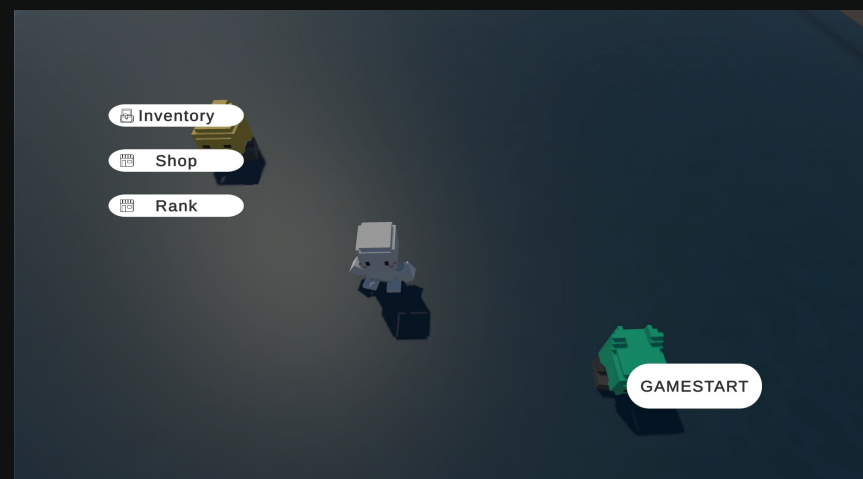




MIRI
INTELLIGENCE

실전 SW 프로젝트 기말 발표

2018320126 김진우



목차



1. 게임 개요

하이퍼 캐주얼 게임의 특징, 개발 환경,
게임 목표 소개

2. 게임 메커니즘

플레이어와 AI의 공통 기능, 기본 스탯,
특수 기술 설명

3. AI 시스템

Enemy 1(직접 추격형)과 Enemy
2(전략적 추격형)의 AI 시스템 상세 설명

4. 게임 데모

AI 파라미터별 행동 패턴 및 실제
게임플레이 시연

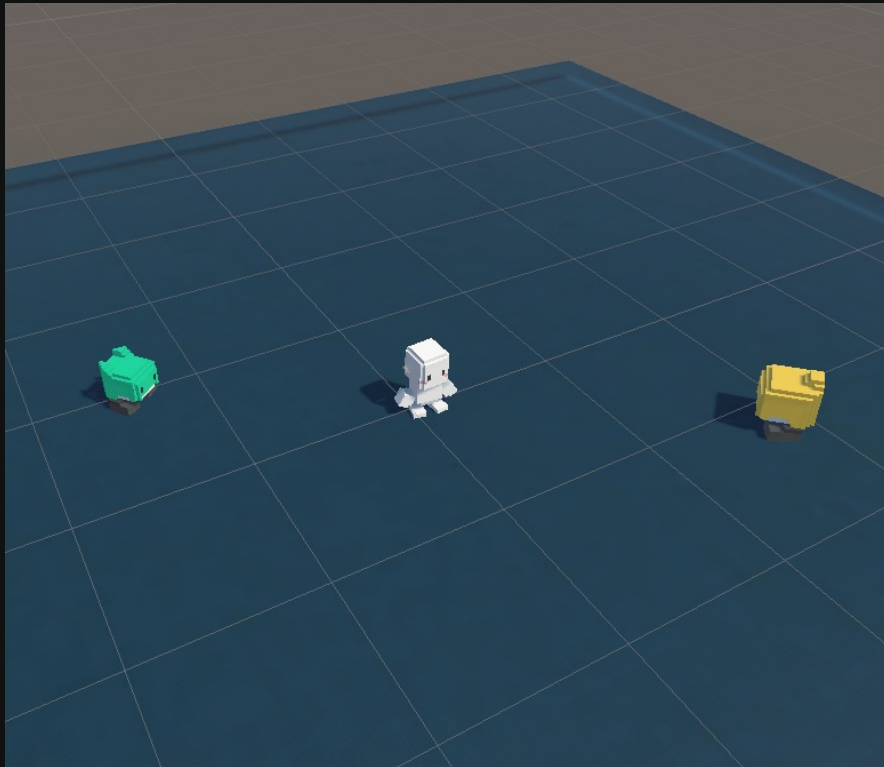
5. 향후 개발 계획

추가 기능, 게임 밸런스 조정, 시각적 효과
개선 계획

6. 결론

프로젝트 요약, 학습 포인트, 향후 발전
방향 및 비전 제시

게임 개요



* Life Runner

.....

장르: 하이퍼 캐주얼 게임

개발 환경: Unity 3D C#

게임 목표: AI 적 2개로부터 오래 생존하기

게임 개요 – Unit 설정



세상에서 제일 빠른 우사인볼트를 두 명이
24시간 안에 잡을 수 있을까?
(잡으면 1억, 못 잡으면 감옥 1년)

내가 잡는다는 생각으로 Enemy Logic 구현

실제 사람의 달리기를 측정 하는 데이터

체력

속도

State 마다 다른 Unit Controll

Run : 체력 소모

Walk: 체력 점점 회복

체력 = 0 이면 못달림

게임 설계



기본 스탯

플레이어와 시 모두에게 적용되는 체력 시스템과 이동 속도.
체력은 지속적으로 감소하며, 이동 속도는 기본값으로 설정됨.

