**JAVA Project UML 다이어그램.**

**- 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.**

**- Item 종류들.**

**- Time class – 게임 제한시간등의 시간 관리.**

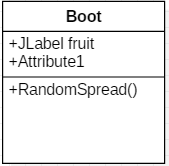
**- Boot class – 초기화면 출력.**

**- Fruit class – 과일종류들 관리.**

**- FruitCheck class – 과일이 3개연속인지, 또 터지면 위에서 새로 내려오는지 등등.**

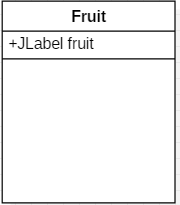
**- Creator class – 새로운 과일 추가, Item 추가.**

start누르면 게임 실행.

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

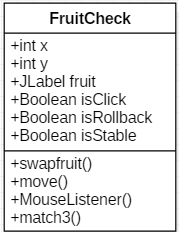
필요 data : JLabel fruit (초기에 깔릴 과일들)

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.

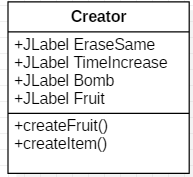
-> MouseListener() : 마우스 입력을 받는 부분.

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)

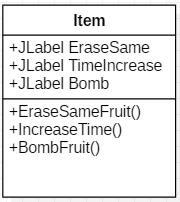
**- Creator class** -> 과일 터졌을 시 새로운 과일 위에서부터 추가. 아이템도 위에서부터 추가.

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.

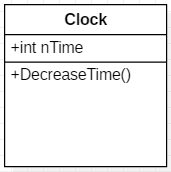
필요한 data : JLabel EraseSame (같은 fruit 제거), JLabel TimeIncrease(시간 증가), JLabel Bomb(주변 fruit제거). 아이콘들.

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.