소프트웨어프로젝트 결과보고서

제출일: 2015년 12월 7일

학년/학번	2 / 20121270	이름	조현우
학년/학번	2 / 20110275	이름	김영현
레벨	3	문제번호	1
문제명	SimFarm		

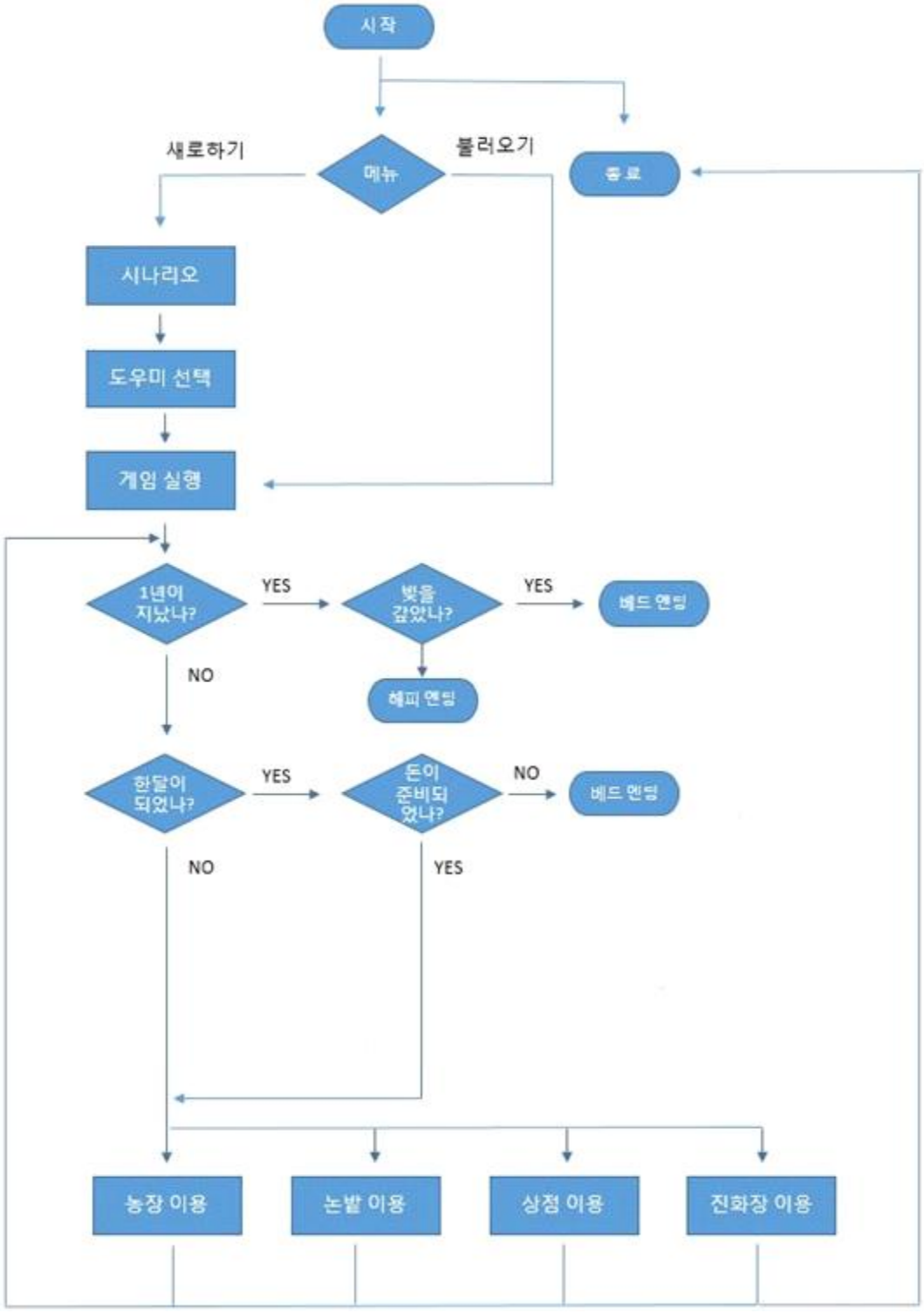
※ 구현 결과 요약

필요한 기능	완성 여부 (○, △, ×)	기본 배점	설명
시뮬레이션 방식 실행	○	3	
메뉴 또는 키 입력에 따른 옵션 처리	○	2	
3가지 이상 분석 결과 제시	○	2	분석 및 설계서에 기술
시나리오 작성	○	2	분석 및 설계서에 기술
화면 설계 결과 제시	○	2	분석 및 설계서에 기술
5가지 이상 캐릭터 포함	○	2	
3단계 이상의 기능 또는 능력 제공	○	2	
2개 이상의 쓰레드 사용	○	2	
GUI 구현	○	3	
배경음악	○		
세이브/로드	△		

