

[졸업논문 최종보고서]

| | | | | |
|-----|------|---------|-------|-------------|
| 신청자 | 성명 | 이륜하 | 학번 | 2019102114 |
| | 소속학과 | 소프트웨어융합 | 신청자확인 | 성명: 이륜하 (인) |

| | | | |
|-----|----------------------------------|------|----------|
| 과제명 | 다중간 생태계 시뮬레이션에서의 초기값 최적화 알고리즘 개발 | 참여학기 | 22년도 2학기 |
|-----|----------------------------------|------|----------|

요약

생태계 시뮬레이션의 가장 큰 맹점은 시뮬레이션 진행 도중 어느 하나의 개체군이 너무나도 쉽게 멸종해 버린다는 것이다. 이렇게 되는 원인들 중 큰 비중을 차지하는 것이 바로 개체들의 초기값이다. 특히 개체군의 종류가 다양할수록, 구현된 개체들의 패러미터가 많을수록 생태계 시뮬레이션을 유지하기 위한 초기값 설정이 어려워진다.

이번 연구에서는 다양한 패러미터를 가지는 다중간 생태계 시뮬레이션에서 생태계를 유지할 수 있는 초기값을 찾아내는 알고리즘을 개발하고자 하였다. 다양한 지형에서 알고리즘을 적용하여 동물군의 수치 변화를 Lotka-Volterra model과 비교하고자 하였다.

실험은 임의로 제작한 3가지의 맵에서, 각 맵마다 1번씩 진행되었다. 또한 한 사이클을 5번의 시뮬레이션을 반복하는 것으로 설정하여 사이클 단위로 실험을 수행하였다. 그러나 실제 실험 결과 시뮬레이션 내에서 고려해야 하는 변수가 너무 많아 실험 결과가 어떤 변수에 영향을 받았는지 명확하지 않았다. 또한 prey-predator의 개체수가 증감을 반복해야 하는데, 결과적으로 Lotka-Volterra model과는 다르게 상승-하강이 반복되는 양상 또한 보여주지 못했다.

이후 연구에서는 딥러닝을 사용해서 개체들에게 학습시키거나, 은신처의 개념을 시뮬레이션에 적용시키는 방향으로 추가 연구를 제안하고자 한다.

1. 서론

1.1 배경지식과 관련 연구

1.1.1. Lotka-Volterra Model

로트카-볼테라 방정식(Lotka-Volterra equation)은 미국의 수학자인 앨프리드 제임스 로트카와 이탈리아의 생물학자인 비토 볼테라가 발표한 연립미분방정식으로, 포식자와 피식자 간의 포식 관계를 수량화한 공식이다. Lotka-Volterra equation은 피식자-포식자 모델의 가장 기초가 되는 방정식으로써 현재까지도 생태계의 성장이나 물질순환, 화학반응 등 광범위한 생명현상을 수리적으로 분석하는 대표적인 모델로 이용되고 있다.

각 생물의 개체수는 다음 방정식에 의해 시간에 따라 변화한다.

$$\frac{dx}{dt} = \alpha x - \beta xy,$$

$$\frac{dy}{dt} = \delta xy - \gamma y,$$

각 변수들은 다음을 의미한다:

- x 는 피식자의 숫자이다.
- y 는 포식자의 숫자이다.
- t 는 시간을 나타낸다.
- $\frac{dx}{dt}$ 와 $\frac{dy}{dt}$ 는 각각 포식자와 피식자의 시간에 따른 개체 수 증가율을 나타낸다.
- $\alpha, \beta, \gamma, \delta$ 는 각각 포식자와 피식자 간의 상호작용에 의한 매개변수들이다. (단, 매개변수는 모두 양수이다.)

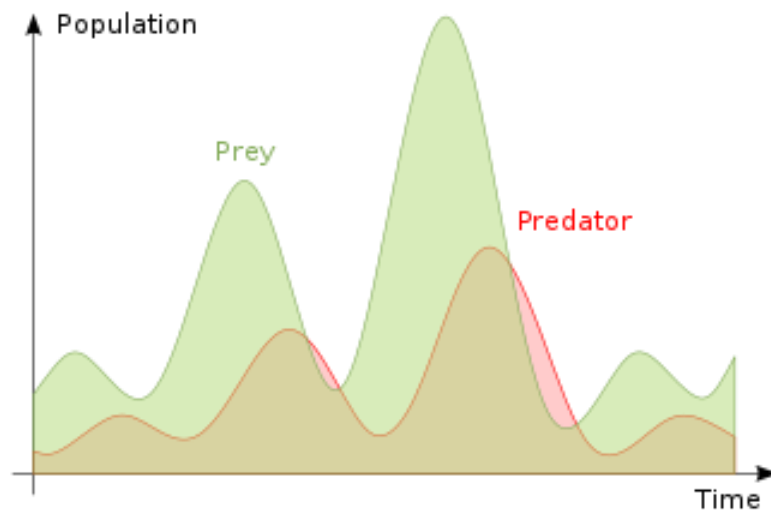


Fig 1. Lotka-Volterra prey-predator model 의 예시.

기본적으로 이 방정식은 포식자-피식자의 두 종의 생물이 상호작용하는 관계를 다루고 있는데, 여기에 새로운 생물종, 특히 식물을 넣어서 3종 간의 개체수 변화 방정식으로 응용할 수 있다. 참조한 논문¹에서 인용한 3종 간의 방정식은 다음과 같다.

$$\frac{dx}{dt} = a(1 - \frac{x}{x_c})x - bxy$$

$$\frac{dy}{dt} = -cy + dxy - eyz$$

$$\frac{dz}{dt} = -fz + gyz$$

- A는 식물의 성장률이다.
- B는 피식자가 식물을 먹는 비율이다.
- C는 피식자가 식물을 먹지 못해 사망하는 비율이다.
- D는 식물 당 피식자의 성장률이다.
- E는 포식자가 피식자를 먹는 비율이다.
- F는 포식자가 피식자를 먹지 못해 사망하는 비율이다.
- G는 피식자 당 포식자의 성장률이다.
- x_c 는 생태계의 식물 수용량이다.

