

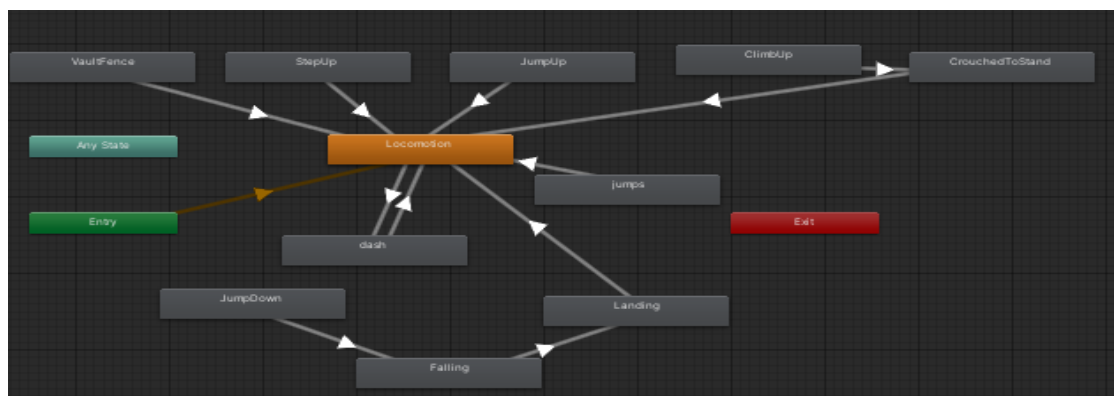
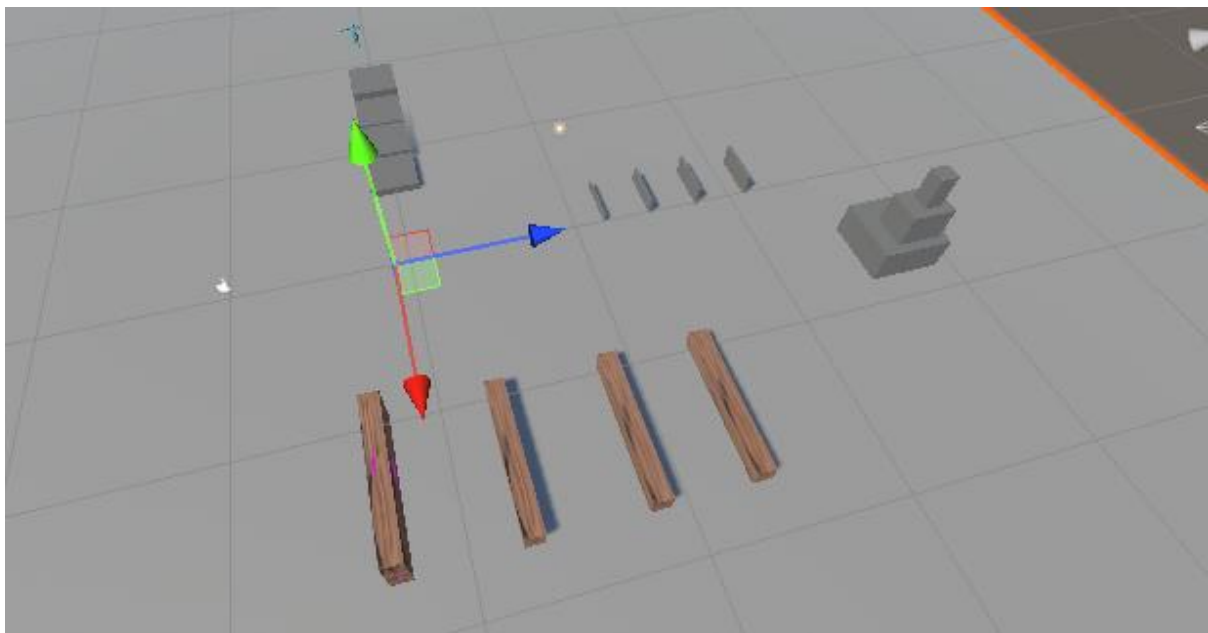
Yuming Su

