UNITY 6000.0.62F1 · COMPLETE OFFLINE DOCUMENTATION

Stretch Cat Developer Guide

Install. Edit. Reskin. Build your release.

szekipapa77@gmail.com



CHAPTER 01

Start here: Stretch Cat

Stretch Cat is a standalone editable Unity game template. Swipe horizontally or vertically to slide Mochi until an obstacle stops her. Her long body stays on visited tiles. Cover every floor tile without trapping the head. Yarn, directional rugs and paired boxes add routing constraints; stars depend on helper use.

Includes 25 curated levels, level generation, a game-specific visual editor and validation, 4 guided helpers, local saving and portrait/landscape scenes. Keep the required private game Kit and publisher-owned Szekipapa77Ads layer. Provider SDK binaries are excluded. There is no bundled HLK, purchases, multiplayer server or remote LiveOps backend. Import only one game Kit into a project because the Kit classes share names.

CHAPTER 02

Requirements, installation and first play

Use Unity **6000.0.62f1**, URP **17.0.4** and Unity UI **2.0.0**. Install matching Android SDK/NDK/OpenJDK and WebGL modules in Unity Hub. Retain the supplied manifest and ProjectSettings.

1. Extract the standalone project to a short writable local path, then open it in the matching editor. Wait for package import and compilation.
2. For native unitypackage import, create a matching URP project and use Assets > Import Package > Custom Package. Import every game, Kit, Szekipapa77Ads and WebGL-template entry.
3. Run Szekipapa77 > Stretch Cat > Setup. It configures rendering and player settings; review its changes when integrating with existing work.
4. Active Input Handling must support the classic Input Manager: choose Old or Both. New Input System only is insufficient.
5. Open `Assets/Games/StretchCat/Scenes/StretchCat_Mobile.unity` and press Play. Follow the level card and first-level instructions.
6. Stop Play mode before editing level assets. Keep all .meta GUIDs. The game uses classic Unity UI Text; do not assume an extra TextMesh Pro import is required.

CHAPTER 03

Project folders and exact source map

All game-relative paths below start at `Assets/Games/StretchCat`. Runtime UI is built in code; editing an unrelated static Canvas does not reskin it. Scripts contains game rules and views; Editor contains authoring tools; Kit supplies UI/audio/save/platform support; Resources/stretchcat contains level JSON, art, music and sound; Settings owns URP assets; Scenes contains both entry scenes.

- `Scripts/ScArt.cs`
- `Scripts/ScCore.cs`
- `Scripts/ScGen.cs`
- `Scripts/ScLevels.cs`
- `Scripts/ScSfx.cs`
- `Scripts/StretchCatGame.cs`
- `Scripts/StretchCatGame.Hud.cs`
- `Scripts/StretchCatGame.View.cs`

Main flow: StretchCatGame.cs. UI and popup consumers: StretchCatGame.Hud.cs. Model/generator/loading: ScCore.cs, ScGen.cs, ScLevels.cs. Authoring: Editor/StretchCatLevelEditor.cs. Shared ad settings live outside the game at Assets/Szekipapa77Ads/Resources/AdsSettings.asset.

CHAPTER 04

Gameplay, controls and progression

Swipe horizontally or vertically to slide Mochi until an obstacle stops her. Her long body stays on visited tiles. Cover every floor tile without trapping the head. Yarn, directional rugs and paired boxes add routing constraints; stars depend on helper use.

The initial bank has 25 curated levels. Read the supplied loader/generator before changing continuation rules. Play the guided opening, ordinary progress, completion and retry paths; preserve saved indexes during a visual reskin. The Mobile scene is portrait and the WebGL-named scene is landscape, but the supplied Build menu selects portrait for both build targets. Select landscape explicitly in your Build Profile if desired. Test a near-startup scene replacement as well as a normal restart.

CHAPTER 05

Tutorials and every helper

Fresh local progress triggers the first-level guidance. Newly unlocked helpers have introduction and guided-use flows. Reset only a test save when reviewing tutorials. A free guided first use can differ from ordinary charge consumption. BoostName/BoostUnlock/BoostStart in StretchCatGame.cs; BoostDesc and tutorial text in StretchCatGame.Hud.cs.

Helper	Unlock	Free charges	Action and effect
Undo	3	3	Rewinds one full swipe, restoring the body route. Needs a previous swipe.
Hint	6	2	Highlights the next swipe on a remaining solvable route.
Yarn Drop	10	2	Arm the tool, then tap a valid empty tile to add a yarn stop.
Peek	15	1	Displays the remaining solution route briefly.

Activate the tool and perform its required target action: arming a tool alone does not demonstrate its effect. Undo needs an eligible previous move, Scoop needs sand, Spin needs a nonsymmetric comb and Drip needs a multi-cell comb. Exhausted charges may offer a rewarded ad; verify success/cancel/failure and grant a reward once only.

CHAPTER 06

Visual level editor: worked editing session

Open **Szekipapa77 > Stretch Cat > Level Editor**. Exact implementation: `Editor/StretchCatLevelEditor.cs`.

Choose Dup to copy a level. Set Name, Room theme, First-time hint, Design difficulty, Breather and Shape tag. Columns/Rows change the room size. Paint floor, furniture, start, yarn, turn rugs and paired boxes. Exactly one start is required; each box letter must occur exactly twice. Right-click paints floor; the rug brush rotates a rug. Use Show solution route, Validate and Save. Regenerate room replaces this room; Clear to floor removes obstacles; Trim removes empty edge rows/columns.