스킬 1 : Run

체력을 소모하여 이동 속도를 일시적으로 증가시키는 기술.
전략적 사용이 중요하며, 체력 관리가 필수적임.

스킬 2 : Dash

높은 체력을 소모하여 순간적으로 빠르게 이동하는 기술. 위기
상황을 탈출하거나 시를 따돌리는 데 유용함.

시와의 상호작용

플레이어는 시의 움직임을 예측하고 대응해야 함. 시의 추적을
피하면서 생존 시간을 최대화하는 것이 핵심 전략임.

게임 설계



데이터 관리(GameDataManager)

각 개체의 스탯을 Json파일로 저장 및 로드

Player

```
{  
  "Stamina": 100,  
  "StaminaRecoveryRate": 5,  
  "StaminaUseRate": 2,  
  "WalkSpeed": 10,  
  "RunSpeed": 15  
}
```

Enemy

```
{  
  "stamina": 80,  
  "maxStamina": 80,  
  "staminaRecoveryRate": 4,  
  "WalkSpeed": 10,  
  "RunSpeed": 15  
}
```

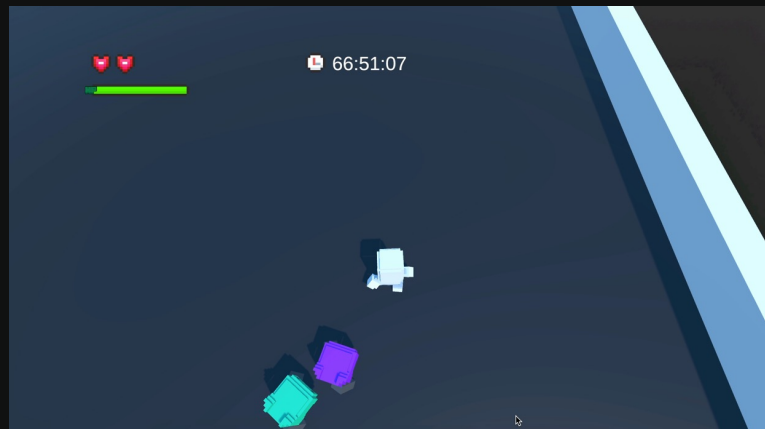
조작키

플레이어 기본 이동: WASD방향키

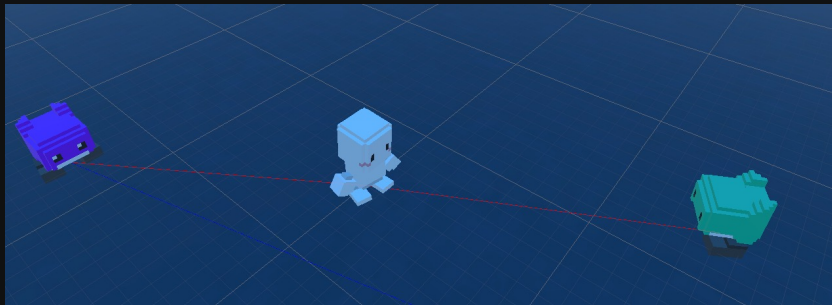
Run: Shift 누르면서 이동

Dash: Space

AI 시스템 - Enemy 1



AI 시스템 - Enemy 1



직접 추격형 AI

플레이어를 직선으로 추적하는 단순하지만 효과적인 AI. 예측 가능한 움직임으로 플레이어에게 지속적인 압박을 가함. 빠른 반응 속도가 특징.

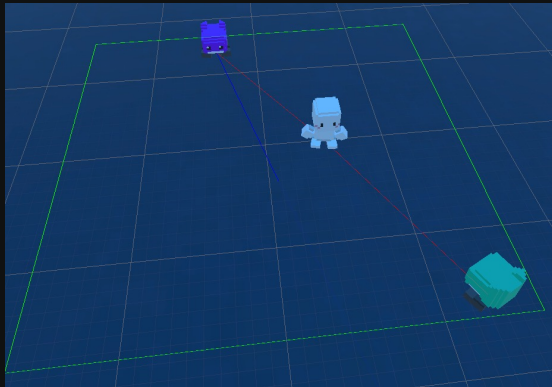
```
private void ChasePlayerDirectly()
{
    Vector3 direction = (player.position - transform.position).normalized;
    transform.position += direction * speed * Time.deltaTime;
    // 회전 업데이트
    RotateTowards(direction);

    MaintainSeparation(); // 적 간 거리 유지
}
```

기본 추격 알고리즘

NavMesh를 활용한 기본적인 경로 탐색 시스템 구현. 장애물을 피해 최단 경로로 플레이어를 추적. 단순하지만 효율적인 추적 로직으로 게임의 난이도 조절에 기여.

