MAGICODE
An Educational Game to Teach Programming Fundamentals

Samuel Levy

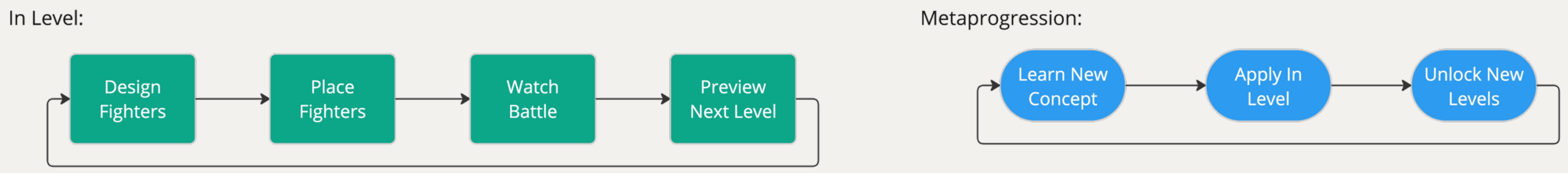
Context

- Games are a valuable tool to **simplify complicated concepts**.
- **Coding games** either focus on abstract concepts or literal language usage.
- Use of simplified **coding tools** is driven by intrinsic motivation.
- **Goal:** Teach programming fundamentals in a fun, simplified way that leaves the player with transferrable skills.

Prior Work

- **Coding Games:** Robocode, Human Resource Machine, Zachtronics Games
- **Coding Tools:** Scratch, Python
- **Educational Games:** Math Blaster, Typing Games, Oregon Trail
- **Game-Based Learning:** Teaching through gameplay, core game mechanics reinforce educational concepts

Game Structure



Players use a **Drag-And-Drop** coding language to program their warriors, then place their warriors into an **Autobattler** where they fight enemies automatically.