1. Back up Resources/stretchcat/Levels/levels.json.
2. Duplicate a simple early level and edit one small element.
3. Use the game-specific validation controls and resolve every reported issue.
4. Save explicitly; stop and restart Play mode, then use Play this level or navigate to the edited index.
5. Check portrait and landscape and rebuild demo outputs after accepting data changes.

The exact current authoring controls, including tooltips and generator switches, follow. These are game-specific source references, not a generic editor feature list.

Source	Exact current code
`StretchCatLevelEditor.cs:45`	`[MenuItem("Szekipapa77/Stretch Cat/Level Editor")]`
`StretchCatLevelEditor.cs:48`	`[MenuItem("Szekipapa77/Stretch Cat/Validate All Levels")]`
`StretchCatLevelEditor.cs:115`	`if (GUILayout.Button(Dirty ? "Save *" : "Save", EditorStyles.toolbarButton, GUILayout.Width(70))) Save();`
`StretchCatLevelEditor.cs:116`	`if (GUILayout.Button("Reload", EditorStyles.toolbarButton, GUILayout.Width(60)) && (!Dirty || EditorUtility.DisplayDialog("Reload", "Discard unsaved changes?", "Discard", "Cancel"))) Load();`
`StretchCatLevelEditor.cs:119`	`if (GUILayout.Button("Validate", EditorStyles.toolbarButton, GUILayout.Width(70)) && Cur != null) ValidateOne(_sel);`
`StretchCatLevelEditor.cs:120`	`if (GUILayout.Button("Validate All", EditorStyles.toolbarButton, GUILayout.Width(85))) ValidateAll();`

StretchCatLevelEditor.cs:123	`if (GUILayout.Button(new GUIContent("Play this level", "Optional quick test: saves, opens the WebGL scene at this level and enters Play mode. Editing never needs it."), EditorStyles.toolbarButton, GUILayout.Width(100)) && Cur != null) Play(_sel + 1);`
StretchCatLevelEditor.cs:151	`if (GUILayout.Button("Add"))`
StretchCatLevelEditor.cs:156	`if (GUILayout.Button("Dup") && Cur != null) { var c = Cur.Clone(); c.name += " copy"; Insert(_sel + 1, c); }`
StretchCatLevelEditor.cs:157	`if (GUILayout.Button("Del") && Cur != null && EditorUtility.DisplayDialog("Delete", $"Delete level {_sel + 1} '{Cur.name}'?", "Delete", "Cancel"))`
StretchCatLevelEditor.cs:161	`if (GUILayout.Button("Up") && _sel > 0) Move(-1);`
StretchCatLevelEditor.cs:162	`if (GUILayout.Button("Down") && _sel < _file.levels.Length - 1) Move(1);`
StretchCatLevelEditor.cs:192	`EditorGUILayout.LabelField($"Level {_sel + 1}", EditorStyles.boldLabel);`
StretchCatLevelEditor.cs:194	`l.name = EditorGUILayout.TextField("Name", l.name);`
StretchCatLevelEditor.cs:196	`l.room = ScGen.Rooms[EditorGUILayout.Popup("Room theme", ri, ScGen.Rooms, GUILayout.Width(300))];`
StretchCatLevelEditor.cs:198	`l.hint = EditorGUILayout.TextField(new GUIContent("First-time hint", "Shown on the level card (introduce new things here)"), l.hint);`
StretchCatLevelEditor.cs:200	`l.difficulty = EditorGUILayout.Slider("Design difficulty", l.difficulty, 0, 1);`
StretchCatLevelEditor.cs:201	`l.breather = EditorGUILayout.ToggleLeft("Breather", l.breather, GUILayout.Width(80));`
StretchCatLevelEditor.cs:203	`l.shape = ScGen.Shapes[EditorGUILayout.Popup("Shape tag", si, ScGen.Shapes, GUILayout.Width(260))];`

StretchCatLevelEditor.cs:208	`int cols = EditorGUILayout.IntSlider("Columns", l.w, 3, ScBoard.MaxSide);`
StretchCatLevelEditor.cs:209	`int rows = EditorGUILayout.IntSlider("Rows", l.h, 3, ScBoard.MaxSide);`
StretchCatLevelEditor.cs:213	`if (GUILayout.Button(new GUIContent("Regenerate room", "Rebuild this level with ScGen from its size, shape tag, features and difficulty (new seed)"), GUILayout.Width(130))) { Regenerate(l); Changed(); }`
StretchCatLevelEditor.cs:214	`if (GUILayout.Button(new GUIContent("Clear to floor", "Every cell becomes floor; the start stays"), GUILayout.Width(100)))`
StretchCatLevelEditor.cs:219	`if (GUILayout.Button(new GUIContent("Trim", "Crop empty void rows/columns"), GUILayout.Width(50))) { Trim(l); Changed(); }`
StretchCatLevelEditor.cs:220	`_showRoute = EditorGUILayout.ToggleLeft("Show solution route", _showRoute, GUILayout.Width(150));`
StretchCatLevelEditor.cs:225	`EditorGUILayout.LabelField($"Tiles {l.FloorCells()} Furniture {l.Count('#')} Yarn {l.Count('Y')} Rugs {l.CountRugs()} Box cells {l.CountBoxes()} Start {(l.Count('S') == 1 ? "ok" : "MISSING / duplicate")}", EditorStyles.miniBoldLabel);`
StretchCatLevelEditor.cs:227	`EditorGUILayout.HelpBox("Solver: " + v + (v.Solvable && v.Solutions > 3 ? "\nMany solutions: fine for early or breather levels, otherwise add furniture." : ""),`
StretchCatLevelEditor.cs:247	`EditorGUILayout.LabelField("Left-click/drag paints, right-click makes plain floor. Clicking a rug with the rug brush turns it. A box letter must appear exactly twice.", EditorStyles.miniLabel);`
StretchCatLevelEditor.cs:365	`EditorGUILayout.LabelField("Difficulty curve (solver score; blue = breather, red = unsolvable). Click a bar to select.", EditorStyles.miniBoldLabel);`
StretchCatLevelEditor.cs:387	`_showGen = EditorGUILayout.Foldout(_showGen, "Auto Generate (seeded, solver-validated, same ScGen code as the runtime)", true);`

