



# Level 2 Agent Experience Map for Teen Learners

Professor Unicorn's Unicorn Warehouse | Working map after Keep Your Brain

**Purpose.** Rank agent experiences by their likely surprise and appeal, then connect each experience to a transferable skill and the tools needed to teach it. This order is a hypothesis for pilot testing. It is not a mandatory route through the program.

**Starting point.** Keep Your Brain supplies the habits of questioning, verification, privacy, consent, and ownership. Level 2 adds an agent that can inspect information and use tools to change or test something. The learner should be able to distinguish what the agent proposed, what it attempted, and what actually happened.

**Teaching rule.** A learner may enter through games, media, science, research, coding, or a personal interest. The shared skills recur across those doors. Tool access expands only when it helps the chosen project and its result can be checked.

## Eight transferable agent skills

**Frame.** Describe the goal, constraints, and what finished work must demonstrate.

**Inspect.** Understand the starting materials and current state before changing anything.

**Direct.** Keep human decisions with the learner and assign the agent a clear role.

**Access.** Choose the information and tools the task requires, with clear limits.

**Task.** Keep one current task record and work in reviewable pieces.

**Check.** Match claims of completion to files, sources, tests, previews, or other evidence.

**Diagnose.** Investigate mismatches and revise a focused part of the work.

**Own.** Explain the result, remaining limits, and what the learner decided.

These names are shorthand for the practices in Agent Use. Some projects exercise all eight; others emphasize a few. A skill is demonstrated through a new task, not by reciting its label.

## Ranked experience overview

| # | Kid interest hook             | Agent reveal                                                                            |
|---|-------------------------------|-----------------------------------------------------------------------------------------|
| 1 | Build the game with me        | The agent changes a real project, runs it, and shows the result.                        |
| 2 | Build my own specialist agent | A learner gives a system a chosen job and can watch tools turn its decisions into work. |



| #  | Kid interest hook                   | Agent reveal                                                                                          |
|----|-------------------------------------|-------------------------------------------------------------------------------------------------------|
| 3  | Make my app real                    | The agent can work inside the actual application instead of handing back isolated code snippets.      |
| 4  | Run my experiment                   | The agent can process real measurements, graph them, and rerun an analysis when the question changes. |
| 5  | Investigate a real question         | An agent follows leads across sources and assembles a packet the learner can verify.                  |
| 6  | Turn my notes into a study machine  | The agent uses a selected set of materials to create practice that adapts to actual mistakes.         |
| 7  | Build an actual website             | The agent turns a concept into pages the learner can visit and revise.                                |
| 8  | Find the story in the data          | The agent can make a large table legible without pretending the chart explains everything.            |
| 9  | Modify a game I already like        | The agent can understand existing project structure before changing a rule or asset.                  |
| 10 | Run my creator workflow             | The agent organizes tedious production work so the learner can spend time making creative choices.    |
| 11 | Make a physical thing respond       | An agent helps write and test code that responds to a sensor or simulated robot.                      |
| 12 | Build a project team around my idea | Several narrow workers can produce separate pieces that the learner combines and judges.              |

## Experience designs

### 1 Build the game with me

**The reveal.** The agent changes a real project, runs it, and shows the result.

**Try it.** Start with a tiny playable game. The learner specifies one mechanic; the agent inspects the project, edits a copy, runs a focused check, and presents the diff and playable result.

**Tools.** Scoped folder read and write; local runner; diff; test output.

**Shared skills.** Frame, Inspect, Direct, Access, Task, Check, Diagnose, Own.

**Proof of learning.** The learner plays the change, finds a mismatch, and can explain the rule and evidence for the repair.

**Boundary.** Prepared local project; no live accounts or unreviewed downloads.



## 2 Build my own specialist agent

**The reveal.** A learner gives a system a chosen job and can watch tools turn its decisions into work.

**Try it.** Choose a narrow role such as lore keeper, study quizzier, art critic, or sports data investigator. Pick its information, two tools, approval points, and a test set; run three realistic tasks and revise the design.

