

AGENT BUILDERS

Build games with AI

A practical guide to building, changing and designing playable games with AI.

August 2026

Before you start

Mistakes are part of the process

Things will occasionally go wrong. Tell the AI what happened, what you expected and ask it to fix that one issue. You can always undo a change or return to the last working version.

What you'll build today

- **Part 1 - Breakout:** build a playable Breakout game, then add faces, reactions and a changing paddle.
- **Part 2 - Modify a game:** choose a well-known 2D game and redesign it with your own gameplay changes.
- **Part 3 - Create your own game:** plan an original idea carefully, build it and improve it without losing sight of the main concept.

How to work

- Start a new chat for each, this allows you to go back to an earlier game if you want to.
- Change one thing – or a few small things – at a time so any problems are easy to identify.
- If you use a workshop account, do not change its password, settings or account information.

GUIDE: The blue boxes are ready-to-use messages for the AI. Copy them as written unless the guide asks you to replace a word or add your own idea.

Part 1: Build and customise Breakout

You'll start with a working Breakout game, then make it your own by adding faces, animated reactions and a gameplay twist.

Step 1 - Open an AI assistant

- Open your own Claude, Microsoft Copilot or ChatGPT Plus account. If you do not have access, use one of the workshop logins on the table.
- In Copilot – select GPT 5.6 Think Deeper as the model
- In ChatGPT – Medium effort level
- In Claude – Sonnet on Medium effort
- Start a fresh chat and keep it open for the whole of Part 1.

Step 2 - Create the Breakout game

Copy this:

```
Make a fully functional clone of Breakout in HTML that I can download
```

When it opens, play briefly to make sure the controls and restart work.

Step 3 - Add faces to the blocks

Copy this:

```
Add simple black faces to every block. Give each face two eyes and a mouth, and make sure the faces stand out clearly against the block colours. Keep the rest of the game working as it already does.
```

Step 4 - Animate the faces

Copy this:

Animate the faces on the blocks. When the ball comes close to a block, make that block's face look shocked. When the ball moves away, return the face to its normal expression. Keep the faces black and keep the rest of the game working as it already does.

Play briefly and watch a few blocks react as the ball approaches.

Step 5 - Add a gameplay twist

Copy this:

Make the paddle periodically change between twice its normal size and half its normal size. Make the size changes clear and smooth, and make sure the ball still collides accurately with the paddle. Keep everything else working.

Try both paddle sizes once. If the change is too fast, ask the AI to slow it down.

Step 6 - Play the finished game

Play one complete round. If anything has stopped working, describe the problem to the AI and ask it to fix that issue without changing the rest of the game.

Part 1 complete.

Part 2: Modify a game of your choice

Start a new chat session.

Pick a 2D game you already know, build a simple working version and then change the way it plays. Use the original idea as your starting point, but create your own simple shapes and colours instead of copying its artwork.

Step 1 - Choose one game

- **Pong:** move a paddle and return a bouncing ball.
- **Snake:** collect food, grow longer and avoid crashing.
- **Flappy Bird-style game:** keep a flying character in the air and guide it through gaps.
- **Space Invaders-style game:** move a spaceship, fire at enemies and avoid their attacks.
- **Frogger-style game:** cross moving lanes while avoiding obstacles.

Step 2 - Build the basic game

Replace **[GAME]** with your chosen game:

Copy this:

Make a simple, fully functional 2D version of **[GAME]** in HTML. Use simple original shapes and colours. Include clear controls, scoring where appropriate, and a way to restart. Make it a downloadable HTML.

Step 3 - Try it

Play one round so you know the basic game works before you change it.

Step 4 - Choose a direction

Choose one change that affects how the game plays. It could alter the movement, controls, difficulty, scoring, enemies, obstacles or rules.

For example: split the Pong ball after a long rally; give different foods in Snake different effects; add wind to a flying game; or vary the enemy movement in Space Invaders.

Step 5 - Describe the modification

Adapt this prompt:

Add [feature] to the existing game. It should activate when [trigger], then [describe what happens], and end when [end condition]. Keep [controls, scoring or other existing features] unchanged. Once it works, tell me how to test it.

EXAMPLE: Add three types of food to Snake. Normal food should grow the snake by one segment, blue food should slow it down for five seconds and gold food should be worth five points. Keep the controls and collision rules unchanged. Tell me how to test each food type.

Step 6 - Try the modification

Trigger the new feature a few times. If it has broken anything, fix that before moving on.

Step 7 - Add one smaller improvement

If there is time, add one smaller improvement such as a visual effect, sound, new obstacle, power-up or clearer score display.

Step 8 - Polish the game

Copy this:

Review the current game without replacing it. Fix any bugs you find and improve the instructions, controls, visual clarity and restart system. Keep my chosen gameplay modifications. Do not add unrelated features. State what you found and changed.

Play one final round after the review.

Part 2 complete.

Part 3: Design and build your own game

Start a new Chat session.

This time, you're starting with your own idea. A short plan before you build will keep the game focused and stop it turning into a pile of unrelated features.

Step 1 - Give the AI your starting idea

Begin with a rough concept. It does not need to be fully worked out yet.

Adapt this prompt:

I want to design an original 2D game. My starting idea is: [describe the idea in one or two sentences]. Do not build it yet. Help me turn it into a clear game plan. Keep it simple enough to build and test quickly using JavaScript and HTML5. It must be a downloadable HTML file.

Step 2 – Review the plan

Send this next, then answer each question it asks.

Copy this:

Review this plan for a game that must be built during this session. Point out anything unclear, contradictory or too ambitious for a first version. Recommend no more than three changes that would make it more achievable without changing the main idea. Ask me no more than 3 clarifying questions. Do not build the game yet.

Step 3 – Build the complete first version

Send this next.

Build it

Step 4 – Iterate on it

Improve your game by asking the AI to make any updates, fix things that are broken. Be clear in describing what you want changed.

Sometimes it's best to make larger changes one by one.

If something breaks

Return to the last version that worked and ask the AI to fix or remove only the most recent change.

Copy this:

The game worked before the most recent change. Please restore the last working behaviour while keeping the features that were already working. Do not start the game again from scratch.

If you are still stuck, ask one of the Agent Builders team to take a look.