최종 보고서

1. 개요

- **주제**

  실제 자연환경과 같은 메타버스 환경을 구축하고 가상의 동물 생태계 구축

  -기본

  - 10종 이상의 종의 특성을 반영한 가상 생명체 시뮬레이션
  - 종간의 상호 작용 및 먹이사슬을 포함한 생태계 구축
  - 가상 환경의 생명체를 간단한 이미지를 활용하여 표현

  -확장

  - 2D 혹은 3D를 활용하여 가상 환경의 시각적 리얼리티를 강화
  - 가상의 날씨 시스템을 구현하고 날씨와 생명체의 상호 영향 작용

- **목표**

  강화학습을 통해 생태계의 평형을 이루는 동물의 개체 수 학습과 생태계를 로블록스 환경에서 시각화 하는 것

  - 제안 발표시 목표
    - 사용자가 동물 개체들에 부여한 초기 속성에 따라, 동물 사이의 생태계 유지에 가장 적합한 초기 개체 수를 파악한다.
    - 동물 간 상호작용 및 먹이사슬, 그리고 날씨를 구현한다.
    - 로블록스를 사용하여 시각적으로 생태계의 시각화하고, 파이썬을 이용해서 강화학습 알고리즘을 개발하고, Flask 웹 프레임워크를 사용하여 로블록스와 서버간 API를 구현한다. (유저의 입력에 따라 학습을 시작한다)
  - 수정된 목표
    - 유저의 입력에 따라 학습을 시작하면, 강화학습에 너무 오랜 시간이 걸려 Flask를 사용하여 API를 구현하는 부분을 삭제했습니다.
    - 초기 목표시 사용할거라 예상했던 DQN 알고리즘에서 A2C 알고리즘으로 선회했습니다.

- **결과**

  강화학습을 통해 얻은 생태계의 초기값으로 로블록스 상에서 시뮬레이션 하는 과정을 시각화하였다.

