Computer
Software

Comercio

1333032 김정태
1333040 김윤환
1433034 김민수

목차

프로젝트 개요

시스템 기능(요약)

시스템 요구분석

개발 환경

역할 분담

DB설계 및 구축

시스템 구성도

시스템 주요 기능

시연

개발 후기

프로젝트 개요 (1/2)

Comercio란?

- 무역,상업,거래의 뜻을 가진 스페인어로 , 자신의 상단을 개설하고 관리하여 무역,거래,사업을 통해 성장하여 무역과 경제에 대한 원리를 이해하고 즐길 수 있는 WindowForm 기반의 게임 시뮬레이션 프로그램.

프로젝트 개요 (2/2)

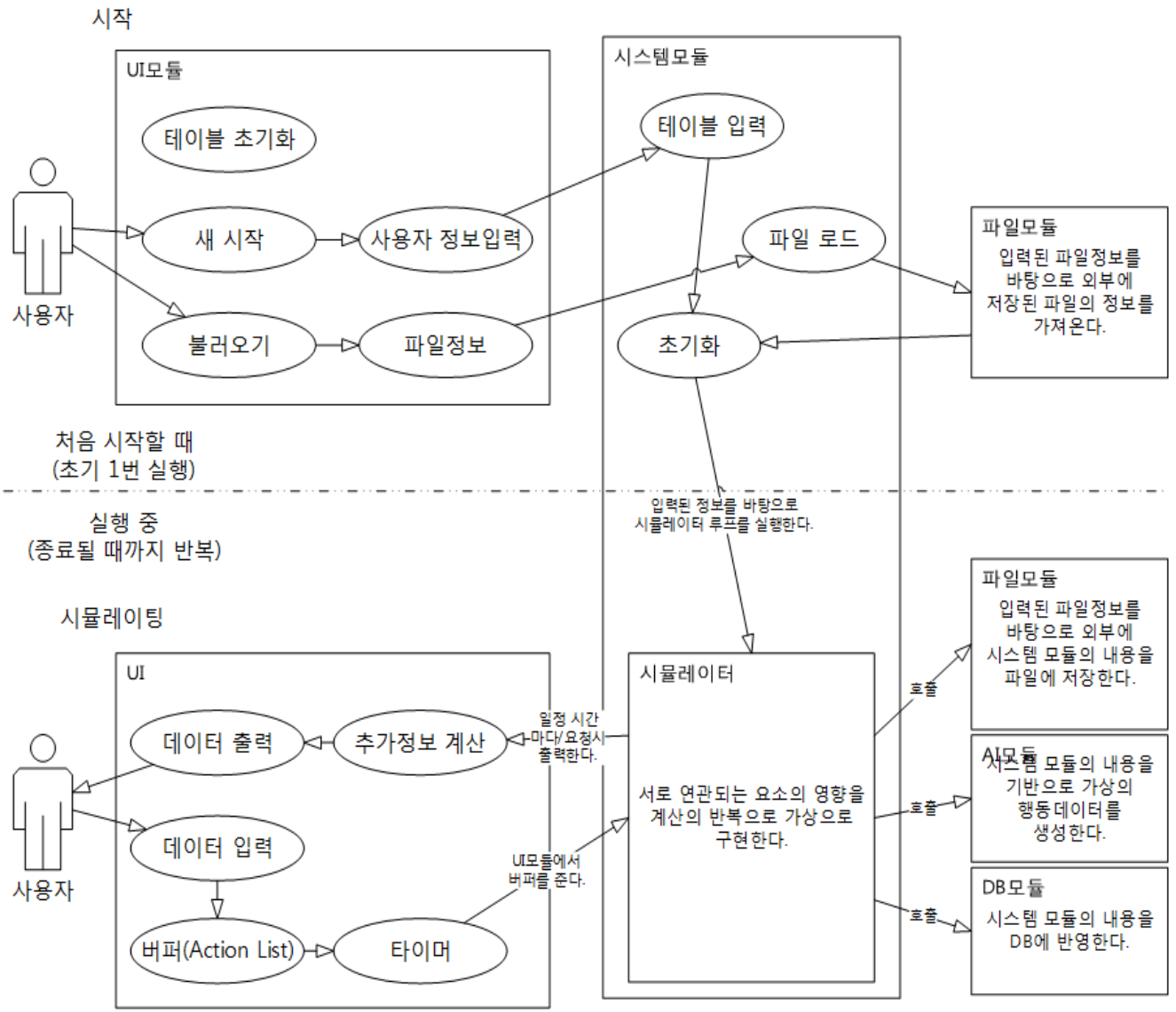
개발 동기

자원 하나 없는 척박한 땅에 살고 있는 대한민국은 무역과 경제활동을 통해 지금까지 왔다. 그래서 무역과 경제활동의 인재가 많으면 대한민국의 미래도 밝아질거라는 생각에 무역과 경제활동의 인재들을 위한 무역 시뮬레이션 프로그램을 재밌게 게임 시뮬레이션으로 만들었다.

개발 목표

RealTime기반의 데이터 입출력 , 플레이어를 상대 할 인공지능을 개발하여 윈도우 폼으로 만든 무역 관련 지식을 얻고 경제 원리를 경험으로 배울 수 있는 실시간 무역 시뮬레이션 게임을 목표로 한다.

시스템 기능 (요약)


시작
UI모듈
테이블 초기화
새 시작
사용자 정보입력
불러오기
파일정보
사용자
시스템모듈
테이블 입력
파일 로드
초기화
파일모듈
입력된 파일정보를 바탕으로 외부에 저장된 파일의 정보를 가져온다.
처음 시작할 때
(초기 1번 실행)
실행 중
(종료될 때까지 반복)
입력된 정보를 바탕으로 시뮬레이터 루프를 실행한다.
시뮬레이팅
UI
데이터 출력
추가정보 계산
데이터 입력
버퍼(Action List)
타이머
사용자
일정 시간마다/요청시 출력한다.
UI모듈에서 버퍼를 준다.
시뮬레이터
서로 연관되는 요소의 영향을 계산의 반복으로 가상으로 구현한다.
호출
호출
호출
파일모듈
입력된 파일정보를 바탕으로 외부에 시스템 모듈의 내용을 파일에 저장한다.
AI모듈
시스템 모듈의 내용을 기반으로 가상의 행동데이터를 생성한다.
DB모듈
시스템 모듈의 내용을 DB에 반영한다.