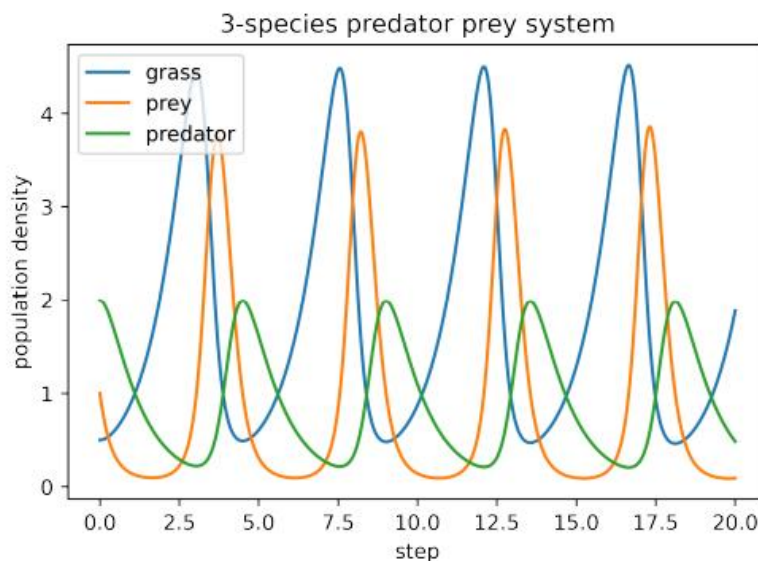


Fig 2. 3종 간의 prey-predator model의 예시. 참조논문오류! 책갈피가 정의되어 있지 않습니다.의 자료 인용.

1.2. 목표

생태계 시뮬레이션의 가장 큰 맹점은 시뮬레이션 진행 도중 어느 하나의 개체군이 너무나도 쉽게 멸종해 버린다는 것이다. 이렇게 되는 원인들 중 큰 비중을 차지하는 것이 바로 개체들의 초기값이다. 특히 개체군의 종류가 다양할수록, 구현된 개체들의 패러미터가 많을수록 생태계 시뮬레이션을 유지하기

위한 초기값 설정이 어려워진다.

이번 연구에서는 다양한 패러미터를 가지는 다종간 생태계 시뮬레이션에서 생태계를 유지할 수 있는 초기값을 찾아내는 알고리즘을 개발하고자 하였다. 알고리즘은 시뮬레이션을 반복할 때마다 특정 패러미터를 자동적으로 조절하여 이상적인 초기값을 찾고자 노력한다. 이 과정에서 각 동물군의 수치 변화를 그래프로 표현, 분석하여 Lotka-Volterra model과 비교하고자 하였다. 또한 이를 다양한 지형에서 적용해 보고자 한다.

2. 본론

2.1. 구현

시뮬레이션 환경을 구축하기 위해 UE4를 사용하였다. 개발의 기본적인 구조는 다음 두 개의 연계된 논문²³을 참조하였다.

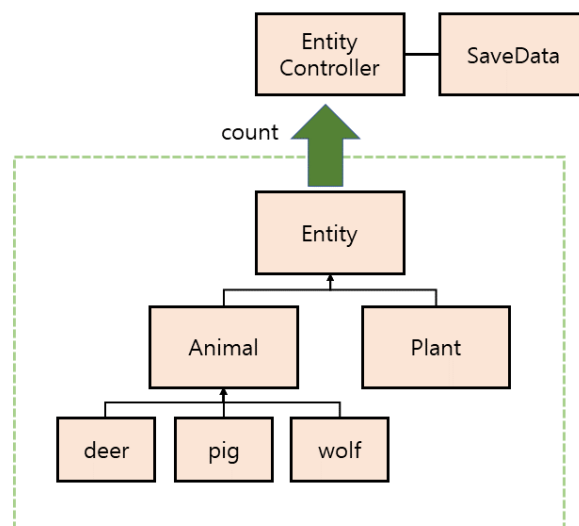


Fig 3. 구성요소 간의 관계 모식도

각 구성요소들의 관계는 다음과 같다. 실제 시뮬레이션의 주체인 생물들은 Entity를 상속받아 만들어졌다. Animal은 동물들의 생명활동을 모방한 함수들로 이루어져 있으며, 총 3종류의 동물종(deer, pig, wolf)을 자식으로 가진다. 반면에 Plant는 식물의 성장 환경을 거의 고려하지 않았다. EntityController에 의해 랜덤생성되며 초식동물(deer, pig)의 먹이로써의 역할만을 중시했다.

EntityController는 시뮬레이션 전반을 관리하며 맵 내의 개체수 변화를 측정, 저장하고 실시간으로 화면에 출력한다. 시뮬레이션이 종료될 때, 해당 시점까지의 개체수 변화값을 그래프로 출력하고 수식에 따라 초기값을 변경한다. 이렇게 변경된 초기값은 SaveData를 통해 외부 파일로 저장되어 다음 시뮬레이션에 반영된다.

