Horse Riding Simulator

Virtual Reality Final Project

Revan Sopher

May 2, 2015

1 Original Proposal

For my final term project I will create a horse-back target shooting game. The player will be riding a horse in a field, using the Falcon in one hand to steer the horse, and either the Hydra (time permitting) or the mouse will be used in the other hand to control a revolver. The Falcon's force feedback will be used to simulate the gallop of the horse, as the display shifts accordingly. This makes the target shooting challenging.

1.1 Modifications

I did not use the Hydra for this project, because I personally find it easier to use a mouse to control looking.

2 Control System

By far the most interesting aspect of this project is the control system. The player controls a modified First Person Controller, with the cylinder collider replaced by a horse model and the button inputs disabled. A pistol model floats to the right as a child of the camera. Thus, what remains is the mouse controlled camera and pistol which rotate independently of the horse.

In the Update() function, the player is advanced by a factor of the current speed. The horse's animation is scaled linearly with the speed, and the camera is rotated up and down slightly in time with the motion. As the gun is attached to the camera, the gun also rotates, making shots inaccurate at high speeds.

```
1 function Update () {
2   transform.Translate(Vector3.forward * speed * Time.deltaTime);
3
4   var anim = transform.GetChild(1).GetComponent.<Animator>();
5   anim.speed = speed / 20.0;
6
7   fpsCamera.transform.Rotate(Vector3.right * frames / 2.0 * Time.deltaTime);
8 }
```

The following function, FixedUpdate(), is responsible for handling the Falcon. The Falcon's servos apply force in a vertical oscillation, the speed of which depends on the player's speed. This creates a vigorous galloping action (the frequency of which is synchronized to the visual shake of the camera).

If the player moves the falcon's knob to the far forward or back, the speed is increased or decreased respectively. There is a one second cool down to these actions registering.

The horse rotates as a linear function of the Falcon's x-axis.

```
1  function FixedUpdate ()
2  {
3    var currentServoPos : Vector3 = plugin.GetServoPos();
4
5    if (frames < -100 || frames > 100) {
6      frame_delta = -1 * frame_delta;
7    }
8    frames += frame_delta * speed;
9
10   plugin.SetServo(Vector3.up * frames / 30.0);
11
12   if (currentServoPos.z < -1.4) {
13     if (Time.time > cooldown)
14       if (speed > 10){
15         speed -= 6;
16         cooldown = Time.time + 1;
17         soundWhinny.Play();
18       } else {
19       frames = 0;
20         if (speed != 0)
21           soundWhinny.Play();
22         speed = 0;
23         transform.GetChild(1).audio.Stop();
24       }
25
26   } else if (currentServoPos.z> 1.4) {
27     if (Time.time > cooldown) {
28       if (speed == 0){
29         transform.GetChild(1).audio.Play();
30         speed += 3;
31       }
32       speed += 3;
33       soundWhip.Play();
34     }
35     cooldown = Time.time + 1;
36   }
37
38   if (speed > 0)
39     transform.Rotate(Vector3.up * Time.deltaTime * (currentServoPos.x / 1.9) *
         100);
40 }
```

3 GUI

In order to make the game challenging, the player has a limited time to destroy all the targets. Progress is displayed in the top left corner, as is the timer.

The game also has an animated main menu.

4 Assets

4.1 Unity Store

The horse and gun models are from the Unity asset store.

4.2 Particle Effects

Firing bullets from the pistol produces a smoke trail particle effect.

The horse produces a dust cloud particle effect when running.

The crates produce a dark smoke particle effect when destroyed.

4.3 Sounds

The horse plays a galloping sound when moving, a whinnying sound when the reins are pulled, and a whipping sound when whipped.

The gun produces a loud gunshot sound when fired.

The crates produce a splintering sound when destroyed.

Figure 1

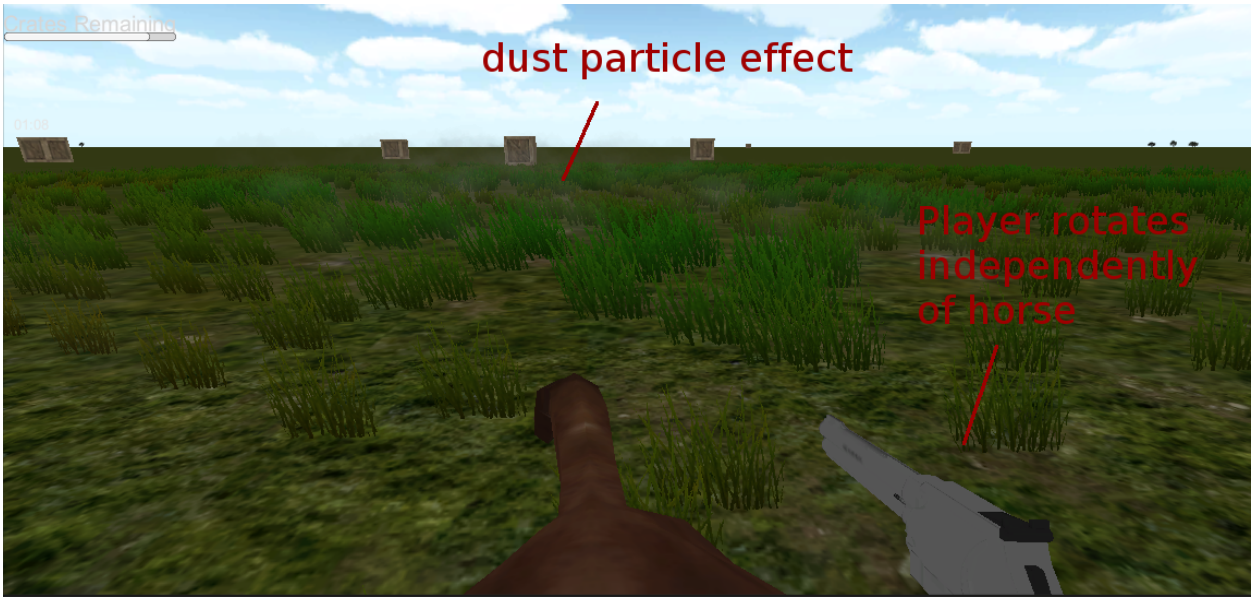

Figure 2

Figure 3

Figure 4