

Can't Catch The Pig

A multiplayer iPhone chase game — one farmer and a herd of very slippery pigs. What I've been building.

Nick May · Status as of September 2026

In one paragraph

Can't Catch The Pig is an asymmetric party game for iOS built in Swift with SpriteKit and SwiftUI. One side plays farmers, the other plays pigs, and the two sides deliberately play like different games: the farmer sees the whole barnyard but can only catch with a committed dive, while pigs only see what's near them and rely on dashes, fence gaps and a haptic heartbeat that speeds up as the farmer closes in. Phase 1 (solo with bots) and Phase 2 (local multiplayer) are done and playable.

Game modes

Mode	Setup	How you win
Herd	1 farmer vs 2–4 pigs	Farmer pens every pig before 2:30 runs out; pigs can break friends out of the pen
Hunt	2–4 farmers vs 1 pig	Farmers tag the pig; the pig survives the 2:00 clock

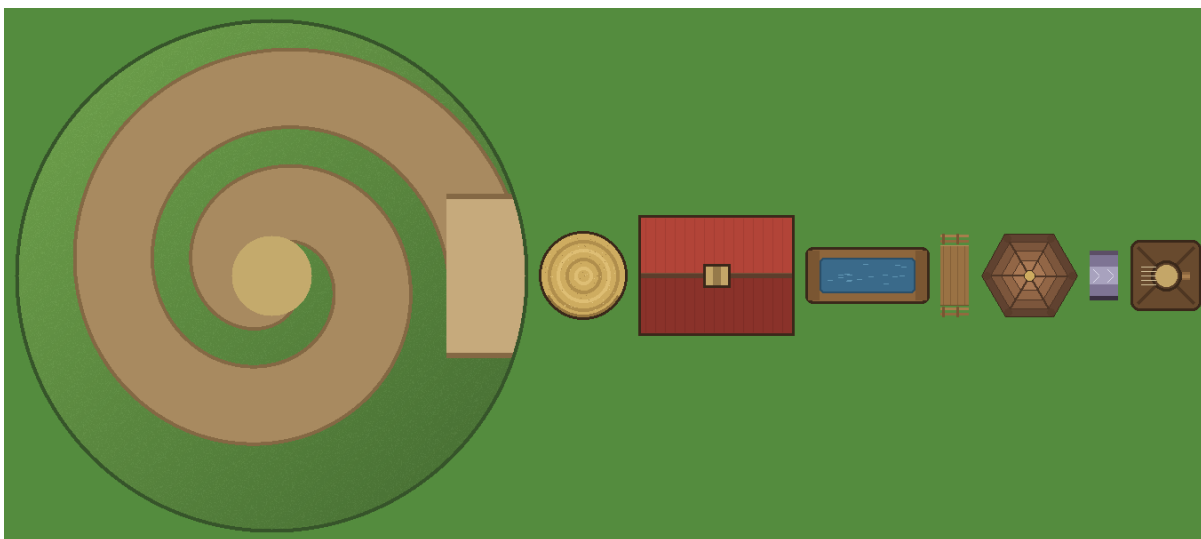


The full 1600×1000 arena, rendered by my own layout tool: spiral hill, barn, haystacks, mud, dirt roads, the pen (top right), a pond with bridges and a gazebo, a slide and a catapult.

What's built so far

- **Core gameplay.** Dive-vs-dash combat, greased pigs that slip the first grab, whiff penalties, pen + jailbreak latch, color-coded zones with callouts, and a prey-side minimap.

- **Local multiplayer.** Up to 5 players over MultipeerConnectivity (Wi-Fi/Bluetooth, no server or accounts). Host-authoritative simulation with 20 Hz snapshots, a lobby with roles and ready-up, rematches, name tags, and bots that take over dropped players.
- **Bots.** Pig and farmer AI so every seat can be filled and the game is fully playable alone on one device.
- **A designed map.** Every feature is tuned for the asymmetry: pig-only gaps vs shared gates, mud that slows only the farmer, roads that speed only the farmer, a dead-end pocket for cornering, a pond with over/under bridges, a gazebo that hides you from view and radar, a climbable spiral hill, plus a slide and catapult either side can use.
- **Feel.** Proximity heartbeat haptics, a floating joystick and action button, and a camera that shows the whole board in landscape and follows in portrait.
- **All original art, generated in code.** Python scripts draw every sprite, tile, prop and the app icon — 340 atlas frames, characters in 16 directions, no licensed assets.



Procedurally generated props: spiral hill, haystack, barn, trough, bridge, gazebo, slide and catapult.

By the numbers

~5,800	lines of Swift across 22 source files
~1,350	lines of Python art and layout tooling (5 generators)
340	sprite and tile frames (pig 208, farmer 112, tiles 20)
16	facing directions per character, drawn rather than rotated
5	max players in a local multiplayer match

Tech stack

Swift · SwiftUI (menus, lobby, results) · SpriteKit (game scene, physics, tile maps) · MultipeerConnectivity (networking) · Core Haptics · Python + Pillow (art pipeline) · Xcode

What's next

Phase	Plan
-------	------

Phase 3	Traps (mud digger, slop bucket, decoy chicken), fake dive, dinner bell, a bull with a host toggle, and a mutator deck to vary rounds
Phase 4	Results screen path replay, automatic highlights, rivalry stats, sound effects and polished sprites
Near-term fixes	Add sound early (the splat and squeal matter), make bot pigs use fence gaps tactically, and playtest the dive/dash balance on device