UNITY 6000.0.62F1 · COMPLETE OFFLINE DOCUMENTATION

Difference Duo Developer Guide

Install. Edit. Reskin. Build your release.

szekipapa77@gmail.com



CHAPTER 01

Start here: your first detective duel

Difference Duo is an editable Unity template built around a simple question: what changed between these two pictures? Find every difference before an animated computer-controlled detective does. The opponent is local game logic, not a connected player. No multiplayer server or account service is supplied.

This handbook follows the work you will do: run the game, edit a round, change the look and wording, then build your release. Paths are relative to the Unity project. Keep the game Kit: it supplies the canvas, audio, saving, popups and platform bridge. Szekipapa77 Ads is publisher-owned integration code; third-party advertising SDK binaries are not in the buyer package.

Included content: 25 curated levels, eight illustrated scene backgrounds, 32 prop stickers, five detective rivals, four helpers, first-play guidance, a visual level editor and seeded generation. After the curated bank, the runtime generates further levels deterministically. Online opponents, cloud saves, purchases, LiveOps events and remote backends are not included.

Use the standalone project for a new game or the native unitypackage for an existing project. Import one game per project: private Kits deliberately share class names. Most screens are created at runtime, not stored as editable Canvas children in the scene. Support: szekipapa77@gmail.com.

CHAPTER 02

Installation and first play

Required tools

Use Unity **6000.0.62f1**, URP **17.0.4** and Unity UI **2.0.0**. Retain the matching official package dependencies and modules listed in the manifest. The game uses Unity UI Text; a separate TextMesh Pro import is unnecessary. Install Android Build Support with matching SDK, NDK and OpenJDK through Unity Hub for Android, and WebGL Build Support for browser builds.

Open the standalone project

1. Extract the complete source archive to a short writable folder such as C:/UnityWork/DifferenceDuo. Open the folder containing Assets, Packages and ProjectSettings.
2. Open through Unity Hub with 6000.0.62f1. Wait for package resolution and compilation. Fix the first Console error before proceeding.
3. Run `Szekipapa77 > Difference Duo > Setup`. It configures the game pipeline, player settings and entry scenes.
4. Under Edit > Project Settings > Player > Other Settings, set Active Input Handling to **Input Manager (Old)**, or Both if your project also installs the new input system. New Input System only does not satisfy this game's classic input path.
5. Open `Assets/Games/SpotDuo/Scenes/SpotDuo_Mobile.unity`. Press Play, dismiss the introduction with Play and tap the difference indicated by the hand.
6. Find the remaining differences. The result offers Next Level. Stop Play mode before changing level data or replacing resources.

Import into an existing project

1. Back up the destination. Install the matching URP and Unity UI versions through Package Manager.
2. Use Assets > Import Package > Custom Package and import the supplied native package, including the game, Szekipapa77Ads and WebGL template folders.
3. Run Setup, check input handling and repeat first play. Setup changes rendering and player settings, so inspect those when combining with an existing project.

Preserve .meta files and GUIDs. If resources seem missing, reimport the complete folder rather than inventing new references. The optional Google provider is deliberately absent; simulated ads are expected without it.

CHAPTER 03

The file map

Game root: `Assets/Games/SpotDuo`. These are actual source files in this product.

Path under the root	Responsibility
Scripts/SpotDuoGame.cs	Flow, tap resolution, rival clock, scoring, helpers and debug hooks.
Scripts/SpotDuoGame.View.cs	Pictures, patch rendering, rings, fog and responsive layout.
Scripts/SpotDuoGame.Hud.cs	HUD, helper captions, settings, help and tutorials.
Scripts/SdData.cs	Serializable level, sticker and difference models; catalog and rivals.
Scripts/SdGen.cs	Seeded generation, hit mathematics, perception estimates and validation.
Scripts/SdLevels.cs	Curated resource cache, clone-on-read and generated continuation.
Scripts/SdArt.cs	Named palette and cached art/material loading.
Scripts/SdPatch.cs / SdSfx.cs	Feathered patch graphics / sound resource constants.
Editor/SpotDuoLevelEditor.cs	Visual authoring, generator and validation window.
Editor/SdImport.cs	Texture and sound import rules.
Kit/Runtime/HzGame.cs / HzCore.cs	Base lifecycle, canvas, orientation / saving and services.
Kit/Runtime/HzAds.cs / HzUI.cs	Publisher ad calls / labels, fonts and icons.
Kit/Editor/HzTools.cs	Setup, rendering configuration, native export and builds.

