```
private void Update()
{
//Check for sight and attack range
playerInSightRange = Physics.CheckSphere(transform.position, sightRange,
whatIsPlayer);
playerInAttackRange = Physics.CheckSphere(transform.position, attackRange,
whatIsPlayer);

           if (!playerInSightRange && !playerInAttackRange) Patroling();
           if (playerInSightRange && !playerInAttackRange) ChasePlayer();
           if (playerInAttackRange && playerInSightRange) AttackPlayer();

           if (!playerInSightRange && !playerInAttackRange)
ChangingAnimationState("Idle_1");
}
void ChangingAnimationState(string newState)
{
    if (currentState == newState) return;
        animator.Play(newState);
        currentState = newState;
}

private void Patroling()
{
    footsteps.Play();
    footSteps.SetActive(true);
    if (!walkPointSet) SearchWalkPoint();

    if (walkPointSet)
       agent.SetDestination(walkPoint);
       Vector3 distanceToWalkPoint = transform.position - walkPoint;

//Walkpoint reached
    if (distanceToWalkPoint.magnitude < 1f)
        walkPointSet = false;
}
private void SearchWalkPoint()
{
    //Calculate random point in range
    float randomZ = Random.Range(-walkPointRange, walkPointRange);
    float randomX = Random.Range(-walkPointRange, walkPointRange);

    walkPoint = new Vector3(transform.position.x + randomX, transform.position.y,
transform.position.z + randomZ);

    if (Physics.Raycast(walkPoint, -transform.up, 2f, whatIsGround))
```

```
          walkPointSet = true;
}
private void ChasePlayer()
{
       agent.SetDestination(player.position);
       ChangingAnimationState("Walk");
}
private void AttackPlayer()
{
     player.GetComponent<MovementProvider>().speed = 0;
     footsteps.Stop();
     jumpScare.SetActive(true);
     footSteps.SetActive(false);
    //Make sure enemy doesn't move
     agent.SetDestination(transform.position);
     transform.LookAt(player);
     ChangingAnimationState("Attack_1");
     endGame.SetActive(true);

     if (!alreadyAttacked)
    {
       alreadyAttacked = true;
       Invoke(nameof(ResetAttack), timeBetweenAttacks);
    }
}
private void ResetAttack()
{
      alreadyAttacked = false;
}
private void OnDrawGizmosSelected()
{
       Gizmos.color = Color.red;
       Gizmos.DrawWireSphere(transform.position, attackRange);
       Gizmos.color = Color.yellow;
       Gizmos.DrawWireSphere(transform.position, sightRange);
}
```