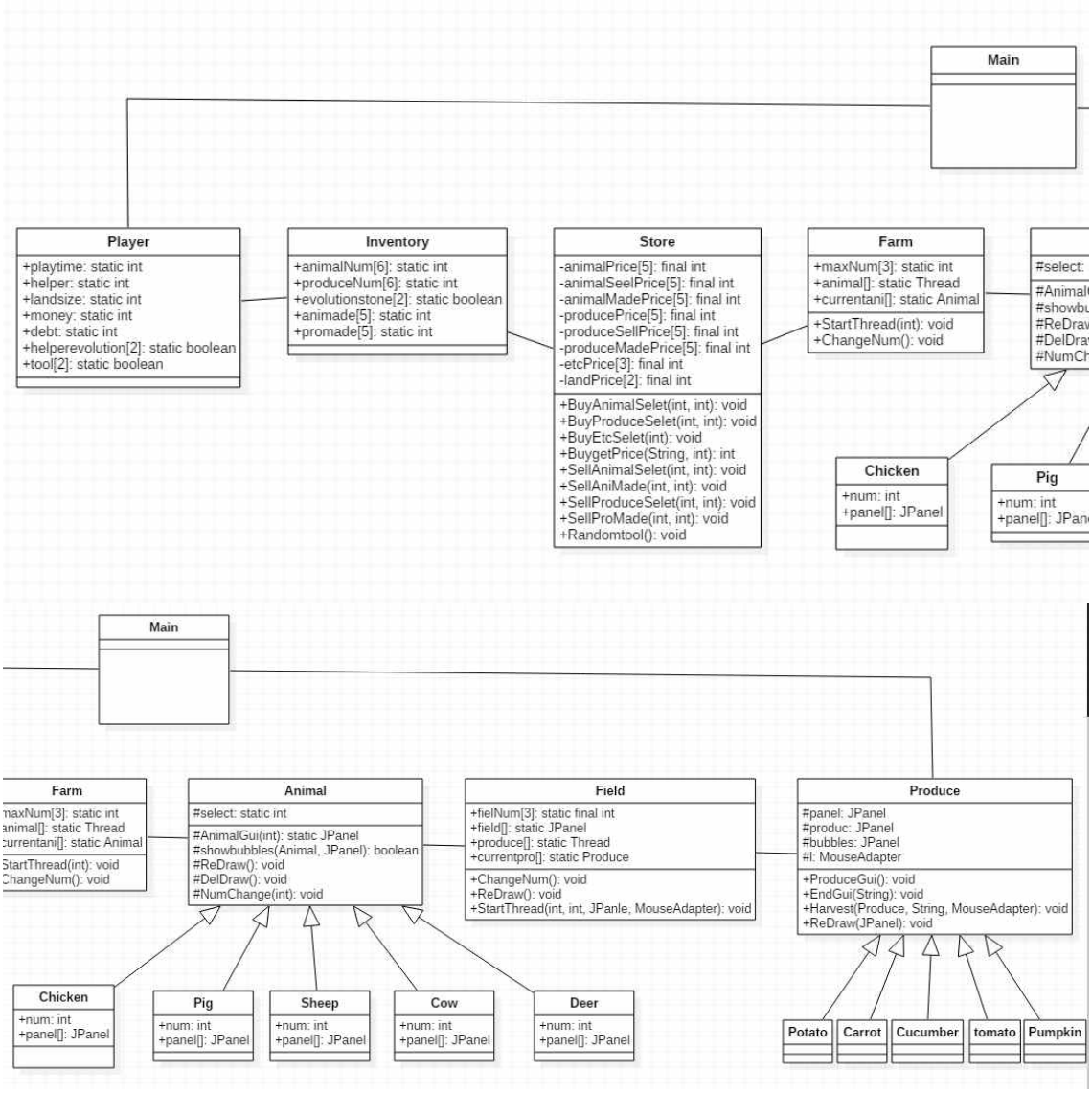
1. 서론

현재 나와 있는 게임들을 분석하여 동물농장을 구성하고 동물들을 키워 판매하거나 부산물을 판매할수 있는 게임을 개발한다.

