🤖 Student Quickstart: Your First Bot

From Zero to Battle-Ready

This guide will help you create **your very first BotBattles bot**.

You don’t need to understand everything yet.
You don’t need to be “good at coding.”

Your goal is simple:

Make a bot that loads into the game and does something in the arena.

Once that works, then you can improve upon it.

🧠 What Is a Bot?

In BotBattles, a bot is a **small program** that controls a robot.

Your bot can:

- Look for other bots
- Turn
- Move
- Shoot
- Make decisions on its own

Once a match starts, **you don’t control it anymore**.
Your bot follows the rules you gave it… for better or worse.

That’s the fun part.

📁 What You’ll Create

Each bot lives in its **own folder** and has **two files** inside.

```
MyFirstBot/
├─ bot.json
└─ behavior.js
```

That’s it.
No extra files. No other hidden steps.

🏷️ Step 1: Create `bot.json`

This file describes **what your bot is like**, not how it thinks.

Start by copying a sample `bot.json` provided by your teacher, or create a new file with this structure:

```
{
  "name": "MyFirstBot",
  "build": {
    "maxSpeedTier": 2,
    "turnRateTier": 3,
    "sightRangeTier": 2,
    "sightFovTier": 3,
    "shotPowerTier": 2,
    "shotSpeedTier": 2,
    "maxHealthTier": 3,
    "memoryTier": 0,
    "wallBehavior": "stop"
  }
}
```

What to change right now

- Change `"name"` to something unique

What *not* to change yet

- Leave the `build` section exactly as it is
  (We'll tune it later.)

Why this matters:
Starting simple makes it much easier to tell if your bot works.

🧠 Step 2: Create `behavior.js`

This file controls **how your bot thinks**.

Create a file called `behavior.js` and put this inside:

```
function tick(api) {
  api.advance(0.5);
}
```

That’s a complete, valid bot brain.

What this does

- Every moment, your bot moves forward a little
- It doesn’t look, turn, or shoot yet - and that’s okay

If your bot moves, **you’ve succeeded**.

▶️ Step 3: Load Your Bot Into the Game

1. Open BotBattles in your browser
2. Use the **Select Bot Folders** tool
3. Choose your bot’s folder (`MyFirstBot/`)
    - NOTE: Don’t go ***into*** that folder when loading, just select the folder itself
4. Check the **Bot Load Report**
5. Click **Start Battle**

If your bot appears and moves, celebrate 🎉

❗ If Something Goes Wrong

This happens to everyone. Seriously.

If your bot doesn’t load

- Check that:
  - The folder has **both** `bot.json` and `behavior.js`
  - The files are named correctly
  - You saved your changes

If you see an error message

- Read it carefully - the game tries to explain what went wrong
- Fix **one thing at a time**
- Reload bots and try again

Good news:
 If you break something, the game will usually fall back to the last working version instead of crashing.

👀 Step 4: Make Your Bot Better

Your bot is capable of more. Some pretty sophisticated behavior is possible with your bot’s **API**. What’s that? API is short for **Application** **Programming** **Interface**. It’s a fancy word for what code commands are available to make things happen.

Our BotBattles API lets your bots do the following actions:

- `api.advance()` - you did this earlier, it moves your bot forward
- `api.scan()` - this let’s your bot “see” in front of itself
- `api.turn(someNumber)` - this rotates your bot
- `api.fire()` - this shoots a laser straight ahead
- And more…

Follow up by reading the Student API Cheatsheet for more info regarding the BattleBots Bot API.

🧪 One Golden Rule

Change one thing at a time.

If something breaks, you'll know *why*.

A good order for experimenting:

1. Movement
2. Turning
3. Scanning
4. Shooting
5. Build values (you'll get to this later)

🧠 What You Can Ignore (For Now)

You do **not** need to worry about:

- Scoring formulas
- Advanced strategies
- Memory slots
- Perfect accuracy
- Winning matches

Right now, success means:

> "My bot works, and I understand why it did that."

✅ You're Officially In

If your bot:

- Loads without errors
- Appears in the arena
- Moves or reacts at all

Then you have built your **first BotBattles bot**.

Next steps:

- Learn new commands in the **Bot API Cheatsheet** (we'll update it soon)
- Watch other bots and borrow ideas
- Improve your design slowly

You're not here to be perfect.
You're here to **think, test, and iterate**.