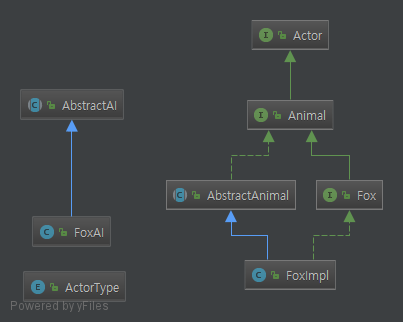
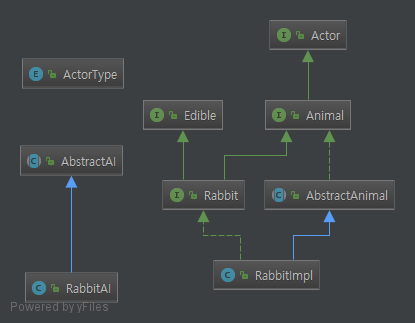
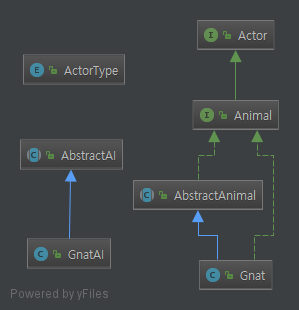
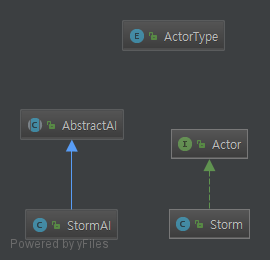
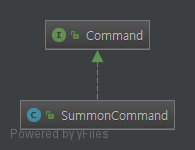
CSE4006 HW2 Report

2015004257 제갈지혜

1. 과제 설명

Fox, Rabbit, Gnat, Grass, Gardener, New creature로 구성된 virtual world를 구현한다. 이 과정에서 코드 재사용성과 캡슐화를 위해 object, 추상화, 상속 등을 최대한 사용하였다. 주어진 skeleton 코드에서는 new creature 추가를 위한 WorldLoader 클래스의 numStorms 변수 추가, addStorm() 함수 추가, WorldPanel 클래스의 아이콘 연결 및 paint 부분(line 44, line 84)이 변경되었다. New Creature Icon은 icons 폴더에 storm.png 라는 파일을 사용한다. 이 외에 skeleton 코드 변경사항은 없다. 전체 UML은 Git Repository VirutalWorld 폴더에 위치한 all.png에 존재한다.