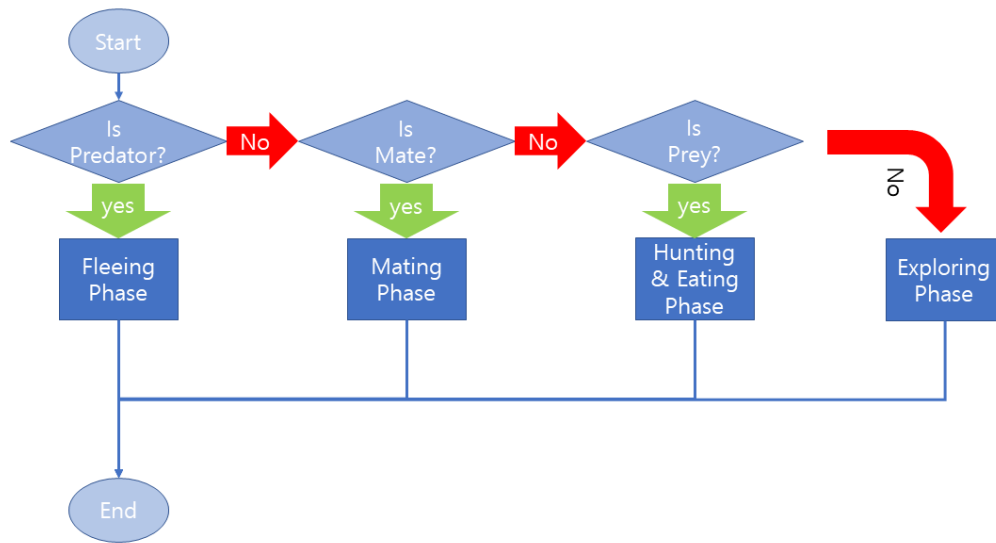


Fig 4. Animal behavior tree의 모식도.

Animal의 behavior tree의 모식도이다. Animal은 총 4가지의 phase로 나뉘며 직사각형으로 표시된다. 각 phase에 진입하는 조건은 마름모로 표시되며, start에서 가까운 순으로 우선순위를 지닌다. 각 phase가 끝나면 end로 진입하고, behavior tree는 초기화되어 행동을 반복한다.



Fig 5. 3종의 Animal의 모델

이번 시뮬레이션에서는 총 3가지의 생물종을 개발하여 적용하였다. 각 동물들은 다양한 패러미터를 통해 독특한 행동양식을 부여하고자 하였다. 각 패러미터들에 대한 설명은 다음과 같다.

| Parameters | 설명 |
|------------------|--|
| Max Hunger | 배고픔 수치. 0이 되면 사망한다. |
| Max destroy time | 사망 이후 맵에 시체가 잔존하는 시간. Predator에 의해 먹히게 되면 더 빠른 속도로 줄어든다. |

| | |
|-------------------|---------------------------|
| Food habit | 초식동물은 0, 육식동물은 1의 값을 가진다. |
| Childbirth | 한번에 최대로 낳을 수 있는 자식의 수 |
| Pregnant time | 임신 후 자식을 낳기까지의 시간 |
| Mate age | 번식 가능한 최소나이 |
| Natural death age | 자연사로 인해 사망하는 나이 |

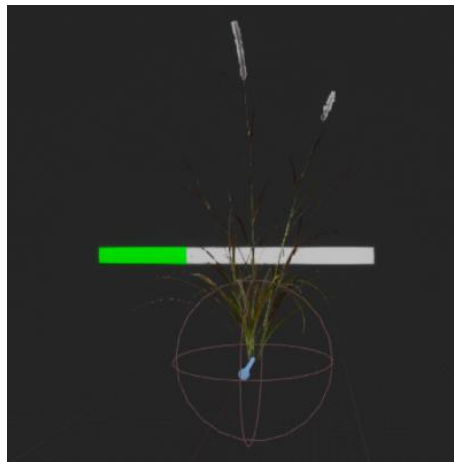


Fig 6. Plant의 모델

이미 동물군이 3종류나 있기 때문에, plant는 초식동물의 먹이가 된다는 개념에 충실하고자 하였다. 식물의 성장이나 번식, 나이 등을 고려하지 않고, 오직 먹이로서 수용력과 랜덤생성되는 시간을 주요 패러미터로 두었다.

| Parameters | 설명 |
|-------------------------------------|---|
| Edible amount (Max destroy time) | 동물에게 먹힐 수 있는 양. 동물과 상호작용해야 줄어든다. 0이 되면 액터는 소멸한다. |
| Regen time | 랜덤생성되는데 걸리는 시간. 작을수록 빈번해진다. |

Lotka-Volterra model에서 변경해 줄 만한 초기값은 피식자의 성장률, 포식자의 성장률, 각 개체군의 초기 개체수 이렇게 세 가지로 볼 수 있다.

시뮬레이션 도중 매 10초마다 각 종의 성장률을 비교한다. 특정 종의 개체수 성장률이 급격하게 떨어지는 구간이 있다면 그 종이 그 시점에서 자연사 또는 포식자에 의한 죽음이 다수 발생했다고 판단하고, 생물종의 성장률에 영향을 끼치는 childbirth의 값을 소량 증가시킨다.

시뮬레이션 종료 시점에서 종마다 개체수를 최솟값과 비교하여 이 종이 멸종했는지 아닌지를 판단한다. 만약 개체수가 최소 개체수 이하라면, 초기 개체수가 충분하지 않다고 판단하여 초기 개체수를 25% 증가시킨다. 이때 최소 개체수는 한 종이 멸종하지 않기 위한 최소 개체수로 정의한다.

3. 실험

3.1. 실험 기준

이 연구는 시뮬레이션을 반복적으로 수행하여 초기값이 바뀌는 것이 핵심이다. 따라서 한 사이클을 5 번의 시뮬레이션을 반복하는 것으로 설정하여 사이클 단위로 실험을 수행하였다. 시뮬레이션은 한 생물 군의 개체수가 최소 개체수 이하가 되는 순간 즉시 종료되고, 초기값을 변경한 뒤 재시작한다. 한 사이클이 끝나면 모든 설정값을 실험 기본값으로 되돌렸다. 실험은 임의로 제작한 3가지의 맵에서, 각 맵마다 1번씩 진행되었다.