시스템 요구분석

프로그램

- 시간이 지날 때마다 물품의 수요와 공급 , 인구 , 가격의 수치가 변한다.
- 플레이어와 상대 할 인공지능의 무역 알고리즘을 구현한다.
- 게임 간의 일어날 수 있는 모든 공급 과 수요에 대한 변수를 구현한다.

데이터베이스

- 실시간으로 프로그램의 시스템 모듈과 데이터를 자주 받으므로 데이터베이스 모듈을 만들어 쉽게 프로그램과 데이터베이스간의 데이터 입출력을 원활히 설계한다.
- 프로그램의 목적에 맞게 테이블간의 관계와 속성을 설계한다.

개발 환경

운영체제

- Window 7

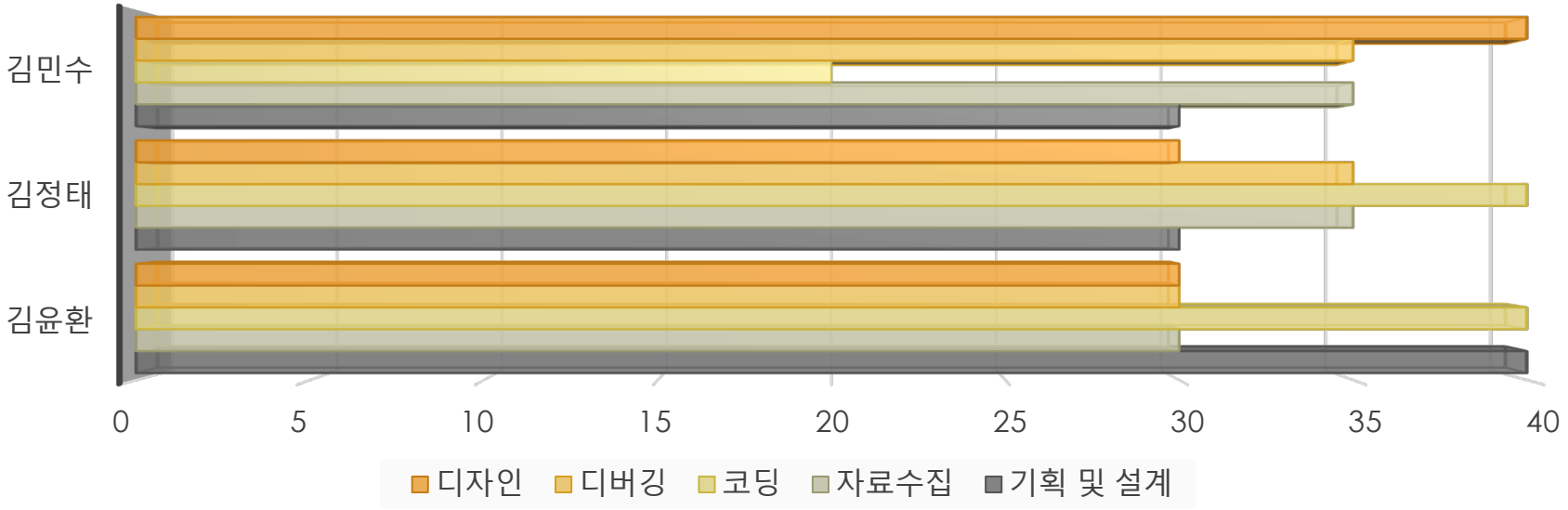
개발도구

- Oracle SQL Developer
- Visual studio 2015

개발언어

- C#

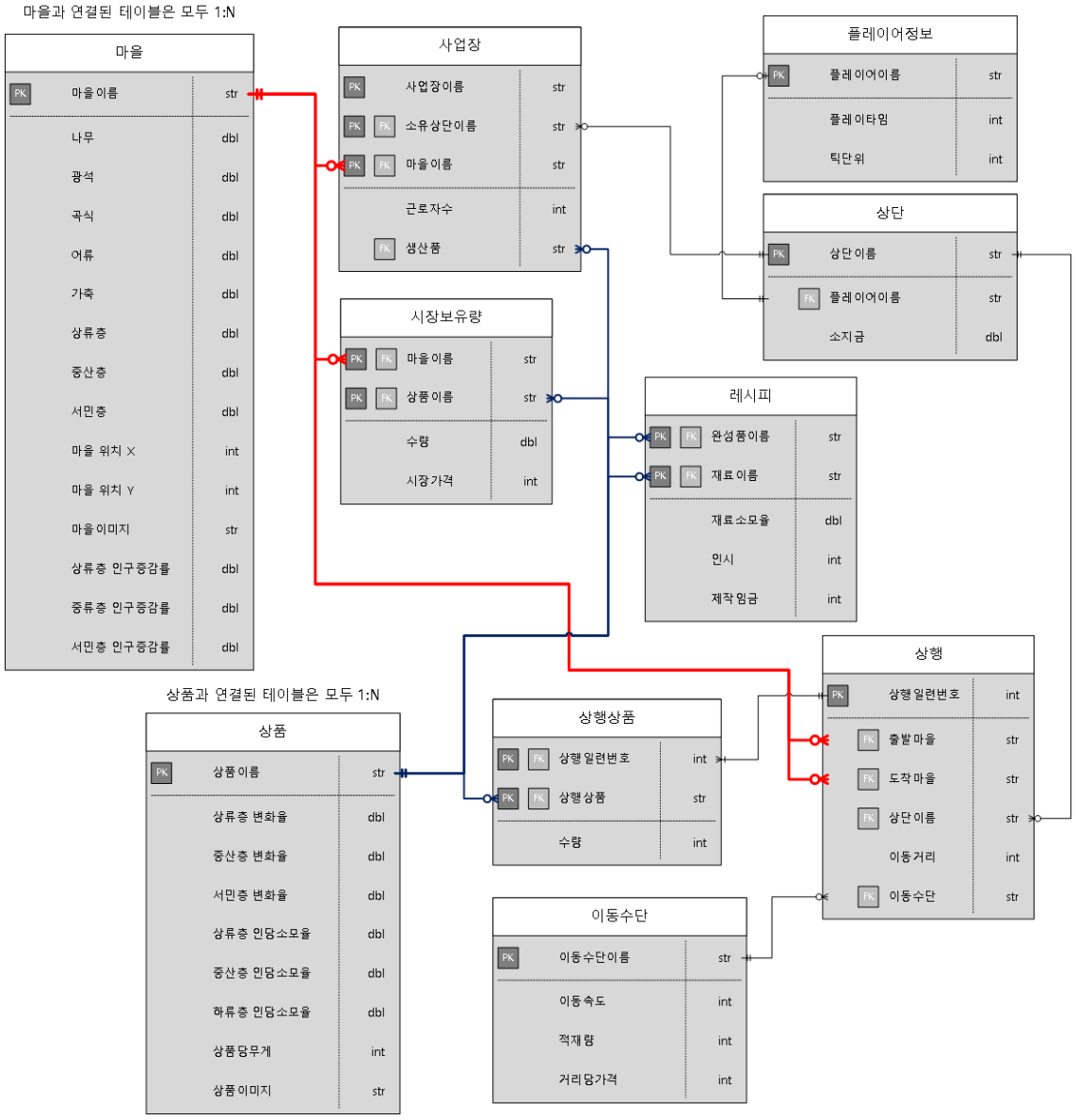
역할 분담



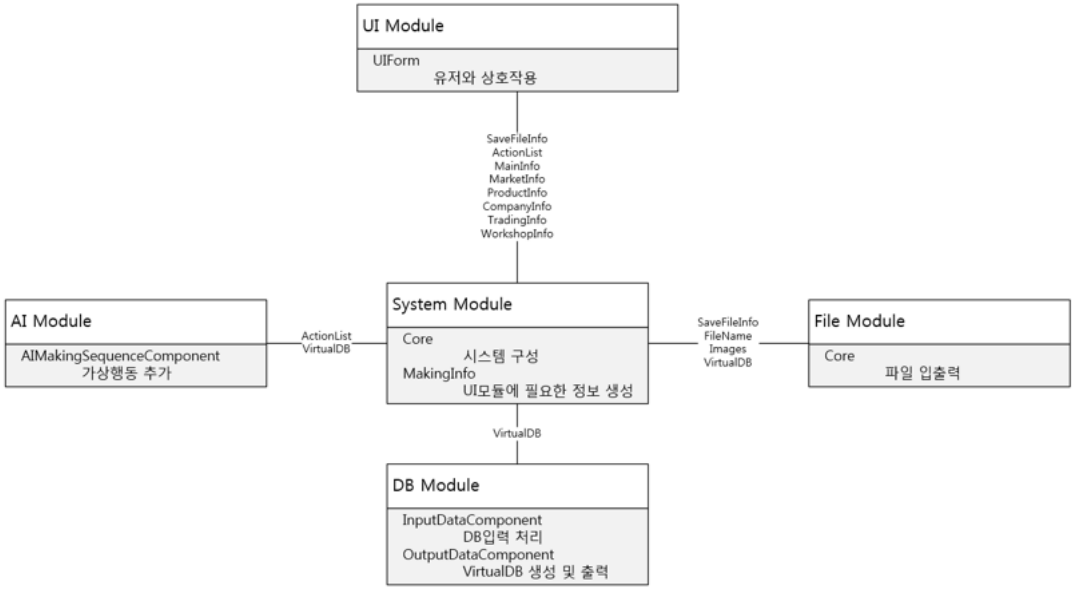
맡은 역할

- 김민수 : DB Module, UI 설계 , 리소스 수집
- 김정태 : File Module , System Module , UI 코딩
- 김윤환 : AI Module, System Module, UI 코딩

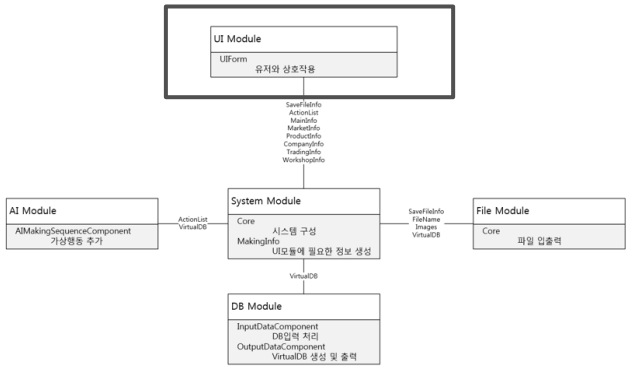