Authored data is `Resources/spotduo/Levels/levels.json`; catalog.json contains scene colour grids and sticker dimensions. Art lives under Resources/spotduo/Art, with Scenes and Props subfolders. Sfx and Music are siblings under Resources/spotduo. Kit fonts have their own path: `Kit/Resources/spotduo/kit/Fonts`.

Entry scenes: `Scenes/SpotDuo_Mobile.unity` is portrait; `Scenes/SpotDuo_WebGL.unity` is landscape. Despite its historical name, the standard current WebGL build command selects the portrait scene. Choose landscape explicitly when making that edition.

The separate ad layer is `Assets/Szekipapa77Ads`; its settings are Resources/AdsSettings.asset. The phone-oriented HTML shell is `Assets/WebGLTemplates/Szekipapa77`. Neither folder is another game.

CHAPTER 04

Controls, rival, scoring and results

Compare and tap

Tap or click a difference on either picture. Portrait stacks the pictures; landscape places them beside each other. A find is circled in both. `SdPicInput.OnPointerDown` forwards to `SpotDuoGame.OnPicTap`, which converts the screen point to picture coordinates before calling Tap. No dragging is required.

A moved object accepts either its original or moved position. Mirrored rounds flip the right picture while keeping hit mapping consistent. Tapping fog clears it without a miss. An already-found target animates its ring without awarding duplicate points. If unfound target areas overlap, the nearest matching difference wins.

Local rival and mistakes

The rival follows a deterministic schedule based on the level. Progress dots show its finds; a speech bubble warns when one remains. Its initial delay and completion time are level data. During the first tutorial it waits for your first find. Game popups stop normal active play through PopupOpen/CanPlay checks.

Three consecutive misses cause Seeing Double and lock the pictures for three seconds while the rival continues. A correct find clears the miss counter. Steady Eyes blocks a limited number of misses, not the entire round.

Points and stars

Each find starts at 100 points, adds a declining speed bonus and can add a streak bonus. Finds less than 2.5 seconds apart build the streak. Winning adds a time-lead bonus. The HUD displays the accumulated total. A win earns one star, another for avoiding Seeing Double and another when the rival found at most half the targets. The best star result for each level is retained.

End and retry

Find every difference before the rival to win. The next level index is saved, then Next Level starts it. If the rival finishes first, Retry repeats the current level. A once-per-round rewarded rescue makes the rival lose its last two finds and continues the round; it depends on a successful earned callback. This is local game state, not a server operation.

CHAPTER 05

First-play and helper tutorials

The tutorial is in `SpotDuoGame.Hud.cs` : StartTutorial, TutorialFound and StopTutorial. A pointing hand identifies the first difference, then a second tip asks for the rest. The completion flag `spotduo_tut` persists, so restarting Play mode alone does not reset the lesson. Use a separate development save or clear only the game's relevant test keys.

Helper introductions are separate. UnlockPopup grants initial charges once, StartGuide/RefreshGuide points to the newly available tool, and CompleteGuide records that its first use was taught. The guided first use is free. A returning player may have completed the main tutorial but still need a helper introduction.

New mechanics use the authored level hint, NewKindHint in SpotDuoGame.cs and SdDiffKinds.IntroHint in SdData.cs. For a single level, edit First-time hint in the visual editor. For general tutorial wording, edit the actual methods, preserving interpolated counts and rival names. Shared font or UI-kit settings do not automatically translate these game sentences.

After a tutorial change, check a fresh save and an existing save, portrait and landscape, a return from settings, and the transition after the first find. A guide that looks correct only before the layout refresh is not sufficient.

CHAPTER 06

Four tools, unlocks and charges

Tool	Unlock / initial charges	Effect
Magnifier	Level 3 / 3	Selects the hardest unfound target, clears fog over it and draws a ring; the player still taps it.
Freeze	Level 6 / 2	Stops the rival clock for 10 seconds with a visible frozen portrait/countdown.
Radar	Level 10 / 2	Pulses the remaining target areas for 2.6 seconds.
Steady Eyes	Level 15 / 2	Blocks the next three misses and displays remaining protection.

In `SpotDuoGame.cs`, BoostUnlock is {3,6,10,15}, BoostStart is {3,2,2,2}. BoostName and BoostArt use the same order. The Hud partial's RefreshBoosters builds locked-level captions and charge badges. Keep parallel arrays aligned.