`StretchCatLevelEditor.cs:390`	`_genCount = Mathf.Clamp(EditorGUILayout.IntField("Count", _genCount), 1, 100);`
`StretchCatLevelEditor.cs:391`	`_genStart = Mathf.Max(1, EditorGUILayout.IntField("Start at level #", _genStart));`
`StretchCatLevelEditor.cs:392`	`_genSeed = EditorGUILayout.IntField("Seed", _genSeed);`
`StretchCatLevelEditor.cs:395`	`_genFrom = EditorGUILayout.Slider("Start difficulty", _genFrom, 0, 1);`
`StretchCatLevelEditor.cs:396`	`_genTo = EditorGUILayout.Slider("End difficulty", _genTo, 0, 1);`
`StretchCatLevelEditor.cs:399`	`_genBreather = EditorGUILayout.IntSlider("Breather every", _genBreather, 0, 8);`
`StretchCatLevelEditor.cs:400`	`_genReplace = EditorGUILayout.ToggleLeft("Replace that range (else insert)", _genReplace);`
`StretchCatLevelEditor.cs:403`	`_genYarn = EditorGUILayout.ToggleLeft("Yarn", _genYarn, GUILayout.Width(70));`
`StretchCatLevelEditor.cs:404`	`_genRugs = EditorGUILayout.ToggleLeft("Turn rugs", _genRugs, GUILayout.Width(90));`
`StretchCatLevelEditor.cs:405`	`_genBoxes = EditorGUILayout.ToggleLeft("Boxes", _genBoxes, GUILayout.Width(70));`
`StretchCatLevelEditor.cs:416`	`if (GUILayout.Button($"Generate {_genCount} level(s)", GUILayout.Height(26))) Generate();`

CHAPTER 07

Generator, insertion and replacement

Count 1–100; Start at level #; Seed; Start/End difficulty; Breather every 0–8; Yarn, Turn rugs and Boxes switches. Replace that range (else insert) overwrites the target range when checked; otherwise generated records are inserted and later levels shift. Start at 26 to append to the initial 25. Generation uses ScGen and solver validation.

Worked extension: back up the bank, generate five candidates after the original 25 at moderate difficulty, validate them, then Save. Reopen the JSON and confirm the count changed. An unchecked replace option inserts rather than silently destroying later data. Runtime continuation does not automatically append new records to project JSON. Validation and bot scores help find problems but do not certify a pleasant human difficulty curve.

CHAPTER 08

Actual level JSON and identifiers

Canonical asset: `Assets/Games/StretchCat/Resources/stretchcat/Levels/levels.json`. Array order controls visible numbering. Preserve numeric/boolean types, root version fields where present and resource/save identifiers. The actual first record is shown below; use the editor for structured coordinates and block/word data.


```
{
  "name": "First Stretch",
  "room": "Living Room",
  "shape": "rect",
  "hint": "Swipe to stretch Mochi. She slides until she bumps into something. Cover every tile!",
  "w": 4,
  "h": 3,
  "rows": [
    " S..",
    "....",
    "...."
  ],
  "breather": false,
  "difficulty": 0.02,
  "seed": 11007,
  "solutions": 1,
  "moves": 5,
  "states": 13,
  "score": 9.192416,
  "randomWin": 0.5
}
```


Root keys: levels. Inspect ScCore.cs, ScGen.cs, ScLevels.cs for field semantics and validation. Keep visible names/hints distinct from technical keys. For Word Hunt, words, masks and placements are gameplay data; interface translation is a separate task.

CHAPTER 09

Reskin recipe: interface colours

Stop Play mode and open `Scripts/ScArt.cs`. Find `Ginger`, currently `#F29A4A`. Change that hex literal to `#42C9B0`, preserve its symbol/type, compile and reopen the scene. Review HUD, menus, helper cards, settings/results and both orientations. Art sprites and procedural fallback shapes can have additional colours; edit them deliberately.

Actual consumers of this symbol are listed first, then all current palette/hex assignments. Editor-only palettes do not control the runtime. Preserve state/goal contrast and rebuild after accepting a change.