  - 강화학습
  - 시뮬레이션
  - 로블록스

2. 추진 체계

- **팀 구성**

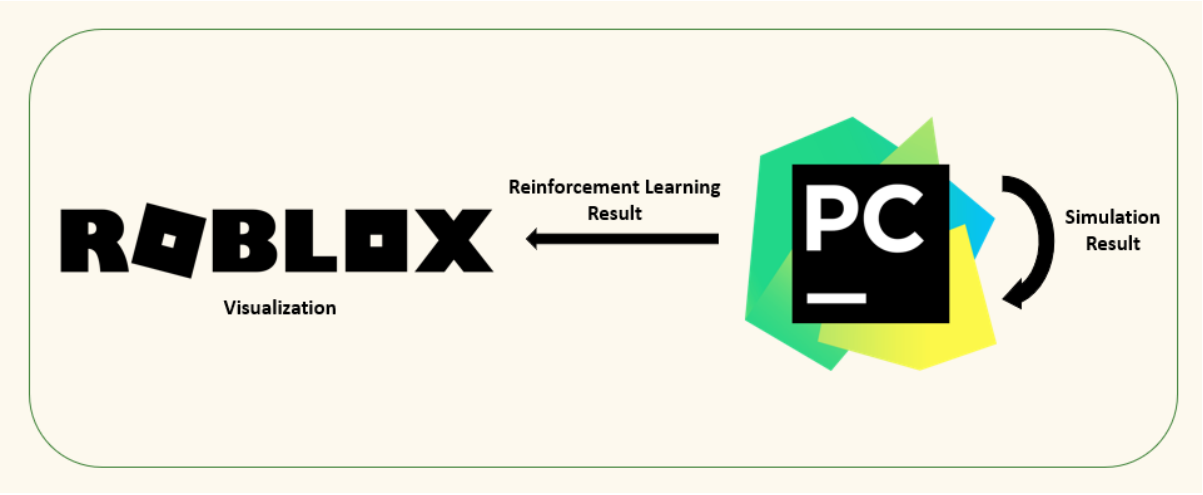
정환주, 이재영, 이호성, 조병화

- **역할 및 기여사항** (* 개인별 작성 부분)
  - 정환주(팀장) : 로블록스, 시뮬레이션
    - 기여 사항
      - 로블록스 및 시뮬레이션 개발 총괄
      - 동물들의 행동 패턴 설정
    - 이슈, 문제해결 내역
      - 로블록스 숙련도 문제로 상호작용 구현에 어려움을 겪었는데, 기존의 로블록스에서 Metable을 이용하여 property 설정 및 추가가 어렵던 상황에서 Table만 이용하도록 수정해서 property 설정 및 추가가 용이하도록 해결했다.
      - 시뮬레이션 코드의 가독성 개선
      - 2차원 환경에서 시뮬레이션을 진행하려고 한 만큼, 동물들이 겹칠 수 있다는 문제가 있었는데 이런 충돌 및 오류 방지 코드 작성
  - 이재영 : 강화학습, wiki 관리
    - 기여 사항
      - 강화학습 환경 개발 및 알고리즘 적용
      - wiki 관리/작성 - 매주 회의 진행될 때마다 wiki 작성
    - 이슈, 문제해결 내역
      - 알고리즘 설정의 문제 (DQN → A2C)
      - Environment를 구성하는 action space에 대해 증감/증가/감소 → 여러 실험과 멘토링 결과를 통해 증감으로 설정
      - 한 번의 강화학습이 8시간 이상 진행 → 적절한 결과가 나오는지 실험하기 위해 보상을 바꿔가며 여러 번의 강화학습을 진행/실험
  - 이호성 : 강화학습
    - 기여 사항
      - 강화학습 환경 개발 및 알고리즘 적용
      - OpenAI/gym의 다양한 environment 코드 분석
    - 이슈, 문제해결 내역
      - 간단한 식을 통한 각 강화학습 알고리즘의 성능 및 적용 가능성 파악
      - Environment의 state, action 및 reward를 주는 방식 구상 및 설정
      - 여러번의 강화학습 실험에 대한 결과 분석 및 개선 방안 모색
  - 조병화 : 로블록스, 시뮬레이션
    - 기여 사항
      - 시뮬레이션 코드 내 세부 요구 사항 구현
      - 로블록스 코드 내 동물 객체 모델과 행동, 움직임 구현
    - 이슈, 문제해결 내역

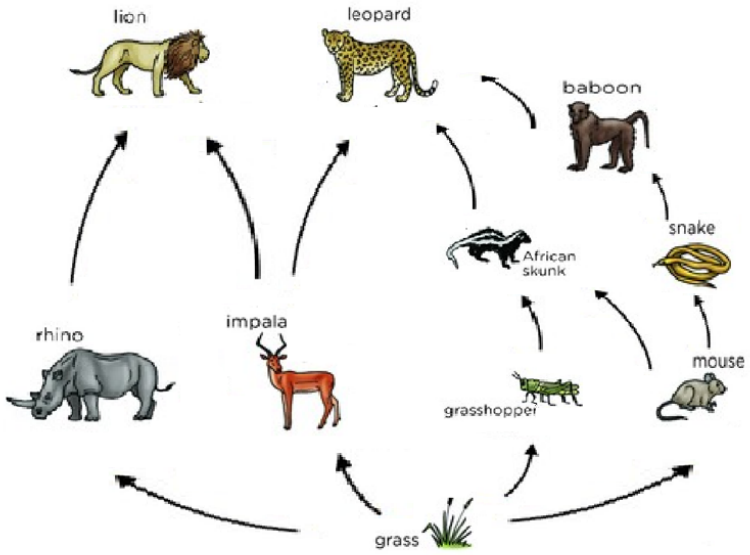
- 추가 기후 반영을 위하여 시뮬레이션 코드에 강수량에 따른 식물들의 생성 비율을 달리하여 자라도록 코드를 수정하였다.
- 사실적이면서도 너무 복잡하지 않은 동물 모델이 필요하여 일부 동물 들을 mesh 타입으로 사용함.

3. 개발 내용

- **전체 시스템 또는 SW 구조**



- **개발범위**
  - **필수항목**
    - 동물 10종 설정(사자, 임팔라, 원숭이, 코뿔소, 표범, 쥐, 메뚜기, 스컹크, 뱀, 풀)한 시뮬레이션 구현



    - 결과를 로블록스 환경에서 시각적으로 표현

- **추가 항목**
  - 강화학습을 통해 생태계가 오래 버틸 수 있는 초기 개체수의 값을 학습하여 이를 시각화한다.
  - 날씨에 따라 달라지는 생태계 환경 구현
  - 로블록스를 이용한 3D 가상 환경 시각적 리얼리티 강화

- **세부개발내용**

  적용 알고리즘, 오픈소스, 프레임웍, 플랫폼 등 포함.