1. 디자인 및 구현
   1. Fox
      1. 디자인 및 구현

* Fox는 Animal이고 Actor이다. (Fox Interface)
* Fox는 초기 에너지가 주어지지 않으면 설정된 Initial Energy(80)에서 시작한다. (FoxImpl constructor, FoxImpl private static final variable)
* FoxImpl 생성자에 초기 에너지가 주어지는 경우는 child를 만들 때만 존재한다.(FoxImpl Constructor, overrode makeChild function)
* Fox는 Breed Limit(최대 에너지의 3/4)을 만족하고 인접 타일에 빈 공간에 있는 경우 Breed한다. 두 조건 중 하나라도 만족하지 않으면 Breed하지 않는다. (FoxAI act function, AbstractAI emptyAdjacentDir function)
* Fox는 인접타일에 Rabbit이 있고 배고픔 정도(max energy / current energy) 가 식욕(0- HUNGRY\_MAX(13) 중 임의의 값) 보다 크면 Rabbit을 원래 있던 위치에서 eat한다. (FoxAI act function, AbstractAI findAdjacentObject function)
* eat을 하면 rabbit의 energy value(20) 만큼의 에너지를 얻는다. 단, fox의 최대 에너지(160)을 넘지는 못한다.(AbstractAnimal earnEnergy function, AbstractAnimal eat function)
* Fox의 Breed Limit(75%)이 Rabbit의 Breed Limit(50%) 보다 높고, eat의 가능성이 rabbit보다 낮기 때문에 전체 breed rate는 fox가 rabbit보다 낮다.
* 위 두 상황에 해당되지 않을 때는 View Range 안에 있는 Rabbit를 찾고, Rabbit를 통해 얻을 수 있는 에너지와 거리에 반비례한 값을 곱해 weight를 구한다. (FoxAI act function)
* Wight가 높은 곳을 찾은 결과 현재 위치와 같다면 임의의 인접한 타일로 이동한다. (AbstractAI moveTo function)
* Weight가 높은 곳(동일한 경우 먼저 탐색한 곳)이 현재위치와 다를 경우 Location Class에서 제공하는 dirTo 함수를 이용하여 weight가 높은 곳으로 향하는 방향을 구한다. World 내에 있는 곳이고 빈 타일일 경우 그 위치로 이동한다. World 내에 있는 곳이 아닌 경우 인접해 있는 빈 타일로 이동한다. (AbstractAI moveTo function)
* 매 행동(act)이 호출될 때 에너지가 1씩 감소하고 나이는 1씩 증가한다. (AbstractAnimal act function)
* 나이가 최대 나이를 넘어서면 죽는다. (AbstractAnimal act function)
* 에너지가 행동을 끝마치고 0이 되면 죽는다. (AbstractAnimal act function)
* 죽을 때는 자기자신을 world에서 지운다.(AbstractAnimal act function)
* 행동(act)는 쿨다운(2 step)만큼 기다리고 일어난다. (FoxImpl private static final variable)
  1. Rabbit
     1. 디자인 및 구현
* Rabbit은 Edible이고 Animal이고 Actor이다. (Rabbit Interface)
* Rabbit은 초기 에너지가 주어지지 않으면 설정된 Initial Energy(10)에서 시작한다. (RabbitImpl constructor, RabbitImpl private static final variable)
* RabbitImpl 생성자에 초기 에너지가 주어지는 경우는 child를 만들 때만 존재한다.(RabbitImpl Constructor, overrode makeChild function)
* Rabbit은 Breed Limit(최대 에너지의 1/2)을 만족하고 인접 타일에 빈 공간에 있으며 BREED\_PROB / BREED\_PROB\_MAX의 확률(50%)을 만족하는 경우 Breed한다. 세 조건 중 하나라도 만족하지 않으면 Breed하지 않는다. (RabbitAI act function, AbstractAI emptyAdjacentDir function)
* Rabbit은 인접타일에 Grass가 있으면 Grass를 원래 있던 위치에서 eat한다. (RabbitAI act function, AbstractAI findAdjacentObject function)
* eat을 하면 Grass의 energy value(5) 만큼의 에너지를 얻는다. 단, Rabbit의 최대 에너지(20)을 넘지는 못한다.(AbstractAnimal earnEnergy function, AbstractAnimal eat function)
* 위 두 상황에 해당되지 않을 때는 View Range 안에 있는 타일들의 Weight를 구한다. (RabbitAI act function, RabbitAI calWeight function)
* View range 안에 있는 타일들의 상하좌우 대각선에 있는 Object에 따라 각 타일의 weight들이 결정된다. Grass의 경우 양의 가중치를, Gnat, Fox, Rabbit의 경우 음의 가중치를 갖는다. 이 외의 것들은 고려하지 않는다.(RabbitAI calWeight function)
* Weight가 높은 곳을 찾은 결과 현재 위치와 같다면 임의의 인접한 타일로 이동한다. (AbstractAI moveTo function)
* Weight가 높은 곳(동일한 경우 먼저 탐색한 곳)이 현재위치와 다를 경우 Location Class에서 제공하는 dirTo 함수를 이용하여 weight가 높은 곳으로 향하는 방향을 구한다. World 내에 있는 곳이고 빈 타일일 경우 그 위치로 이동한다. World 내에 있는 곳이 아닌 경우 인접해 있는 빈 타일로 이동한다. (AbstractAI moveTo function)
* 매 행동(act)이 호출될 때 에너지가 1씩 감소하고 나이는 1씩 증가한다. (AbstractAnimal act function)
* 나이가 최대 나이를 넘어서면 죽는다. (AbstractAnimal act function)
* 에너지가 행동을 끝마치고 0이 되면 죽는다. (AbstractAnimal act function)
* 죽을 때는 자기자신을 world에서 지운다.(AbstractAnimal act function)
* 행동(act)는 쿨다운(4 step)만큼 기다리고 일어난다. (RabbitImpl private static final variable)
  1. Gnat
     1. 디자인 및 구현
* Gnat은 Animal이고 Actor이다. (Gnat Class)
* Gnat은 초기 에너지와 나이가 주어지지 않는다. (Gnat constructor)
* Gnat은 eat하거나 breed 하지 않으며 오로지 임의의 위치로 move만 한다.(GnatAI act function)
* 다른 Abstract Animal과 달리 매 행동(act)이 호출될 때 에너지나 나이가 감소하지 않는다. (Gnat overrode act function)
* 행동(act)는 쿨다운(0 step)만큼 기다리고 일어난다.(Gnat private static final variable)
  1. Storm
     1. 디자인 및 구현
* Storm은 Actor이다. (Storm Class)
* Storm은 에너지와 나이가 없다. insideObject를 null로 초기화한다.(Storm constructor)
* Storm은 eat하거나 breed하거나 move 하지 않으며 Summon만 한다.(StormAI act function)
* Storm이 Summon할 때 임의의 위치를 정하고 그 위치에 있는 Object를 빨아들인다. 이 후 다시 Summon 될 때 이전 위치에 빨아들인 Object를 뱉고 새로운 위치의 Object를 빨아들인다.(Storm Summon function)
* 빨아 들여진 Object는 world에서 remove된 상태이므로 energy가 감소하지 않는다. Age도 증가하지 않는다.(Storm summon function)
* 특정 object에게 time freezing과 같은 효과를 준다. Storm의 cool down 동안 정지상태가 되는 것과 같다.
* 행동(act)는 쿨다운(20 step)만큼 기다리고 일어난다.(Storm private static final variable)
  1. Others(SummonCommand, ActorType)
     1. 디자인 및 구현
* SummonCommand는 Storm의 act와 execute를 위해 생성하였다. (SummonCommand Class)
* ActorType은 AbstractAI에서 Actor를 받아 type을 구별하여 반환해 주기 위해 생성하였다. (ActorType enum class, AbstractAI typeActor function)