```
/// <summary>
    /// Handles casting of spells if requirements are met.
    /// </summary>
    private void HandleMagic()
    {
        if (!Input.GetKeyDown(KeyCode.LeftControl) || controller.you_are_death) return;

        var spells = inventory.get_all_incantationMagic();
        if (spells.Count == 0) return;

        var spell = spells[inventory.get_Iandex_ofIncantation()];
        if (stats.all_curentStats[2] >= spell.MP_substract && incantationCooldown <= 0)
            CastSpell(spell);
    }

    private void CastSpell(magic_incantation_Manager spell)
    {
        incantationCooldown = spell.cooldown;

        GameObject newSpell = Instantiate(spell.gameObject, magicIncantation.transform);
        newSpell.transform.localPosition = Vector3.zero;

        stats.all_curentStats[2] -= spell.MP_substract;
        stats.all_curentStats[9] += spell.temporary_arrmor;

        Destroy(newSpell, spell.cooldown - 1);
        StartCoroutine(ResetIncantationBuff(spell.cooldown, spell.temporary_arrmor));
    }

    private IEnumerator ResetIncantationBuff(float cooldown, float armor)
    {
        yield return new WaitForSeconds(cooldown);
        stats.all_curentStats[9] -= armor;
    }

    /// <summary>
    /// Handles shield activation and block timing.
    /// </summary>
    private void HandleShield()
    {
        if ((Input.GetMouseButton(1) && !controller.you_are_death && shieldBlockTimer <= 0 &&
inventory.cooldownEchipe <= 0)
            && (attackCooldown <= 0.7 && attackCooldown >= 0.3 || attackCooldown <= 0))
        {
            if (inventory.sword_echipe_now != null)
                inventory.sword_echipe_now.boxCollider.enabled = false;

            if (canUseShield) { shieldDuration = 1; canUseShield = false; }

            attackCooldown = 0;
            swordAnimator.SetBool("shield", true);
            equipCooldown = 1.5f;
        }
        else
        {
            canUseShield = true;
            swordAnimator.SetBool("shield", false);
        }

        if (Input.GetMouseButtonUp(1))
            shieldBlockTimer = shieldBlockCooldown;
    }

    /// <summary>
    /// Handles melee combo attacks in 3 steps, consuming stamina.
    /// </summary>
    private void HandleAttack()
    {
        if (controller.you_are_death || inventory.sword_echipe_now == null || isInventoryOpen ||
!hasSwordEquipped) return;

        if (attackStep == 1 && Input.GetMouseButtonDown(0) && attackCooldown <= 0 &&
stats.all_curentStats[1] > 10)
            PerformAttack(10, 2);
        else if (attackCooldown <= 0.7 && attackCooldown >= 0.5 && attackStep == 2 &&
Input.GetMouseButtonDown(0) && stats.all_curentStats[1] > 12.5f)
            PerformAttack(12.5f, 3);
        else if (attackCooldown <= 0.5 && attackCooldown >= 0.3 && attackStep == 3 &&
Input.GetMouseButtonDown(0) && stats.all_curentStats[1] > 15)
            PerformAttack(15, 1);
    }

    private void PerformAttack(float staminaCost, int nextStep)
    {
        attackCooldown = savedAttackCooldown;
        swordAnimator.SetTrigger("attack");
        inventory.sword_echipe_now.boxCollider.enabled = true;

        attackStep = nextStep;
        SubtractStamina(staminaCost);

        stats.generate_voidSound(transform, stats.sword_attack_clip, 0.5f);
        equipCooldown = 1.2f;
    }

    private void SubtractStamina(float amount)
    {
        if (stats.numbers_of_enemy_is_in_battle_with_player > 0)
            stats.all_curentStats[1] -= amount;
    }

    /// <summary>
    /// Updates cooldown timers for spells, attacks, shield and equips.
    /// </summary>
    private void UpdateCooldowns()
    {
        if (equipCooldown > 0) equipCooldown -= Time.deltaTime;
        if (incantationCooldown > 0) incantationCooldown -= Time.deltaTime;
        if (shieldBlockTimer > 0) shieldBlockTimer -= Time.deltaTime;
        if (shieldDuration > 0) shieldDuration -= Time.deltaTime;
        if (attackCooldown > 0) attackCooldown -= Time.deltaTime;
        else attackStep = 1;

        if (attackCooldown <= 0.5 && inventory.sword_echipe_now != null)
            inventory.sword_echipe_now.boxCollider.enabled = false;
    }

    // ---------------- PUBLIC HELPERS ----------------
    /// <summary> Increases base damage permanently. </summary>
    public void UpdateBaseDamage(float value) => baseDamage += value;

    /// <summary> Returns current attack cooldown. </summary>
    public float GetAttackCooldown() => attackCooldown;

    /// <summary> Opens/closes inventory and locks combat if true. </summary>
    public void SetInventoryOpen(bool open) => isInventoryOpen = open;

    /// <summary> Returns true if inventory is currently open. </summary>
    public bool GetInventoryOpen() => isInventoryOpen;

    /// <summary> Updates sword damage value and logs new total damage. </summary>
    public void SetSwordDamage(float dmg)
    {
        swordDamage = dmg;
        Debug.Log($"Sword Damage: {dmg}, Total: {dmg + baseDamage}");
    }

    /// <summary> Returns remaining spell cooldown. </summary>
    public float GetIncantationCooldown() => incantationCooldown;
```