GO BADUK WEIQI

Game Template

Documentation

by Fly Studios Games

Scene overview, script reference & customisation guide
for Unity developers.

1. Overview

Go Baduk Weiqi – Game Template is a complete, mobile-ready Unity implementation of the classic board game Go (Baduk / Weiqi). The template ships ready to build and publish: AI opponent, first-time tutorial, matchmaking simulation, sound effects, background music, and a clean scalable UI.

Supported Platforms

- Android
- iOS
- WebGL (portrait orientation supported)
- PC / Mac (standalone)

2. Quick Start

Step 1 — Import

Import the package into your Unity project. Unity 2021.3 LTS or newer is recommended.

Step 2 — Open the Demo Scene

The project contains one scene. Open it from:
Assets / Fly Studios Games / Go Baduk Weiqi / Scenes / Go Demo.unity

Step 3 — Press Play

On first launch the tutorial appears automatically. On return visits it is skipped (state saved via PlayerPrefs) and matchmaking starts directly.

3. Scene Overview

The demo scene (**Go Demo.unity**) is the only scene in the project and is set as Build Index 0. All game systems live on a single root object called **GoGameSystemsRoot**, keeping the hierarchy clean and easy to extend.

GameObject	Role
MainCamera	Orthographic camera. Renders the board and all UI canvases.
GoBoardTable	Board background sprite with a BoxCollider2D for touch and click input.
GoGameSystemsRoot	Host object for all core logic: GoGameManager.cs, MatchmakingController.cs, BotNameManager.cs, TutorialManager.cs, BannerOverlayManager.cs.
GoGameUICanvas	Primary HUD canvas — score, prisoners, player profiles, Pass and Confirm buttons.
GoGameOverlayCanvas	Overlay canvas for result banners (Black wins, White wins, Draw).
GoGameTutorialCanvas	Tutorial slides canvas. Inactive by default; activated by TutorialManager.cs.
WebGLPortraitSupport	Hosts WebGLPortraitEmulator.cs — forces portrait mode on WebGL builds.

DevelopmentDebugController	Debug utilities only. Keep this object INACTIVE before publishing.

4. Script Reference

All scripts are in **Assets / Fly Studios Games / Go Baduk Weiqi / Scripts/**.

GoGameManager.cs

The core of the template. Manages the full lifecycle of a game: board initialisation, stone placement, move validation (captures, Ko rule), AI logic across three difficulty levels, score calculation (Chinese and Japanese rules), turn timer, and end-game detection (two consecutive passes).

Method	Description
StartGame(difficulty, useAI, randomColor, startingPlayer)	Initialises and starts a new game with the given configuration.
ConfirmMove()	Commits the pending stone to the board and advances the turn.
PassMove()	Passes the current player's turn. Two consecutive passes end the game.
Enum	**Description**
AIDifficulty	Easy | Normal | Smart
StartingPlayer	Black | White
Inspector Field	**Description**
turnTimeSeconds	Per-turn timer in seconds. Set to 0 to disable.
minMoveReactionAI / maxMoveReactionAI	Random delay range before the AI plays (seconds).

MatchmakingController.cs

Simulates an online opponent search with an animated countdown before launching the game. Works with BotNameManager.cs to display a random bot name as the opponent found.

Method	Description
StartMatchOpponentFind()	Starts the animated search sequence and countdown timer.
StartGameDefault()	Launches a 9x9 Normal-difficulty game immediately. Useful for a quick-play button or for testing.
GetCurrentOpponentName()	Returns the currently assigned opponent name as a string.

TutorialManager.cs

Manages a slide-based first-time tutorial. Completion state is saved via PlayerPrefs so returning players skip straight to matchmaking. Each slide is a TutorialStep with a name, a Sprite, and a text description.

Method	Description
TryStartTutorial()	Shows the tutorial if not yet completed; otherwise goes straight to matchmaking.
ForceShowTutorial()	Always shows the tutorial from slide 1. Wire this to a How to Play button.
NextStep()	Advances to the next slide, or finishes the tutorial on the last step.
PreviousStep()	Goes back one slide.

■ *To reset tutorial progress: PlayerPrefs.DeleteKey("TutorialCompleted");*

AudioManager.cs

Controls all in-game audio. Background music loops with configurable pause intervals and smooth fade in/out. Sound effects use randomised clip arrays for natural variation.

Method	Description
PlayCapture()	Plays a random stone-capture sound effect.
PlayPlace()	Plays a random stone-placement sound effect.
PlayPass()	Plays the turn-pass sound effect.
PlayButtonClick()	Plays the UI button-click sound effect.
Inspector Field	**Description**
loopsBeforePause	Number of music loops before a silence break.
pauseDuration	Seconds of silence between music loops.
fadeDuration	Seconds for music fade in/out transitions.

BotNameManager.cs

Provides a random human-like name for the AI opponent shown during matchmaking. The name list is fully editable in the Inspector without any code changes.

BannerOverlayManager.cs

Displays result banners on the overlay canvas (Black wins, White wins, Draw). The companion EditorBannerOverlay.cs adds Inspector buttons so you can preview banners without entering Play mode.

NotificationManager.cs

Shows brief in-game notifications such as Illegal move, Your turn, or Opponent passed. Called automatically by GoGameManager.cs.

ConfirmMoveUI.cs

Shows a Confirm / Cancel panel after a stone is tapped but before it is committed to the board, letting the player review their placement.

PlayerProfileUI.cs

Renders a player card (avatar, name, prisoner count) for both the local player and the opponent. Updated automatically by GoGameManager.cs after every move.

SafeArea.cs

Adjusts the UI canvas to respect the device safe area, preventing elements from being hidden behind notches or rounded corners on iOS and Android.

WebGLPortraitEmulator.cs

Forces portrait orientation in WebGL builds. Has no effect on other platforms. Already attached to the WebGLPortraitSupport object in the scene.

ButtonPulseAnimation.cs

Applies a repeating scale-pulse animation to a UI button to draw the player's attention. Speed and scale range are configurable in the Inspector.

RestartCurrentScene.cs

Utility with one public method that reloads the active scene. Assign it to a Restart button via the Inspector OnClick event.

TimeFast.cs

Developer-only utility that multiplies Time.timeScale to speed up gameplay during testing. Keep the DevelopmentDebugController object INACTIVE before publishing.

5. Customisation Tips

AI Reaction Speed

Adjust **minMoveReactionAI** and **maxMoveReactionAI** on GoGameManager.cs to control how quickly the AI responds (in seconds).

Turn Timer

Change **turnTimeSeconds** on GoGameManager.cs. Set to 0 to disable it entirely.

Tutorial Slides

Select GoGameSystemsRoot, expand TutorialManager.cs in the Inspector, and edit the **tutorialSteps** list. Each step has a name, a Sprite image, and a text description.

Bot Names

Edit the name list on BotNameManager.cs directly in the Inspector — no code needed.

Music & SFX

Assign audio clips to AudioManager.cs in the Inspector. Swap any clip without touching code — just drag the new file into the slot.

Scoring Mode

Scoring mode is a parameter of GoGameManager.StartGame(). Wire a UI toggle to switch between Chinese and Japanese rules before a match.

---

6. Support

If you have questions, found a bug, or need help integrating the template, reach out through any of the channels below.

Email	flystudiosgames@gmail.com
Telegram	https://t.me/fly_studios_games
Discord	https://discord.gg/9sMmQ7Eg

Thank you for purchasing Go Baduk Weiqi – Game Template!

Fly Studios Games | Unity Asset Store