Tank Man Demo

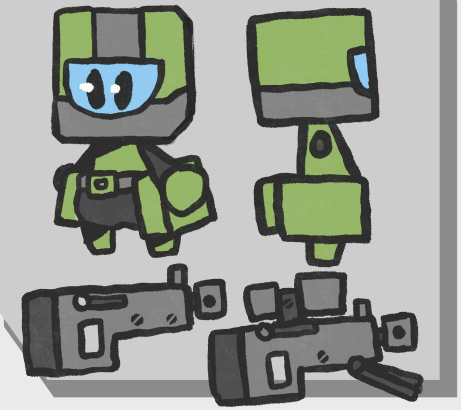
Created by Ethan, Frank & Jacob

CONTENTS OF THIS PRESENTATION

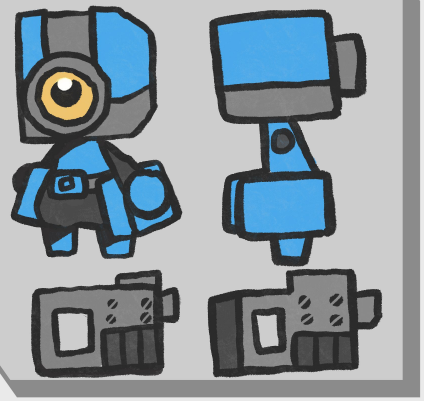
Character Design

Something cute, yet deadly…

Armor: RECON



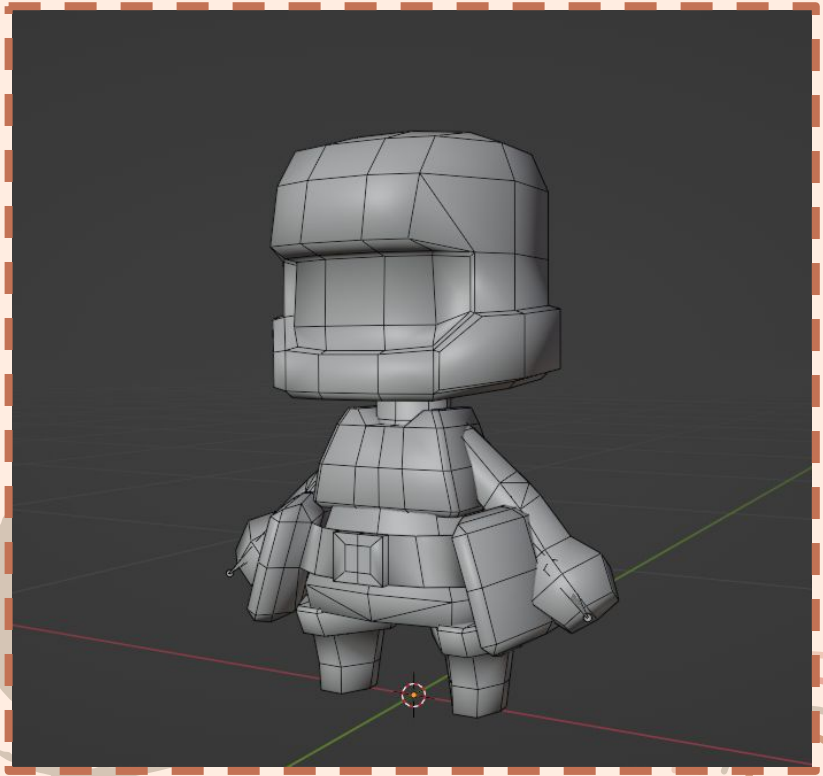
Armor: CQB



Concept Art for Player

Default player appearance