Gameplay

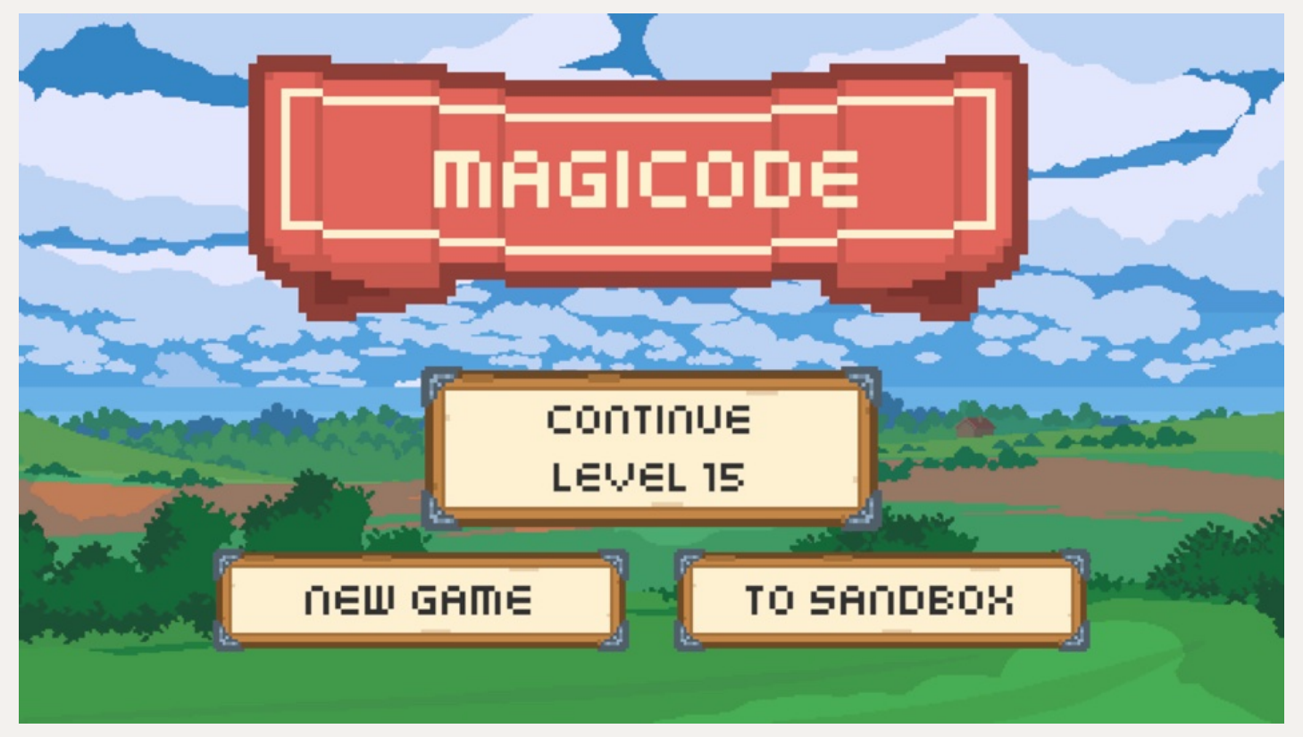


Figure 1. Main menu

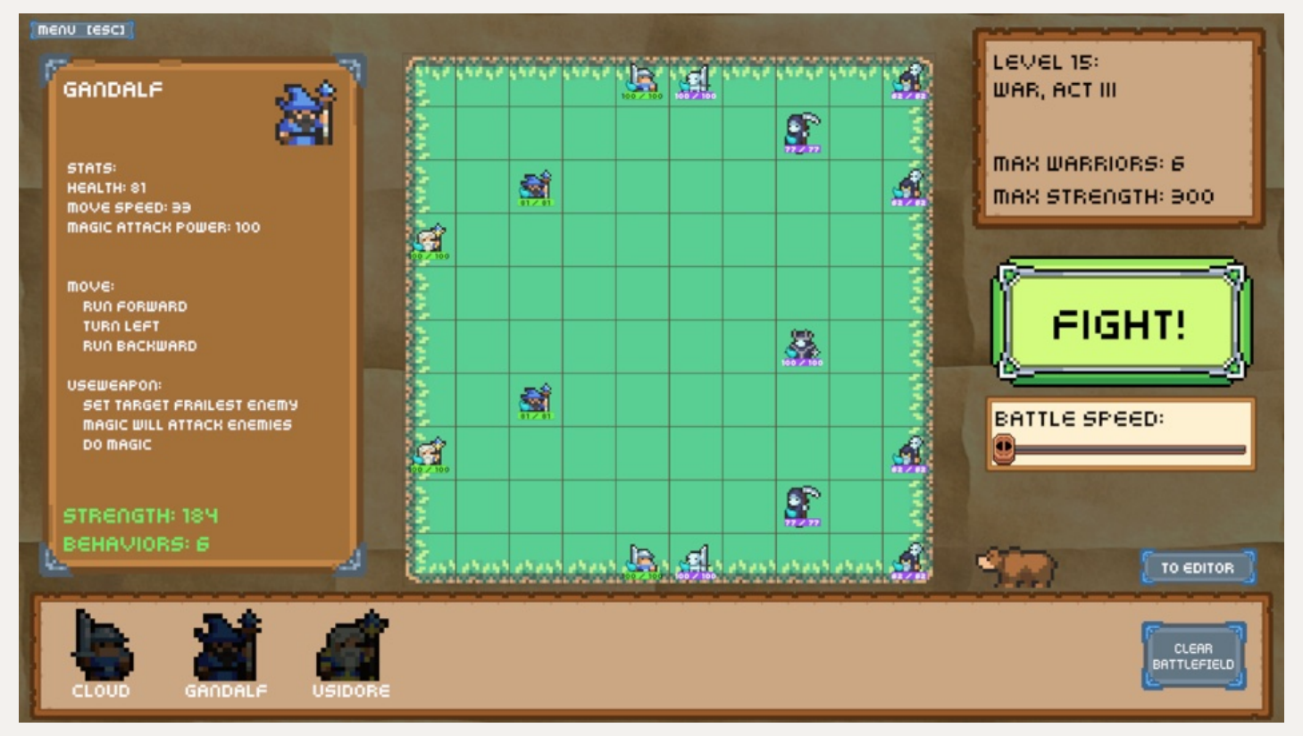