Db 설계 및 구축


마을과 연결된 테이블은 모두 1:N
마을
마을이름 str
나무 dbl
광석 dbl
곡식 dbl
어류 dbl
가축 dbl
상류층 dbl
중산층 dbl
서민층 dbl
마을 위치 X int
마을 위치 Y int
마을이미지 str
상류층 인구증감률 dbl
중류층 인구증감률 dbl
서민층 인구증감률 dbl
사업장
사업장이름 str
소유상단이름 str
마을이름 str
근로자수 int
생산품 str
시장보유량
마을이름 str
상품이름 str
수량 dbl
시장가격 int
플레이어정보
플레이어이름 str
플레이타임 int
틱단위 int
상단
상단이름 str
플레이어이름 str
소지금 dbl
레시피
완성품이름 str
재료이름 str
재료소모율 dbl
인시 int
제작임금 int
상품과 연결된 테이블은 모두 1:N
상품
상품이름 str
상류층 변화율 dbl
중산층 변화율 dbl
서민층 변화율 dbl
상류층 인당소모율 dbl
중산층 인당소모율 dbl
하류층 인당소모율 dbl
상품당무게 int
상품이미지 str
상행상품
상행일련번호 int
상행상품 str
수량 int
이동수단
이동수단이름 str
이동속도 int
적재량 int
거리당가격 int
상행
상행일련번호 int
출발마을 str
도착마을 str
상단이름 str
이동거리 int
이동수단 str