AI 시스템 - Enemy 2



```
// 적 2: 플레이어의 이동 방향 예측 및 추적  
참조 1개  
private void ChasePlayerWithPrediction()  
{  
    // 플레이어의 예측 위치 계산  
    Vector3 futurePosition = player.position + player.forward * predictionDistance;  
    Vector3 direction = (futurePosition - transform.position).normalized;  
    // 회전 업데이트  
    RotateTowards(direction);  
  
    // 적 간 거리 유지  
    MaintainSeparation();  
  
    // 이동  
    transform.position += direction * speed * Time.deltaTime;  
}
```

전략적 추격형 AI

플레이어의 이동 경로를 예측하여 효율적으로 추적합니다. 직선 추적이 아닌 지능적인 경로 선택으로 플레이어를 효과적으로 압박합니다.

전략적 포지셔닝 구현

Enemy 1과 적절한 거리를 유지하며 플레이어를 포위하는 전략을 구사합니다. 이를 통해 플레이어의 탈출 경로를 차단하고 생존을 어렵게 만듭니다.

AI 시스템



```
private void ChasePlayerDirectly()
{
    Vector3 direction = (player.position - transform.position).normalized;
    transform.position += direction * speed * Time.deltaTime;
    // 회전 업데이트
    RotateTowards(direction);

    MaintainSeparation(); // 적 간 거리 유지
}
```

```
// 적 2: 플레이어의 이동 방향 예측 및 추적
참조 1개
private void ChasePlayerWithPrediction()
{
    // 플레이어의 예측 위치 계산
    Vector3 futurePosition = player.position + player.forward * predictionDistance;
    Vector3 direction = (futurePosition - transform.position).normalized;
    // 회전 업데이트
    RotateTowards(direction);

    // 적 간 거리 유지
    MaintainSeparation();

    // 이동
    transform.position += direction * speed * Time.deltaTime;
}
```

전략적 추격형 AI

플레이어의 이동 경로를 예측하여 효율적으로 추적합니다. 직선 추적이 아닌 지능적인 경로 선택으로 플레이어를 효과적으로 압박합니다.

전략적 포지셔닝 구현

Enemy 1과 적절한 거리를 유지하며 플레이어를 포위하는 전략을 구사합니다. 이를 통해 플레이어의 탈출 경로를 차단하고 생존을 어렵게 만듭니다.

게임 데모 - AI 행동 패턴



Enemy 1 조정

직접 추격형 AI의 속도와 반응 시간을
조절하여 다양한 난이도 설정이 가능

높은 속도 설정 시 긴박한 추격전이,
낮은 속도에서는 전략적 회피가 가능한
게임플레이를 제공함

Enemy 2 조정

전략적 추격형 AI의 예측 정확도와
포지셔닝 빈도를 조정할 수 있습니다.

높은 정확도 설정은 숙련된 이동 및
체력 관리, 낮은 정확도는 초보자에게
적합한 난이도를 제공

AI 상호작용

두 AI 간의 협력 수준을 조절하여
게임의 복잡성을 높일 수 있다

향후 AI 모델링을 통해 학습된
데이터로 더욱 체계적이고 정교한
움직임을 구현할 수 있음

게임 DEMO - 실제 게임플레이



향후 개발 계획 - 추가 기능



캐릭터마다 다른 스킬

투사체 발사, 장애물 설치, 체력 회복 등 각 캐릭터의 특색에 맞게 스킬 구현
처음 게임 컨셉에 맞게 스포츠 종목으로 구성
우사인볼트: 전력질주
마라톤 선수: 체력 무한
양궁, 사격선수: 투사체 발사

다양한 맵 구현

도시 환경, 숲, 미로 등 다양한 테마의 맵 구현

상점 및 업그레이드

플레이어가 게임Gold를 통해 스텟 업그레이드 및 아이템 구매를 하고 더 어려운 남이도에 도전할 수 있도록 한다.

향후 개발 계획 - 밸런스 조정



능력치 조정

플레이어와 시의 속도, 체력, 스킬 사용 빈도 등을 세밀하게 조정

데이터 분석을 통해 승률과 생존 시간을 모니터링하여 최적의 밸런스를 찾아가고 특히 시의 지능과 플레이어의 기술 사용 간의 균형을 맞추는 데 중점

난이도 시스템

각스태이지에서 시의 행동 패턴, 반응 속도, 예측 능력을 조절하여 플레이어의 실력에 맞는 도전을 제공하고 업그레이드 기능을 통해 높은 난이도의 스테이지 도전 유도

AI

현재는 Enemy1, Enem2의 역할이 처음부터 정해지는데 Player와의 거리 및 방향에 따라 각자의 Strategy가 알맞게 변경되도록



MIRI
INTELLIGENCE