OnBooster dispatches UseHint, UseFreeze, UseRadar and UseShield. Consume takes a charge except during the free guide. An empty button opens OfferCharge; a successful Watch ad result adds one charge, then the player activates the helper normally. There is no cash purchase system here.

To reduce Freeze duration, change FreezeSeconds from 10 to 8, update matching explanation strings and test the countdown. To move an unlock, edit its array element and start a controlled fresh save; existing seen/guided flags may suppress the first-use flow. Verify locked captions, free grant and tutorial together.

CHAPTER 07

Make one level in the editor

1. Exit Play mode. Open `Szekipapa77 > Difference Duo > Level Editor`. Select a level and press Dup to preserve the original.
2. Set Name, Scene, Rival and First-time hint. Names and hints are player-visible; scene/rival selectors use catalog identifiers.
3. Choose + Missing and a prop. Click a clear picture area. Switch to Select and drag to adjust. Use the inspector for size, side and hit radius.
4. Add a Recolour target elsewhere. Choose an obvious alternate tint. Do not cover targets with decoration stickers.
5. Set Rival finishes at (s) and Rival start delay (s), or use Auto pace as a starting estimate.
6. Validate the level and resolve warnings. Save writes levels.json and refreshes Unity. Play this level saves, opens the landscape scene at that index and enters Play mode.
7. Test portrait separately, including small-screen clue visibility. Back up the JSON before large changes.

Tools and gestures

Tools cover Missing, Recolour, Moved, Flipped, Resized, Tinted and Vanished differences, shared Sticker decoration and Fog. Drag a moved difference's second point to change its destination; a clone patch's second point chooses its sampled source. Mouse wheel or square brackets resize; right-click deletes. Keys 1–7 change type, R rotates, F flips, S swaps the changed side, arrows nudge and Delete removes the selected item.

The list provides Add, Dup, Del, Up and Down. Its order is runtime order. Reordering can change the content a saved level number resolves to. Undo, the dirty marker and save-on-close prompt support authoring, but keep independent backups of intentionally large edits.

The inspector shows every selected difference/sticker/fog field, its tap radius and perception estimates. Validate All inspects the whole bank. The curve highlights difficulty, easier breathers and failures; do not interpret bot estimates as guaranteed human difficulty.

CHAPTER 08

Generate and extend safely

Control	Meaning and initial setting
Count	1–30 generated entries; initially 5.
Start level #	Progression number for allowed mechanics, initially 26; not always the insertion index.
Seed	Initially 1234; subsequent seeds add index × 7919.
Start / End difficulty	0–1 design range, initially 0.45 / 0.85.
Breather every	0 disables easier breathers; otherwise 1–10, initially 5.
Replace levels from start #	Off inserts after selection; on replaces from the clamped start index and appends beyond the end.

Worked append

1. Back up levels.json. Select the final authored level, 25.
2. Set Count 3, Start level # 26, Seed 9127, Start difficulty 0.45, End difficulty 0.60 and Breather every 0.
3. Leave Replace unchecked. Generate inserts three entries after the selection, retaining the existing 25.
4. Inspect all pairs and rival pacing, run Validate All and Save. Restart Play mode to avoid cached data.

Worked replacement

For levels 10–12, use Start level # 10, Count 3 and enable Replace. The implementation clamps the insertion start to the bank's valid bounds, replaces existing entries and appends only if the range runs beyond the current end. It does not create another parallel bank.

Runtime continuation after the curated bank uses SdGen.ForLevel. Editor generation uses the same SdGen family with explicit seeds and a difficulty curve. Validation checks bounds, overlaps, subtlety and estimated bot times/win rates; it cannot replace human inspection of ambiguous clues.

