

Documento Explicativo del Proyecto

Maria Eduarda Beckers

1. Instrucciones de ejecución y controles

Objetivo del juego: Mover la partícula jugador y experimentar con fuerzas, vientos y explosiones. Al alcanzar altura 1600 se pasa a la escena de victoria (WinScene) con un sistema de fuegos artificiales.

Teclas del juego:

Tecla Acción
P / p (StartScene) Transición a la GameScene.
A / a (GameScene) Aplicar fuerza lateral izquierda al jugador.
D / d (GameScene) Aplicar fuerza lateral derecha al jugador.
W / w (GameScene) Activar/desactivar viento 1.
E / e (GameScene) Activar/desactivar viento 2.
R / r (GameScene) Activar/desactivar viento 3.
T / t (GameScene) Activar/desactivar viento 4.
Y / y (GameScene) Crear explosión al lado izquierdo del jugador (Firework).
P / p (WinScene) Reinicia el juego, vuelve a GameScene.
P / p (Scene1) Crea Particulas normales con solo gravedad.
P / p (Scene2) Crea Bala ligera.
B / b (Scene2) Crea Bala pesada.
P / p (Scene3) Crea Sistema de partículas tipo lluvia.
P / p (Scene4) Crea Sistema de partículas tipo lluvia con torbellino.
P / p (Scene5) Crea Sistema de partículas tipo lluvia con explosion.
1 Transición a la Scene1.
2 Transición a la Scene2.
3 Transición a la Scene3.
4 Transición a la Scene4.
5 Transición a la Scene5.
G / g Transición a la StartScene.

2. Requisitos de programación y localización en código

A) Clase Particle

- 1) **Método de integración** : Particle::integrate(double t) en Particle.cpp (líneas 48-83 aprox.).
- 2) **Masa y varias partículas** : Particle::_mass en Particle.h y definición de partículas con distintos pesos en cada escena.
- 3) **Gestión de fuerzas y aceleración** : _force_registry->updateForces() en ParticleSystem::update() (ParticleSystem.cpp línea 21) o en las propias escenas, considerando masa en Particle::integrate() .

B) Clase Projectile

- **Clase:** Bullet
- **Método de escalado de velocidad y fuerza:**
 - Se pasa la velocidad real (velReal) y la velocidad de simulación (velSim) al constructor de Bullet.
 - Esto permite ajustar la velocidad, energía y fuerzas de la bala según su tipo.
- **Tipos de proyectiles implementados:**
 - **Bala ligera** : masa 0.008 kg, tecla 'p' (Scene2::keyPressed)
 - Código: Scene2.cpp líneas 75-80
 - **Bala de cañón** : masa 10.0 kg, tecla 'b' (Scene2::keyPressed)
 - Código: Scene2.cpp líneas 81-86
- **Fuerzas aplicadas:**
 - La gravedad se aplica a todos los proyectiles mediante _force_registry->add(bull, _gravity)
 - Código: Scene2.cpp línea 90
- **Notas de funcionamiento:**
 - La dirección se obtiene de la cámara (cam->getDir()) y se normaliza.
 - Cada proyectil se integra en update() usando integrate(t) con la gravedad aplicada.

C) Clase ParticleSystem

- Definición y gestión de conjuntos de partículas : ParticleSystem.cpp líneas 12-154.

- Se encarga de actualizar partículas (`ParticleSystem::update()`), registrar fuerzas (`registerAllForces()`), y manejar generadores y partículas (`addGenerator()` , `createParticles()`).

D) Generadores de Partículas

- 1) **Partícula modelo** : atributo `prop` en `FireworkGen/FogGen/RainGen` (`FireworkGen.cpp` líneas 15-23). No he usado una Partícula modelo para hacer copia, si no que un struct de información.
- 2) **Posición variable con distribuciones probabilísticas:** `_u` y `std::uniform_real_distribution` en `FireworkGen.cpp` línea 33, en `FogGen.cpp` línea 23 y `RainGen.cpp` línea 23.
- 3) Activación/desactivación desde `ParticleSystem` `ParticleSystem::createParticles()` (`ParticleSystem.cpp` líneas 39-65).

E) Generadores de Fuerza

- 1) **Cálculo independiente por partícula:** `Particle::addForce()` es llamado por el propio sistema de fuerza despues de que se actualicen `ForceGenerator::calculateForce()` (`ExplosionsForce.cpp`, `WindForceM.cpp`, `WindForce.cpp`, `TorbellinoForce.cpp`).
- 2) **Múltiples generadores por partícula:** `_force_registry->add(p, generator)` en `ParticleSystem.cpp`.
- 3) **Registro en ParticleSystem:** `ForceRegistry` en `ParticleSystem::_force_registry` (`ParticleSystem.cpp`) y también están referenciados en la propias escenas.
- 4) **Tipos de fuerza:**
 - Variando con la posición: `WindForceM` (`WindForceM.cpp` líneas 15-50).
 - Variando con el tiempo: `ExplosionsForce` (`ExplosionsForce.cpp` líneas 10-50).