시스템 구성도 (1/6)


UI Module
UIForm
유저와 상호작용
SaveFileInfo
ActionList
MainInfo
MarketInfo
ProductInfo
CompanyInfo
TradingInfo
WorkshopInfo
AI Module
AIMakingSequenceComponent
가상행동 추가
ActionList
VirtualDB
System Module
Core
시스템 구성
MakingInfo
UI모듈에 필요한 정보 생성
SaveFileInfo
FileName
Images
VirtualDB
File Module
Core
파일 입출력
VirtualDB
DB Module
InputDataComponent
DB입력 처리
OutputDataComponent
VirtualDB 생성 및 출력

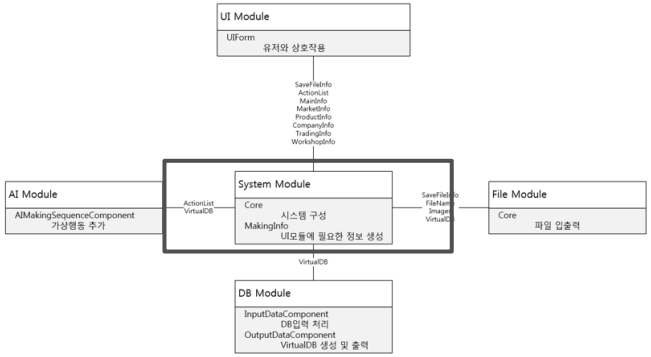
시스템 구성도 (2/6)



UI 모듈

1. 시작화면에서 사용자의 정보를 입력받고 초기 설정을 하거나 이미 있는 사용자 정보를 불러온다.
2. 시스템 모듈의 필요한 초기값을 주고 생선한다.
3. DB의 값을 불러와서 메인화면을 띄워주고 메인 타이머를 설정한다.
4. 사용자의 상단행동데이터를 입력받는다.
5. 타이머시간(100ms)마다 시스템모듈에 행동데이터를 전달한다.
6. To 폼이 올라오는 경우 , 즉시 데이터가 필요한 경우 시스템에 요청해서 값을 가져온다.
7. 사용자가 저장요청시 저장에 필요한 정보를 전달하면서 시스템에 저장을 요청한다.

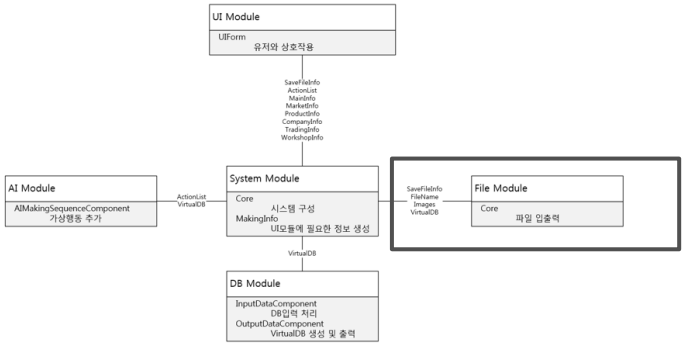
시스템 구성도 (3/6)



시스템 모듈

1. UI모듈에서 행동데이터를 전달하면서 프로그램을 동작시킨다.
2. 마을 소모를 반영한다.
3. 사업장 소모와 생산을 반영한다.
4. 상품이동을 반영한다.
5. 상품의 영향에 따라 마을인구를 증감시키고 상품의 수요에 따라서 가격을 증감시킨다.
6. 변경된 DB정보를 일정 횟수마다 DB Module에 전달한다.
7. UI에서 DB정보를 요청할 경우 DB Module에서 받은 DB데이터에 추가적인 정보를 입력 후 전달한다.

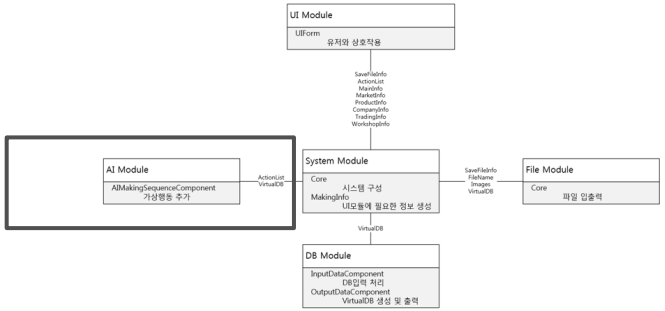
시스템 구성도 (4/6)



파일 모듈

1. 초기화된 VirtualDB (시스템내부에 있는 가상 데이터베이스)를 요구 받는 경우 (사용자가 게임을 새로 시작하는 선택을 한 경우), 프로그램과는 별도의 리소스 형태로 존재하는 JSON 방식으로 직렬화된 기초 VirtualDB 데이터 파일을 읽어 VirtualDB객체로 반환한다.
2. 저장된 플레이어 목록을 요구받는 경우, 게임 저장 폴더에서 저장된 파일들을 읽어 저장된 플레이어 목록을 PlayerInfo 객체 배열로 반환한다.
3. 기존 게임을 불러오기 위해 파일로 저장되어있던 VirtualDB를 요구받는 경우, VirtualDB 객체가 JSON 방식으로 직렬화된 저장 데이터를 읽어들여 VirtualDB 객체로 반환한다.
   이 때, 이미지가 존재하지 않는 경우에는 null을 반환한다.
4. 현재 게임을 저장하기 위해 VirtualDB 객체를 전달받는 경우 , 저장할 플레이어 이름도 함께 전달받고 VirtualDB 객체를 JSON 방식으로 직렬화하여 파일 형태로 저장한다. 저장하는 작업은 다른 스레드에서 처리하고 저장하는 작업이 끝나면 UI 모듈에 정의된 이벤트 핸들러를 호출한다.
5. DB 상의 상품 테이블을 튜플 형태로 존재하는 이미지가 요구될 때, 그 튜플의 "주소" 필드에 할당된 문자열을 전달받아 프로그램과 별도로 존재하는 리소스 폴더에서 해당하는 이미지를 동적으로 읽어오고 Image 객체 형태로 반환하여 준다. 이 때, 이미지가 존재하지 않는 경우에는 null을 반환한다.