3.2. 실험 기본값

각 구성요소들의 실험 기본값은 다음과 같다. 기본값의 수치는 다음 논문을오류! 책갈피가 정의되어 있지 않습니다. 참조하였다

Animal Farm

73

| Parameters | Chicken | Pig | Cow | Dog | Fox | Lion |
|------------------|--------------|--------------|--------------|---------|-------------|------|
| Avarage lifetime | 375 | 700 | 800 | 650 | 550 | 700 |
| Most childbirth | 3 | 3 | 1 | 1 | 2 | 1 |
| Pregnancy dur. | 55 | 70 | 80 | 65 | 65 | 70 |
| Spermatogenesis | 50 | 30 | 30 | 30 | 30 | 30 |
| Food | Grain | Grain | Grain | Chicken | Chicken,Pig | All |
| Enemy | Dog,Fox,Lion | Dog,Fox,Lion | Dog,Fox,Lion | Lion | Lion,Dog | None |

| Viewrange | Chicken | Pig | Cow | Dog | Fox | Lion | Walkingspeed | Chicken | Pig | Cow | Dog | Fox | Lion |
|-----------|---------|-----|-----|-----|-----|------|--------------|---------|-----|-----|------|------|------|
| Puppy | 20 | 22 | 22 | 23 | 26 | 23 | Puppy | 3.3 | 3.2 | 3.2 | 3.55 | 3.65 | 3.5 |
| Juvenile | 22 | 24 | 24 | 25 | 28 | 25 | Juvenile | 3.6 | 3.6 | 3.4 | 3.85 | 3.85 | 3.65 |
| Young | 23 | 24 | 26 | 27 | 30 | 27 | Young | 4.1 | 3.9 | 3.6 | 4.05 | 4.15 | 3.85 |
| Adult | 25 | 25 | 28 | 29 | 32 | 29 | Adult | 4.3 | 4.1 | 4 | 4.2 | 4.25 | 4.1 |
| Aged | 22 | 24 | 24 | 25 | 28 | 25 | Aged | 3.7 | 3.6 | 3.4 | 3.7 | 3.85 | 3.55 |
| Elder | 18 | 22 | 21 | 21 | 24 | 21 | Elder | 3.2 | 3.1 | 2.9 | 3.2 | 3.35 | 3.3 |

| Thirst/Hunger | Chicken | Pig | Cow | Dog | Fox | Lion | AgeStage bound. | Chicken | Pig | Cow | Dog | Fox | Lion |
|---------------|---------|------|------|-----|------|------|-----------------|---------|-----|-----|-------|-------|------|
| Puppy | 1.3 | 1.1 | 1 | 1.1 | 15 | 1.1 | Puppy | 37.5 | 105 | 120 | 97.5 | 82.5 | 105 |
| Juvenile | 1.55 | 1.35 | 1.35 | 1.3 | 1.35 | 1.22 | Juvenile | 75 | 154 | 176 | 143 | 121 | 154 |
| Young | 1.65 | 1.5 | 1.5 | 1.4 | 1.45 | 1.3 | Young | 112.5 | 231 | 264 | 214.5 | 181.5 | 231 |
| Adult | 1.55 | 1.4 | 1.35 | 1.3 | 1.35 | 1.2 | Adult | 187.5 | 350 | 400 | 325 | 275 | 350 |
| Aged | 1.25 | 1.3 | 1.1 | 1.1 | 1.15 | 1.1 | Aged | 281.25 | 525 | 600 | 487.5 | 412.5 | 525 |
| Elder | 1 | 0.9 | 0.8 | 0.7 | 0.7 | 0.7 | Elder | 375 | 700 | 800 | 650 | 550 | 700 |

Fig 6. 참조 논문오류! 책갈피가 정의되어 있지 않습니다.의 시뮬레이션 초기값. 다음 표를 참조하여 아래의 초기값들을 설정하였다.

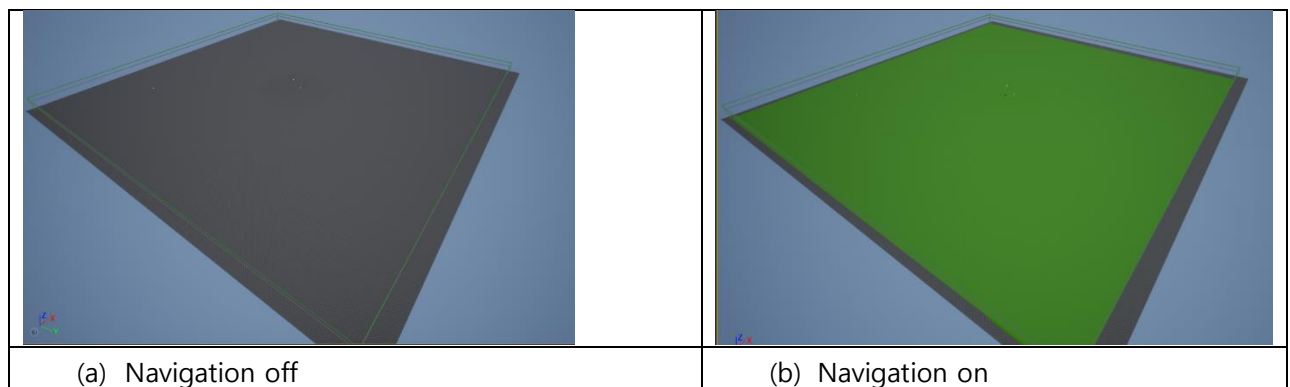
이 중 사이클 중 EntityController에 의해 변경 가능한 초기값을 굵은 글씨로 표시하였다. 앞에서 제시하였듯이 plant, animal의 시작 시점 개체값과 성장률에 영향을 주는 plant의 regen time, animal의 childbirth 이렇게 4가지 패러미터를 변경한다.

| Parameters | Plant |
|--|-----------|
| Initial population (시작 시점 개체값) | 50 |
| Edible amount (Max destroy time) | 30 |
| Regen time | 5 |

