Exercise 2 — Behavior Tree Design

Behavior Tree Structure

```
Root
→ Selector
    → Sequence (Attack Branch)
        Condition: Player Visible
        Condition: Player In Attack Range
        Action: Attack

    → Sequence (Chase Branch)
        Condition: Player Visible
        Condition: Player Out of Range
        Action: Chase Player

    → Sequence (Search Branch)
        Condition: Player Seen Recently
        Action: Move to Last Known Position

    → Action: Patrol
```

Node Labels

Selector (priority-based decision)

Sequences:

- Attack Sequence
- Chase Sequence
- Search Sequence

Condition Nodes:

- IsPlayerVisible
- IsPlayerInRange
- WasPlayerSeenRecently

Action Nodes:

- Attack
- Chase

- Search
- Patrol

Exercise 3 — Unity Practice AI (FSM + NavMesh)

Required States

Idle

- Guard waits for a few seconds.

Patrol

- Guard walks between waypoints using NavMeshAgent.

Chase

- Guard runs toward player if detected.

Return

- Guard goes back to patrol route if player lost.

FSM Logic

Idle → Patrol
Condition: Timer ends.

Patrol → Chase
Condition: Player enters detection radius.

Chase → Patrol
Condition: Player lost for X seconds.

Patrol → Idle
Optional short pause at waypoint.

Unity Setup

1. Create Plane (Ground).
2. Create Player (Capsule + CharacterController).
3. Create Guard (Capsule + NavMeshAgent).
4. Bake NavMesh from Navigation window.
5. Add detection radius (Physics.OverlapSphere).

6. Use NavMeshAgent.SetDestination() for movement.

Expected Result

- Guard moves naturally between waypoints.
- Reacts when player approaches.
- Chases logically.
- Returns to patrol if player escapes.
- Behavior feels structured and predictable.