 해야 하는데 이를 진행하기 위해 MouseListener를 implements한 클래스를 만들어 두고 그 안의 mouseClicked(MouseEvent e) 메소드를 사용하여 버튼의 이름에 따라 다른 기능을 하도록 구현하였습니다.

진행 결과는 Display manager의 Over class는 Rank Manager의 Rank class를 상속 받습니다. 그 이유는 Rank class의 Input(int score, String nickname)메소드를 사용하기 위해서입니다. Over class는 만두가 젓가락에 닿아 게임이 오버된 후 나타나는 화면이며 오버 되었을 시 점수를 Score class의 getScore()라는 메소드를 사용하여 얻어옵니다. 닉네임은 Over class에 구현되어 있는 enterNickname()이란 메소드를 사용하여 얻게 됩니다. enterNickname() 메소드는 Over class에서 점수 저장 여부 버튼 중 YES 버튼을 누르면 나타나는 작은 창을 구현한 메소드인데 닉네임을 입력할 수 있는 TextField가 있으며 이

창의 OK버튼을 누르면 Rank class의 Input(int score, String nickname) 메소드가 실행되어 위에서 얻어온 점수와 입력한 닉네임이 인자로 들어감과 동시에 창의 닫히며 showNS_end class의 생성자가 동시에 실행됩니다. enterNickname()의 OK버튼은 MouseListener를 implements한 class를 사용하지 않고 ActionListener를 innerclass로 사용하여 구현하였습니다. 점수 저장여부 버튼 중 NO버튼을 누르면 showNS_end class의 생성자가 바로 실행됩니다. showNS_end()는 랭킹 차트를 보여주는 작은 창입니다. 랭킹 차트 창의 구성은 상위 5명의 닉네임과 점수를 보여주고 창 하단에 MAIN 버튼과 REGAME 버튼이 있습니다. MAIN 버튼은 게임의 메인화면을 돌아가게 하는 버튼이고 REGAME 버튼은 캐릭터선택 화면으로 돌아가게 하는 버튼입니다. showNS_end()에서 상위 5명의 닉네임과 점수를 얻어오는 방법은 Rank class를 상속하여 OpenFile()이라는 메소드를 통해서입니다. Rank class의 OpenFile() 메소드를 설명하기 이전에 Rank class의 Input(int score, String nickname) 메소드를 알아야 하는데 이 메소드는 점수와 닉네임을 저장하고 내림차순으로 정리하여 주는 메소드입니다. 닉네임과 점수를 각각 Ranksave(nick).txt과 Ranksave(score).txt이란 파일에 저장해주며 닉네임은 하수연/나희지/처럼 저장되어 들어가고 점수는 4567A5213A처럼 저장되게 됩니다. Input(int score, String nickname)에서 파일에서 얻어온 점수와 닉네임을 배열에 일시적으로 저장하기 때문에 배열의 범위를 벗어나지 않기 위해 범위가 벗어나기 전 파일에 다시 내림차순으로 정리한 일정 배열을 넣어줍니다. 다시 OpenFile() 메소드를 설명하자면 Input(int score, String nickname)에서 쓰고 정리하고 저장된 Ranksave(nick).txt과 Ranksave(score).txt 파일을 다시 열어 한 문자열 배열로 묶어 저장해 반환합니다. 예를 들어 OpenFile()[0]을 호출하게 되면 상위 1명의 "닉네임 1234"처럼 나타나게 됩니다. 그러므로 랭크 차트에서 OpenFile() 메소드를 사용하여 상위 5명의 닉네임과 점수를 나타나게 할 수 있습니다. Rank class의 output() 메소드는 파일에 있던 이전 내용을 스트링으로 반환해주는 기능을 합니다. 이렇게 Rank Manager와 Display Manager의 Over class를 만들어 실행시켜 보면 가끔 한 닉네임과 점수가 두 번나오는 경우가 있는데 그 것을 고치지 못하였습니다. 그 것 이외에 내림차순 정리 등은 잘 되었습니다. 그러므로 구현 상황은 80%정도라고 생각합니다.