Design Aspects



01

Low Poly

Keeping the player under 1000 vertices

02

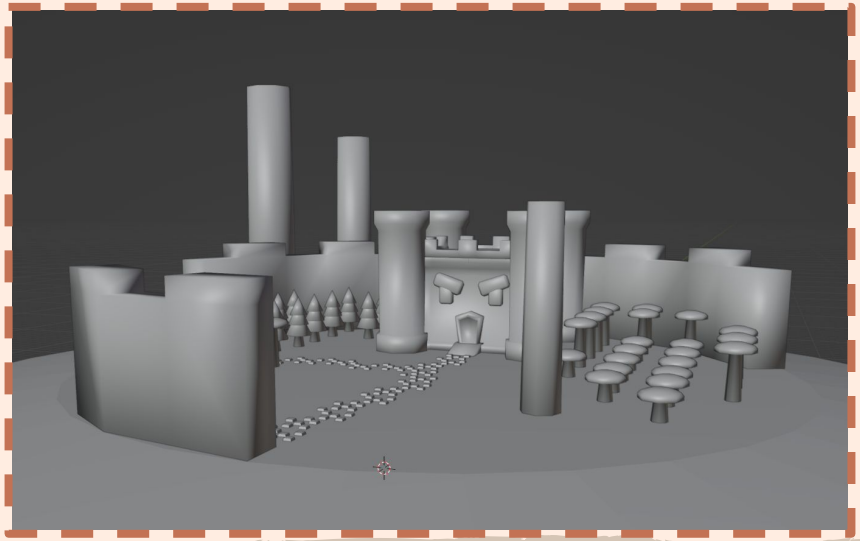
Conclusion

Keeping frame rates higher and able to play with low end hardware.

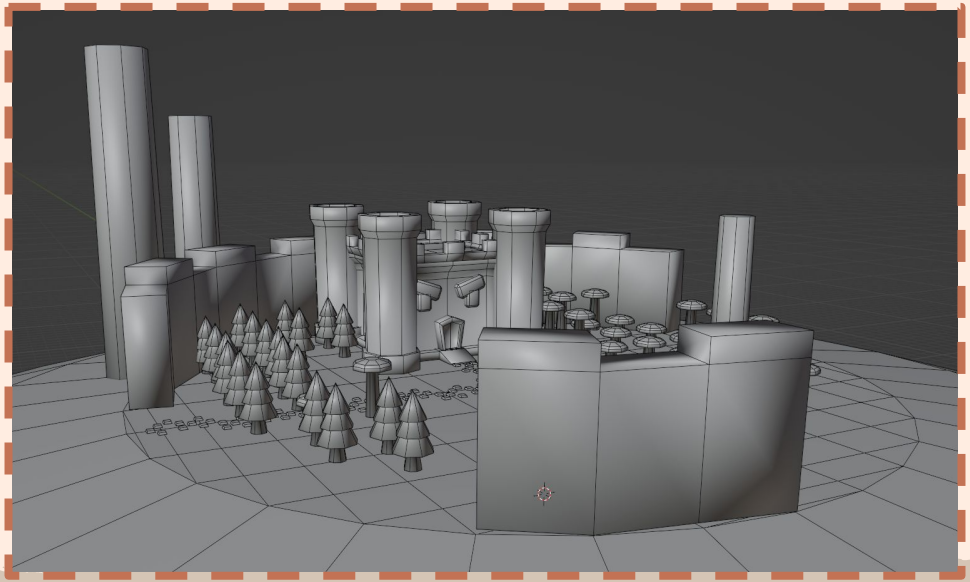
Level Design

Cartoon inspired environments
with a lot of personality.