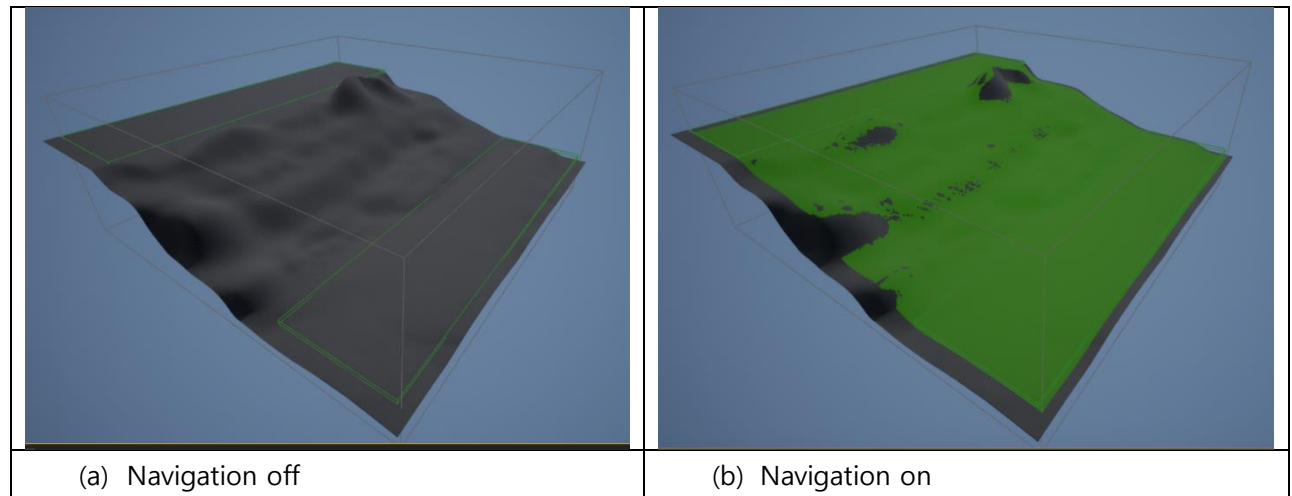
| Parameters | Animal |
|---------------------------|--------|
| Max Hunger | 150 |
| Max destroy time | 30 |
| Min population (최소개체수) | 5 |

| Parameters | Deer | Pig | Wolf |
|---------------------------|---------------|---------------|---------------|
| Initial population | 25 | 25 | 14 |
| Walk speed | 320 | 300 | 350 |
| Chase speed | 380 | 350 | 400 |
| Food habit | 0 (herbivore) | 0 (herbivore) | 1 (carnivore) |
| Childbirth | 2 | 3 | 2 |
| Pregnant time | 15 | 13 | 20 |
| Mate age | 80 | 70 | 150 |
| natural death age | 600 | 500 | 600 |

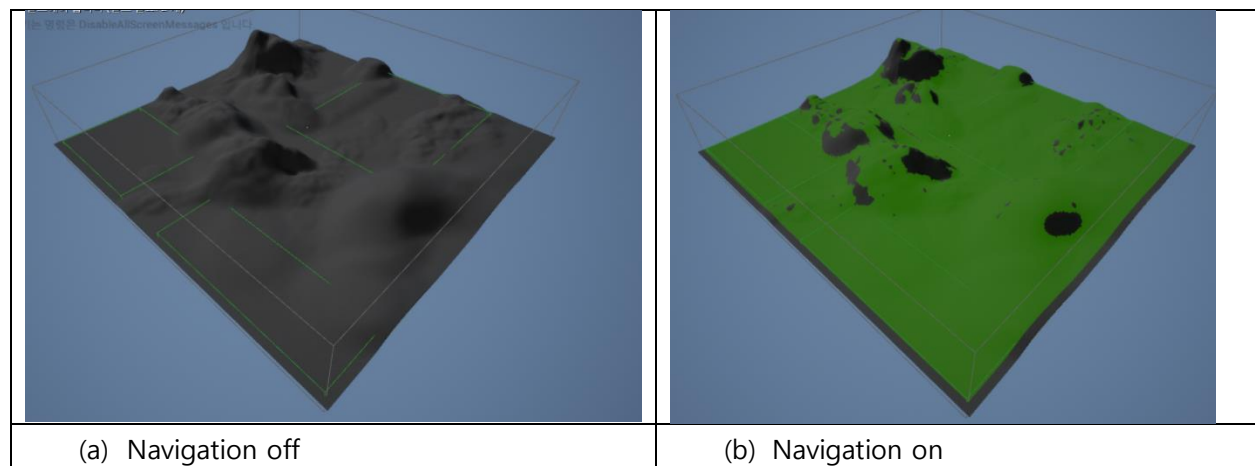
실험의 배경이 되는 맵의 구조는 다음과 같다. 평평한 지형에서만 생물이 생성되도록 box를 설정하였다.



Map 1. 평평한 지형



Map 2. 높이의 고저차가 있는 지형

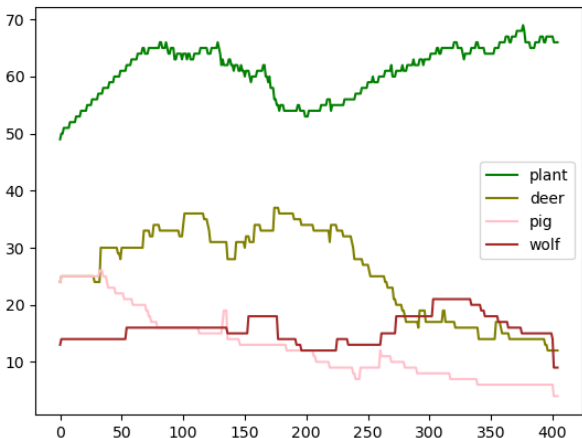
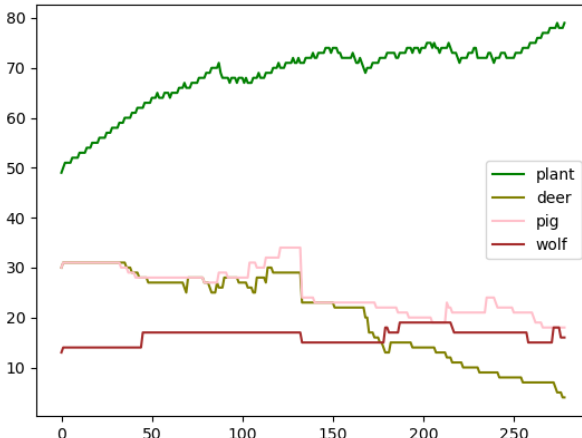