2. 실행 흐름도



3. 시스템 구성



4. 설계구성요소 및 설계제한 요소

설계 구성요소						설계 제한요소						
목표 설정	합성	분석	구현/ 제작	시험/ 평가	결과 도출	성능	규격/ 표준	경제 성	미 학	신뢰 성	안정성/ 내구성	환경
●		●	●	●	●	●	●			●	●	

1) 설계구성요소

① 목표 설정

- 현재 나와 있는 시뮬레이션 게임들을 분석하여 동물농장을 구성하고 동물들을 키워 판매하거나 부산물을 판매할수 있는 게임을 개발한다. 게임의 최종목표는 시나리오에

의해 정해진 빚을 모두 갚는 것으로 한다.

② 분석

- 기존의 전략시뮬레이션의 중 문명이나 스타크래프트의 경우 처음 사용자가 플레이할 국가, 종족등에 따라 고유의 유닛 혹은 특수능력을 가지고 있다. 이를 이용해 게임실행시 5가지 캐릭터중 하나를 골라서 각 캐릭터 마다 고유의 능력을 부여한다.
  또한 위의 게임들은 같은 게임지만 사용자에 따라 다른 플레이를 할수 있다. 예를 들어 스타크래프트의 경우 빌드에 따라 다른 플레이가 나오는 것을 볼수 있다. 이를 이용하여 돈을 모으는 방법에 있어서 사용자가 여러 가지 방법중 좋은 방법으로 게임을 플레이 할수 있도록 한다. 처음 시작시 초기 자본으로 동물을 선택하는데 있어 가격이 저렴한 동물 많이 살지 비싼 동물을 조금 사서 시작을 하는등 여러 방법을 제시할수 있도록 한다.

③ 구현/제작

- eclipse를 사용 자바프로그래밍을 한다.

④ 시험/평가

- 만들어진 프로그램을 실행하여 확인을 한다.

⑤ 결과도출

- 농장을 운영 및 경영할수 있는 게임을 만든다.

2) 설계제한요소

① 성능

- 최종목표인 빚을 갚기 위해 무작정 기다릴 수 없으므로 게임 제한을 1년으로 둔다.

② 규격/표준

- 한달의 시간을 실제시간 10분으로 잡는다.
- 캐릭터 레벨에 따라 살 수 있는 동물 및 곡물수에 제한을 둔다.
- 동물의 종류 사슴(녹용),양(양털),소(우유),돼지(고기),닭(달걀)
- 곡물의 종류 벼, 보리, 감자, 호박, 콩