Concept art of Castle



Material Preview of Castle map

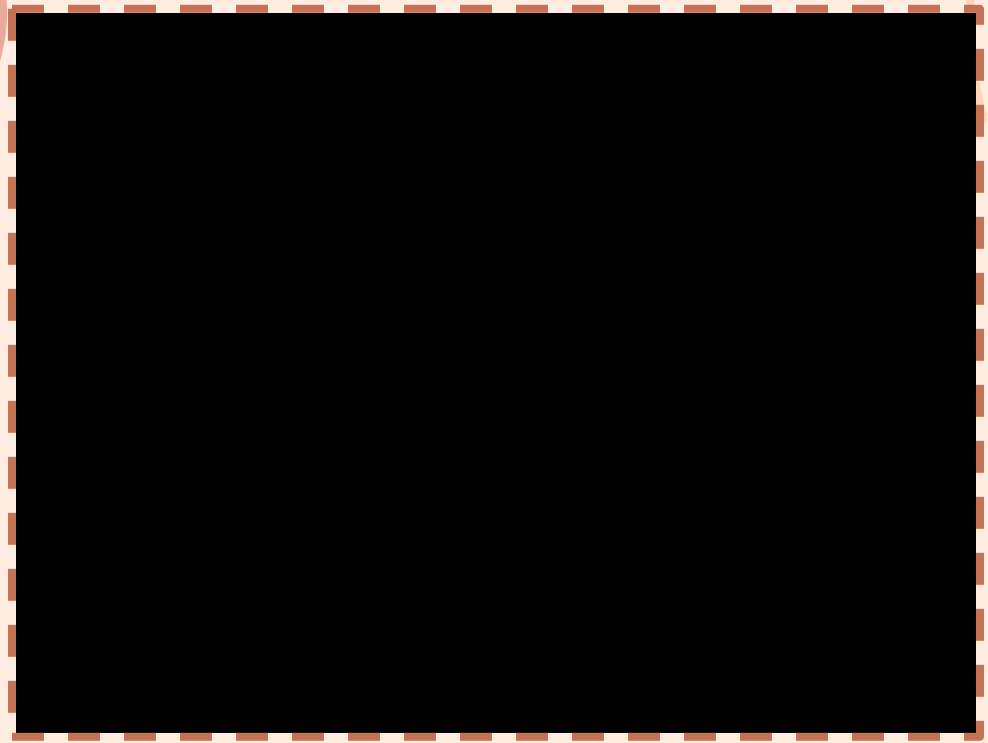


WireFrame Preview of Castle map

Gameplay

Testing walking & jumping, camera controls and Idle

Short Video Demo



Walking & Jumping

```
if (direction.magnitude >= 0.1f)
{
    // Playerboy point in movement direction
    float targetAngle = Mathf.Atan2(direction.x, direction.z) * Mathf.Rad2Deg + playerCamera.transform.eulerAngles.y;
    float angle = Mathf.SmoothDampAngle(transform.eulerAngles.y, targetAngle, ref turnVelocity, turnSmoothing);

    //Responsible for moving character in movement direction
    Vector3 tempMoveDir = Quaternion.Euler(0f, angle, 0f) * Vector3.forward;
    moveDir = tempMoveDir.normalized;
    cont.Move(tempMoveDir.normalized * playerSpeed * Time.deltaTime);
    playerAnimator.SetBool("Walking", true);
    Debug.Log("WALK");


    if (tempMoveDir != Vector3.zero)
    {
        transform.forward = tempMoveDir * Time.deltaTime;
        //transform.position = Vector3.Lerp(transform.position, target.position, Time.deltaTime);

    }
}
//if (direction.magnitude <= 0.01f)
else
{
    playerAnimator.SetBool("Walking", false);
   // Debug.Log("NO WALK");
}
```