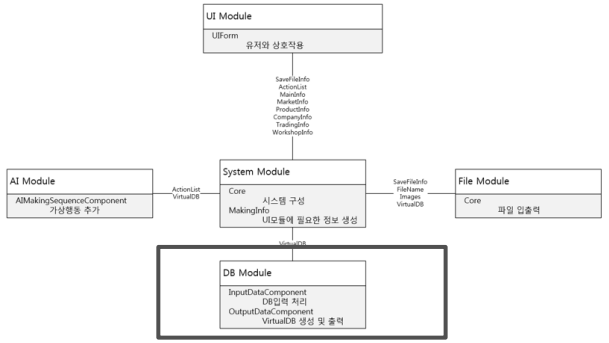
시스템 구성도 (5/6)



AI 모듈

1. 각 상단들의 현재 취급상품의 수량을 변경한다.
 - VirtualDB에서 Workshops와 Towns를 이용해서 수요를 Market에 저장한 뒤에 사업장 출하량을 계산 후 초과하거나 미달일 경우 사업장의 장인수를 변경한다.
 - 사업장의 상품을 만드는데 필요한 상품들(Recipes)의 재고를 알아내고(Towns) Market을 변경한다.
 - 변경사항을 ChangingWorkshopAction으로 만들고 ActionList에 추가한다.
2. 사업장 추가와 폐쇄 여부 결정 후 ActionList에 추가한다.
 - 사업장을 추가할 경우 CreatingWorkshopAction을 ActionList에 추가한다.
 - 사업장을 폐쇄할 경우 DestroingWorkshopAction을 ActionList에 추가한다.
3. 상행 추가
 - 마을의 초과 공급 상품을 알아내서 수요가 더큰 도시로 가는 상행을 CreatingTradingAction에 저장하고 ActionList에 추가한다.

시스템 구성도 (6/6)



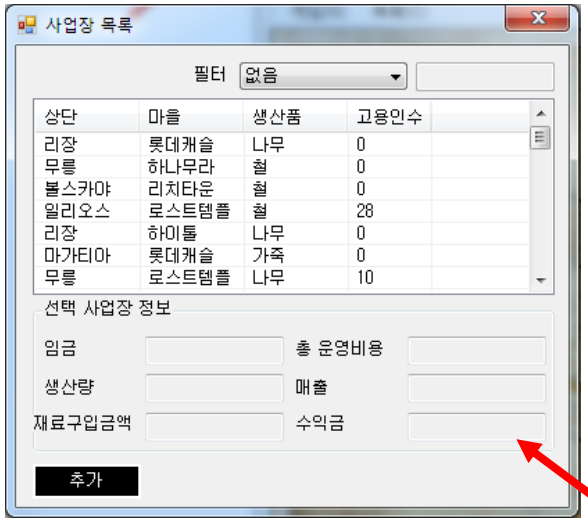
DB 모듈

1. 시스템모듈에서 가공된 데이터를 VirtualDB로 받아서 쿼리문을 통해 실 데이터베이스에 저장한다.
2. 시스템모듈에서 실 데이터베이스에있는 데이터를 요청할 경우 실 데이터베이스에 있는 데이터를 VirtualDB형으로 데이터가공을 하여 전달한다.
3. 사용자가 프로그램을 실행할때 실 데이터베이스에있는 데이터를 전부 비운다.

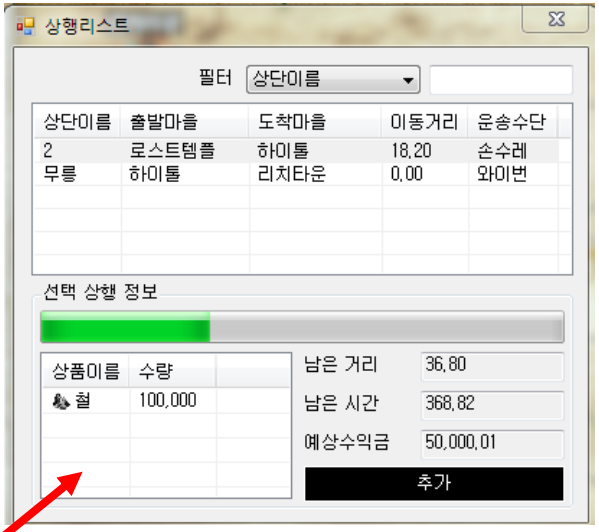
시스템 주요기능 (1/4)

€
€omércio
새로하기
불러오기
끝내기
새로하기
홍길동
홍길동상단
게임 속도(%)
100
AI 상단 수
5
마을 수
5
시작
취소
불러오기
이름
소지금
플레이팁
1
10000000
169
홍길동
10000000
22
시작
삭제
취소
Comercio
게임(G)
목록(C)
리치타운
하나무라
하이톨
로스트템플
롯데캐슬
comercio
리치타운 마을
인구
68.3%
13.4%
소유상단
생산품
마가티아
도끼
볼스카야
철
마가티아
나무
무릉
가죽
상세 사업장 정보
품명
시세
수량
나무
0,71
2,303
철
0,72
109
가죽
0,00
256
상세 시장 정보
내 정보
재산 :
10,000,000,00
수익률 :
0,00
사업장수 :
0
상행수 :
1
내 사업장
나의 사업장 수익가치 :
마을
생산품
수익률
선택 상행
운송상품
수량
운송 상품 수량
상행 기대수익
게임 시간
0001-01-05 오후 8:00:00
게임속도배율(%) :
100
Comercio v1.0
상행 총 127개
사업장 총 120개
상단 총 10개