CHAPTER 09

Understand the actual JSON

The outer object is `{ "levels": [ ... ] }`, represented by SdLevelFile. This real first-level excerpt retains only its first difference/decoration/fog items. Do not overwrite the full bank with the shortened example.


```
{
  "name": "Breakfast Mix-up",
  "scene": "kitchen",
  "rival": "owl",
  "hint": "Tap anything that differs between the two pictures: on either side.",
  "seed": 1037,
  "mirror": false,
  "breather": false,
  "difficulty": 0.02,
  "rivalTime": 27,
  "rivalDelay": 4.3,
  "deco": [
    {
      "prop": "sunflower",
      "x": 0.8819567,
      "y": 0.22110534,
      "s": 0.107650526,
      "rot": -7.230486,
      "tint": "#F1E6FF",
      "flip": true
    }
  ],
  "diffs": [
    {
      "IsProp": true,
      "kind": "missing",
      "prop": "cupcake",
      "x": 0.49182767,
      "y": 0.40271395,
      "s": 0.105339,
      "rot": -7.3461733,
      "tint": "",
      "flip": true,
      "side": 0,
      "alt": "",
      "dx": 0,
      "dy": 0,
      "val": 0,
      "r": 0
    }
  ],
  "fog": [],
  "score": 16.473146,
  "botTime": 9.222903,
  "botRate": 1
}
```


Field	Meaning
name / hint	Visible player wording.
scene / rival	Exact catalog keys; do not translate technical IDs.
seed / difficulty / breather	Generation identity, design difficulty and easier-level marker.
rivalTime / rivalDelay	Rival completion and initial waiting seconds.
mirror	Mirrors the right picture and its input mapping.
deco	SdProp: prop, x, y, s, rot, tint, flip.
diffs	SdDiff: kind, prop, x, y, s, rot, tint, flip, side, alt, dx, dy, val, r.

fog	SdFog: x, y, r.
score / botTime / botRate	Validation estimates, not the player save score.

x/y are normalized 0–1 picture coordinates, y down. Size and radius use picture width. Aspect is 1.5. Difference side 0 places the change on the right picture, side 1 on the left. Exact kind keys are missing, color, moved, flip, size, hue and clone.

Color uses alt for the changed tint; moved uses dx/dy for its displacement; clone uses them for the sampled source offset. Size uses val as changed scale and hue uses it as hue shift. Positive r overrides the automatic hit radius. Use the editor to see these fields before making broad text changes.

Catalog scene colour grids and sticker dimensions support visibility estimates. After a substantial image replacement, inspect catalog assumptions and validate every affected level. Replacing a JPG does not update its perception model automatically.

CHAPTER 10

Reskin the interface palette

Named colours are actual fields in `Assets/Games/SpotDuo/Scripts/SdArt.cs`. The following values connect directly to current UI consumers.

Symbol / current value	Actual use
Teal / #2EC4B6	Hud.BuildHud level pill, help accent, player ring and shared themed accents.
Coral / #FF6B6B	RefreshBoosters charge badges, failure accents and picture motifs.
Cream / #FFF5E1	SdPopupFull outer card.
CardC / #FFFBF2	Duel panel and popup inner card.
Ink / #3D2C4A	Duel labels and unlocked helper captions.
Sub / #8C7A99	Secondary and locked captions.
BgTop / #7FD8CF; BgBot / #F6C6D0	View.BuildView background via HzUI.Gradient.
FoundC / #33D17A	Found progress pips and success feedback.
MissC / #FF4D5E	RefreshHud miss-eye feedback.
Leaf / #3DBE6E	Play, Next Level and positive actions.
Plum / #6D597A	Settings/help button background and neutral actions.
Sun / #FFD166	Tutorial outlines and guided/active helper ring.
BoostColor in SpotDuoGame.cs	Helper UI accents: #F2B33D, #4FA8E8, #2EB8A6, #F0647A.

Worked colour edit

1. Exit Play mode and back up SdArt.cs. Find `Teal = HzUI.Hex("#2EC4B6")`.
2. Replace only that hex string with `#557A68`. Keep the symbol and declaration syntax.
3. Recompile and start portrait. The level pill, player ring and other teal accents should change. Open help and an introduction popup; confirm white text remains readable.
4. Stop and run landscape, then advance a level. This tests rebuild and refresh paths as well as initial construction.
5. If only the pill should change, create a dedicated palette field and use it in that BuildHud Card call. Editing shared Teal intentionally affects every consumer.

Change BgTop/BgBot for the backdrop, then restart Play mode: static fields and cached graphics are not a live theme editor. BoostColor changes button and popup accents, not baked PNG colours. To change icons, replace their artwork separately.

Shadows, disabled-eye states, popup dimming and some transparency use inline new Color values in the Hud/View partials. Search those exact files and edit a specific consumer when a named field is insufficient. Preserve meaningful alpha values for overlay legibility.

CHAPTER 11

Reskin visible words without breaking IDs

The interface combines hard-coded Unity UI strings, per-level wording and rival labels. Shared-kit styling does not automatically translate game text. No word dictionary is involved here.