Source	Exact current code
`StretchCatGame.cs:243 / void ShowLevelCard()`	`var p = ScPopup(LevelLabel(), msg, Ginger, LevelCardArt(), D.name + " - " + D.room, ("Play", Teal, (Action)Begin));`
`StretchCatGame.cs:421 / void Stuck()`	`("Restart room", Ginger, () => Restart()));`
`StretchCatGame.Hud.cs:103 / void BuildHudPortrait(float W, float H)`	`var pill = CardAt("LevelPill", 40 + 260, SafeTop + 70, 520, 132, Ginger, 48);`
`StretchCatGame.Hud.cs:137 / void BuildHudLandscape(float W, float H)`	`var title = HzUI.Label("Title", _layer, "Stretch Cat", 82, Ginger, TextAnchor.MiddleLeft);`
`StretchCatGame.Hud.cs:143 / void BuildHudLandscape(float W, float H)`	`var pill = CardAt("LevelPill", px, 285, pw, 132, Ginger, 48);`
`StretchCatGame.Hud.cs:159 / void BuildHudLandscape(float W, float H)`	`var rb = HzUI.Button("Restart", _layer, "", Ginger, Color.white, () => { if (CanPlay || _playing) Restart(); }, 40, 36);`
`StretchCatGame.Hud.cs:275 / void IconRow(float rightX, float y, float size)`	`var restart = HzIcons.IconButton(_layer, HzIcons.Replay, Ginger, () => { if (_playing) Restart(); }, size);`
`StretchCatGame.Hud.cs:507 / HzPopup ScPopupFull(string title, string message, Color accent, Sprite art, string subtitle, int stars, (string label, Color color, Action action)[] buttons)`	`var paw = HzIcons.Icon(card.transform, PawIcon, 70, new Color(Ginger.r, Ginger.g, Ginger.b, 0.45f));`
`StretchCatGame.Hud.cs:583 / HzPopup ScPopupFull(string title, string message, Color accent, Sprite art, string subtitle, int stars, (string label, Color color, Action action)[] buttons)`	`var st = HzUI.Label("Sub", card.transform, subtitle, 40, GingerDark); st.rectTransform.Place(new Vector2(0.5f, 1), new Vector2(cw - 80, 50), new Vector2(0, y - 25));`

StretchCatGame.Hud.cs:693 / void ShowSettingsPopup()	("Restart Room", Ginger, () => Restart()),
StretchCatGame.View.cs:157 / void BuildBoard(RoomTheme theme)	var l = HzUI.Panel("Link", _bodyLayer, Ginger, 0); l.raycastTarget = false; l.gameObject.SetActive(false); _link[i] = l;
StretchCatGame.View.cs:164 / void BuildBoard(RoomTheme theme)	var s = HzUI.Panel("Seg", _bodyLayer, Ginger, rad); s.raycastTarget = false;
StretchCatGame.View.cs:166 / void BuildBoard(RoomTheme theme)	var st = HzUI.Panel("Stripe", s.transform, GingerDark, Mathf.Max(2, (int)(_cs * 0.05f))); st.raycastTarget = false;
StretchCatGame.View.cs:173 / void BuildBoard(RoomTheme theme)	_tailImg = tailSp ? HzUI.Img("Tail", _bodyLayer, tailSp, Color.white) : HzUI.Panel("Tail", _bodyLayer, GingerLight, Mathf.Max(4, (int)(_cs * 0.3f)));
StretchCatGame.View.cs:180 / void BuildBoard(RoomTheme theme)	_headImg = HzUI.Panel("Head", _bodyLayer, GingerLight, Mathf.Max(4, (int)(_cs * 0.45f)));
StretchCatGame.View.cs:231 / void ShowBodyAt(int k)	s.color = (k & 1) == 0 ? Ginger : GingerLight;

Source	Exact current code
ScArt.cs:9 / class/static declarations	public static readonly Color Cream = HzUI.Hex("#FFF4E2"), Ginger = HzUI.Hex("#F29A4A"), GingerLight = HzUI.Hex("#F8B873"), GingerDark = HzUI.Hex("#D9772C"),
ScArt.cs:10 / class/static declarations	Cocoa = HzUI.Hex("#4B2E2A"), Teal = HzUI.Hex("#3BB7B0"), Berry = HzUI.Hex("#E2557A"), Mustard = HzUI.Hex("#F7C548"), Lilac = HzUI.Hex("#9C7BD8"),
ScArt.cs:11 / class/static declarations	Sky = HzUI.Hex("#4FA3D9"), Leaf = HzUI.Hex("#5DBB63"), CardBg = HzUI.Hex("#FFF8EC"), CardInk = HzUI.Hex("#5A3A2E"), Sub = HzUI.Hex("#9A7B68"),
ScArt.cs:12 / class/static declarations	Locked = HzUI.Hex("#B9A796"), PopupCard = HzUI.Hex("#FFF6E8"), PopupInner = HzUI.Hex("#FDEBD3");