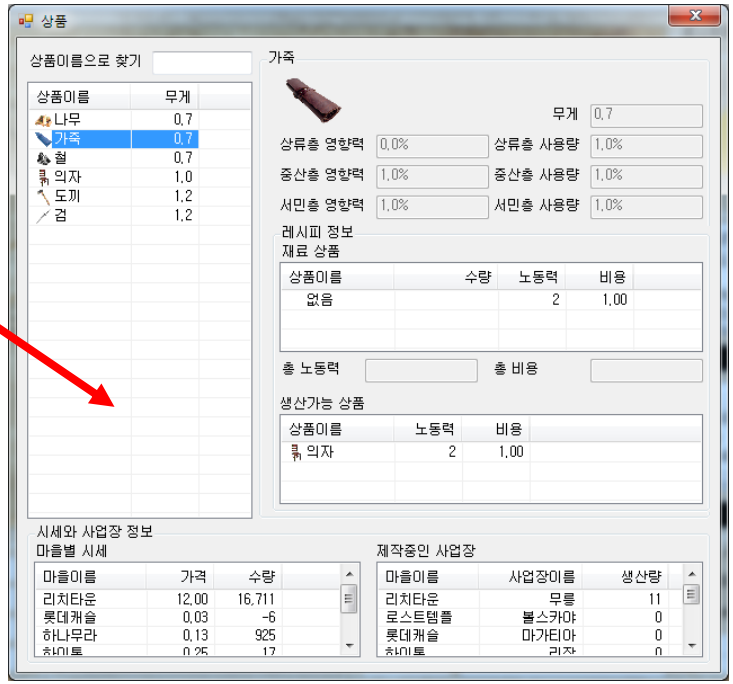
시스템 주요기능 (2/4)


사업장 목록
필터
없음
상단
마을
생산품
고용인수
리장
롯데캐슬
나무
0
무릉
하나무라
철
0
볼스카야
리치타운
철
0
일리오스
로스트템플
철
28
리장
하이톨
나무
0
마가티아
롯데캐슬
가죽
0
무릉
로스트템플
나무
10
선택 사업장 정보
임금
총 운영비용
생산량
매출
재료구입금액
수익금
추가

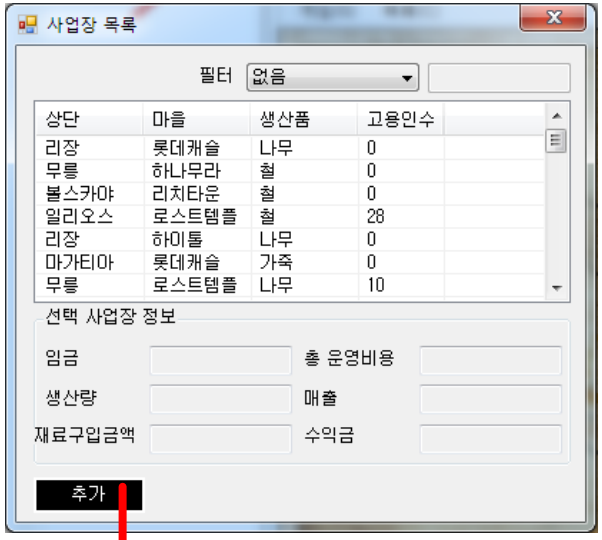
Comercio
리치타운
하나무라
로스트템플
해어톤
롯데캐슬

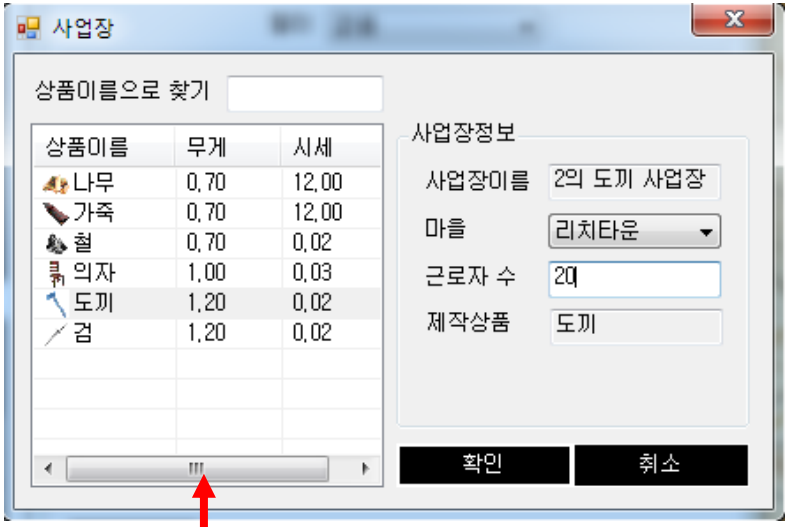

상행리스트
필터
상단이름
상단이름
출발마을
도착마을
이동거리
운송수단
2
로스트템플
하이톨
18,20
손수레
무릉
하이톨
리치타운
0,00
와이번
선택 상행 정보
상품이름
수량
철
100,000
남은 거리
36,80
남은 시간
368,82
예상수익금
50,000,01
추가

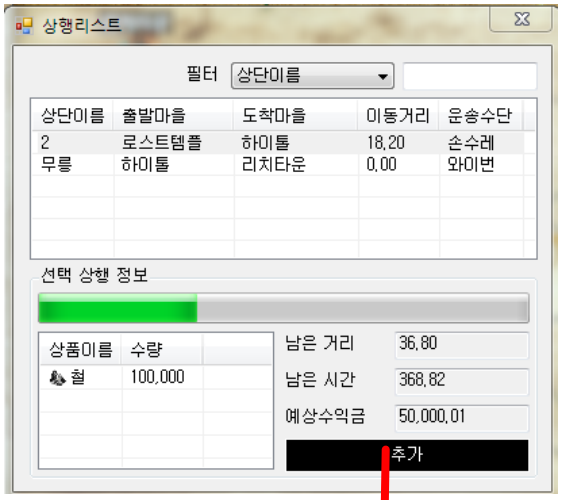
상단
상단이름으로 찾기
플레이어
상단이름
소지금
1
2
77,419,292,23
무릉
13,719,846,692,22
볼스카야
1,557,695,574,00
일리오스
3,395,228,672,066,
리장
8,441,145,43
마가티아
10,401,945,32
상단정보
상단이름
무릉
소지금
13,719,846,692,22
수익금
232,00
사업장수
5
사업장가치
232,0
상행수
0
상행가치
0,0
상단 사업장 정보
사업장이름
소재마을
생산품
고용인수
무릉의 철 사업장
하나무라
철
0
무릉의 나무 사업장
로스트템플
나무
10
무릉의 가죽 사업장
리치타운
가죽
11
무릉의 가죽 사업장
하나무라
가죽
0
무릉의 의자 사업장
리치타운
의자
0
상세 사업장 정보
상단 상행 정보
출발마을
도착마을
이동거리
운송수단
상세 상행 정보

