HW2#Virtual World

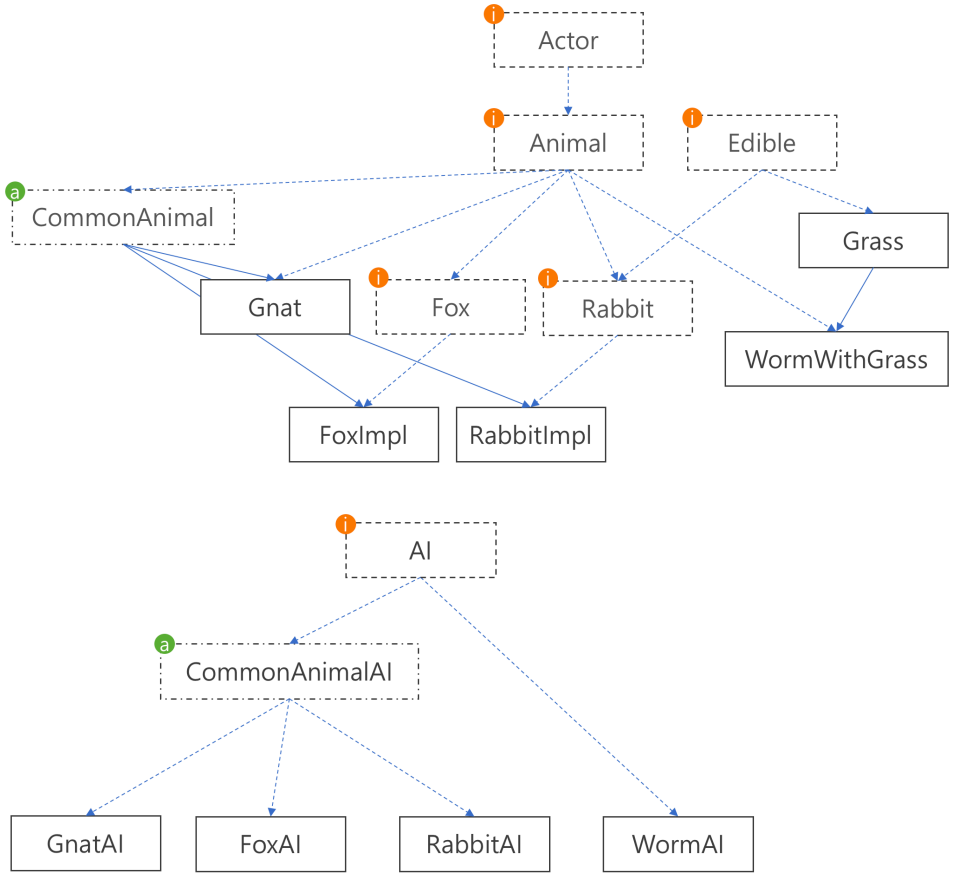
2015004857 이영수

1. Summary

1.1. 구현 된 것

- Gnat (Creature 및 AI)
- Rabbit (Creature 및 AI)
- Fox (Creature 및 AI)
- Third-animal (Creature 및 AI)

1.2. 전체적인 상속 구조



기존 소스의 구조와 거의 유사하지만 비슷한 동작을 하는 동물을 *CommonAnimal* 이라는 추상 클래스로 묶었다. 이 클래스에서는 동물들이 공통적으로 동작하는 move, breed, eat 등에 대한 implementation 코드가 작성되어 있다.
AI 또한 animal 과 비슷하게 동물들이 공통적으로 행동하는 로직은 묶어서 작성하였다.

2. Creature Implementations

2.1. CommonAnimal

Gnat, Fox, Rabbit 은 기본적으로 비슷한 동작을 하며, 내부적으로도 비슷한 코드를 사용하게 된다. 코드의 재사용성을 극대화 하며 동시에 확장성을 고려하여 일반적인 동작을 하는 '흔한 동물' 을 의미하는 *CommonAnimal* 추상 클래스 만들었으며, 동물들이 이를 상속받을 수 있도록 설계하였다.

abstract public boolean **edibleWith**(Edible prey)	인자를 받아 해당 Edible 객체를 먹을 수 있는지를 정의
protected void **actWithAI**(World world, AI ai)	AI 인스턴스를 바탕으로 act 동작을 실행한다.
protected int **getConsumingEnergy**()	매 step 마다 소모되는 에너지를 반환 (기본 값: 1)
public int **getEnergy**()	
protected void **setEnergy**(World world, int energy)	
public void **eat**(World world, Direction dir)	
public void **move**(World world, Direction dir)	
public void **breed**(World world, Direction dir)	

2.2 Gnat

Gnat 은 아무 오브젝트도 먹지 않는다. 따라서 ` *edibleWith* ` 메소드는 항상 false 를 반환하도록 재 정의 되었다. 또한 Gnat 은 에너지를 잃지 않기에 `*getConsumingEnergy*` 메소드를 0 을 반환하도록 재 정의하여 에너지 소모가 없도록 하였다.

2.3 FoxImpl 및 RabbitImpl

Fox 는 *Rabbit* 만을 먹으며, *Rabbit* 은 *Grass* 만을 먹는다. 이를 바탕으로 ` *edibleWith* ` 메소드를 재 정의 하였다.

3. AI Implementations

3.1. CommonAnimalAI

CommonAnimal 은 몇 가지 순서에 따라 조건을 확인하고, 조건을 만족했을 때 해당 Command 를 실행하도록 되어있다. 첫 번째 조건으로는 개체의 에너지가 breedLimit 보다 높고, 주변에 어떤 오브젝트도 없으며, 특정 확률을 만족하면 breed 를 실행한다. 다음으로 근처에 먹이가 있다면 먹이를 먹는다. 그렇지 않고 먹이가 시야에 보인다면, 먹이를 향해 이동한다.

3.2. RabbitAI 및 FoxAI

Rabbit 은 위의 *CommonAnimalAI* 를 그대로 실행하되 마지막 상황(어떤 조건도 만족하지 못하는 경우)에는 랜덤하게 움직인다. *Fox* 는 마지막 상황에 랜덤한 목적지를 정하고, 목적지를 향해 이동한다. 단, 목적지가 있어도 *CommonAnimalAI* 에 의한 결정이 우선시된다.

4. Third Animal Implementation

4.1. 컨셉

Grass 와 *Animal* 의 속성을 동시에 갖는 오브젝트이다. 이름은 `*WormWithGrass*`로, 풀 위에 지렁이가 살고 있는 개체를 의미한다. *WormWithGrass* 는 *Grass* 보다 energy value 가 1 적으며, *WormWithGrass* 은 *Grass* 타입을 상속받으므로 *Rabbit* 은 *WormWithGrass* 개체도 먹을 수 있다. Worm 은 주변 타일에 Grass 가 있으면 해당 오브젝트로 이동한다.

 (아이콘)

4.2. 구현

WormWithGrass 클래스는 *Grass* 클래스를 상속받으며 *Animal* 인터페이스를 implement 한다. `*WormAI*` 라는 클래스를 통해서 결정을 내리며, AI 인스턴스는 개체의 인접 타일을 탐색하고, *Grass* 가 있으면 움직이도록 명령을 내린다.

5. 기타

5.1. 스켈레톤 코드 수정

아이콘 경로 변경, 신규 크리처 추가 등의 이유로 일부의 스켈레톤 코드가 수정되었다.

5.2. 문서화

대부분의 클래스 및 인터페이스에 Javadoc 주석을 달았으며, 주요 로직에는 로직 설명용 주석을 달아 두었다.