ScArt.cs:14 / class/static declarations	public static readonly string[] BoxColors = { "#3BB7B0", "#E2557A", "#9C7BD8", "#F7C548", "#4FA3D9", "#5DBB63", "#F29A4A", "#4B2E2A" };
ScArt.cs:23 / class/static declarations	new RoomTheme { Name = "Living Room", FloorA = HzUI.Hex("#FBE3C4"), FloorB = HzUI.Hex("#F6D3AA"), Rim = HzUI.Hex("#8A5A3C"), Tint = new Color(1f, 0.85f, 0.65f, 0.10f), BlockBase = HzUI.Hex("#D9B48C") },
ScArt.cs:24 / class/static declarations	new RoomTheme { Name = "Kitchen", FloorA = HzUI.Hex("#E3F4EC"), FloorB = HzUI.Hex("#CBEAD9"), Rim = HzUI.Hex("#3F7A6A"), Tint = new Color(0.6f, 1f, 0.85f, 0.12f), BlockBase = HzUI.Hex("#A9D2C2") },
ScArt.cs:25 / class/static declarations	new RoomTheme { Name = "Garden", FloorA = HzUI.Hex("#DDF0B8"), FloorB = HzUI.Hex("#C9E59A"), Rim = HzUI.Hex("#5B7A33"), Tint = new Color(0.7f, 1f, 0.5f, 0.14f), BlockBase = HzUI.Hex("#A8C87A") },
ScArt.cs:26 / class/static declarations	new RoomTheme { Name = "Library", FloorA = HzUI.Hex("#EADDF6"), FloorB = HzUI.Hex("#DCC9EE"), Rim = HzUI.Hex("#5E4682"), Tint = new Color(0.75f, 0.6f, 1f, 0.14f), BlockBase = HzUI.Hex("#BCA6D8") },
ScArt.cs:27 / class/static declarations	new RoomTheme { Name = "Attic", FloorA = HzUI.Hex("#D7E3F5"), FloorB = HzUI.Hex("#C3D3EE"), Rim = HzUI.Hex("#34446E"), Tint = new Color(0.25f, 0.3f, 0.65f, 0.32f), BlockBase = HzUI.Hex("#9FB0D6") },
StretchCatGame.cs:63 / class/static declarations	public static readonly string[] BoostColor = { "#3BB7B0", "#F2A93B", "#9C7BD8", "#4FA3D9" };
StretchCatGame.Hud.cs:445 / void RefreshBoosters()	v.Ring.color = sel || _guide == b ? HzUI.Hex("#FFD23F") : new Color(1, 1, 1, 0);
StretchCatGame.View.cs:61 / void BuildView()	var g = HzUI.Img("Bg", _layer, HzUI.Gradient(HzUI.Hex("#FBE3C4"), HzUI.Hex("#E9B98A")), Color.white); g.preserveAspect = false; g.raycastTarget = false;
StretchCatGame.View.cs:144 / void BuildBoard(RoomTheme theme)	Image img = sp ? HzUI.Img("Box", ring.transform, sp, Color.white) : HzUI.Panel("Box", ring.transform, HzUI.Hex("#C99A62"), Mathf.Max(4, (int)(_cs * 0.14f)));

CHAPTER 10

Reskin recipe: visible menu and HUD text

Open `Scripts/StretchCatGame.Hud.cs`, find the displayed literal `"Stretch Cat"`, and replace its displayed value with `"Cozy Cat"`. Keep hierarchy names and technical IDs unchanged. In HzUI.Label/Button calls the first string may be an object name; change the displayed text argument. Read the method references below to distinguish construction from later refresh assignments.

Preserve interpolation braces, invariant number formatting, ON/OFF updates and line breaks. Update both initial and refresh paths, portrait and landscape. BoostName/BoostUnlock/BoostStart in StretchCatGame.cs; BoostDesc and tutorial text in StretchCatGame.Hud.cs. Level names/hints remain separate JSON content. If a title appears in art, Title property or another screen, update those visible uses deliberately without renaming GameId, namespace, resources or save keys.

Hard-coded game strings do not automatically use shared localization. To translate the interface, route every display assignment through your translation table. Dictionaries, word banks and theme pools are independent gameplay assets and need separate validation. Test help, settings, results, tutorial and helper introductions after a fresh launch.

Source	Exact current code
`ScArt.cs:23 / class/static declarations`	`new RoomTheme { Name = "Living Room", FloorA = HzUI.Hex("#FBE3C4"), FloorB = HzUI.Hex("#F6D3AA"), Rim = HzUI.Hex("#8A5A3C"), Tint = new Color(1f, 0.85f, 0.65f, 0.10f), BlockBase = HzUI.Hex("#D9B48C") },`
`ScArt.cs:24 / class/static declarations`	`new RoomTheme { Name = "Kitchen", FloorA = HzUI.Hex("#E3F4EC"), FloorB = HzUI.Hex("#CBEAD9"), Rim = HzUI.Hex("#3F7A6A"), Tint = new Color(0.6f, 1f, 0.85f, 0.12f), BlockBase = HzUI.Hex("#A9D2C2") },`
`ScArt.cs:25 / class/static declarations`	`new RoomTheme { Name = "Garden", FloorA = HzUI.Hex("#DDF0B8"), FloorB = HzUI.Hex("#C9E59A"), Rim = HzUI.Hex("#5B7A33"), Tint = new Color(0.7f, 1f, 0.5f, 0.14f), BlockBase = HzUI.Hex("#A8C87A") },`
`ScArt.cs:26 / class/static declarations`	`new RoomTheme { Name = "Library", FloorA = HzUI.Hex("#EADDF6"), FloorB = HzUI.Hex("#DCC9EE"), Rim = HzUI.Hex("#5E4682"), Tint = new Color(0.75f, 0.6f, 1f, 0.14f), BlockBase = HzUI.Hex("#BCA6D8") },`
`ScArt.cs:27 / class/static declarations`	`new RoomTheme { Name = "Attic", FloorA = HzUI.Hex("#D7E3F5"), FloorB = HzUI.Hex("#C3D3EE"), Rim = HzUI.Hex("#34446E"), Tint = new Color(0.25f, 0.3f, 0.65f, 0.32f), BlockBase = HzUI.Hex("#9FB0D6") },`