Figure 2. Level setup

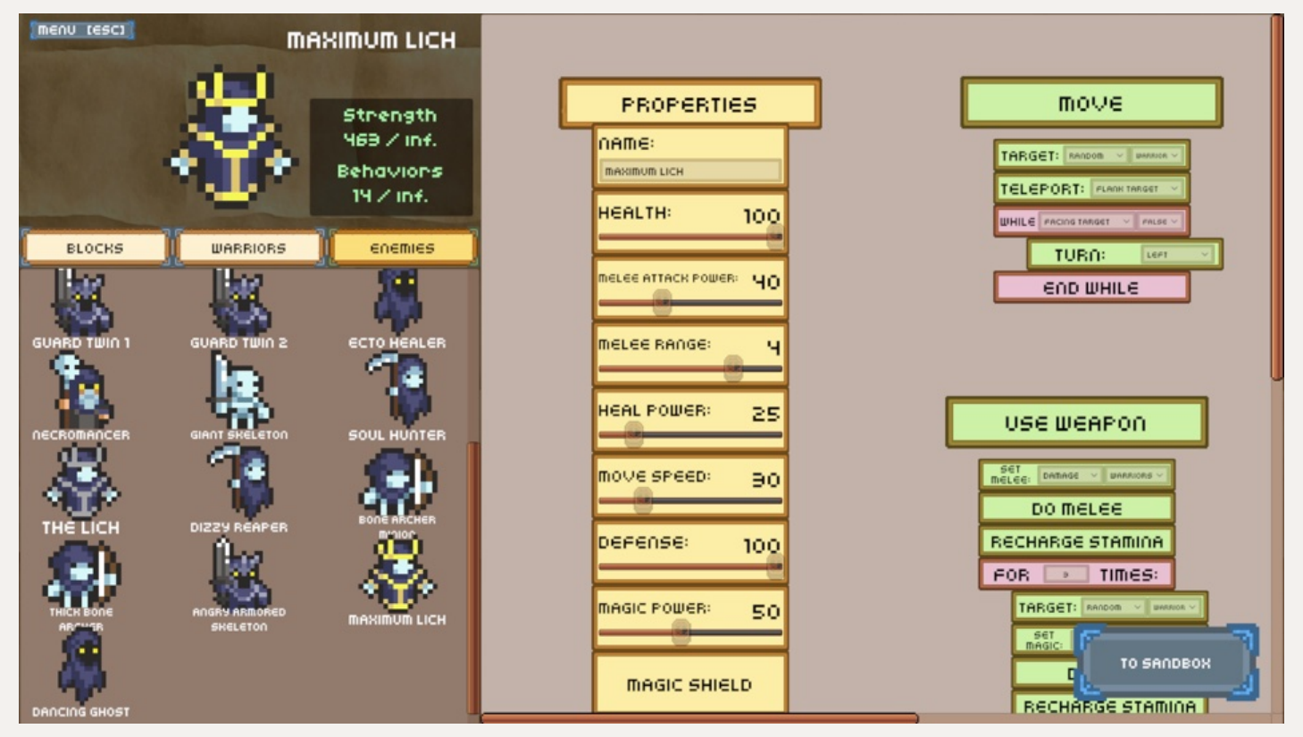