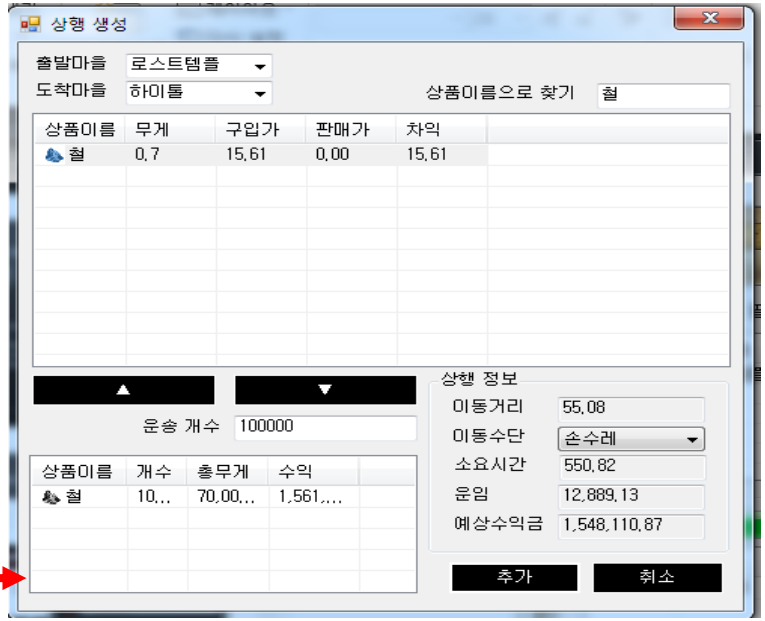

상품
상품이름으로 찾기
상품이름
무게
나무
0,7
가죽
0,7
철
0,7
의자
1,0
도끼
1,2
검
1,2
가죽
무게
0,7
상류층 영향력
0,0%
상류층 사용량
1,0%
중산층 영향력
1,0%
중산층 사용량
1,0%
서민층 영향력
1,0%
서민층 사용량
1,0%
레시피 정보
재료 상품
상품이름
수량
노동력
비용
없음
2
1,00
총 노동력
총 비용
생산가능 상품
상품이름
노동력
비용
의자
2
1,00
시세와 사업장 정보
마을별 시세
마을이름
가격
수량
리치타운
12,00
16,711
롯데캐슬
0,03
-6
하나무라
0,13
925
하이톨
0,25
17
제작중인 사업장
마을이름
사업장이름
생산량
리치타운
무릉
11
로스트템플
볼스카야
0
롯데캐슬
마가티아
0
하이톨
리장
0

시스템 주요기능 (3/4)

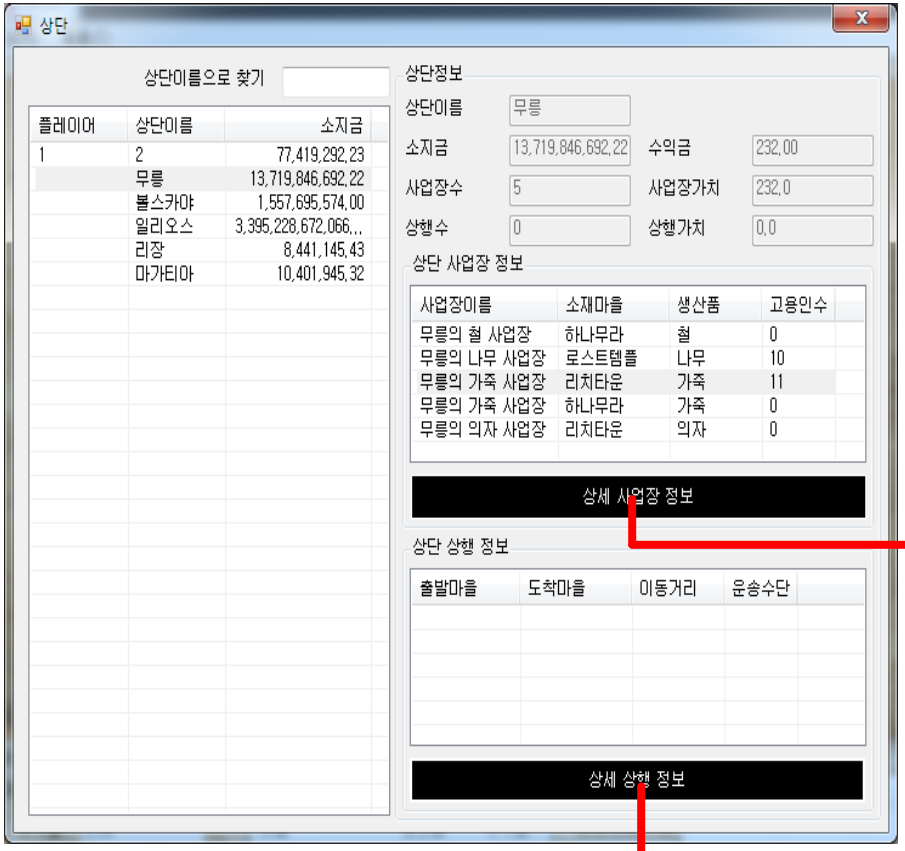

사업장 목록
필터 없음
상단 마을 생산품 고용인수
리장 롯데캐슬 나무 0
무릉 하나무라 철 0
볼스카야 리치타운 철 0
일리오스 로스트템플 철 28
리장 하이톨 나무 0
마가티아 롯데캐슬 가죽 0
무릉 로스트템플 나무 10
선택 사업장 정보
임금
총 운영비용
생산량
매출
재료구입금액
수익금
추가

