README

This is a mini game where the player must steal treasure from caves guarded by 2 ogres, and carry the treasure to spawn position. Ogres are controlled by a custom Hierarchical Task Network (HTN) AI system.

Scenes: Main (gameplay), Win (win UI), Lose (lose UI)
Controls: WASD: Movement, Mouse: Look around, Space: Toggle invisibility (10 seconds total)

Features:

1. Level Generation (Script: ObstacleSpawner)

On each playthrough,10-20 boulder obstacles and 5-10 mushrooms spawn randomly. The overlapping check is mainly adapted from Assignment 3 - checking bounding circles on the XZ plane. The boulders are generated with varying size and rotation.

2. Player (Script: PlayerController)

The controls are mainly adapted from Assignment 1, first person camera with WASD movement and mouse look. Invisibility is toggled with the space key, limited to 10 seconds total. The acquisition of treasure is by colliding with the treasure collider, which is set to be larger than the actual mesh. Losing lives is done by a method that will be called in the ogre's behaviours. UI displays the number of lives left, Treasure number, invisibility remaining time and status.

3. Ogre AI (Script: OgreController)

Two ogres use the same HTN-based AI. They patrol near their caves, forage for mushrooms when hungry, and attack the player on sight. They have 4 idle moves: 1. walking to a random position near the cave, 2. Stop and turn around, 3. Picking up a boulder, 4. Foraging mushrooms. In idle, their speed is slower, 5f. They also have 2 attack moves, 1. Throwing the boulder, 2. Melee attack. When attacking, their speed is set to 12f (slightly higher than player). The FOV of ogres has limited angle (120°) and range (~50 units). They can not see invisible players or players behind an obstacle (boulder). The ogres are immediately aware when treasure is stolen, regardless of their location. After the treasure is stolen, they will rush to the stealing position, know the player's location permanently (but will lose target if the player toggles invisibility), and will only choose melee attacks. See next page for HTN design.

4. HTN Visual Display (Script: SplitScreenManager)

For each ogre, displays:

- Current world state flags (sees player, has boulder, treasure stolen, hungry, etc.)
- Complete task plan with status indicators (done, current, pending)
- Sub-steps within the current task showing internal state machine progress

Generative AI is used to help construct SplitScreenManager.cs.

HTN System Design

World State:

Variable	Type	Description
treasureStolen	bool	Whether player has taken any treasure
treasureInvestigated	bool	Whether ogre has checked the theft location
seesPlayer	bool	Player currently visible (in FOV, in range, not invisible)
playerInAttackRange	bool	Player within melee attack distance
playerInvisible	bool	Player's invisibility is active
distanceToPlayer	float	Distance to player position
distanceToNearestBoulder	float	Distance to closest pickup-able boulder
boulderCloserThanPlayer	bool	Whether nearest boulder is closer than player
hasBoulderInHand	bool	Ogre currently holding a boulder
hungry	bool	Ogre needs to eat (triggers random 15-25 seconds)
mushroomAvailable	bool	At least one mushroom exists in the level
isAlert	bool	Ogre is in heightened awareness state
lastSeenPlayerPos	Vector3	Last known player position
lastTreasureSpot	Vector3	Location where treasure was stolen
timeSinceLastSeen	float	Time elapsed since player was last visible
homePosition	Vector3	Ogre's cave position

HTN Structure and preconditions:

ManageOgre (Root Compound Task)

- Method: AttackPlayer (seesPlayer && !playerInvisible)
    - EngagePlayer
        - Method: EngageThrow (seesPlayer && hasBoulderInHand)
            - ThrowBoulder
        - Method: EngageMelee (seesPlayer && !hasBoulderInHand)
            - MeleeAttack
- Method: InvestigateTreasure (treasureStolen && !treasureInvestigated)
    - GoToTreasureSpot

- Method: HuntPlayer (treasureStolen && treasureInvestigated && !playerInvisible)
    - EngagePlayer
- Method: IdleAfterTreasure (treasureStolen && treasureInvestigated && playerInvisible)

Idle

- Method: EatWhenHungry (!treasureStolen && hungry && mushroomAvailable)
    - GoToMushroom
    - ForageMushroom
    - ReturnHome
- Method: Idle (!treasureStolen && !seesPlayer && !hungry)
    - Idle (randomly choose one of three)
        - Method: Walk (always)
            - WalkRandomly
        - Method: LookAround (always)
            - LookAround
        - Method: PickUpBoulder (!hasBoulderInHand && distanceToNearestBoulder < infinity)
            - PickUpBoulder

Tasks details:

WalkRandomly

- Preconditions: None
- Effects: None
- Behavior: Navigate to random point within patrol radius of cave
- Sub-steps: Choosing Destination → Walking → Reached Destination

LookAround

- Preconditions: None
- Effects: None
- Behavior: Turn left, right, then forward
- Sub-steps: Turn Left → Turn Right → Look Forward → Complete

GoToMushroom

- Preconditions: hungry && mushroomAvailable
- Effects: None
- Behavior: Navigate to nearest mushroom
- Sub-steps: Finding Mushroom → Moving to Mushroom → Arrived

ForageMushroom

- Preconditions: hungry && mushroomAvailable
- Effects: hungry := false
- Behavior: Wait 1 second, then destroys mushroom object
- Sub-steps: Starting to Eat → Eating Mushroom → Finished Eating

ReturnHome

- Preconditions: None
- Effects: None
- Behavior: Navigate back to cave entrance

- Sub-steps: Finding Home → Walking Home → Arrived Home

MeleeAttack

- Preconditions: seesPlayer && !hasBoulderInHand
- Effects: None
- Behavior: Chase player, attack on contact. If the player escapes, go to the last seen position and search.
- Sub-steps: Chasing Player → Going to Last Seen → Searching Area → Complete

PickUpBoulder

- Preconditions: !hasBoulderInHand && distanceToNearestBoulder < infinity
- Effects: hasBoulderInHand := true
- Behavior: Navigate to nearest boulder and pick it up
- Sub-steps: Running to Boulder → Picking Up → Complete

ThrowBoulder

- Preconditions: seesPlayer && hasBoulderInHand
- Effects: hasBoulderInHand := false
- Behavior: Approach player until in throw range (25m), then throw boulder
- Sub-steps: Approaching Player → Throwing Boulder → Complete

GoToTreasureSpot

- Preconditions: treasureStolen && !treasureInvestigated
- Effects: treasureInvestigated := true
- Behavior: Navigate to location where treasure was stolen
- Sub-steps: Locating Theft Spot → Going to Location → Investigating

HTN Planner Implementation

Methods are registered in priority order. The first method whose preconditions are satisfied is selected, except for Idle which randomly selects among valid methods to create varied behavior.
The ogre replans when:

- Current plan completes
- World state changes significantly (player visibility, treasure stolen, hunger)
- Current task's preconditions become invalid

Limitations

1. Boulder throw range is fixed at 25m; ogres will chase until in range.
2. The HTN UI covers part of the in game screen.