Figure 3. Code editor with an enemy

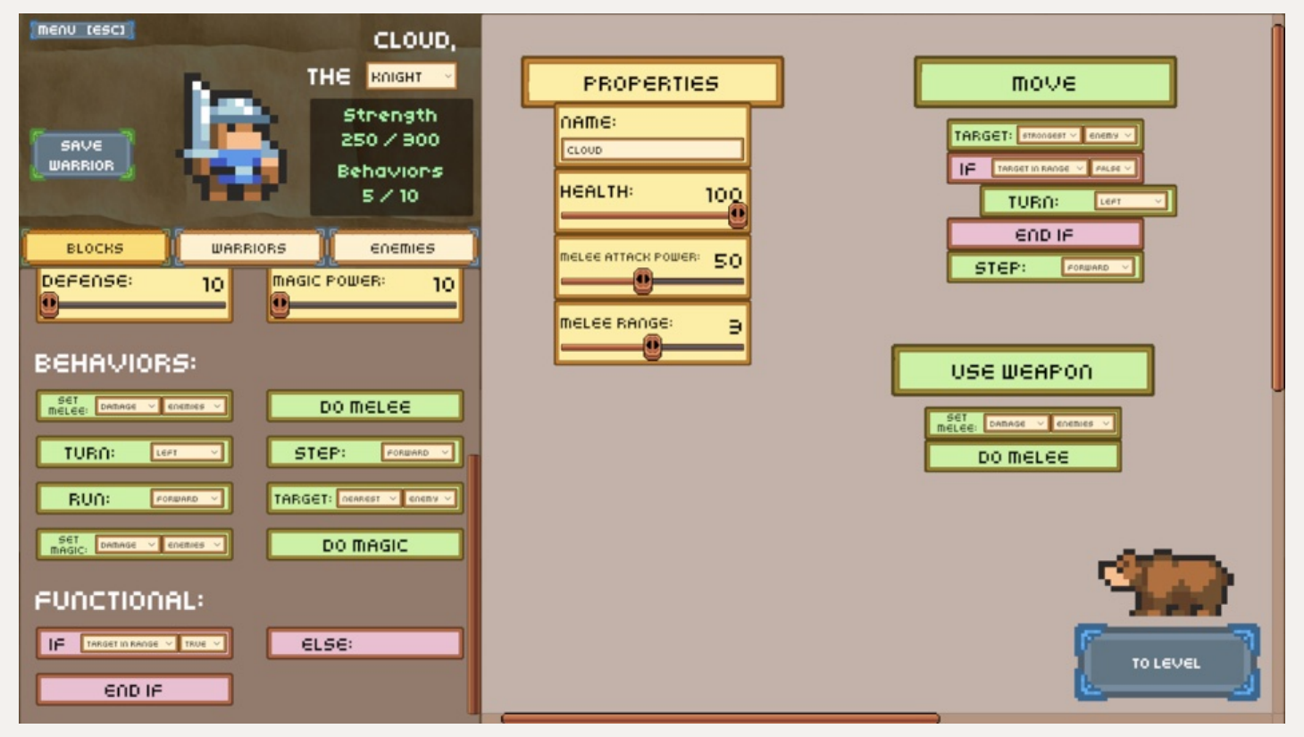


Figure 4. Code editor with a