

```
○○○

///<Summary>
/// Handles player interaction with the locker
///<Summary>
public void Interact()
{
    if (Cooldown) return;

    if (_doorAnimator.GetBool("open"))
    {
        HandleClose();
    }
    else
    {
        HandleOpen();
    }
    Cooldown = true;
}

///<Summary>
/// Opens the locker, triggers animation and sound, starts cooldown
///<Summary>
private void HandleOpen()
{
    _doorAnimator.SetBool("open", true);
    StartCoroutine(CooldownForInteract(_cooldownForOpen));
    PlaySoundEffect(_openClip, 1f, 0.65f);
    _lastSoundPointRepeatCount += _gameEngineScript.CheckDollDistance(transform,
_dollHearingRange, _dollHearingRange + _lastSoundPointRepeatCount);
}

///<Summary>
/// Closes the locker, triggers animation and sound, starts cooldown
///<Summary>
private void HandleClose()
{
    _doorAnimator.SetBool("open", false);
    StartCoroutine(CooldownForInteract(_cooldownForClose));
    PlaySoundEffect(_closeClip, 1f, 0.85f);
    _gameEngineScript.CheckDollDistance(transform, _dollHearingRange, _dollHearingRange +
_lastSoundPointRepeatCount);
}

///<Summary>
/// Coroutine to manage cooldown between interactions
///<Summary>
private IEnumerator CooldownForInteract(float time)
{
    yield return new WaitForSeconds(time);
    Cooldown = false;
}

///<Summary>
/// Plays a specific sound effect with volume and pitch settings
///<Summary>
private void PlaySoundEffect(AudioClip clip, float volume, float pitch)
{
    _audioSource.volume = volume;
    _audioSource.pitch = pitch;
    _audioSource.clip = clip;
    _audioSource.Play();
}

///<Summary>
/// Triggers the locker animation when catching the player
///<Summary>
public void CatchPlayer()
{
    _doorAnimator.SetTrigger("catch_player");
}
```