2026 Humanoid Robotics Learning Roadmap

1 January - February (Winter Vacation / Junior Year, Second Semester)

Foundation Building Month

1. C++ Introduction: Complete *Accelerated C++* or equivalent courses, master classes, STL containers, and modern C++ fundamentals.
2. Mathematics & Tools: Review linear algebra and probability theory; proficiently master Linux commands and Git.
3. RL Enlightenment: Start learning the first 4 chapters of Sutton's *Reinforcement Learning* book or Li Hongyi's RL course.

Milestones:

1. Rewrite a C program in an object-oriented style using C++.
2. Establish a well-organized personal repository on GitHub for recording study notes and code.
3. Manually derive and implement a simple tabular RL algorithm (e.g., Q-Learning).

2 March - April (Semester Term)

Algorithm & Simulation Introduction Month

1. C++ Advancement: Learn the basics of the Eigen library, implement robot forward kinematics with C++. Master CMake.
2. Python Enhancement: Proficiency in NumPy and basic PyTorch.
3. Core RL Learning: In-depth study of the principles of Policy Gradient and PPO algorithms.
4. Initial Simulation Exploration: Install PyBullet, run and understand its examples.

Milestones:

1. A C++ Eigen project capable of calculating the position of a robot end-effector.
2. Implement PPO with PyTorch in simple environments such as CartPole.
3. Load a standard bipedal model (e.g., Cassie) in PyBullet and control it via keyboard or scripts.

3 May - June (End of Semester)

Project Rehearsal Month

1. C++ Integration: Learn LibTorch C++, implement loading PyTorch models and performing inference.
2. RL Simulation Project: Train a walking policy for a bipedal robot (non-G1) in PyBullet using Python (Stable-Baselines3).
3. G1 Documentation Research: In-depth study of Unitree G1 official documentation, SDK/API, and URDF model (if available).

Key Milestone 1:

1. A complete sample code repository for ”Loading RL Models with C++”.
2. A bipedal RL policy capable of walking in simulation, with training curves and demonstration videos attached.
3. A preliminary technical research report on how to control the G1 robot.

4 July - August (Summer Vacation)

Physical Robot Breakthrough Month

Fully commit to the core ”G1-RL” project:

1. Build G1 Simulation Environment: Import the G1 model into PyBullet and write an RL training environment.
2. Sim2Real Training: Train G1’s balance/stepping policy in simulation, apply domain randomization.
3. Physical Deployment Exploration: Connect to the real G1 using C++ (or ROS 2 if needed), attempt to deploy the simplest policy (e.g., PD standing).

Key Milestone 2 - Core Achievement:

1. Open-source ”Unitree G1 RL Simulator”.
2. A successfully trained G1 policy model in simulation.
3. Experimental report and video of physical deployment (even for a simple policy).
4. A detailed project document and GitHub repository.

5 September - October (Senior Year, First Semester)

Deepening & Optimization Month

1. Project Review & Improvement: Analyze problems in the summer project and attempt improvements (e.g., imitation learning, better reward functions).
2. C++ Performance Optimization: Optimize the real-time performance of deployment code, learn multi-threading.
3. Knowledge Gap Filling: Learn ROS 2 basics if helpful for deployment, based on project needs.
4. Resume & Application Preparation: Complete the first draft of technical resume by the end of September, highlighting the G1 project.

Milestones:

1. Version 2 of the project with improved performance or stability.
2. Final resume with quantified project outcomes (e.g., ”Training speed increased by X
3. Start targeted LeetCode practice (C++).