1. 커피메뉴정하기. 크레이지 아케이드(미니판). 자판기에서 메뉴뽑기. 현금인출기 돈 예금, 출금. 음식 요리하기.
2. 코딩 , ppt , 결과보고서 작성, 자료 조사, 코드에 주석달기. 발표.

GUI 디자인. \

오늘 회의결과입니다.

회의 내용은 다음 주에 제출할 제안서 구성이었습니다.

1. 프로젝트 주제 결정: 이전에 진행하고 있던 타일맵 아케이드 게임을 수정해 타일퍼즐 게임을 만들기로 했습니다. 프로젝트명은 Orchard입니다.

제안된 의견: 지렁이 키우기, 분리수거 게임

2. 클래스 구조도: 프로젝트 구현을 위한 대략적인 클래스들을 생각해봤습니다. 구체적인 UML 다이어그램은 진행할 프로젝트 구현을 위해 필요한 내용들을 조사, 분석 후 11월 10일(일요일) 12시까지 제출하기로 했습니다.

3. 역할분담:

이성우 - 프로젝트 제안서 및 PPT 제작, 발표

그 외 - 프로젝트 설계 및 코딩(아직까지 구체적으로 결정된 바는 없고, 클래스 구조도가 어느정도 완성되면 세부적으로 분담할 예정입니다.)

4. 조장 선정: 이번 프로젝트 조장은 제가 하기로 결정됐습니다. 추가적으로 학기말에 점수 분배는 특별한 일이 없는 이상 모두 동일하게 하도록 합니다.

다들 능력자분들이어서 든든합니다! 이번 프로젝트 잘 해서 좋은 결과 얻도록 합시다 ^^

한 가지 더 저희가 만드는 게임 종류는 구체적으로 3 match game입니다! 관련된 키워드로 검색하시면서 찾아보시면 어떻게 클래스를 나누어야할지, 어떤 알고리즘이 필요할지 조금 더 쉽게 알 수 있을 듯 합니다!

**- 필요한 것들.**

**- 어떤 크기의 격자.**

**- 다른 과일 종류들.**

**- 두 과일을 바꾸는 메소드. -> swap() 메소드 구현.**

**- 만약 세개가 같은 과일이면 제거. -> 과일이 제거되면 다른 과일이 내려옴.**

**-> 하늘에서부터 새로운 과일이 내려옴.**

**- 필요한 것들.**

**- 어떤 크기의 격자.**

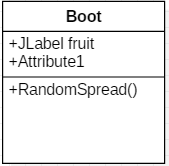
**- 다른 과일 종류들.**

**- 두 과일을 바꾸는 메소드. -> swap() 메소드 구현.**

**- 만약 세개가 같은 과일이면 제거. -> 과일이 제거되면 다른 과일이 내려옴.**

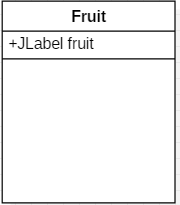
**-> 하늘에서부터 새로운 과일이 내려옴.**

start누르면 게임 실행.

**- Boot class** -> 초기에 과일들이 깔려있는 화면 출력.

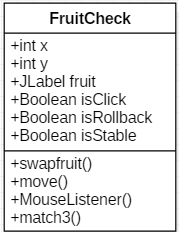
필요 data : JLabel fruit (초기에 깔릴 과일들)

필요 메소드 : RandomSpread() -> fruit을 랜덤하게 뿌려줌.

**- Fruit class** -> 과일의 종류들 관리.

필요 data : JLabel fruit(과일의 종류 개수만큼 할당.)

필요 메소드 :

**- FruitCheck class** -> 과일 check하여 3개가 연속이면 터지기. -> 위에서 다른 과일 내려옴. (터지는 동작 수행 후)

필요 data : JLabel fruit(과일 종류만큼) int x, int y (좌표값), Boolean isClick(클릭했는지 여부), Boolean isRollback(다시 돌아갈지 여부), Boolean isStable

필요 메소드 :

-> swapfruit() : 이 게임의 유일한 입력. Swap은 마우스 입력을 받고 드래그 감지하여 클릭한 fruit과 드래그 한 방향의 fruit이 swap 되면서 이루어진다.

두개의 과일 위치를 교환. 입력 한 즉시가 아닌 fruit이 실제로 화면에서 움직이는 행위가 이루어지고 난 후 다음 동작(match 또는 rollback)이 이루어져야 한다.

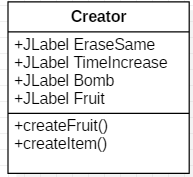
-> MouseListener() : 마우스 입력을 받는 부분.

mouse 입력이 발생하면, MouseListener 이벤트 핸들러가 호출되고 해당 이벤트와 마우스 좌표가 매개변수로 입력. MouseDown 발생시 현재 좌표 저장. MouseMouve 발생할 때마다 저장한 좌표와 현재 좌표 비교하여 드래그 감지. 드래그 감지 했으면 Swap() 호출.

-> Move() : 실제로 fruit을 움직이게 하는 부분.

자신이 가지고 있는 x,y변수를 변경.

-> Match3() : 연속된 fruit이 3개면 삭제. (가로나 세로로)

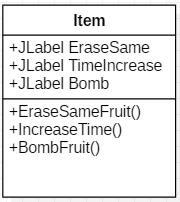
**- Creator class** -> 과일 터졌을 시 새로운 과일 위에서부터 추가. 아이템도 위에서부터 추가.

필요한 data : JLabel EraseSame (같은 fruit 제거), JLabel fruit(추가할 과일들), JLabel TimeIncrease(시간 증가), JLabel Bomb(주변 fruit제거).

필요 메소드 :

-> createFruit() : 위에서부터 새로 추가될 fruit을 생성해주는 메소드. (어떤 fruit이 추가될지는 random)

-> createItem() : 위에서부터 새로 추가될 Item을 생성해주는 메소드. (어떤 Item이 추가될지는 random)

**- Item class** -> 게임에 도움이 되는 Item관리.

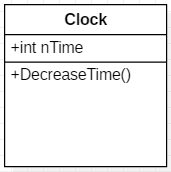
필요한 data : JLabel EraseSame (같은 fruit 제거), JLabel TimeIncrease(시간 증가), JLabel Bomb(주변 fruit제거). 아이콘들.

필요한 메소드 :

EraseSameFruit() -> 같은 과일을 없애주는 메소드.

IncreaseTime(int t) -> 제한시간을 일정시간만큼 추가해주는 메소드

BombFruit() -> 주변 fruit을 제거해주는 메소드.

**- Clock class** -> 게임이 종료하기까지의 시간 제한 관리.

필요한 data : int nTime(제한시간).

필요한 메소드 :

DecreaseTime() -> 시간이 초당 1씩 감소하는 메소드.