Map 3. 다양한 언덕이 분포하고 있는 지형

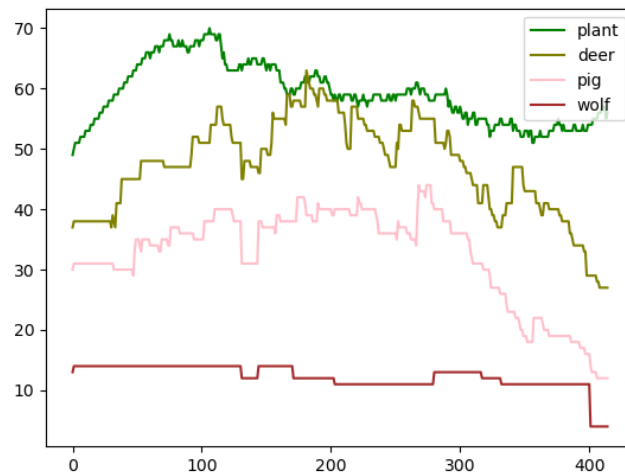
3.3 실험 결과

시뮬레이션 종료 시점까지의 각 생물군의 개체수 변화량을 그래프로 표현하였다. 데이터는 1초 단위로 저장되어 그래프 상으로는 계단식으로 표현된다. 각 시뮬레이션의 결과로 변화하는 초기값을 표로 나타냈다. Sim.1에서의 초기값은 항상 앞에서 제시한 초기값이기 때문에 실질적인 변화가 있는 Sim.2부터 표기하였다.

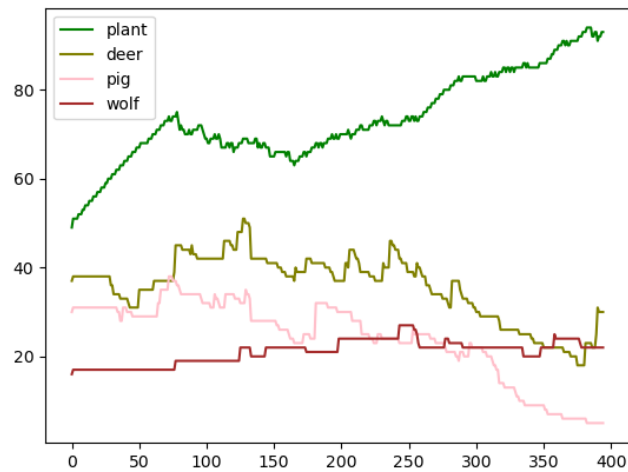
3.3.1 Map 1

| Simulation | graph |
|------------|---|
| 1 |  <p>This line graph shows the population dynamics of four species over 400 time steps. The y-axis ranges from 10 to 70. The 'plant' population (green line) starts at approximately 50, rises to a peak of about 65 around time step 100, dips to 55 at time step 200, and then fluctuates between 60 and 70. The 'deer' population (olive line) starts at 25, peaks at 35 around time step 100, and then declines to about 15 by time step 400. The 'pig' population (pink line) starts at 25 and steadily declines to about 5. The 'wolf' population (red line) starts at 15, remains relatively stable until time step 100, and then fluctuates between 15 and 20.</p> |
| 2 |  <p>This line graph shows the population dynamics of four species over 300 time steps. The y-axis ranges from 0 to 70. The 'plant' population (green line) starts at 50, peaks at 65 around time step 100, and then fluctuates between 55 and 65. The 'deer' population (olive line) starts at 25 and declines steadily to about 5 by time step 300. The 'pig' population (pink line) starts at 30, peaks at 40 around time step 100, and then declines to about 15. The 'wolf' population (red line) starts at 15, remains stable until time step 100, and then fluctuates between 15 and 20.</p> |
| 3 |  <p>This line graph shows the population dynamics of four species over 250 time steps. The y-axis ranges from 10 to 80. The 'plant' population (green line) starts at 50, rises to about 70 by time step 100, and then fluctuates between 70 and 80. The 'deer' population (olive line) starts at 30 and declines steadily to about 5 by time step 250. The 'pig' population (pink line) starts at 30 and declines to about 15. The 'wolf' population (red line) starts at 15, remains stable until time step 100, and then fluctuates between 15 and 20.</p> |

4



5



| Sim. 2 | Deer | Pig | Wolf | Plant |
|--------------------|------|-----|------|-------|
| Initial population | 25 | 30 | 14 | 50 |
| Childbirth | 2 | 3 | 2 | |
| Regen time | | | | 5 |

| Sim. 3 | Deer | Pig | Wolf | Plant |
|--------------------|------|-----|------|-------|
| Initial population | 31 | 31 | 14 | 50 |
| Childbirth | 2 | 3 | 2 | |
| Regen time | | | | 4 |

| Sim. 4 | Deer | Pig | Wolf | Plant |
|--------|------|-----|------|-------|
|--------|------|-----|------|-------|

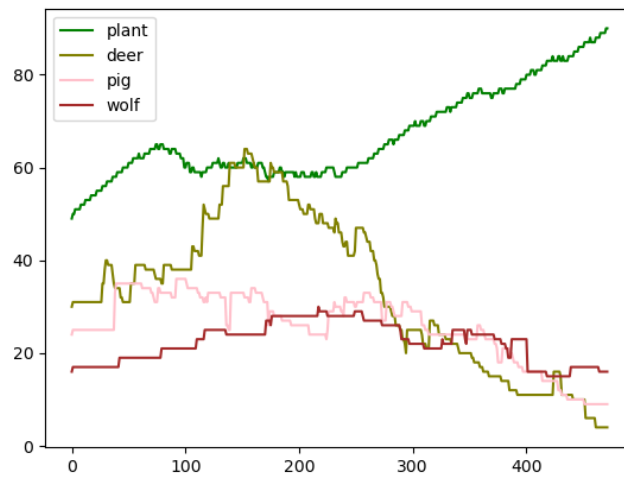
| | | | | |
|--------------------|----|----|----|----|
| Initial population | 37 | 31 | 14 | 50 |
| Childbirth | 4 | 5 | 2 | |
| Regen time | | | | 4 |

| | | | | |
|--------------------|------|-----|------|-------|
| Sim. 5 | Deer | Pig | Wolf | Plant |
| Initial population | 37 | 31 | 17 | 50 |
| Childbirth | 6 | 7 | 2 | |
| Regen time | | | | 3.2 |

