南通大學人工智能与计算机学院
School of Artificial Intelligence and Computer Science, Nantong University
（阿里云大数据学院）
（人工智能研究院合署）

UI 设计基础

Introduction to UI Design Course

REPORT ON

The Basic Principles of Collision Detection and Simulation with Applications in FPS Gaming

Supervisor

Dr. Gao Zhan

Professor

School of Artificial Intelligence and Computer Science

Nantong University

Submitted By

Vaskar Chakma

2130130204

2024 年 12 月 07 日

Introduction

Collision detection is a critical component in computer graphics, gaming, and simulation, enabling virtual objects to interact realistically within a digital environment. It ensures that characters don't walk through walls, objects respond naturally to forces, and interactions like hits and impacts are accurately registered. Beyond gaming, collision detection is widely applied in robotics, virtual reality, and even autonomous vehicle systems.

In gaming, particularly in First-Person Shooter (FPS) titles, collision detection plays an indispensable role. It allows for seamless navigation, realistic combat interactions, and dynamic environmental responses, creating an immersive and responsive player experience. Understanding the principles of collision detection is vital for designing games that feel grounded and engaging.

Basic Principles of Collision Detection

Collision detection operates on a systematic process to identify and respond to interactions between objects in a virtual space. It is broadly categorized into three main phases:

Broad Phase

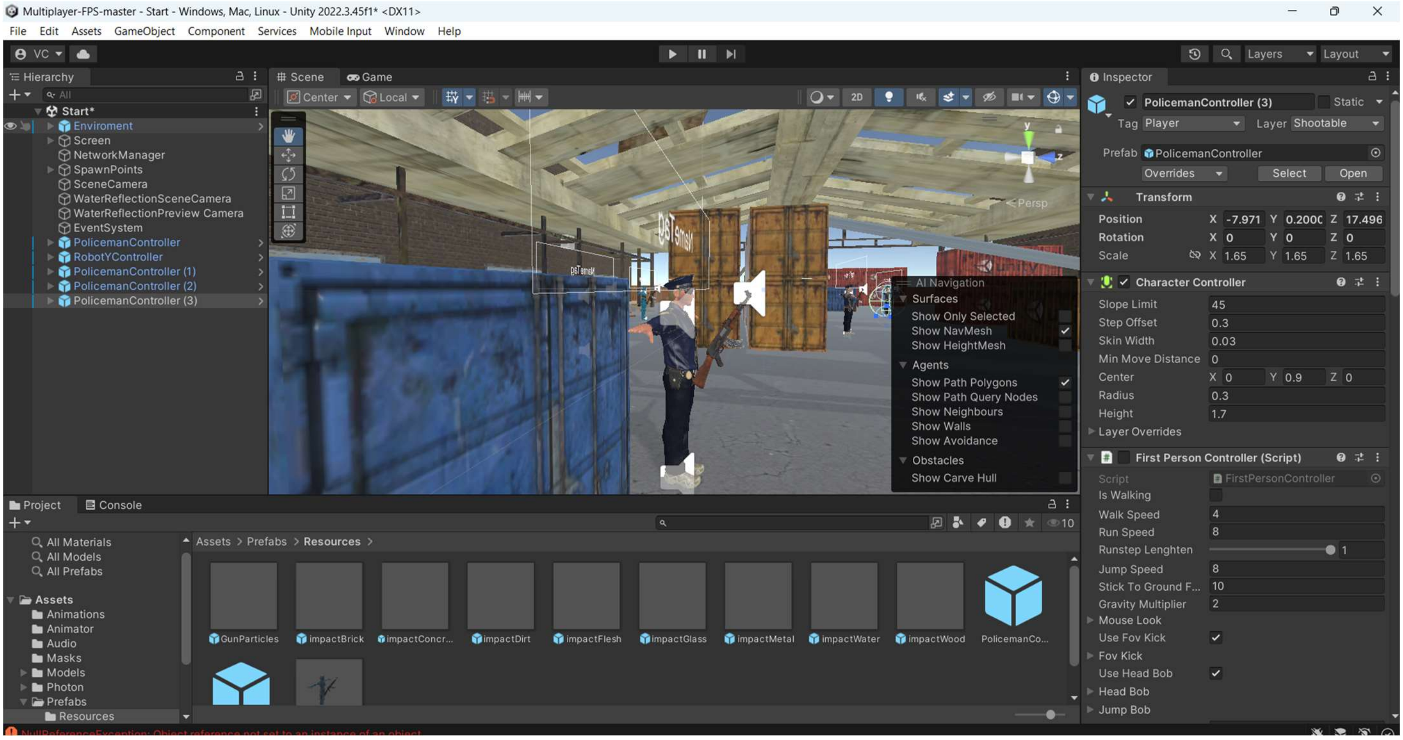
The broad phase is the initial step, designed to efficiently filter out pairs of objects that are too far apart to collide. This step prevents unnecessary computational checks and ensures the system focuses on likely collisions.