  - **강화학습 Algorithm** : A2C 알고리즘

    A2C 이유 : Advantage Actor Critic 알고리즘으로 actor와 critic 2개의 neural network를 사용하여 학습하는 알고리즘으로 on-policy방식으로 에피소드가 끝나지 않아도 step마다 학습 가능

  - **강화학습 Environment :**

    목적에 맞게 강화학습을 진행할 environment를 생성했다.

    - observation space : 10가지 동물의 초기 개체 값으로 gym 의 Box type을 활용하였다.
    - action space : 10가지 개체수의 값 중 한가지를 증가/감소 하는 것으로 총 20가지
    - state : simulation을 진행할 초기 개체수의 배열
    - done : simulation 결과 목표 tick을 버티었거나 초기 개체수가 설정 범위를 넘어가는 경우 done 이 true 가 되어 한 episode가 종료된다.
    - reward ：목표 시간을 버틴 경우 보상을,목표 시간을 버티지 못하였거나 설정 범위를 넘어간 경우 페널티를 주었다.

  - **Platform**
    - Roblox : 능동적인 게임 제작 플랫폼으로 현재 VR, AR 시장에서 180여개국의 사용자를 확보하고 있는 플랫폼으로 시각적 표현을 목적으로 사용
    - Pycharm : IDE for python
  - 시뮬레이션
    - 가능한 현실적으로 보이도록 동물들의 패턴들을 설정
    - 동물들의 기본적인 패턴은
      1. 시야 범위 내부를 검색
      2. 포식자 감지 시도
      3. 먹이 검색
      4. 무작위 이동
      5. 번식

      의 5가지 패턴을 보인다.
    - 각 동물들은 시간이 흐름에 따라서 가지고 있는 에너지를 잃고, 에너지가 0이 되거나, 수명이 다하면 무대에서 사라지도록 했다.
    - 동물들은 먹이를 사냥해 먹음으로써, 에너지를 얻고, 본인이 지니고 있을 수 있는 최대 에너지의 일정 비율을 넘기면 번식율에 따라 번식하도록 설정했다.

        - 동물들의 이동방향은 포식자, 혹은 먹이가 존재하는 사분면의 반대 혹은 해당 방향으로 가도록 설정하여, 동물들의 이동방향을 되도록 합리적으로 구현했다.
        - 생태계의 유지를 위해서 초식동물들은 에너지가 부족해서 죽는 일이 없도록 하는 대신, 번식할 수 있는 경우도 풀을 섭취했을 때만 시도하도록 했다.
        - **시뮬레이션 종료조건** : 한 종이 멸종하면, 그 생태계의 안정성이 깨졌다고 판단하고, 해당 시뮬레이션을 종료하고 버틴 tick을 반환한다.
- **이슈사항 및 해결방안** (* 중요)
    - 별도 파일 첨부

4. 성과

- **목표치**
    - 기능 : 로블록스 상에서 특성값을 입력 받아 강화학습 시뮬레이션 하는 과정을 시각화 하여 안정적인 생태계를 구축하는 과정을 보여준다.
    - 목표 : 가상 생태계에서 특정 tick동안 멸종하지 않는 10종의 개체 수를 구한 후 로블록스 상에서 시각화
- **수행결과**
    - 생태계 시뮬레이션으로 강화학습을 진행하여 목표치를 안정적으로 달성하는 초기 개체수의 값을 구한 후 해당 생태계의 초기값으로 시뮬레이션 하는 과정을 로블록스 상에서 표현한다.
- **목표 대비 성과**
    - 강화학습하는 과정이 10시간 이상 걸리기 때문에 이를 모두 시각화 하기에는 무리가 있었다.

      → 강화학습을 별도로 진행하여 안정적인 생태계를 시각화 하는것으로 변경하였다.
    - 로블록스 내에서 똑같이 시뮬레이션을 실행하기에는 로블록스가 너무 느려서, 시각화를 위해서 동물들의 수를 조정하는 결과가 있었다.
    - 시뮬레이션이 매번 거의 일정한 값이 나오면 학습 속도를 더 줄일 수 있을텐데, 무작위적 요소가 강해서 같은 값에 대해서도 조금씩 다른 값이 나와 학습에 걸리는 시간이 늘어났다.