simple warrior

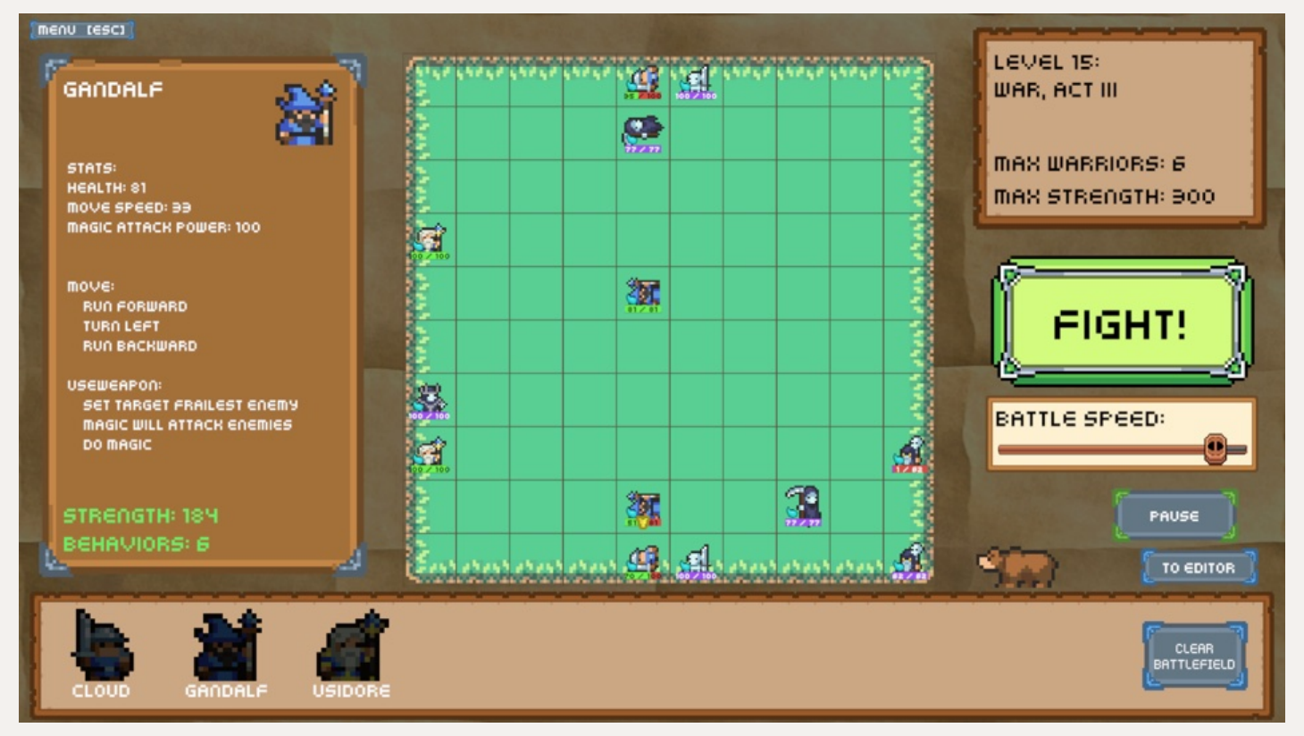


Figure 5. Level 15 with a battle in progress