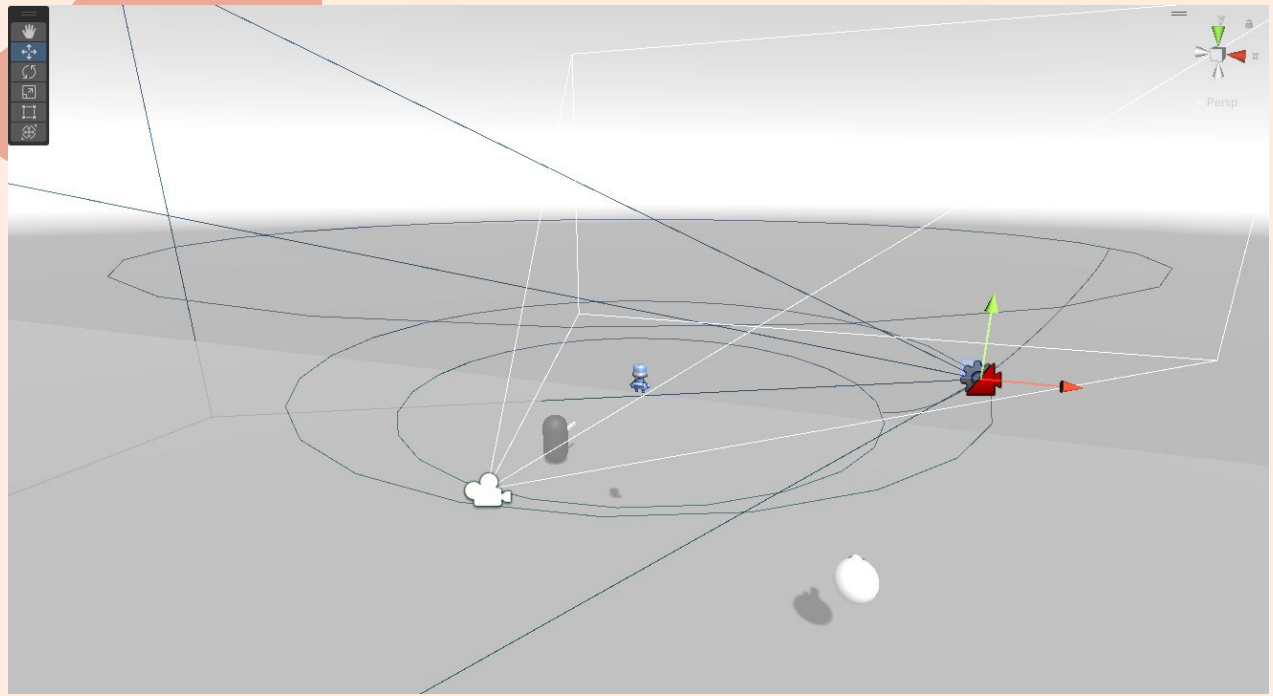
```
grounded = Physics.CheckSphere(groundCheck.position, groundDistance, groundMask);
if (Input.GetButtonUp("Jump")) isJumping = false;
if (velocity.y < 0 && Input.GetButton("Jump") && isJumping == true) isJumping = false;

if (grounded && velocity.y < 0)
{
    jumpCount = 0;
}

if (Input.GetButtonDown("Jump"))
{
    if (jumpCount < 2) { initialVelocity = jumpForce; isJumping = true; } //Will not add a
    if (jumpCount <= 2) jumpCount++; //jumpCount will be allowed to go over the max jump c
      over the max by 1, to prevent overflow.
}

if (grounded && Input.GetButtonDown("Jump") && jumpCount <= 2)
{
    //The first jump.
    playerAnimator.SetBool("Jump", true);
    Debug.Log("Jumping true");
}
else if (!grounded && Input.GetButtonDown("Jump") && jumpCount <= 2) //&& jumpCount < 2)
{
    //Every jump beyond the first jump.
    playerAnimator.SetBool("ReJump", true);
    Debug.Log("JUMMMMMMMMMMMMMMMMMMMP");
}
//Keep separate.
if (grounded && !Input.GetButtonDown("Jump") && !isJumping)
{
    playerAnimator.SetBool("Jump", false);
}
```

Camera Control



Dashing

```
IEnumerator Dash()
{
    float startTime = Time.time;

    canDash = false;

    playerAnimator.SetBool("dashing", false);

    while (Time.time < startTime + dashTime)
    {
        cont.Move(moveDir.normalized * dashSpeed * playerSpeed * Time.deltaTime);

        playerAnimator.SetBool("dashing", true);
        yield return null;
    }
    yield return new WaitForSeconds(dashCoolDown);

    Debug.Log("Dash Cool Down Over");

    canDash = true;

    playerAnimator.SetBool("dashing", false);
}
```

Conclusion