Bounding Volumes: Simple geometric shapes such as spheres, axis-aligned bounding boxes (AABB), or oriented bounding boxes (OBB) are used to approximate objects. These shapes are computationally inexpensive to test for overlaps.

Spatial Partitioning: Techniques like grids, quadtrees, or octrees divide the virtual space into smaller regions, narrowing down the objects that need to be considered in detail.

Narrow Phase

After the broad phase identifies potential collision candidates, the narrow phase performs detailed checks to confirm whether and where collisions occur.



☑ PolicemanController (3) □ Static

Tag Player Layer Shootable

Prefab PolicemanController

Overrides Select Open

Transform

Position X -7.971 Y 0.200C Z 17.496

Rotation X 0 Y 0 Z 0

Scale X 1.65 Y 1.65 Z 1.65

☑ Character Controller

Slope Limit 45

Step Offset 0.3

Skin Width 0.03

Min Move Distance 0

Center X 0 Y 0.9 Z 0

Radius 0.3

Height 1.7

Layer Overrides

□ First Person Controller (Script)

Script FirstPersonController

Is Walking □

Walk Speed 4

Run Speed 8

Runstep Lenghten 1

Jump Speed 8

Stick To Ground F... 10

Gravity Multiplier 2

Mouse Look

Use Fov Kick ☑

Fov Kick

Use Head Bob ☑

Head Bob

Jump Bob

NullReferenceException: Object reference not set to an instance of an object.

Intersection Tests: Mathematical formulas are applied to determine precise overlap points. For example, line-line intersection or polygonal mesh tests can be used for complex shapes.

Separating Axis Theorem (SAT): This algorithm is commonly used to test collisions between convex shapes by checking for gaps along potential separating axes.

Collision Response

Detecting a collision is only part of the process. The system must also resolve how objects react to the collision. This involves:

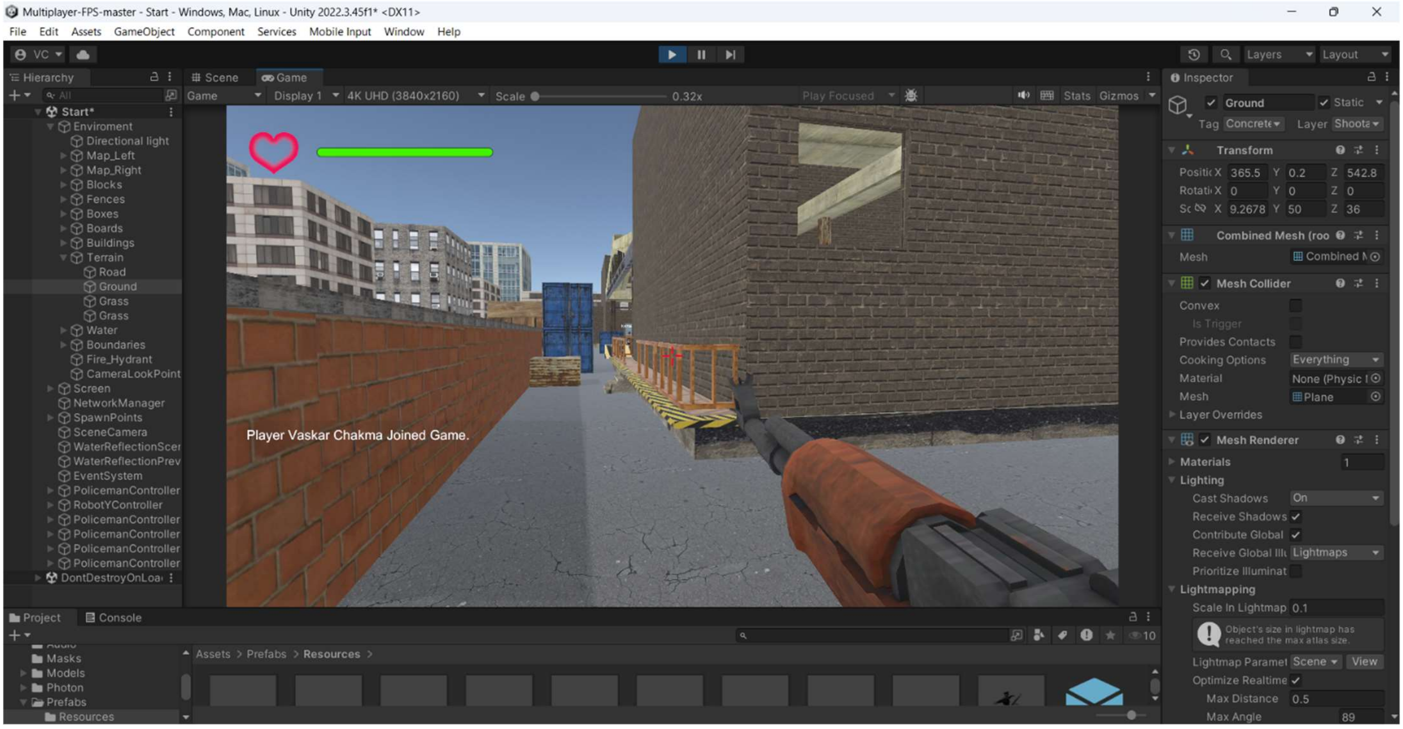
Penetration Resolution: Adjusting object positions to resolve overlaps.