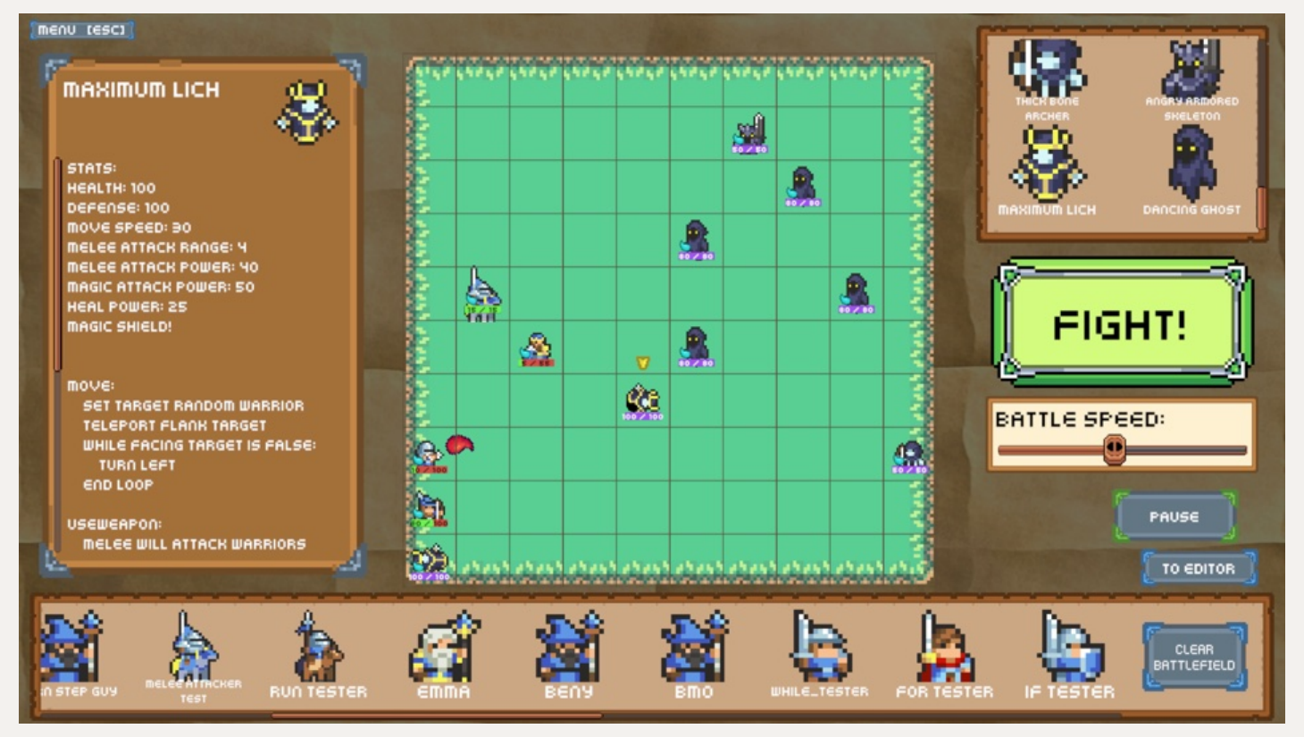


Figure 6. Sandbox with a battle in progress