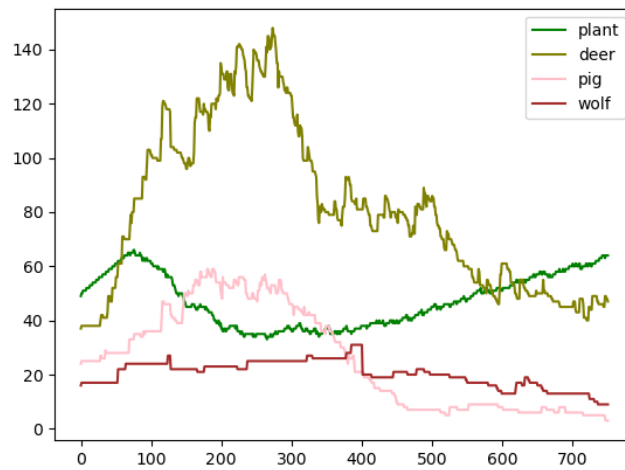
3.3.2 Map 2

| Simulation | graph |
|------------|-------|
| 1 | |
| 2 | |

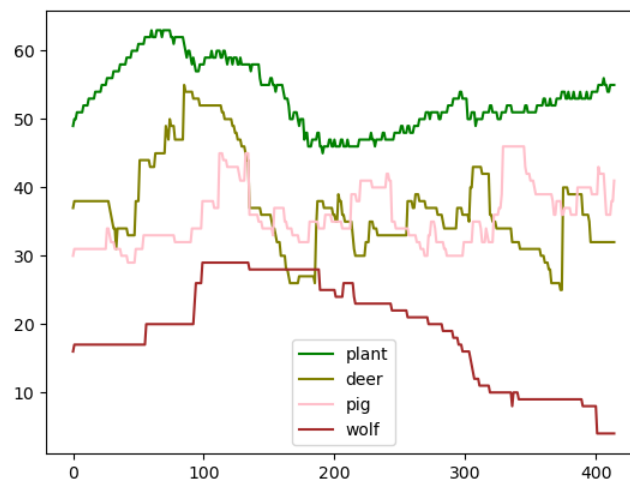
3



4



5



| Sim. 2 | Deer | Pig | Wolf | Plant |
|--------------------|------|-----|------|-------|
| Initial population | 30 | 25 | 14 | 50 |
| Childbirth | 4 | 5 | 2 | |
| Regen time | | | | 5 |

| Sim. 3 | Deer | Pig | Wolf | Plant |
|--------------------|------|-----|------|-------|
| Initial population | 31 | 25 | 17 | 50 |
| Childbirth | 6 | 5 | 2 | |
| Regen time | | | | 5 |

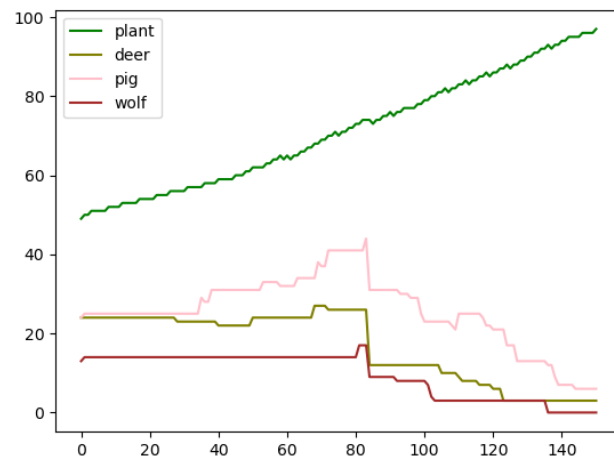
| Sim. 4 | Deer | Pig | Wolf | Plant |
|--------------------|------|-----|------|-------|
| Initial population | 38 | 25 | 17 | 50 |
| Childbirth | 8 | 5 | 2 | |
| Regen time | | | | 5 |

| Sim. 5 | Deer | Pig | Wolf | Plant |
|--------------------|------|-----|------|-------|
| Initial population | 38 | 31 | 17 | 50 |
| Childbirth | 10 | 7 | 2 | |
| Regen time | | | | 5 |