**Tools.** Agent configuration; curated information set; two bounded tools; action record; repeatable test cases.

**Shared skills.** Frame, Inspect, Direct, Access, Check, Diagnose, Own.

**Proof of learning.** The learner demonstrates both a task it handles well and one it refuses or escalates, then explains why.

**Boundary.** This is a capstone by skill readiness, even though it ranks high for surprise. It does not require training a model.

## 3 Make my app real

**The reveal.** The agent can work inside the actual application instead of handing back isolated code snippets.

**Try it.** Give it a small existing app and a feature request such as a filter or score tracker. The learner writes acceptance checks; the agent inspects, changes, runs, and shows the working feature.

**Tools.** Project files; local development server; focused tests; browser preview; diff.

**Shared skills.** Frame, Inspect, Direct, Access, Task, Check, Diagnose, Own.

**Proof of learning.** The learner tests both a normal path and a failure case and can identify what files changed.

**Boundary.** Local test data only; no secrets, payment flows, or production deployment.

## 4 Run my experiment

**The reveal.** The agent can process real measurements, graph them, and rerun an analysis when the question changes.

**Try it.** Use an age appropriate provided dataset or learner collected observations. The learner names a question and units; the agent cleans a copy, computes a result, makes a chart, and saves the method.

**Tools.** Scoped data files; calculator or code runner; chart creation; saved analysis steps.

**Shared skills.** Frame, Inspect, Direct, Access, Task, Check, Diagnose, Own.

**Proof of learning.** The learner checks one calculation, identifies a suspicious data point, and states what the graph does and does not show.

**Boundary.** Avoid sensitive participant data; supervised rules for physical experiments.



## 5 Investigate a real question

**The reveal.** An agent follows leads across sources and assembles a packet the learner can verify.

**Try it.** Pick a low stakes question with competing answers. The learner defines the claim; the agent searches, opens sources, records excerpts and links, and groups supported and unresolved points.

**Tools.** Search; page read; source capture; research folder with provenance.

**Shared skills.** Frame, Inspect, Direct, Access, Task, Check, Own.

**Proof of learning.** The learner opens the strongest source, finds a claim it does not support, and revises the conclusion.

**Boundary.** Public sources only; outside pages are evidence to assess, not instructions to obey.

## 6 Turn my notes into a study machine

**The reveal.** The agent uses a selected set of materials to create practice that adapts to actual mistakes.

**Try it.** Provide permitted lesson notes. The learner selects a topic; the agent builds practice questions, records missed concepts in a small local progress file, and generates a second round that tests understanding.

**Tools.** Read only course folder; quiz generator; scoped progress file; comparison of attempts.

**Shared skills.** Frame, Inspect, Direct, Access, Task, Check, Diagnose, Own.

**Proof of learning.** The learner answers without the notes, explains a corrected mistake, and can challenge a bad question.

**Boundary.** Permission to share course materials matters; no grades, student records, or hidden school accounts.

## 7 Build an actual website

**The reveal.** The agent turns a concept into pages the learner can visit and revise.

**Try it.** Choose a fan site for an original fictional world or a club demo. The learner sketches the pages and audience; the agent creates a local site, previews it, and checks one small screen and one large screen.

**Tools.** Scoped project files; local server; browser preview; screenshot; diff.

**Shared skills.** Frame, Inspect, Direct, Access, Task, Check, Diagnose, Own.

**Proof of learning.** The learner navigates the site, catches a broken link or layout problem, and explains one design decision.

**Boundary.** Preview locally; publication and public data collection require a separate decision.



## 8 Find the story in the data

**The reveal.** The agent can make a large table legible without pretending the chart explains everything.

**Try it.** Use a public sports, weather, or game statistics table. The learner asks one comparison; the agent inspects columns, calculates, charts, and flags missing data or outliers.

**Tools.** CSV or spreadsheet read; calculator or code runner; chart; saved method.

**Shared skills.** Frame, Inspect, Direct, Access, Check, Diagnose, Own.

**Proof of learning.** The learner checks the chart scale and a sampled value, then explains a plausible alternative reading.