- **프로젝트 수행 산출물**
- 프로젝트 성과
    - 기존의 사람이 시뮬레이션을 돌려서 적절한 초기값을 추론하던 방식에서, 시뮬레이션만 적절히 작성하면 강화학습 알고리즘이 해당 종들과 생태계에 적합한 초기값을 학습하여, 개발자가 시뮬레이션 구현에 집중할 수 있도록 해준다.
    - 위에서 보인 생태계와 시뮬레이션에 대해서 강화학습 알고리즘이 목표로 한 tick 이상을 버티는 초기값을 반환하는 것을 확인

463	Success, Ecosystem's Animal Number : [[40 150 100 90 40 90 120 140 70 1500]]
464	Success, Ecosystem's Animal Number : [[40 150 100 90 40 90 120 120 70 1500]]
465	Success, Ecosystem's Animal Number : [[40 150 100 90 40 90 120 120 70 1500]]
466	Success, Ecosystem's Animal Number : [[40 150 100 90 40 90 120 120 70 1500]]
467	Success, Ecosystem's Animal Number : [[40 150 100 90 40 90 120 120 70 1500]]
468	Success, Ecosystem's Animal Number : [[40 150 100 90 40 90 120 120 70 1500]]
469	Success, Ecosystem's Animal Number : [[40 150 100 90 40 90 120 120 70 1500]]
470	

170 tick 이상을 버티는 초기값을 반환하는 강화학습 알고리즘

- 로블록스에서 시각화하여 사용자가 직접 생태계를 눈으로 확인할 수 있게 함

- 한계 / 목표 미달성 부분
  - 로블록스 가상환경 상에서 동물들이 동시에 움직이도록 하는 부분에 어려움이 있음
    - 로블록스의 느리고, 무거운 환경을 고려하지 않고 시뮬레이션을 작성하여, 일정 이상의 수의 동물들에 대해서는 현실적으로 coroutine을 적용할 수 없음을 뒤늦게 2차 멘토링에서 알 수 있었다.
    - 강화학습 알고리즘 및 시뮬레이션 구현에 집중하여 시간이 부족하여 생소한 lua script 언어를 사용하는 로블록스 내부의 동물들의 상호작용을 정교하게 구현하지는 못 했다.
  - 날씨 상호작용 부분에서 아쉬운 점
    - 간단한 건기, 우기 정도의 상호작용밖에 구현하지 못했다.
    - 프로젝트 마감이 다가와서 구현하여, 건기, 우기를 적용한 시뮬레이션의 데이터를 충분히 얻지 못 했다.

5. 검증

- **검증 방법**
  - 강화학습의 결과로 나온 개체수로 시뮬레이션을 여러번 돌려서 경험적으로 확인했다.
  - 임의의 초기 개체수으로 시뮬레이션을 돌려, 강화학습의 결과로 나온 값과의 시뮬레이션 결과 차이를 확인할 수 있었다.

```
simulate with array :  [25, 100, 75, 40, 25, 50, 90, 75, 50, 1250]
simulation 0 : 155
simulation 1 : 187
simulation 2 : 175
simulation 3 : 152
simulation 4 : 154
```

강화학습 결과로 얻은 초기 개체 수

```
simulate with array :  [35, 80, 75, 40, 35, 50, 50, 75, 50, 1250]
simulation 0 : 129
simulation 1 : 155
simulation 2 : 115
simulation 3 : 119
simulation 4 : 156
```

임의로 정한 개체수

- **검증 결과**
  - 강화학습의 결과로 나온 개체수로 시뮬레이션을 돌려본 결과 150tick 이상의 값이 반복적으로 나타나는 반면 임의의 초기 개체수으로 시뮬레이션을 돌리면 150tick을 못 넘는 경우가 많았다. 이를 통해 강화학습을 통해 안정적인 초기 개체수 값을 구했음을 알 수 있다.

6. 향후 개선방안

- 특성을 로블록스에서 입력 받은 후 시뮬레이션을 배속하여 강화학습 하는 과정 전체를 시각화하는 것으로 발전시킬 수 있을 것이다.
- 사용자 임의의 생태계 입력값을 입력 받아 게임처럼 조절가능하게 발전 시킬 수 있을 것이다.

- 현재 시뮬레이션에서는 바위, 산, 강 등의 환경 요소들이 없는데, 이후 이런 요소들을 더 추가하여 다양한 지형과 상호작용을 만들면 더 현실적인 시뮬레이션을 구현할 수 있을 것이다.
- 사자 같은 종의 경우 무리 사냥을 하고, 임팔라의 경우 귀가 좋고, 이런 동물들의 특징을 더 살리는 패턴을 구현하면 더 현실적인 시뮬레이션을 작성할 수 있었을 것 같다.
- 로블록스에서 종들의 상호작용만 구현했는데, 이를 발전시켜 유저가 동물들과 상호작용하는 일종의 게임과도 같이 개선할 수 있을 거 같다.