Visible words	Exact source
Product title	SpotDuoGame.cs: HzGameInfo display name and Title property; Kit/Runtime/HzKit.cs: Title constant; Setup applies productName.
LEVEL, SCORE, You	SpotDuoGame.Hud.cs: RefreshHud assigns _levelTxt, _scoreTxt, _youTxt.
Rival names and count	SdData.cs: SdRivals.Names; Hud.RefreshRival; SpotDuoGame.cs derives RivalShort.
Level name / introduction hint	levels.json name/hint; editor Name and First-time hint; ShowPreLevel and RefreshHud display them.
Play / round goal	SpotDuoGame.cs: ShowPreLevel and NewKindHint.
Results, Next Level and Retry	SpotDuoGame.cs: LevelComplete / GameOver.
Rival speech	SpotDuoGame.cs: RivalFinds.
Settings and its actions	SpotDuoGame.Hud.cs: ShowSettingsPopup, SoundLabel and MusicLabel.
Help paragraphs	Hud.cs: HelpPages and HelpPage.
First tutorial	Hud.cs: StartTutorial and TutorialFound.
Helper names / locked labels	SpotDuoGame.cs: BoostName; Hud.cs: RefreshBoosters and UnlockPopup.
Helper guidance	Hud.cs: RefreshGuide and CompleteGuide.
Ad refill wording	SpotDuoGame.cs: OfferCharge.
Difference labels / hints	SdData.cs: SdDiffKinds.Titles and IntroHint.

Worked HUD edit

In Hud.RefreshHud, replace:

```
_scoreTxt.text = "SCORE " + Total.ToString(CultureInfo.InvariantCulture);
```

with:

```
_scoreTxt.text = "POINTS " + Total.ToString(CultureInfo.InvariantCulture);
```

Keep the numeric expression and formatting. Recompile, find a difference and confirm the updated total appears under POINTS. Open settings, return, switch to the other supplied scene and advance a level. The wording must survive BuildView and the next RefreshHud call.

Worked helper and title edits

To rename Magnifier to Clue Lens, change the first BoostName value. Captions and guides using that array follow. Inspect help/unlock explanations for any hard-coded wording. Keep booster_hint as the resource key and retain the same array order.

For a product rename, update the display Title, HzGameInfo display name, HzKit.Title, listing copy and this manual. Run Setup to apply productName. MenuItem strings in HzTools.cs and SpotDuoLevelEditor.cs are developer-facing editor menu labels and can also be updated. Do not rename spotduo, SpotDuo, namespace or save/resource keys just to change visible wording.

Hierarchy names such as Ribbon/Title, LevelPill and YouTxt are lookup names, not translations. Keep technical difference kinds, resource IDs and save keys unchanged unless implementing a planned migration. Preserve interpolation braces, number formatting and newline escapes when translating. Check long words, punctuation and counts in both orientations, including locked captions, charge badges, help pages and results.

CHAPTER 12

Artwork, fonts and audio

Backgrounds, props and rivals

Scenes load from `Resources/spotduo/Art/Scenes/scene_<id>.jpg`. Keep their 3:2 shape. Stickers are `Art/Props/prop_<id>.png` with transparency. To replace an existing image, overwrite it while retaining its .meta file. New identifiers need art and catalog entries before being selected in the editor.

Do not leave excessive transparent borders around stickers. Their visual footprint must agree with the catalog. A flipped clue needs an asymmetric prop; a colour clue needs a clearly visible difference from the base art. Test at actual phone sizes, not only a large editor preview.

Rival keys are rival_fox, rival_cat, rival_owl, rival_penguin and rival_raccoon. BoostArt identifies helper files. SdArt caches sprites and scenes; restart Play mode after replacing resources.

Scenery patches

Keep `SdPatch.shader` and `SdPatchMat.mat`, including their GUID relationship. Feathered hue and clone-stamp differences depend on the material and extra canvas texture-coordinate channels. Harsh rectangles or missing scenery differences warrant checking shader/material import before replacing backgrounds.

Type and licences

`Kit/Resources/spotduo/kit/Fonts` contains LilitaOne.ttf, Nunito.ttf and MaterialIcons.ttf. HzUI.Display and Body load the first two; HzIcons.Font loads the icon font. Replace text fonts with suitable licensed fonts and preserve paths or update loaders. An ordinary font cannot replace the icon font's codepoints.