**Boundary.** Use public or fictional data; no personal profiles or private datasets.

## 9 Modify a game I already like

**The reveal.** The agent can understand existing project structure before changing a rule or asset.

**Try it.** Use a small open source or learner owned game project with modding permitted. The learner chooses one alteration; the agent locates the relevant files, explains its proposed edit, changes a copy, and tests it.

**Tools.** Project search; scoped file edits; local runner; diff; version snapshot.

**Shared skills.** Frame, Inspect, Direct, Access, Task, Check, Diagnose, Own.

**Proof of learning.** The learner identifies the original behavior, demonstrates the changed behavior, and can reverse the change.

**Boundary.** Respect game licenses and community rules; avoid untrusted mod installers.

## 10 Run my creator workflow

**The reveal.** The agent organizes tedious production work so the learner can spend time making creative choices.

**Try it.** Provide a small folder of original video or art assets. Have the agent inventory it, produce a shot list and captions draft, and prepare a proposed rename plan for review.

**Tools.** Scoped asset folder; metadata reader; draft document editor; preview and diff.

**Shared skills.** Frame, Inspect, Direct, Access, Task, Check, Diagnose, Own.

**Proof of learning.** The learner rejects a poor shot order or caption, approves a useful file organization, and checks the final files.

**Boundary.** Use copies of original or permitted assets; get consent before handling another person's likeness.



## 11 Make a physical thing respond

**The reveal.** An agent helps write and test code that responds to a sensor or simulated robot.

**Try it.** Begin with a simulator or low power educational kit. The learner defines an input and expected output; the agent edits the control code and checks the behavior in simulation before supervised hardware use.

**Tools.** Simulator; scoped code folder; local runner; device connection only with adult supervision.

**Shared skills.** Frame, Inspect, Direct, Access, Task, Check, Diagnose, Own.

**Proof of learning.** The learner predicts a response, tests a failure case, and identifies the difference between simulated and physical results.

**Boundary.** A specialized hardware branch; physical motion, power, heat, and connectivity need appropriate supervision.

## 12 Build a project team around my idea

**The reveal.** Several narrow workers can produce separate pieces that the learner combines and judges.

**Try it.** Give research, creation, and review roles to separate agents on an already understood project. Define who owns each artifact; compare their outputs, resolve a disagreement, and assemble one final result.

**Tools.** Multiple scoped agents or sequential roles; separate work folders; shared spec; comparison record.

**Shared skills.** Frame, Direct, Access, Task, Check, Diagnose, Own.

**Proof of learning.** The learner catches a conflict between outputs and can justify the final integration using evidence.

**Boundary.** Optional advanced branch; more agents add coordination cost and do not automatically improve the result.

# How to choose the teaching order

**Entry.** Offer several hooks and let the learner choose. A high surprise rank does not mean a beginner should start there. The specialist agent at rank 2 works as a capstone after simpler tool experiences.

**Progress.** Move from reading a small curated workspace to changing a copy, then running or checking the result. Research, data, media, and code tools branch by project. Publishing, shared accounts, purchases, and physical devices require their own decisions.

**Pilot evidence.** For each experience record the learner's prediction, the task specification, tool actions, one observed result, a learner correction, and whether the learner can apply the same skill in a different area. Change the ranking when actual learners choose differently.

**Optional branches.** Building a specialist agent, using several agents, and controlling physical hardware can be compelling projects. They are not prerequisites for every learner. Outside documents and web pages also introduce prompt injection; teach that skill when those materials enter the work.

## Source map

Agent Use: suggestion, action, result; scoped access; clear specs; inspection; reviewable work; tests; diffs; outside content; connected actions; ownership.



AI Literacy Teaching Map and Keep Your Brain materials: prediction, comparison, verification, privacy, cognitive ownership, transfer, and controlled workflows.

Level 1 Interest Map for Teen AI Learning: the twelve interest hooks that can continue into agent projects.