

○○○

```
///  
///<Summary>  
/// Handles the interaction logic when the player interacts with the cabinet.  
/// Checks cooldown, door state, and doll hearing range, and triggers object spawning if needed.  
///  
public void Interact()  
{  
    if (Cooldown) return;  
  
    if (_doorAnimator.GetBool("open")) CloseDoor();  
    else OpenDoorAndSpawn();  
  
    _lastSoundPointRepeatCount += _gameEngineScript.CheckDollDistance(transform,  
_dollHearingRange, _dollHearingRange + _lastSoundPointRepeatCount);  
    Cooldown = true;  
}  
  
///  
///<Summary>  
/// Opens the door with animation and sound, starts cooldown.  
///  
private void OpenDoor()  
{  
    SetDoorState(true, _cooldownForOpen, _openClip, 0.7f);  
}  
  
///  
///<Summary>  
/// Closes the door with animation and sound, starts cooldown.  
///  
private void CloseDoor()  
{  
    SetDoorState(false, _cooldownForClose, _closeClip, 1f);  
}  
  
///  
///<Summary>  
/// Handles opening the door and spawning objects if cabinet is empty.  
///  
private void OpenDoorAndSpawn()  
{  
    OpenDoor();  
  
    if (!_spawnRandomObjects || !_spawnRandomObjects.HasItems)  
    {  
        StartCoroutine(_gameEngineScript.SetNewReplieForEmpty());  
    }  
}  
  
///  
///<Summary>  
/// Sets the door state, plays corresponding audio clip, and starts cooldown coroutine.  
///  
private void SetDoorState(bool isOpen, float cooldownTime, AudioClip clip, float fallbackPitch)  
{  
    _doorAnimator.SetBool("open", isOpen);  
    StartCoroutine(CooldownForInteract(cooldownTime));  
  
    if (clip) _audioSource.clip = clip;  
    else  
    {  
        _audioSource.clip = _openCloseClip;  
        _audioSource.pitch = fallbackPitch;  
    }  
    _audioSource.Play();  
}  
  
///  
///<Summary>  
/// Coroutine that resets interaction cooldown after a delay.  
///  
private IEnumerator CooldownForInteract(float time)  
{  
    yield return new WaitForSeconds(time);  
    Cooldown = false;  
}  
  
///  
///<Summary>  
/// Draws a visual representation of the doll hearing range in the editor.  
///  
private void OnDrawGizmosSelected()  
{  
    Gizmos.color = Color.yellow;  
    Gizmos.DrawWireSphere(transform.position, _dollHearingRange);  
}
```