Keep the supplied font notices, including OFL and MaterialIcons licence files. Use a font with glyph coverage for your language and inspect accented letters, punctuation and numbers. Do not put emoji into a font that does not supply them. For any newly added art, music or font, retain the rights evidence outside your runtime assets and supply applicable redistribution notices.

Sound

SdSfx.cs names resource events. Sfx contains find/streak/miss, rival, fog, the four helpers, popup and result sounds. Music uses `Music/duo_loop.ogg`. Retain basenames or update the matching constants. HzAudio supplies playback and combo pitch. Settings exposes separate sound/music toggles; verify both after replacement.

Balance music beneath clue feedback and test loud transitions for clipping. Marketing voiceover is separate from in-game sound and is not a runtime dependency.

CHAPTER 13

Saving and compatible updates

HzSave in Kit/Runtime/HzCore.cs wraps PlayerPrefs and coalesces writes. Progress uses `spotduo_level`, score uses `spotduo_total` and tutorial completion uses `spotduo_tut`. Charges are spotduo_b0 through spotduo_b3. Initial unlock flags are spotduo_seen0 through spotduo_seen3; guided-use flags are spotduo_guided0 through spotduo_guided3. Best stars use spotduo_stars plus the level number.

Other flags track introduced difference types, fog, mirror and the first Seeing Double explanation. These are an update contract. A visible title rename is safe; replacing every spotduo string can reset progress or break Resources loading. Test migration before making that kind of change.

PlayerPrefs is local, not secure currency storage or cloud synchronization. Browser storage can be cleared or restricted. Android uninstall can remove progress. There is no supplied cross-device account. Cloud saving requires your own provider and migration logic.

A saved level refers to its bank position, not a stable content GUID. Append where possible. Reordering or replacing earlier levels can show different content to returning players at the same saved index. Back up development saves before testing a new bank.

CHAPTER 14

Advertising: included integration and external provider

The buyer source contains Szekipapa77 Ads integration, settings and an optional adapter. It does not contain Google's Mobile Ads SDK or External Dependency Manager. This distinct source edition does not include HyperCasual LiveOpsKit, purchases, LiveOps stores or a remote backend. Do not follow older templates' HLK configuration instructions.

The call path is game → HzPlatform → game Kit HzAds → Szekipapa77.Ads.AdsService. The optional adapter detects a separately installed Google plugin. Real rewards depend on an earned callback; an installed provider failing to show an ad does not silently become a successful simulated reward.

AdsSettings.asset field	Supplied meaning
backend: 2	Requests AdMob. This enum is None=0, Dummy=1, AdMob=2; it is not the HLK enum.
missingSdkFallback: 1	Dummy simulation when the provider is absent.
useGoogleTestIds: 1	Test application and ad unit IDs if an optional SDK is installed.
dummyDelaySeconds: 0.4	Development simulation delay.
pauseGameDuringFullscreenAds: 1	Pauses active game flow during full-screen ads.
loadTimeoutSeconds: 25	Ad load waiting budget.

Development

Open `Tools > Szekipapa77 Ads > Setup & Diagnostics`. With no Google plugin, fallback is simulated. WebGL always simulates because this AdMob adapter does not support browser ads. Simulation tests UI/callback flow; it is not revenue. A separate test-ad Android demo can be compiled with an externally imported SDK without redistributing that SDK in the buyer source.

Configure an optional Google provider

1. Back up the project. Obtain the official Google Mobile Ads Unity plugin and its dependency manager from Google; import them separately and follow their current platform instructions.
2. Check detection in the publisher diagnostics window. Keep test IDs enabled while testing. Exercise success, cancellation/failure, earned reward exactly once, banners and interstitial continuation.
3. Enter your own app ID and format-specific unit IDs in `Assets/Szekipapa77Ads/Resources/AdsSettings.asset`. Use your account identifiers for your release.

4. Before a production build, set Missing SDK Fallback to None and disable Use Google Test IDs only after entering valid production IDs. The supplied build tooling refuses a missing-provider release or a release still configured for simulated fallback.
5. Provide your own privacy policy, consent flow, Privacy Options and required disclosures. Check them on physical devices in the relevant regions. Template configuration alone does not establish compliance or revenue.

Interstitials are throttled to at most one per 60 seconds. Failure/skipping continues game flow once. Phone banners reserve bottom space; inspect camera/layout clearance and device safe areas after reskinning. The game has no supplied production IAP configuration.