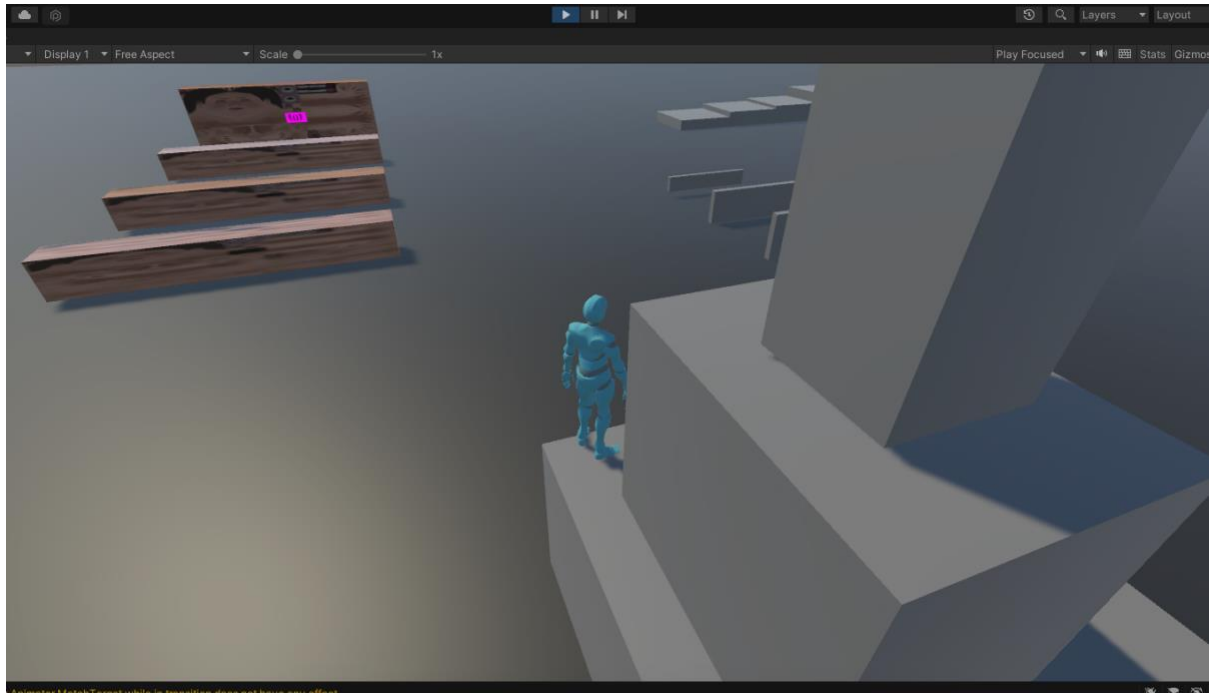
CS512

Assignment2: Parkour

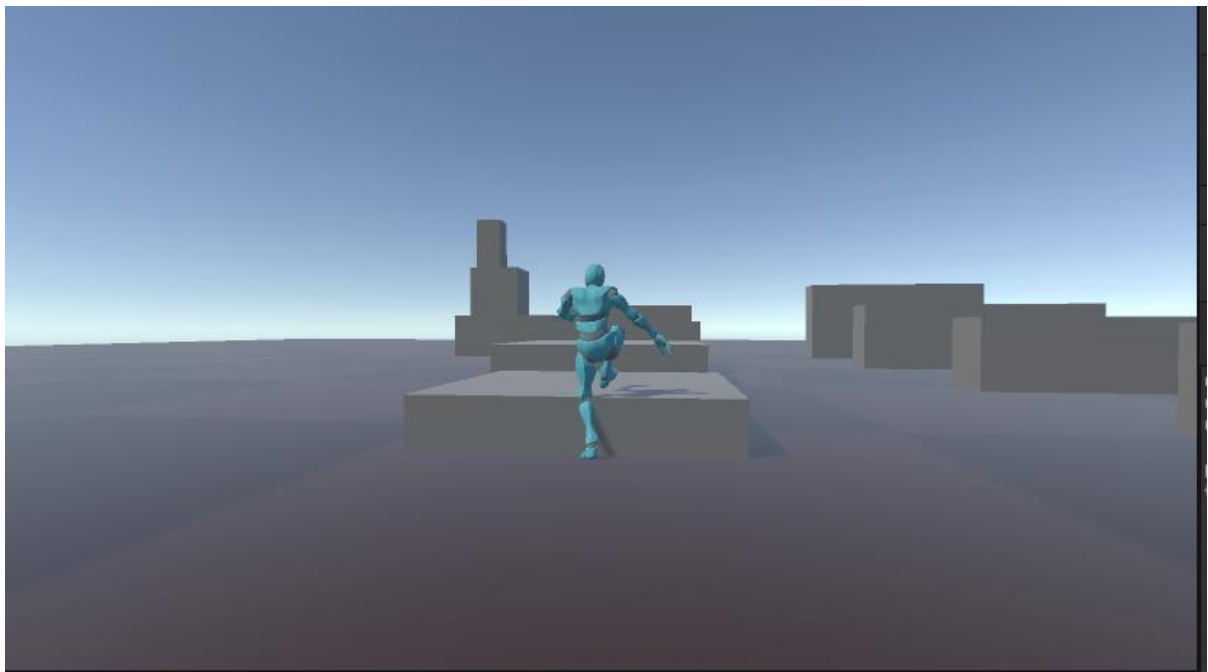
Overview:

This is a 3th person camera parkour game. It relies on keyboard(wsad). I worked on it about 1 week. Press “shift” could run dash. When you close to the obstacle, press “space”(jump), it will automatically show the jump and climb movement. Since I have no Windows machine now. All test made on Mac.