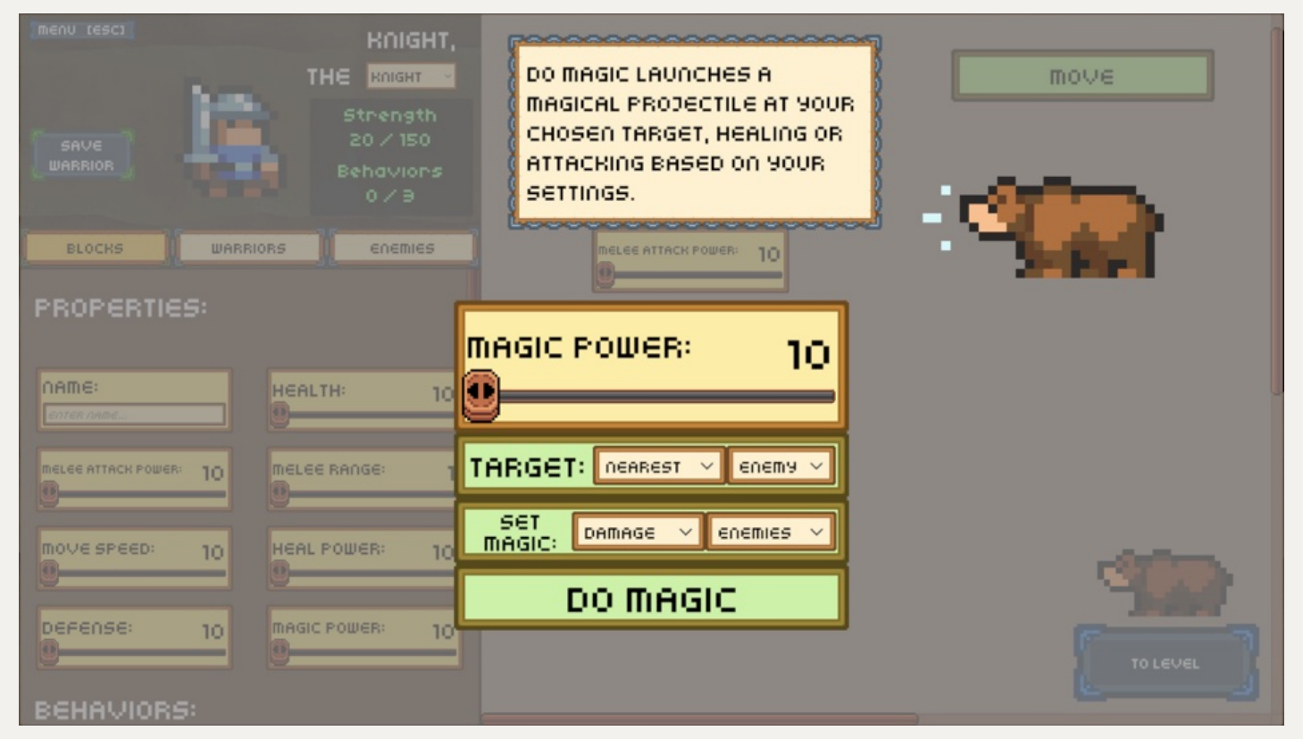


Figure 7. Tutorial in progress

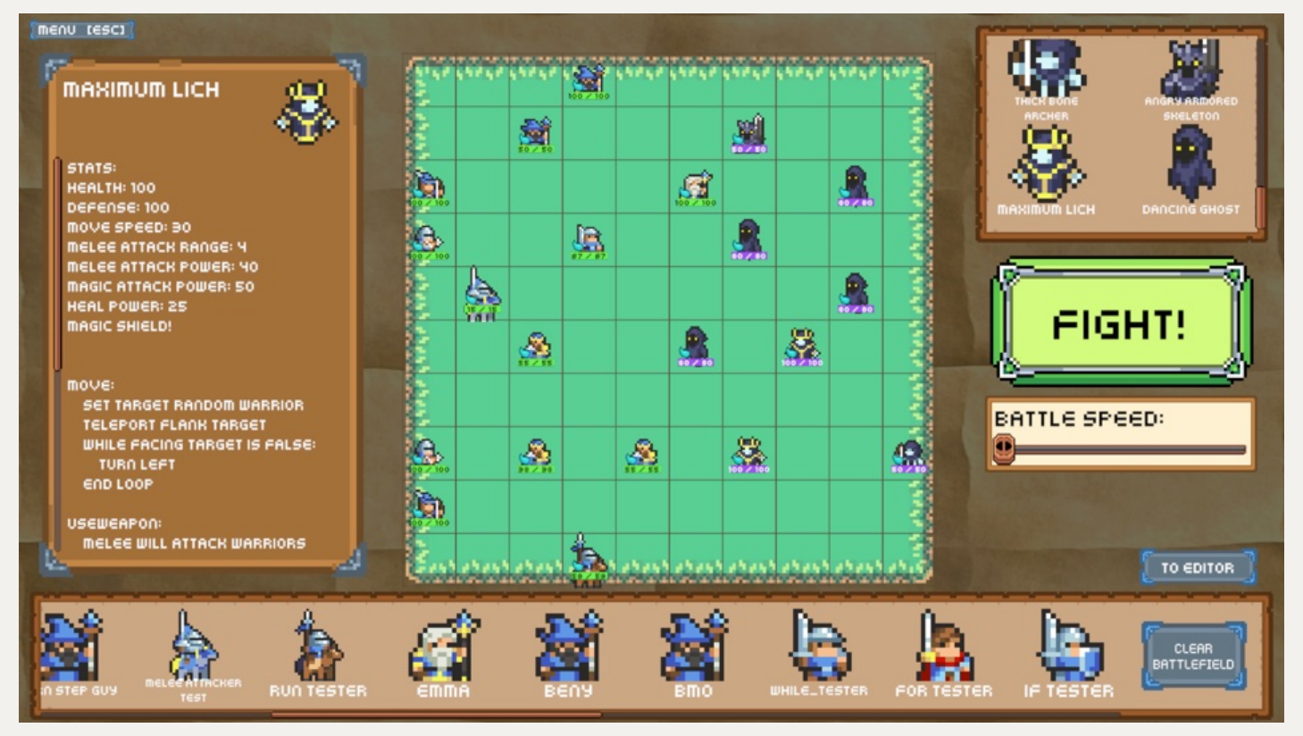