`ScCore.cs:91 / public int FloorCells() { int n = 0; foreach (var s in rows) foreach (var x in s) if (ScBoard.IsOpenChar(x)) n++; return n; }`	`public static readonly string[] Name = { "up", "right", "down", "left" };`
`ScGen.cs:124 / static float Clamp01(float v) => v < 0 ? 0 : v > 1 ? 1 : v;`	`p.Name = NameA[rng.Range(0, NameA.Length)] + " " + NameB[rng.Range(0, NameB.Length)];`
`StretchCatGame.cs:57 / class/static declarations`	`public static readonly string[] BoostName = { "Undo", "Hint", "Yarn Drop", "Peek" };`
`StretchCatGame.cs:243 / void ShowLevelCard()`	`var p = ScPopup(LevelLabel(), msg, Ginger, LevelCardArt(), D.name + " - " + D.room, ("Play", Teal, (Action)Begin));`
`StretchCatGame.cs:251 / string LevelLabel() => ScLevels.IsGenerated(_playLevel) ? "Room " + _playLevel.ToString(CultureInfo.InvariantCulture) : "Level " + _playLevel.ToString(CultureInfo.InvariantCulture);`	`string LevelLabel() => ScLevels.IsGenerated(_playLevel) ? "Room " + _playLevel.ToString(CultureInfo.InvariantCulture) : "Level " + _playLevel.ToString(CultureInfo.InvariantCulture);`
`StretchCatGame.cs:387 / void Stuck()`	`Toast("Tangled! Let's try that again", Cream, 1.2f, ToastY);`
`StretchCatGame.cs:409 / void Stuck()`	`ScPopup("Tangled!", $"Mochi can't move and {Left} tile{(Left == 1 ? " is" : "s are")} still empty.", Berry, ScArt.Get("cat_head"), null,`
`StretchCatGame.cs:645 / void OnBooster(int b)`	`Toast(BoostName[b] + " unlocks at level " + BoostUnlock[b], null, 1.3f, ToastY); HzAudio.Play(Sfx.Cancel, 0.4f);`
`StretchCatGame.cs:665 / void OnBooster(int b)`	`if (_moveStart.Count == 0) { Toast("Nothing to undo yet", null, 1.1f, ToastY); HzAudio.Play(Sfx.Cancel, 0.4f); return; }`
`StretchCatGame.cs:681 / void OnBooster(int b)`	`else Toast("Tap an empty tile to drop a yarn ball", null, 1.4f, ToastY);`
`StretchCatGame.cs:733 / void OfferCharge(int b)`	`ScPopup("+1 " + BoostName[b], "Out of " + BoostName[b] + ".\nWatch a short ad for one more?", HzUI.Hex(BoostColor[b]), ScArt.Get(BoostArt[b]), null,`
`StretchCatGame.cs:797 / void UseHint()`	`if (!r.Solvable || r.Route == null || r.Route.Count == 0) { Toast("Try restarting this room", null, 1.4f, ToastY); return; }`

`StretchCatGame.cs:809 / void UseHint()`	`Toast(rewound == 1 ? "Rewound one swipe - now:" : $"Rewound {rewound} swipes - now:", Mustard, 1.4f, ToastY);`
`StretchCatGame.cs:849 / void UsePeek()`	`if (!r.Solvable) { Toast("Mochi is tangled - Undo or restart first", Berry, 1.6f, ToastY); HzAudio.Play(Sfx.Cancel, 0.4f); return; }`
`StretchCatGame.cs:871 / void DropYarn(int cell)`	`Toast("Pick an empty plain tile", null, 1.1f, ToastY);`
`StretchCatGame.cs:1131 / string PopupTitle()`	`return t ? t.GetComponent<Text>().text.Replace(' ', '_') : "popup";`
`StretchCatGame.Hud.cs:105 / void BuildHudPortrait(float W, float H)`	`_levelTxt = HzUI.Label("Level", pill.transform, "", 58, Color.white); _levelTxt.AddOutline(new Color(0.4f, 0.15f, 0, 0.35f), 3);`
`StretchCatGame.Hud.cs:109 / void BuildHudPortrait(float W, float H)`	`_nameTxt = HzUI.Label("Name", pill.transform, "", 32, new Color(1, 1, 1, 0.95f), TextAnchor.MiddleCenter, false); _nameTxt.fontStyle = FontStyle.Bold;`
`StretchCatGame.Hud.cs:137 / void BuildHudLandscape(float W, float H)`	`var title = HzUI.Label("Title", _layer, "Stretch Cat", 82, Ginger, TextAnchor.MiddleLeft);`
`StretchCatGame.Hud.cs:145 / void BuildHudLandscape(float W, float H)`	`_levelTxt = HzUI.Label("Level", pill.transform, "", 58, Color.white); _levelTxt.AddOutline(new Color(0.4f, 0.15f, 0, 0.35f), 3);`
`StretchCatGame.Hud.cs:149 / void BuildHudLandscape(float W, float H)`	`_nameTxt = HzUI.Label("Name", pill.transform, "", 32, new Color(1, 1, 1, 0.95f), TextAnchor.MiddleCenter, false); _nameTxt.fontStyle = FontStyle.Bold;`
`StretchCatGame.Hud.cs:159 / void BuildHudLandscape(float W, float H)`	`var rb = HzUI.Button("Restart", _layer, "", Ginger, Color.white, () => { if (CanPlay || _playing) Restart(); }, 40, 36);`
`StretchCatGame.Hud.cs:163 / void BuildHudLandscape(float W, float H)`	`ButtonIconLabel(rb, HzIcons.Replay, "Restart");`
`StretchCatGame.Hud.cs:165 / void BuildHudLandscape(float W, float H)`	`var sb = HzUI.Button("Settings", _layer, "", Lilac, Color.white, ShowSettingsPopup, 40, 36);`
`StretchCatGame.Hud.cs:169 / void BuildHudLandscape(float W, float H)`	`ButtonIconLabel(sb, HzIcons.Settings, "Settings");`