Physical Reactions: Simulating effects such as bouncing, friction, or deformation, depending on the nature of the objects involved.

Collision Detection in FPS Games

Collision detection plays a crucial role in FPS games, ensuring realistic gameplay and player immersion. Key areas of application include:

1. **Player and Environment Interaction**
   Collision detection prevents players from walking through walls, falling through floors, or climbing impassable terrain. Most FPS games use a capsule-shaped bounding volume for player models to optimize collision checks while maintaining realism.
2. **Projectile Impacts**
   - **Raycasting:** This technique is used to simulate bullets and lasers. A virtual ray is cast from the gun barrel to check if it intersects any object, ensuring accurate hit detection.
   - **Hitboxes:** Simplified shapes around characters or objects are used to register projectile hits efficiently.
3. **Dynamic Objects and Physics**
   Interactive elements like destructible walls, movable crates, or bouncing grenades rely on collision detection to behave realistically in the game world.
4. **AI Navigation**
   Collision detection is essential for non-player characters (NPCs) to navigate the game world. It helps them avoid obstacles, take cover, and interact with the player strategically.



Multiplayer-FPS-master - Start - Windows, Mac, Linux - Unity 2022.3.45f1* <DX11>

Game | Display 1 | 4K UHD (3840x2160) | Scale 0.32x

Player Vaskar Chakma Joined Game.

Ground ✓ Static
Tag Concrete Layer Shoota

Transform
Position X 365.5 Y 0.2 Z 542.8
Rotation X 0 Y 0 Z 0
Scale X 9.2678 Y 50 Z 36

Combined Mesh (roo
Mesh: Combined M

✓ Mesh Collider
Convex □
Is Trigger □
Provides Contacts □
Cooking Options: Everything
Material: None (Physic M
Mesh: Plane
Layer Overrides

✓ Mesh Renderer
Materials 1
Lighting
Cast Shadows: On
Receive Shadows ✓
Contribute Global ✓
Receive Global Illu: Lightmaps
Prioritize Illuminat □
Lightmapping
Scale In Lightmap 0.1
Object's size in lightmap has reached the max atlas size.
Lightmap Paramet: Scene | View
Optimize Realtime ✓
Max Distance 0.5
Max Angle 89

Collision Detection Methods in FPS Games

1. **Bounding Boxes and Spheres**
   These simple shapes are efficient for broad-phase detection and are commonly used for static objects like walls and obstacles.
2. **Raycasting**
   Fast and efficient, raycasting is ideal for detecting bullet impacts and line-of-sight checks.
3. **Continuous Collision Detection (CCD)**
   Ensures fast-moving objects, such as bullets, don't "tunnel" through thin barriers by checking for collisions between their start and end points in each frame.
4. **Mesh-Based Detection**
   Used for highly detailed interactions, where collision detection must be precise. However, due to its computational intensity, this method is typically reserved for critical interactions.

Challenges in Collision Detection

Implementing collision detection in FPS games comes with challenges:

- **Performance Optimization:** Detailed collision checks can strain computing resources, especially in large, dynamic game environments.
- **Precision vs. Speed:** Developers must balance the accuracy of collision detection with the need for real-time performance.

Conclusion

Collision detection is an essential component in computer graphics, simulations, and gaming, forming the backbone of realistic and interactive digital environments. Its systematic process—from identifying potential collisions to resolving interactions—ensures that objects in virtual spaces behave according to physical rules, enhancing both functionality and immersion.

In FPS gaming, collision detection plays a critical role in various aspects, including character movement, projectile accuracy, dynamic interactions, and AI navigation. Techniques such as bounding volumes, raycasting, and spatial partitioning strike a balance between performance and precision, allowing games to handle complex environments efficiently while maintaining realism. As gaming technology evolves, collision detection will continue to advance, paving the way for more immersive, dynamic, and responsive experiences. Its integration ensures that virtual worlds feel authentic, creating an engaging environment for users across industries.