

Dungeon

```
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    public static PlayerController instance;

    public float moveSpeed;
    private Vector2 moveInput;

    public Rigidbody2D theRB;

    public Transform gunArm;

    public Animator anim;

    public SpriteRenderer bodySR;

    private float activeMoveSpeed;
    public float dashSpeed = 8f, dashLength = .5f, dashCooldown = 1f, dashInvincibility
= .5f;
    [HideInInspector]
    public float dashCounter;
    private float dashCoolCounter;

    [HideInInspector]
    public bool canMove = true;

    public List<Gun> availableGuns = new List<Gun>();
    [HideInInspector]
    public int currentGun;

    private void Awake()
    {
        instance = this;

        DontDestroyOnLoad(gameObject);
    }

    // Start is called before the first frame update
    void Start()
    {
        //theCam = Camera.main;

        activeMoveSpeed = moveSpeed;

        UIController.instance.currentGun.sprite = availableGuns[currentGun].gunUI;
        UIController.instance.gunText.text = availableGuns[currentGun].weaponName;
    }

    // Update is called once per frame
    void Update()
    {
        if (canMove && !LevelManager.instance.isPaused)
        {
            moveInput.x = Input.GetAxisRaw("Horizontal");
            moveInput.y = Input.GetAxisRaw("Vertical");

            moveInput.Normalize();

            theRB.velocity = moveInput * activeMoveSpeed;

            Vector3 mousePos = Input.mousePosition;
```

```

        Vector3 screenPoint =
CameraController.instance.mainCamera.WorldToScreenPoint(transform.localPosition);

        if (mousePos.x < screenPoint.x)
        {
            transform.localScale = new Vector3(-1f, 1f, 1f);
            gunArm.localScale = new Vector3(-1f, -1f, 1f);
        }
        else
        {
            transform.localScale = Vector3.one;
            gunArm.localScale = Vector3.one;
        }

        Vector2 offset = new Vector2(mousePos.x - screenPoint.x, mousePos.y -
screenPoint.y);
        float angle = Mathf.Atan2(offset.y, offset.x) * Mathf.Rad2Deg;
        gunArm.rotation = Quaternion.Euler(0, 0, angle);

        if(Input.GetKeyDown(KeyCode.Tab))
        {
            if(availableGuns.Count > 0)
            {
                currentGun++;
                if(currentGun >= availableGuns.Count)
                {
                    currentGun = 0;
                }

                SwitchGun();
            }
            else
            {
                Debug.LogError("Player has no guns!");
            }
        }

        if (Input.GetKeyDown(KeyCode.Space))
        {
            if (dashCoolCounter <= 0 && dashCounter <= 0)
            {
                activeMoveSpeed = dashSpeed;
                dashCounter = dashLength;

                anim.SetTrigger("dash");

                PlayerHealthController.instance.MakeInvincible(dashInvincibility);

                AudioManager.instance.PlaySFX(8);
            }
        }

        if (dashCounter > 0)
        {
            dashCounter -= Time.deltaTime;
            if (dashCounter <= 0)
            {
                activeMoveSpeed = moveSpeed;
                dashCoolCounter = dashCooldown;
            }
        }
    }
}

```

```

if (dashCoolCounter > 0)
{
    dashCoolCounter -= Time.deltaTime;
}

```

```

if (moveInput != Vector2.zero)
{
    anim.SetBool("isMoving", true);
}
else
{
    anim.SetBool("isMoving", false);
}
} else
{
    theRB.velocity = Vector2.zero;
    anim.SetBool("isMoving", false);
}
}

```

```

public void SwitchGun()
{
    foreach(Gun theGun in availableGuns)
    {
        theGun.gameObject.SetActive(false);
    }

    availableGuns[currentGun].gameObject.SetActive(true);

    UIController.instance.currentGun.sprite = availableGuns[currentGun].gunUI;
    UIController.instance.gunText.text = availableGuns[currentGun].weaponName;

    AudioManager.instance.PlaySFX(6);
}
}

```