



Abstract

The rapid advancement in virtual reality (VR) technology has empowered developers to craft highly immersive experiences that captivate users' imaginations and deliver engaging, interactive content. In the "VR DINO" project we developed a VR parkour game, inspired by the popular Chrome "DINO" game. Players control a cute animated dinosaur to avoid crashing into its animal friends. Utilizing VR technology, this game seeks to enhance the player's experience through immersive interaction.

1. Game Description

First-person VR games can easily lead to motion sickness, especially when the character in the scene moves while the actual player does not. To mitigate this, we designed our game so that the character remains stationary, with obstacles moving towards the character for higher stability. In our game, the player controls a dinosaur standing on a brick path through the fields. Various animal friends, such as birds, horses, and cats, approach the dino. The player uses the "up, down, left, and right" keyboard keys to make the dino jump, crouch, slide sideways, or dive quickly to avoid crashing into the animals. Each time the dino successfully avoids a line of animals, one point is added. If the dino collides with an animal, the animal will scatter into pieces and the game round ends. The objective is to achieve the highest score possible.

2. Artistic Design

In this section, we will introduce our design for shapes, appearances, individual animations, and environment.

2.1. Dino Body Design

Since only the dino's body will be visible from the player's view when they rotate their heads, we decided to create a body without a head. The dino body's mesh and

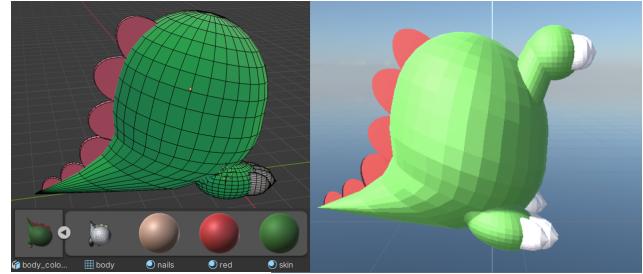


Figure 1. Dino Body Design



Figure 2. Animal Friends Design

skin materials are built in Blender and imported into Unity, as shown in Figure 1.

2.2. Animal Friends Design

The animal friends consist of three types: birds, horses, and cats. Birds and cats each occupy a 1×1 grid, while horses occupy a 2×1 grid from the front view. The materials for each animal type are defined individually, and the mesh and animations for quadruped walking and running are imported from Blender Rigify Animbox [3] animation scripts and the Unity Living Birds Asset [1] package. The integrated effect is shown in Figure 2.

2.3. Ending Design

When the dino collides with any animal, the game ends. A scattering glitter effect appears for each animal friend,

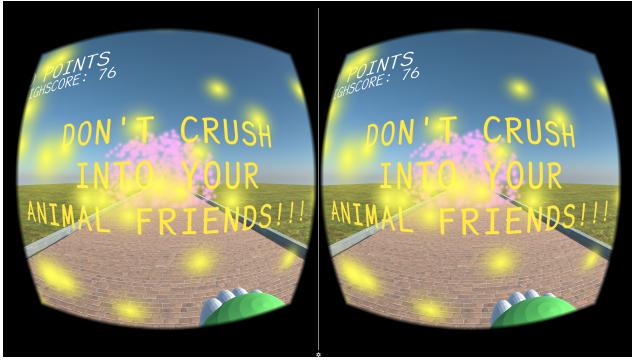


Figure 3. Ending and Scoreboard Design

and they disappear. The text “DON’T CRASH INTO YOUR ANIMAL FRIENDS!!!” spans across the canvas as shown in Figure 3. To restart the game, the player presses the spacebar.

2.4. Scoreboard Design

The scoreboard is located in the top left corner of the screen and displays the current and highest scores as shown in Figure 3.

2.5. Environment Design

The environment is constructed from a plane of grass, a plane of a brick pathway, and two rows of curbs on both sides of the path. These elements are created by mapping texture images by Quixel Megascans [2] onto planes and meshes.

3. Logic Design

In this section, we will introduce our design for the logistics of the game.