`StretchCatGame.Hud.cs:171 / void BuildHudLandscape(float W, float H)`	`var help = HzUI.Label("Help", _layer, "Swipe, drag or use the arrow keys", 32, CardInk, TextAnchor.MiddleCenter, false);`
`StretchCatGame.Hud.cs:187 / void ButtonIconLabel(Button b, string icon, string label)`	`var t = HzUI.Label("L", b.transform, label, 40, Color.white); t.rectTransform.Stretch(); t.rectTransform.offsetMin = new Vector2(80, 0);`
`StretchCatGame.Hud.cs:219 / void ProgressCard(Image card, float w, float h)`	`_leftTxt = HzUI.Label("Left", card.transform, "", tall ? 46 : 42, CardInk, TextAnchor.MiddleLeft);`
`StretchCatGame.Hud.cs:223 / void ProgressCard(Image card, float w, float h)`	`_swipesTxt = HzUI.Label("Swipes", card.transform, "", 30, Sub, TextAnchor.MiddleLeft, false); _swipesTxt.fontStyle = FontStyle.Bold;`
`StretchCatGame.Hud.cs:319 / void BoosterBar(float cx, float cy, float w, float h, bool horizontal)`	`var btn = HzUI.Button("Booster" + b, parent, null, Color.white, Color.white, () => OnBooster(bb), 40, (int)(size * 0.32f));`
`StretchCatGame.Hud.cs:365 / void BoosterBar(float cx, float cy, float w, float h, bool horizontal)`	`v.BadgeText = HzUI.Label("N", bd.transform, "", (int)(size * 0.24f), Color.white); v.BadgeText.rectTransform.Stretch();`
`StretchCatGame.Hud.cs:369 / void BoosterBar(float cx, float cy, float w, float h, bool horizontal)`	`v.Caption = HzUI.Label("Cap", v.Root, "", 30, CardInk);`
`StretchCatGame.Hud.cs:389 / void RefreshHud()`	`_nameTxt.text = D.name + " - " + D.room + (D.breather ? " - breather" : "");`
`StretchCatGame.Hud.cs:393 / void RefreshHud()`	`_leftTxt.text = left == 0 ? "All covered!" : left.ToString(CultureInfo.InvariantCulture) + (left == 1 ? " tile left" : " tiles left");`
`StretchCatGame.Hud.cs:395 / void RefreshHud()`	`if (_swipesTxt) _swipesTxt.text = "Swipes: " + _moveDir.Count.ToString(CultureInfo.InvariantCulture);`
`StretchCatGame.Hud.cs:433 / void RefreshBoosters()`	`v.BadgeText.text = c > 0 ? c.ToString(CultureInfo.InvariantCulture) : "+";`
`StretchCatGame.Hud.cs:437 / void RefreshBoosters()`	`v.Caption.text = un ? BoostName[b] : "Level " + BoostUnlock[b].ToString(CultureInfo.InvariantCulture);`

StretchCatGame.Hud.cs:573 / HzPopup ScPopupFull(string title, string message, Color accent, Sprite art, string subtitle, int stars, (string label, Color color, Action action)[] buttons)	var t = HzUI.Label("Title", ribbon.transform, title, 72, Color.white); t.rectTransform.Stretch(6);
StretchCatGame.Hud.cs:583 / HzPopup ScPopupFull(string title, string message, Color accent, Sprite art, string subtitle, int stars, (string label, Color color, Action action)[] buttons)	var st = HzUI.Label("Sub", card.transform, subtitle, 40, GingerDark); st.rectTransform.Place(new Vector2(0.5f, 1), new Vector2(cw - 80, 50), new Vector2(0, y - 25));
StretchCatGame.Hud.cs:593 / HzPopup ScPopupFull(string title, string message, Color accent, Sprite art, string subtitle, int stars, (string label, Color color, Action action)[] buttons)	p.Message = HzUI.Label("Msg", card.transform, message, 44, CardInk, TextAnchor.MiddleCenter, false);
StretchCatGame.Hud.cs:669 / void UnlockPopup(int b, Action done)	ScPopup("New: " + BoostName[b] + "!", BoostDesc[b] + "\nYou get " + BoostStart[b].ToString(CultureInfo.InvariantCulture) + " free!", HzUI.Hex(BoostColor[b]), ScArt.Get(BoostArt[b]), null,
StretchCatGame.Hud.cs:685 / void ShowSettingsPopup()	var p = ScPopup("Settings", "", Lilac, null, null,
StretchCatGame.Hud.cs:691 / void ShowSettingsPopup()	("How to Play", Mustard, () => HelpPage(0)),
StretchCatGame.Hud.cs:713 / static string SoundLabel() => HzAudio.Muted ? "Sound: OFF" : "Sound: ON";	static string SoundLabel() => HzAudio.Muted ? "Sound: OFF" : "Sound: ON";
StretchCatGame.Hud.cs:715 / static string MusicLabel() => HzAudio.MusicMuted ? "Music: OFF" : "Music: ON";	static string MusicLabel() => HzAudio.MusicMuted ? "Music: OFF" : "Music: ON";
StretchCatGame.Hud.cs:741 / void HelpPage(int i)	ScPopup($"How to Play {i + 1}/{HelpPages.Length}", HelpPages[i], Sky, i == 0 ? ScArt.Get("cat_head") : null, null,
StretchCatGame.Hud.cs:743 / void HelpPage(int i)	(last ? "Got it" : "Next", Teal, last ? (Action)null : () => HelpPage(i + 1)));
StretchCatGame.Hud.cs:883 / void ShowTip(string text)	_tipText = HzUI.Label("T", tip.transform, text, IsPortrait ? 38 : 36, Cream, TextAnchor.MiddleCenter, false);