CHAPTER 15

Android and browser releases

Android build

1. Install the Unity-matched Android tools. Run Setup and inspect the portrait scene.
2. Replace the template application identifier `com.szekipapa77.spotduo` with your own unique ID. Choose product/company labels, app icon, version and version code.
3. Review scripting backend and architectures in the actual Android Build Profile. Do not assume a supplied demo is configured for your target marketplace.
4. Configure the provider as above and run `Szekipapa77 > Difference Duo > Build Android`. Output is `Builds/Android/spotduo.apk`.
5. Use your own production keystore and keep its secrets outside distributable source. A test-signed demo is not production signing. Build an App Bundle if your destination requires it.
6. Install on physical devices. Check tap edges, tutorials, every helper, miss locks, pause/resume, sound toggles, ads and save persistence.

WebGL build

1. Install WebGL Build Support and run Setup. The supplied Szekipapa77 template frames a portrait phone layout.
2. Run `Szekipapa77 > Difference Duo > Build WebGL`. Output is `Builds/WebGL/spotduo`.
3. Upload index.html together with all Build and TemplateData content. Serve over HTTP(S), not file://. Preserve filenames referenced by the loader.
4. The setup uses compression with decompression fallback. If loading stalls, inspect Console errors and server compression/content headers.
5. For landscape, explicitly select SpotDuo_WebGL.unity in your Build Profile or adapted build scene list. The standard menu intentionally selects the Mobile portrait scene.
6. Test desktop mouse, mobile taps, audio after user interaction, storage/reload behavior and the actual hosted page. Browser ads remain simulated.

For each final build, retain size and SHA-256. Verify a complete anonymous download rather than an owner preview. A browser demo does not prove native provider behavior. Your accounts, publishing destinations and signing keys are not runtime dependencies.

CHAPTER 16

Troubleshooting by symptom

Symptom	Action
Input does nothing	Dismiss the intro, wait for any Seeing Double lock, check classic input settings and preserve the EventSystem/private Kit.
Tutorial is absent	Use a controlled fresh save; spotduo_tut persists across Play mode restarts.
Helper remains locked	Check BoostUnlock, current level and seen/guided flags. Test balance changes with a controlled save.
Edited data is unchanged	Exit Play mode, Save the editor and restart. SdLevels caches Resources.
Missing picture/font	Retain Resources folders, exact names, .meta files and appropriate font glyphs.
A clue cannot be tapped	Check fog, changed side, target overlap and hit radius; validate and tap either picture.
Scenery patch fails	Check shader/material GUIDs and extra canvas UV channels, then test an actual player.
WebGL is portrait	Expected for the standard build menu. Build the landscape entry scene explicitly when wanted.
Android ads simulate	No SDK is bundled. Inspect fallback/backend settings, diagnostics and your separately installed plugin.
WebGL has no AdMob	Expected; a separate web provider would need your implementation.
Release build stops	Use missing-SDK fallback None and a valid installed provider/production IDs. Do not bypass the release guard.
Duplicate class errors	Restore a clean one-game project. Different private game Kits are not designed to be imported together unchanged.
Pink rendering	Check URP 17.0.4, Graphics/quality pipeline selection and the first shader/package error. Run Setup.
Browser loading stalls	Serve the full directory via HTTP(S), keep loader filenames and inspect compression headers.
Exception after scene replacement	Reproduce the exact sequence and capture the first full stack. Delayed callbacks must not access destroyed game/UI objects.

Report the exact Unity version, platform, entry scene, level, reproduction sequence and first relevant Console stack. Do not send passwords, ad account secrets or signing keys.

CHAPTER 17

Before shipping and getting help

1. Run both supplied scenes on a fresh installation. Complete a round, restart and verify saved progression.
2. Exercise first-play and helper guides, all helper effects, miss lock, win/loss/retry and both rewarded success/failure paths.
3. Validate edited/generated content. Inspect clue readability and refresh paths in both layouts after a reskin.
4. Review licences, new artwork/font permissions, privacy/consent requirements and current platform rules for your release.
5. Replace application identity/signing configuration. Import the final package into another clean project to prove independence from caches, private tooling and other games.
6. Provide this offline handbook before import and inside the project. Verify public demo/manual/video links after publication.

Support: szekipapa77@gmail.com. Publisher: Szekipapa77. This is an editable development template; your final accounts, release choices, additional content, legal disclosures and store submission remain your responsibility.