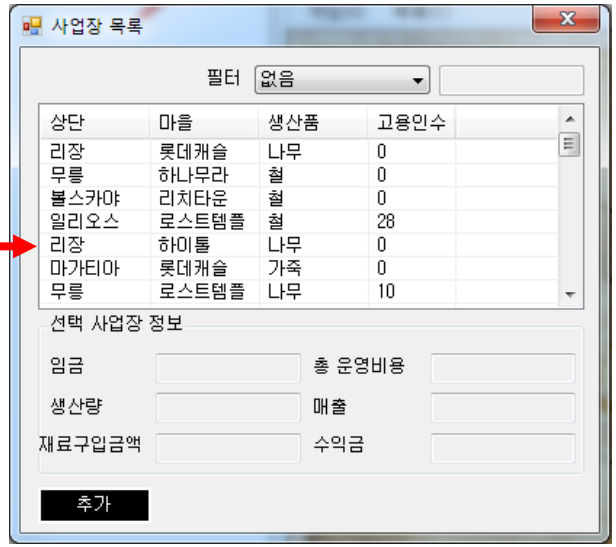

사업장
상품이름으로 찾기
상품이름 무게 시세
나무 0,70 12,00
가죽 0,70 12,00
철 0,70 0,02
의자 1,00 0,03
도끼 1,20 0,02
검 1,20 0,02
사업장정보
사업장이름 2의 도끼 사업장
마을 리치타운
근로자 수 20
제작상품 도끼
확인
취소

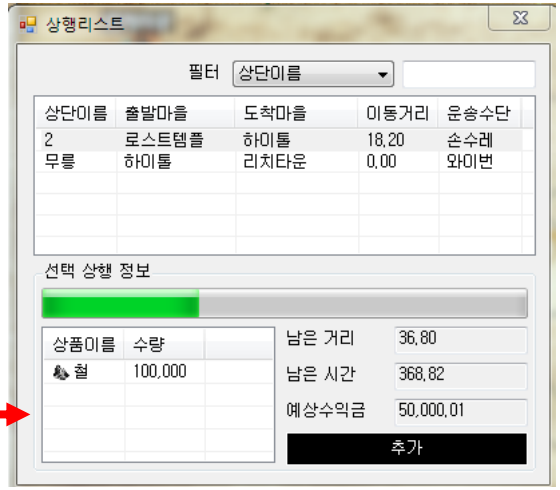

상행리스트
필터 상단이름
상단이름 출발마을 도착마을 이동거리 운송수단
2 로스트템플 하이톨 18,20 손수레
무릉 하이톨 리치타운 0,00 와이번
선택 상행 정보
상품이름 수량
철 100,000
남은 거리 36,80
남은 시간 368,82
예상수익금 50,000,01
추가


상행 생성
출발마을 로스트템플
도착마을 하이톨
상품이름으로 찾기 철
상품이름 무게 구입가 판매가 차익
철 0,7 15,61 0,00 15,61
운송 개수 100000
상품이름 개수 총무게 수익
철 10... 70,00... 1,561,...
상행 정보
이동거리 55,08
이동수단 손수레
소요시간 550,82
운임 12,889,13
예상수익금 1,548,110,87
추가
취소

시스템 주요기능 (4/4)


상단
상단이름으로 찾기
플레이어 상단이름 소지금
1 2 77,419,292,23
무릉 13,719,846,692,22
볼스카야 1,557,695,574,00
일리오스 3,395,228,672,066,..
리장 8,441,145,43
마가티아 10,401,945,32
상단정보
상단이름 무릉
소지금 13,719,846,692,22
수익금 232,00
사업장수 5
사업장가치 232,0
상행수 0
상행가치 0,0
상단 사업장 정보
사업장이름 소재마을 생산품 고용인수
무릉의 철 사업장 하나무라 철 0
무릉의 나무 사업장 로스트템플 나무 10
무릉의 가죽 사업장 리치타운 가죽 11
무릉의 가죽 사업장 하나무라 가죽 0
무릉의 의자 사업장 리치타운 의자 0
상세 사업장 정보
상단 상행 정보
출발마을 도착마을 이동거리 운송수단
상세 상행 정보


사업장 목록
필터 없음
상단 마을 생산품 고용인수
리장 롯데캐슬 나무 0
무릉 하나무라 철 0
볼스카야 리치타운 철 0
일리오스 로스트템플 철 28
리장 하이톨 나무 0
마가티아 롯데캐슬 가죽 0
무릉 로스트템플 나무 10
선택 사업장 정보
임금
총 운영비용
생산량
매출
재료구입금액
수익금
추가


상행리스트
필터 상단이름
상단이름 출발마을 도착마을 이동거리 운송수단
2 로스트템플 하이톨 18,20 손수레
무릉 하이톨 리치타운 0,00 와이번
선택 상행 정보
상품이름 수량
철 100,000
남은 거리 36,80
남은 시간 368,82
예상수익금 50,000,01
추가

시 연

COMERCIO

개발 후기

처음으로 혼자가아닌 프로젝트 팀으로써 프로그램 개발을 하게 되었는데 작업 효율성에 큰 영향을 주는 역할분담의 중요성을 알게 되었으며, 실시간으로 데이터를 주고받고 인공지능적 요소가 들어간 프로그램을 개발하면서 많은 오버플로우, 밸런싱문제로 디버깅과정에서 많은 고난과 역경을 겪었습니다. 또한 , 기획과 설계를 크게 멀리 보고 해야 실 코딩기간과 디버깅의 기간을 확 줄일 수 있다는 것을 이번 프로젝트를 통해 깨달았습니다.

Thank you