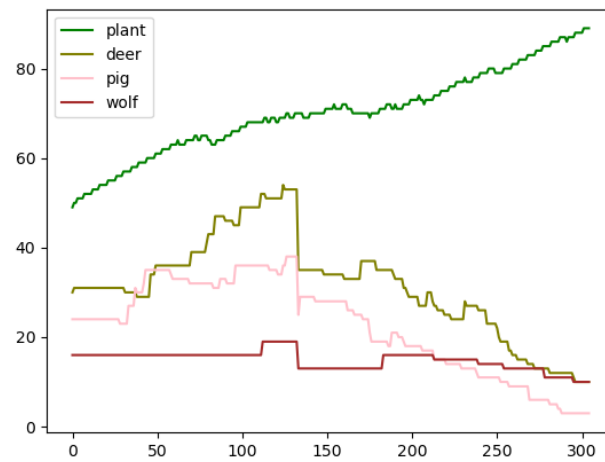
3.3.3 Map 3

| Simulation | graph |
|------------|-------|
| 1 | |

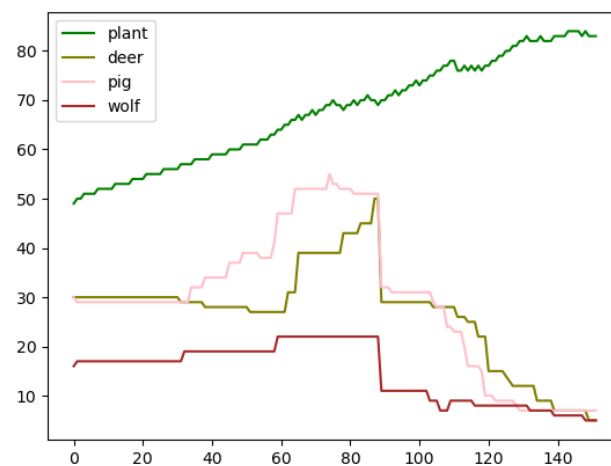
2



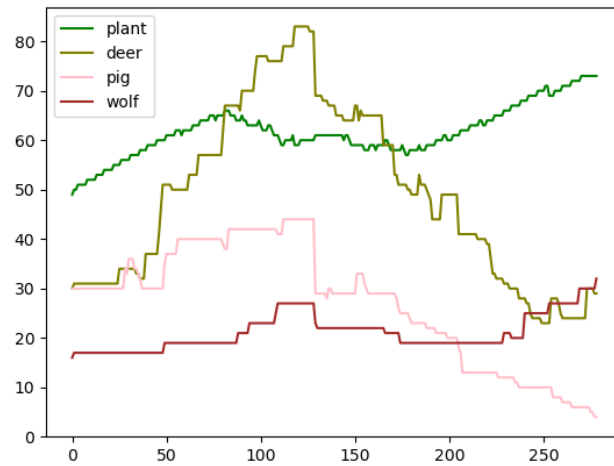
3



4



5



| Sim 2 | Deer | Pig | Wolf | Plant |
|--------------------|------|-----|------|-------|
| Initial population | 25 | 25 | 14 | 50 |
| Childbirth | 2 | 5 | 2 | |
| Regen time | | | | 5 |

| Sim 3 | Deer | Pig | Wolf | Plant |
|--------------------|------|-----|------|-------|
| Initial population | 30 | 25 | 16 | 50 |
| Childbirth | 4 | 7 | 2 | |
| Regen time | | | | 5 |

| Sim 4 | Deer | Pig | Wolf | Plant |
|--------------------|------|-----|------|-------|
| Initial population | 30 | 25 | 16 | 50 |
| Childbirth | 6 | 9 | 2 | |
| Regen time | | | | 5 |

| Sim 5 | Deer | Pig | Wolf | Plant |
|--------------------|------|-----|------|-------|
| Initial population | 30 | 30 | 16 | 50 |
| Childbirth | 8 | 11 | 2 | |
| Regen time | | | | 5 |

4. 결론

이번 연구에서는 다양한 패러미터를 가지는 다종간 생태계 시뮬레이션에서 생태계를 유지할 수 있는 초기값을 찾아내는 알고리즘을 개발하고자 하였다. 그러나 실제 실험 결과 시뮬레이션 내에서 고려해야 하는 변수가 너무 많아 실험 결과가 어떤 변수에 영향을 받았는지 명확하지 않았다.

생성된 그래프를 살펴보면 개체수가 급락하는 구간 (80~100초, 100~150초)이 존재하는데, 이는 시작 시점에서 생성된 개체가 그 시점까지 먹이를 찾지 못해 사망하는 것이라 추측된다. 지형지물이 많은 Map 3에서 이러한 급격한 사망이 두드러지게 나타나는 것 또한 지형으로 인해 먹이에 접근하기 어려웠기 때문이라고 생각된다.

또한 안정적으로 늘어나던 deer, pig의 개체수가 wolf의 개체수가 미세하게 상승하기만 해도 큰 폭으로 하락하는 모습을 보였다. 이렇게 하락한 deer, pig의 개체수는 먹이가 떨어져 wolf의 숫자가 줄어든 후에도 회복되지 않았다. Lotka-Volterra model에 따르면 predator의 개체수가 줄어들면 자연스럽게 prey의 개체수가 늘어나며 prey-predator의 개체수가 증감을 반복해야 하는데, 그 부분에 있어서 미흡함이 있었다. 결과적으로 Lotka-Volterra model과는 다르게 상승-하강이 반복되는 양상 또한 보여주지 못했다.

이번 실험의 문제점과 추가 연구를 할 만한 소재는 다음과 같다. 첫 번째는 animal에 부여한 패러미터가 너무 많았다는 것이다. 다종 간의 실험이라 종마다 서로 다른 행동양상을 보여주고 싶어 너무 많은 패러미터들을 부여했더니 결과 분석에 어려움이 있었다. 결과에 크게 유의미하지 않은 패러미터들을 적절히 걸러내든지, 아예 딥러닝을 사용해서 개체들에게 학습시키는 방향으로 개발을 진행했다면 좀 더 나은 결과물이 나오지 않았을까 하는 아쉬움이 있다.

두 번째는 predator의 개체수에 prey가 너무 큰 영향을 받는다는 것이다. 특히 이 시뮬레이션에서 predator는 거의 반드시 prey 사냥에 성공하는데, 이것이 문제가 발생한 가장 큰 원인이라고 추측된다. Prey가 predator의 공격을 회피할 수단이 있다면 늘어난 predator에도 불구하고 일정 비율은 살아남아 번식할 수 있었을 것이다. 즉 은신처의 개념을 시뮬레이션에 적용시킬 수 있었다면 실험의 결과가 크게 달라질 것이라 생각된다.

지도교수: _____ (인)

¹ Tobias Karlsson. Multi-Agent Deep Reinforcement Learning in a Three-Species Predator-Prey Ecosystem. 2021

² Attila Kiss and Gábor Pusztai. Using the Unity Game Engine to Develop a 3D Simulated Ecological System Based on a Predator–Prey Model Extended by Gene Evolution. *Informatics* 2022, 9, 9

³ Kiss, A. Pusztai, G. Animal Farm—A complex artificial life 3D framework. *Acta Univ. Sapientiae Inform.* 2021, 13, 60–85.