3.1. Animal Friends Spawn

We studied the movement mechanics of the original Chrome DINO game and discovered that the obstacles come in three different heights, with the DINO’s jump being approximately four times the height of the smallest obstacles. Since our game is in 3D, we have added an additional axis of movement, allowing the DINO to move left or right. Combining these features, we designed our VR DINO game with a 3x3 grid, as shown on the white plane with grid lines in Figure 4.

For clarity in referring to specific locations on the grid, we define row 1 as the bottom row and column 1 as the leftmost column. In our game, birds, horses, and cats are selected randomly to spawn at different times. Birds can appear in either row 2 or row 3. Horses and cats, however,

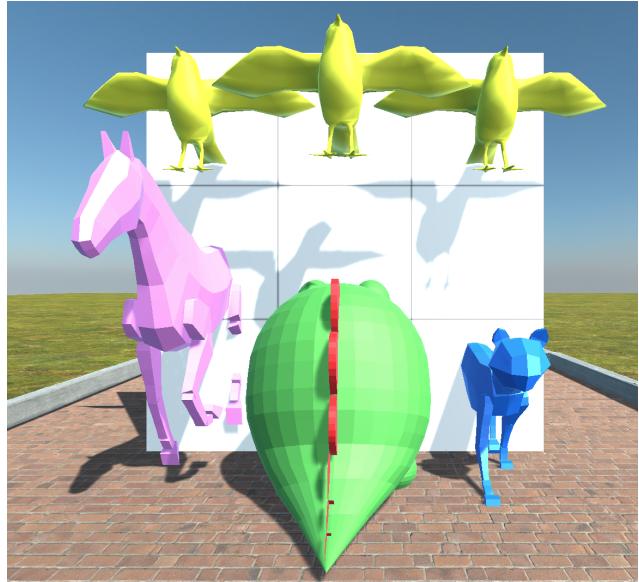


Figure 4. Animal Friends Control

are only spawned in row 1, and can appear in any number of columns, allowing them to form groups of two or three.

3.2. Dino Control

To enhance gameplay, we implemented several controls:

1. **Slide Left or Right:** We tried various control schemes for moving left and right. Initially, we allowed small incremental movements with each key press, but this was unintuitive given our design of three distinct columns. We adopted a three-lane movement (Left, Middle, Right), where pressing the left or right arrow key switches the DINO to the adjacent lane. The movement is smooth, starting quickly and slowing as it reaches the target, using linear interpolation. The DINO can move while performing other actions like jumping, crouching, or sliding, ensuring a seamless experience.
2. **Crouching:** We added a crouching mechanic to avoid higher animal friends. Pressing the down arrow key lowers the DINO’s body, allowing it to pass under animal friends. The transition into and out of crouching is smooth, with a quick drop followed by slower stabilization, using linear interpolation. The DINO can move left or right while crouching, integrating seamlessly with other actions.
3. **Jumping and Quick Fall:** Pressing the up arrow key triggers a jump, allowing the DINO to leap over obstacles. The down arrow key in mid-air activates a quick fall, letting the DINO descend rapidly, useful for dodging.

ing obstacles after a jump. Both actions are smooth, with natural acceleration and deceleration.

3.3. Difficulty Control and Score Tracking

We implemented difficulty control by gradually increasing the speed of the animal friends over time and increasing their spawn rate to match this speed. The score increases by 1 each time the DINO passes a line of animals. The highest score is stored at the project level, so it is retained even after restarting the app.

3.4. Sound Effect

We implemented sound effects to enhance the gaming experience. Sounds play for actions such as jumping, landing after a jump, crouching, quick fall, and game over. To add variety, we randomly select from several sound options for each action, ensuring a dynamic and engaging auditory experience.

References

- [1] Living birds | | unity asset store.
- [2] Quixel megascans - the world's largest scanned 3d asset library.
- [3] Rigify animbox | animation - coding / released scripts and themes. Section: Coding.