`StretchCatGame.Hud.cs:1029 / void CompleteGuide(int b)`	`Toast("Nice! " + BoostName[b] + " is yours", Mustard, 1.3f, ToastY);`

CHAPTER 11

Artwork, fonts, audio and licences

Replace game images under Resources/stretchcat through Unity while retaining resource keys, .meta GUIDs, Sprite/UI import settings, alpha, pivots and slicing. Procedural graphics in the art/view sources may need separate colour changes. Replacing one PNG does not change every fallback. Keep third-party font and other licence notices with the corresponding shipped assets.

Music is Resources/stretchcat/Music/cozy_loop.ogg. Preserve its load key when replacing it, check loop boundaries/compression and test Sound and Music toggles separately. Kit UI uses supplied fonts; validate glyph coverage for new languages and retain the actual font licence files.

CHAPTER 12

Local saves, reset and migration

HzSave wraps PlayerPrefs and flushes local state on focus loss/quit. WebGL persistence uses browser storage tied to the hosting origin. Clearing site data or changing origin starts a new local save; this is not cloud synchronization. Inspect HzSave/PlayerPrefs calls in the actual game source before altering progression, stars, helper charges or tutorial flags.

Verify fresh launch, a completed puzzle, helper use, sound options, focus loss and restart. Use a dedicated test profile for resets. Keep technical save keys stable during colour/text reskins. If changing schema or keys, implement and test a migration rather than silently discarding existing users' progress.

CHAPTER 13

Advertising integration and configuration

Asset: `Assets/Szekipapa77Ads/Resources/AdsSettings.asset`. Inspect backend, useGoogleTestIds and missingSdkFallback in the publisher-owned Ads configuration. Buyer projects include adapter/integration code and exclude external provider SDK binaries. WebGL and a missing-provider fallback simulate ads; these are not live commercial impressions.

For real Android ads install the supported provider SDK from its official source into your working project, enter your own app/placement IDs and use test ads first. Verify banner/interstitial/rewarded callbacks on supported hardware. For production use your own IDs and disable simulated missing-SDK fallback. Build tooling rejects unsafe production fallback settings. No IAP, remote backend or HLK is included in this publisher edition.

CHAPTER 14

Android build and release

Install Unity 6000.0.62f1 Android modules through Hub. Select Android, run Setup and review application ID `com.szekipapa77.stretchcat`, version, scene orientation and signing. The build command writes `Builds/Android/stretchcat.apk`. Confirm BuildResult.Succeeded and the actual output file; intermediate report totals are not the final APK size.

Use your own keystore/signing and distribution settings. A test-signed demo is not a signed store release. A provider-free build simulates ads; a provider test-demo does not prove production ad-account configuration. Test real hardware touch, safe areas, focus/audio, save/relaunch and reward callbacks. Keep passwords, keystores and private service credentials out of buyer packages.

CHAPTER 15

WebGL build and hosting

Install matching WebGL support. Run Setup and Build WebGL; output is `Builds/WebGL/stretchcat/`. Serve index.html with all loader/framework/data/WASM/StreamingAssets files over HTTP(S), not file://. Match compression, Content-Encoding and MIME types to the host. Outer page sizing and actual scene orientation are different settings.

The publisher's playable demo is hosted on HyperCodeStore. Upload the actual runtime files rather than merely changing a URL. Test a fresh anonymous load, real pointer controls, audio after interaction, saved progress/reload and phone/desktop sizing. Rebuild and rehost when code/assets change. A successful HTTP page alone is not gameplay verification.

CHAPTER 16

Troubleshooting

Compile errors: restore matching packages and the full private Kit/Ads layer; only one game Kit belongs in a project. **Pink/invisible rendering:** run Setup and inspect Graphics and every Quality URP assignment. **No input:** classic input handling must be enabled; check canvas focus and blocking modals. **Missing resources:** restore exact paths, .meta GUIDs and importer settings.

Edits do not appear: stop Play, Save in the level editor, reload the canonical bank and check current saved level. **Text reverts:** update construction and refresh assignments from the exact table. **No live ads:** provider binaries are deliberately excluded; inspect your installed provider and fallback mode. **Scene reload exception:** retain owner-lifetime and generation guards on delayed callbacks. **WebGL fails:** inspect browser Console, compression headers and missing payloads.

CHAPTER 17

Support and final release checklist

Publisher: Szekipapa77. Support: szekipapa77@gmail.com. Include game title, exact Unity version, platform, first Console error and reproduction steps. Never send passwords/tokens.

1. Clean-import the standalone project and native package.
2. Play both scenes and replace a scene near startup.
3. Review tutorial, normal completion/retry and every helper activation plus effect.
4. Validate editor changes and test reskin text/colours in both layouts.
5. Rebuild affected Android/WebGL outputs.
6. Verify complete public APK bytes, actual browser demo, documentation and video links.
7. Keep licences and exclude all internal prompts, planning/capture/build/QA tooling from buyers.