

Warmup 2

Due: Feb. 28, 2021

Introduction

Warmup 2 will add world organization functionality to your game engine, as well as a basic collision system and a third-person camera. By the end of the week, you'll have your first fully playable game!

Design Check

- How will your camera support both first and third person?
- Describe the steps to determine if, and by how much, two cylinders are colliding.
- What are the steps involved with determining a collision? Which are engine-side, and which are game-side?
- What kind of gameplay will you implement?

Basic Requirements

- Code never crashes
- Code runs on your machine at 20+ FPS
- All game logic is contained in a GameWorld class; Screens have no game logic
- There is some non-trivial game logic (e.g., jumping, colliding with ground, etc.)

Engine Requirements

Like Warmup1, all code written this week will be a part of your “Common” engine:

- World/System/GameObject/Component hierarchy
 - GameWorld class representing representing the game world which supports:
 - * Timed updates (tick)
 - * Render events (draw)
 - * Adding and removing Systems
 - * Adding and removing GameObjects
 - System class representing one type of functionality that your world supports, containing:
 - * Timed updates (tick)

- * Adding and removing GameObjects
- GameObject class representing one entity within your game, supporting:
 - * Adding and removing Components
- Component class representing the smallest unit of functionality in a single entity, supporting:
 - * Timed updates (tick)
- Engine supports an input map
- Engine has a TickSystem
- Engine has a DrawSystem
- Engine supports a third-person camera
- CollisionSystem that can handle Cylinder-Cylinder collisions
 - Detection - determine that two cylinders are overlapping
 - Resolution - translate the cylinders out of detection using the MTV
 - Response - dispatches collision callback

Game Requirements

Now that you have some basic collisions, you can complete your first game! We're intentionally vague in the following requirements to encourage you to think outside the box: what can you do with just cylinder-cylinder collisions and a flat floor? We're looking forward to seeing the results.

Your game must fulfill the following set of requirements:

- The player should be able to jump (but only when on the ground, or under other specific conditions)
- The player shouldn't fall through the ground
- The player should fall under the effects of gravity
- Camera and player movement should be continuous
- Cylinder-cylinder collisions are used somewhere
- The game has some non-arbitrary win or loss condition
- The game must be resettable
- The game cannot enter an unwinnable or unlosable scenario (or resets upon doing so)