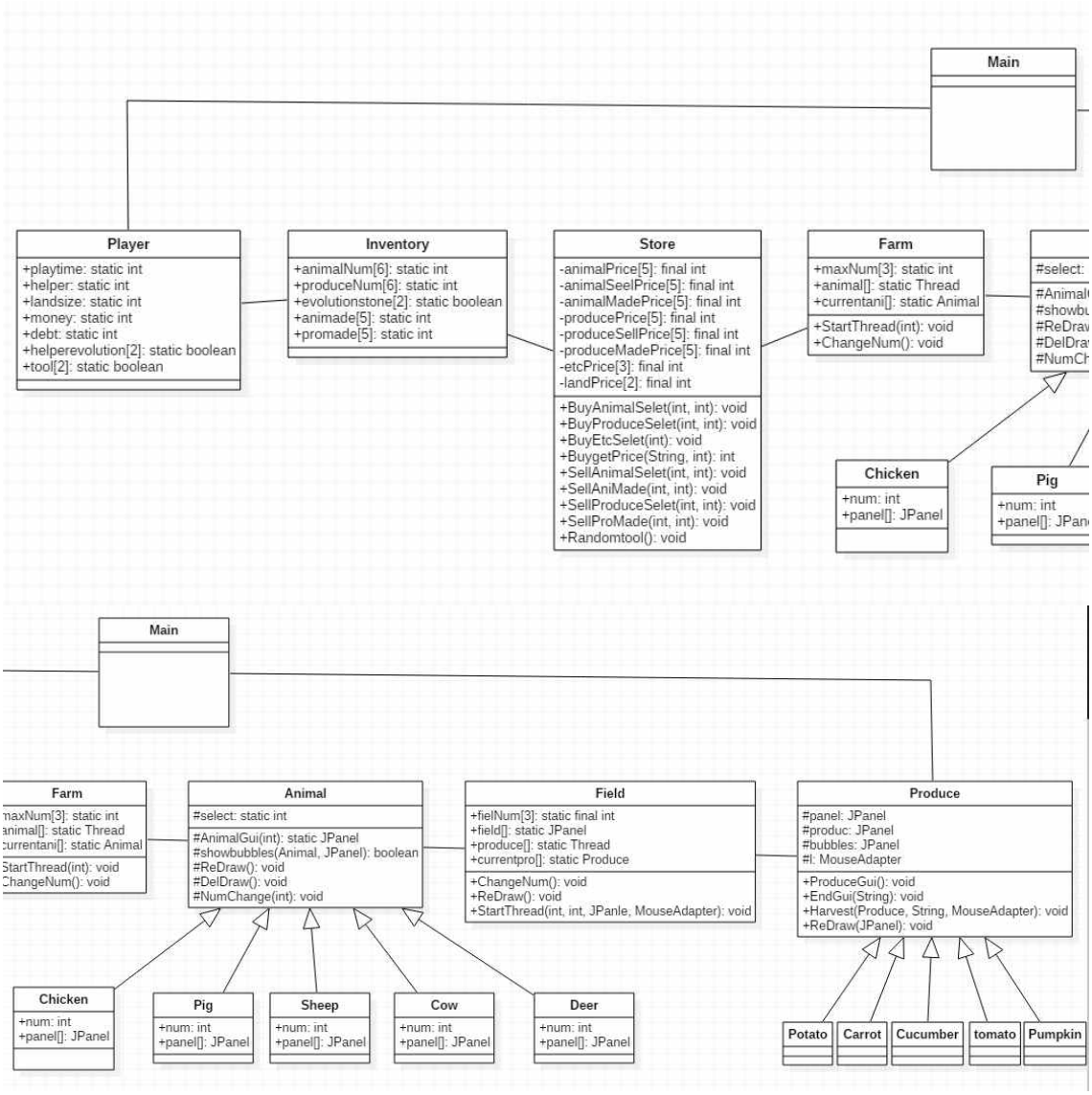
③ 신뢰성

- 이벤트이후 상태창을 확인함으로 버그가 있는지 확인을 한다.

④ 안정성/내구성

- 반복적 실행으로 여러 이벤트를 확인하여 버그의 유무를 확인한다.

5. 자료구조 및 모듈 설계



Main class : 메인 스레드가 존재하는 클래스

Player class : 플레이어의 정보를 저장하는 클래스

Inventory class : 게임상 아이템의 수량을 저장하는 클래스

Store class : 상점에서 구입과 판매를 관리하는 클래스, 각 상품의 가격을 저장

Farm class : 농장에 대한 클래스

Animal class : 각 동물에 대한 스레드를 가지고 있는 클래스들의 슈퍼클래스, 동물의 시간에 따른 변화를 GUI를 이용해 표현

Field class : 논밭에 대한 클래스

Produce class : 각 작물에 대한 스레드를 가지고 있는 클래스들의 슈퍼클래스, 작물의 시간에 따른 변화를 GUI를 이용해 표현

7. 주요 알고리즘

메인 스레드

```
public void run(){
    while(true){
        try{
            if(Player.playtime == 360){ //1년이 되었는지 확인
                if(Player.money >= Player.debt){ //가진돈이 남은 빚보다 많으면 해피엔딩
                    Player.debt = 0;
                    new CreditorDialog(true, Player.debt);
                }
                else ////가진돈이 남은 빚보다 많으면 해피엔딩
                    new CreditorDialog(false, Player.debt);
            }
            else if(Player.playtime%30 == 0){ //1달이 지났는지 확인
                if(Player.money >= 800000){ //가진돈이 한달 남부금액인 800000보다 많으면 게임 계속진행
                    Player.money -= 800000;
                    Player.debt -= 800000;
                    new CreditorDialog(true, 800000);
                    ChangeLabel();
                }
                else //부족하면 배드엔딩
                    new CreditorDialog(false, 800000);
            }

            Thread.sleep(20000); //20초당 하루
            Player.playtime++;
        }
        catch(InterruptedException e){
            return;
        }
        ChangeLabel();
    }
}
```