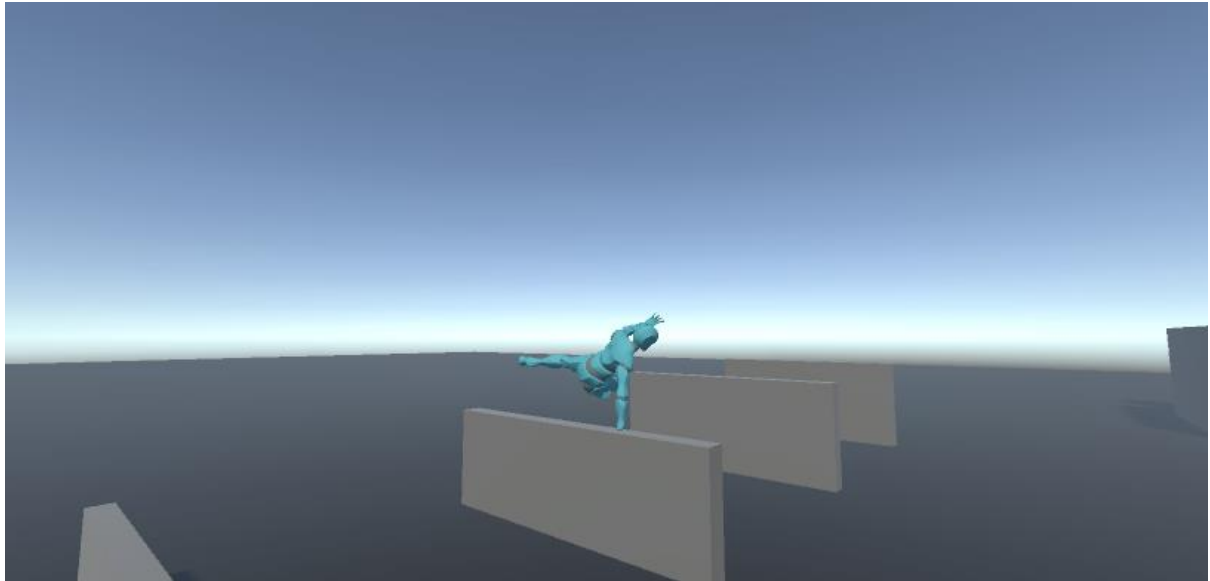




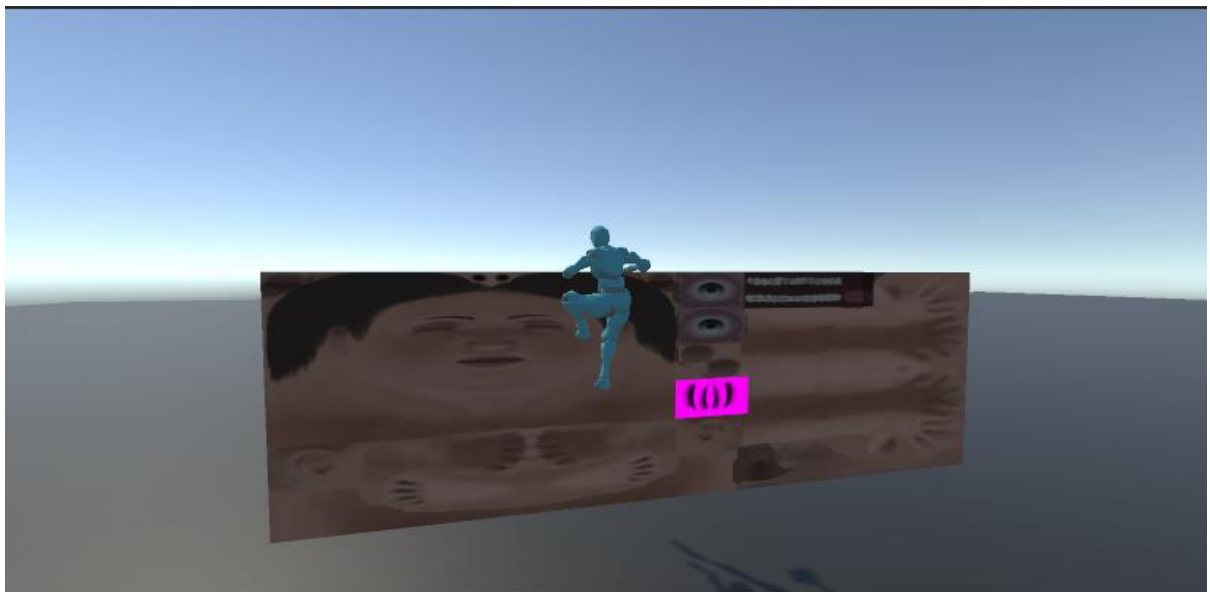
Step on stairs:

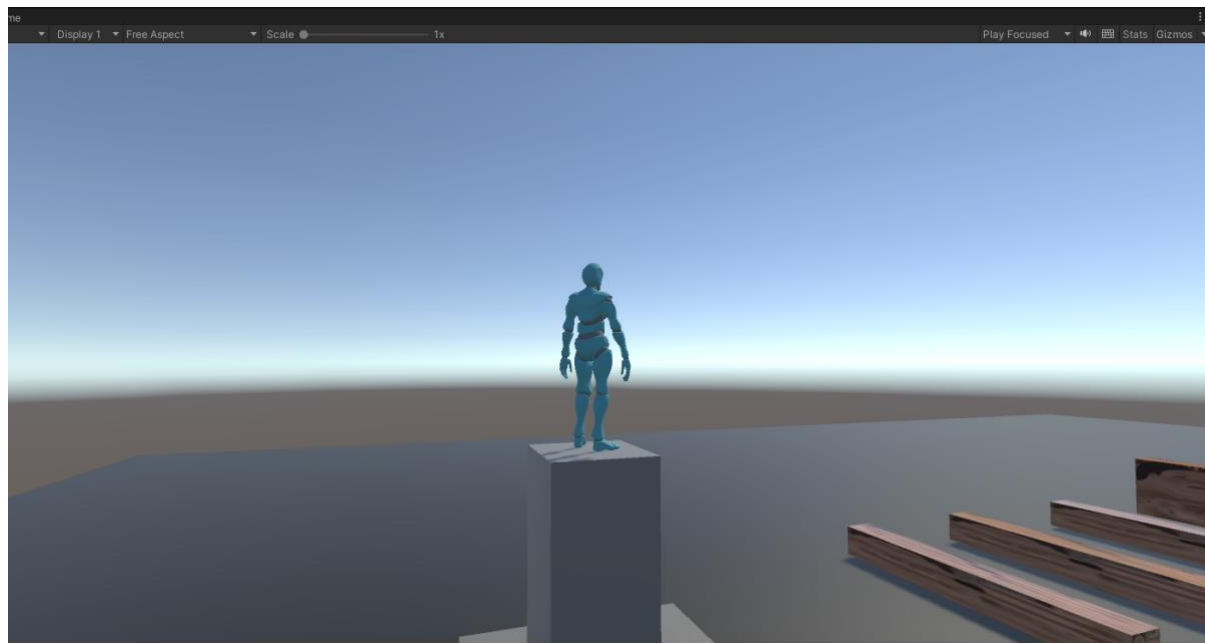
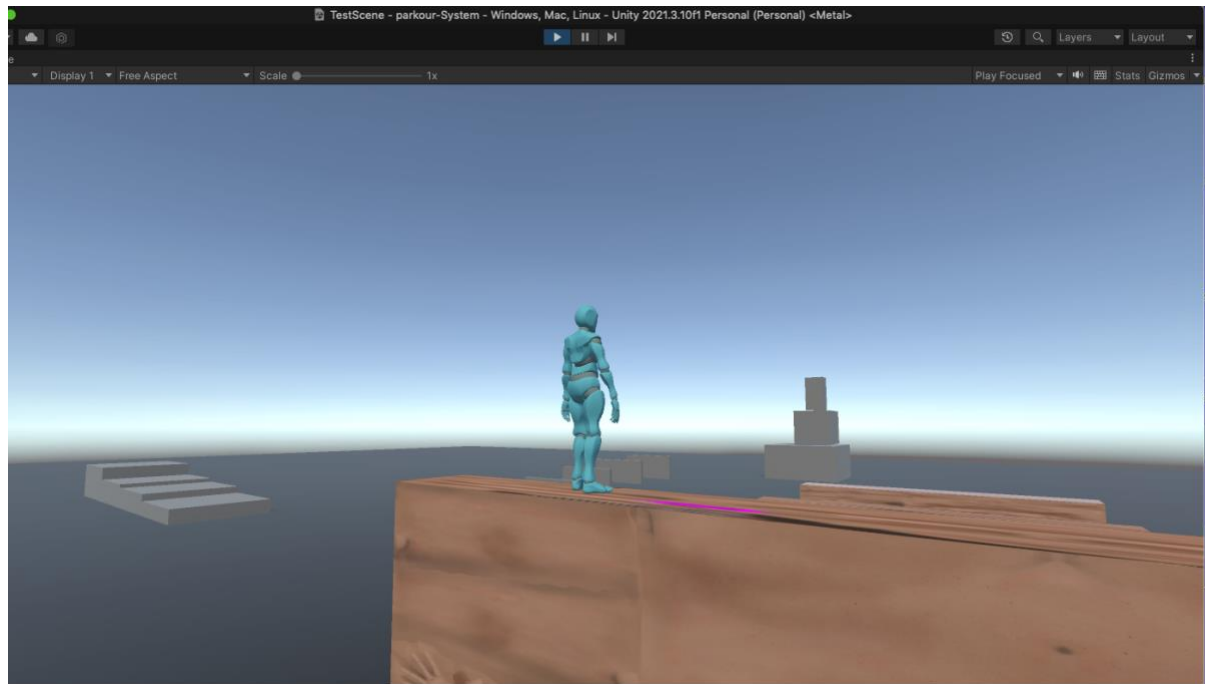


Jump over obstacle (hurdle):



climb on:





Basic code:

```
< > ParkourController.cs PAction.cs NPlayerController1.cs
NPlayerController1 ▶ Update()
21
22 public class NPlayerController1 : MonoBehaviour
23 {
24     [SerializeField] float moveSpeed = 5f;
25     public float rotationSpeed = 500f;
26
27     //GroundCheck()中使用的3个变量
28     [Header("Ground Checking:")]
29     [SerializeField] Vector3 groundCheckOffset; //接地检查
30     [SerializeField] float groundCheckRadius=0.2f;
31     [SerializeField] LayerMask groundLayer;
32
33
34     NCameraControl cameraController;
35
36
37
38     //用于rotation的
39     Quaternion targetRotation;
40
41     //用于控制角色的animator, 在界面中player中的
42     Animator animator;
43
44     //用于Unity player 中的characterController, 用于collusion (collusion通常可以用rigidbody 或者 characterController来实现)
45     CharacterController characterController;
46
47     float ySpeed;
48
49
50     bool isGrounded;
51     bool hasControl = true;
52
53
54     public void Awake()
55     {
56         cameraController= Camera.main.GetComponent<NCameraControl>();
57
58         animator = GetComponent<Animator>();
```

```
velocity.y = ySpeed;

//dash and jump
if (Input.GetButton("Fire3"))
{
    animator.SetBool("dash", true);
    velocity = velocity * 2;
}
else {
    animator.SetBool("dash", false);
}

if (Input.GetButton("Jump"))
{
    animator.SetBool("jump", true);
}
else
{
    animator.SetBool("jump", false);
}

characterController.Move(velocity * Time.deltaTime);

if (moveAmount > 0 && moveDir.magnitude > 0.2f)
{
    targetRotation = Quaternion.LookRotation(moveDir);
}

transform.rotation = Quaternion.RotateTowards(transform.rotation, targetRotation,
rotationSpeed * Time.deltaTime);
}
```

```

public class PAction : ScriptableObject
{
    [SerializeField] string animName;

    [SerializeField] float minHeight;
    [SerializeField] float maxHeight;

    [SerializeField] bool rotateToObstacle;

    [Header("Target Matching")]
    [SerializeField] bool enableTargetMatching = true;
    [SerializeField] AvatarTarget matchBodyPart;
    [SerializeField] float matchStartTime;
    [SerializeField] float matchTargetTime;

    //可以创建一个属性, 无法看到的
    public Quaternion TargetRotation { get; set; }

    //
    public Vector3 MatchPos{get; set;}

    public bool CheckIfPossible(Envoriment.obstacleHitData hitData, Transform player)
    {
        // Height Tag
        float height = hitData.heightHit.point.y - player.position.y;
        if (height < minHeight || height > maxHeight)
            return false;

        if (rotateToObstacle)
            TargetRotation = Quaternion.LookRotation(-hitData.forwardHit.normal);

        if (enableTargetMatching == true)
            MatchPos = hitData.heightHit.point;

        return true;
    }
}

```

```

ParkourController.cs  PAction.cs  NPlayerController1.cs
ParkourController ▶ DoParkorAction(PAction action)
7      [SerializeField] List<PAction> ParkourActions;
8
9      Envoriment envorimentScanner;
10     Animator animator;
11
12     bool inAction;
13
14     NPlayerController1 playerController;
15
16     private void Awake()
17     {
18         envorimentScanner = GetComponent<Envoriment>();
19
20         //获取属性从界面
21         animator = GetComponent<Animator>();
22
23         playerController = GetComponent<NPlayerController1>();
24
25         // playerController.GetComponent<NPlayerController1>();
26     }
27
28     // Update is called once per frame
29     void Update()
30     {
31
32
33         //这种按键都是固定的控制Jump means space
34         if (Input.GetButton("Jump") && !inAction )
35         {
36             var hitData = envorimentScanner.obstacaleCheck();
37             if (hitData.forwardHitFound)
38             {
39                 foreach (var action in ParkourActions)
40                 {
41                     if (action.CheckIfPossible(hitData, transform))
42                     {
43                         StartCoroutine(DoParkorAction(action));

```

I had tried my best combine this with NavMeshAgent, but I failed. Due to some crashed and hardware reasons, my project crashed a few times. I lost a lots of works. I waste much of time. I didn't finish the Point-and click mode. Anyway, it's a very challenging work to me. Thank you to understand. I had used most of time on this assignment recently.