Topic 3 Mid-Term Assignment 3D Graphics and Animation Report

Description: Interactive Physics Scene in Unity

Task 1: Physics-Based Object

Computer Graphics and Animation Techniques Used

The penguin acts under physics using Unity's Rigidbody component, which allows simulation of mass, drag, and gravity. The terrain, composed of rocks, trees, and bushes, forms the alien environment, created using Unity's terrain tools and mesh assets.

Mathematically, Unity's Rigidbody integrates the principles of Newtonian mechanics:

1) **Force:** Applied to the penguin for movement, governed by:
   - Force = Mass x Acceleration

2) **Collision Detection**: Implemented using Unity's collider system, which resolves object interactions through physical bounding volumes.

Explanation of Use

The Rigidbody was chosen for its native support in Unity, enabling seamless application of realistic forces and gravity. The penguin moves naturally on the terrain, showcasing rolling and sliding mechanics when interacting with objects like rocks. Aesthetic choices, for example, the green hat and alien landscape. Enhance immersion while testing the physics interactions.

Evaluation of Methods

The Rigidbody's integration simplifies physics management, but alternatives like custom physics scripts could allow more control over behaviour. However, Unity's native physics engine strikes an ideal balance between realism and efficiency for this project's scale. The terrain assets effectively support the alien planet's aesthetic but could be further optimized with procedural generation.

Task 2: Control Scheme

Computer Graphics and Animation Techniques Used

A control scheme allows the user to move the penguin using keyboard input. The implementation relies on Unity's input system combined with Rigidbody's, "AddForce( )" and "AddTorque( )" methods to apply directional and rotational forces. The camera follows the penguin using Unity's system, adjusting based on mouse focus.

Mathematical Principles

1) **Force Application**: Movement directions correspond to force vectors relative to the penguin's orientation.

2) **Rotation**: Torque applied around the y-axis modifies the penguin's angular velocity.

3) **Camera Movement**: Achieved using transformations and angles to match the penguin's direction and mouse input.

Explanation of Use

Forces provide realistic movement while avoiding the limitations of direct position changes. The camera improves player immersion by dynamically adjusting its position and focus.

Evaluation of Methods

Unity's physics-based controls allow smooth, natural navigation. Alternative techniques like character controllers lack the physics realism required for this project. Future improvements could include fine-tuning force application for more nuanced control and exploring smoother camera transitions.

Extensions : Hinge Joints for Windmills

Windmill rotors use hinge joints, a physics component in Unity that constrains motion to a single axis. The rotors rotate around the joint's axis based on applied torque, simulating wind-powered movement.

Evaluation: The hinge joint was chosen for its simplicity and effectiveness in creating realistic rotational movement. Further refinement could include dynamic wind zones affecting rotation speed.

Gravitational Pull for Planet: A custom gravitational force affects objects on the planet.

Gravitational pull in Unity by calculating a direction vector toward a specified centre of gravity. It applies a force proportional to the object's mass and a tuneable gravity strength using Rigidbody, "AddForce( )". This enables realistic motion as objects are pulled toward the centre, replicating celestial gravity. It's ideal for creating dynamic, physics-based environments like planetary orbits.

Evaluation: This mechanic adds depth to the scene, but fine-tuning is required to balance realism and gameplay. Objects occasionally behave unpredictably under extreme gravitational forces.

Particle System for Waterfall: The waterfall is created using Unity's particle system, simulating water droplets with attributes like velocity, lifetime, and gravity influence.

Evaluation: The particle system is visually effective but lacks interaction with other objects. Integrating splashes or mist effects could enhance realism.

Conclusion: This project effectively demonstrates Unity's physics capabilities and animation techniques. While the implementation meets and exceeds requirements, further refinement in interactivity and control precision could elevate the user experience.

Credits: Polygonal Alien Cave Nature Asset Package - Aligned Games

https://assetstore.unity.com/packages/3d/environments/sci-fi/polygonal-alien-cave-nature-asset-package-299156

Wikipedia – Outer Space

https://en.wikipedia.org/wiki/Outer_space