동물스레드

```
public void run() {
    panel  = AnimalGui(num); //농장에 동물의 크리를 그림
    while(true){
        try{
            if(Player.helper == 1){ //양 도우미를 선택하였는 지 확인
                if(Player.helperevolution[1] == true) //2단계 진화 하였는지 확인
                    Thread.sleep(5000);
                else if(Player.helperevolution[0] == true) //1단계 진화 하였는지 확인
                    Thread.sleep(7500);
                else //진화 하지 않았을 경우
                    Thread.sleep(9000);
            }
            else
                Thread.sleep(10000); //부산물 생성까지 기본 10초

            if(false == showbubbles(this, panel[1])){ //부산물을 생성하였다고 화면에 표현
                return;
            }

        }
        catch(InterruptedException e){
            return;
        }
    }
}
```

작물스레드

```
public void run() {
    ProduceGui(); //농장에 해당 식물의 그림을 그림
    try{
        if(Player.tool[0] == true){ //감자 도구가 있는지 확인
            Thread.sleep(7500);
        }
        else
            Thread.sleep(10000); //부산물 생성까지 기본 10초
        EndGui(Thread.currentThread().getName()); //농장을 다합므슬그 부산물이 생성되었다고 화면에 표현
        Harvest(this, Thread.currentThread().getName(), 1); //플레이어가 수확하였는지 확인하고 진행
    }
    catch(InterruptedException e){
        return;
    }
}
```

8. 개발 내용 및 실행 화면


SimFarm
new game
load
exit
skip
그는 잘나가는 회사의 사장이였다. 그러던 어느 날...
도우미 선택
돼지를 키울 때 고기를 10%(->25%->50%)더 빨리 획득 가능
선택
나가기
소유 금액 : 400000
0년 0월 1일
메 뉴
농 장
논 밭
상 점
진화장
MARKET
소유 금액 : 374000
0년 0월 6일
소유 금액 : 323000
0년 0월 10일
동물
작물
진화석&장비
구매
판매


SimFarm
소유 금액 : 323000
0년 0월 12일
메 뉴
조합
나가기
SimFarm
소유 금액 : 273000
0년 0월 15일
메 뉴
소유 금액 : 273000원
플레이 시간 : 0년 0월 15일
남은 빚 : 10000000원
도우미 : 돼지도우미(1단계)
땅의 크기 : 1단계(밭 9개, 동물 10마리)
장비 수 : 1개
동물 수 : 5마리
작물 수 : 0개
메 뉴
주인공 정보
도우미 정보
인벤토리
저장하기
게임종료
닫기
SimFarm
소유 금액 : 273000
0년 0월 17일
메 뉴
돼지를 키울 때 고기를 25% 더 빨리 획득 가능
메 뉴
주인공 정보
도우미 정보
인벤토리
저장하기
게임종료
닫기
SimFarm
소유 금액 : 273000
0년 0월 18일
메 뉴
인벤토리
닭 1
달걀 0
돼지 1
고기 0
양 1
양털 0
소 1
우유 0
사슴 1
녹용 0
감자씨앗 0
감자 0
당근씨앗 0
당근 0
메 뉴
주인공 정보
도우미 정보
인벤토리
저장하기
게임종료
닫기
SimFarm
메 뉴
빚 쟁 이 등 장
돈을 갚으셔야 합니다.
갚으실 금액은 8000000원 입니다.
예
designed by freepik.com
SimFarm
빚을 갚지 못하여 빚쟁이에게 잡혀갔습니다.
메뉴로 돌아가기
SimFarm
축하합니다. 빚을 다 갚았습니다.
메뉴로 돌아가기

9. 결론

시간에 흐름에 따라 동물의 부산물을 얻고 작물등 을 키워 돈을 벌어 빚을 갚는 게임을 만들었다.

아직 세이브와 로드부분이 완벽하게 되지 않지만 좀 더 노력해서 완벽하게 만들어야 한다.

10. 느낀 점

GUI를 이용한 게임을 처음 만들었는데 생각보다 힘이 들었다. 실제 상용화되고 있는 게임을 보고 대단하다는 생각이 들었다.