Figure 8. Setup for a sandbox battle

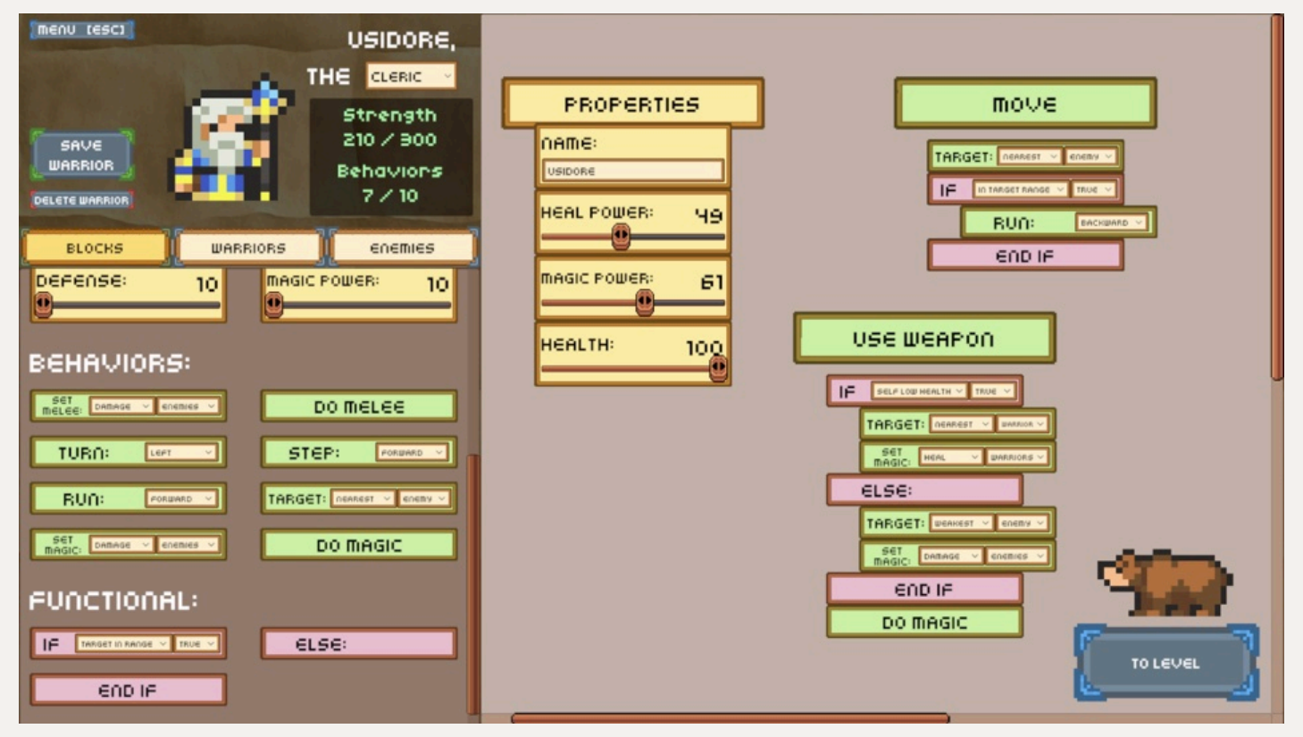
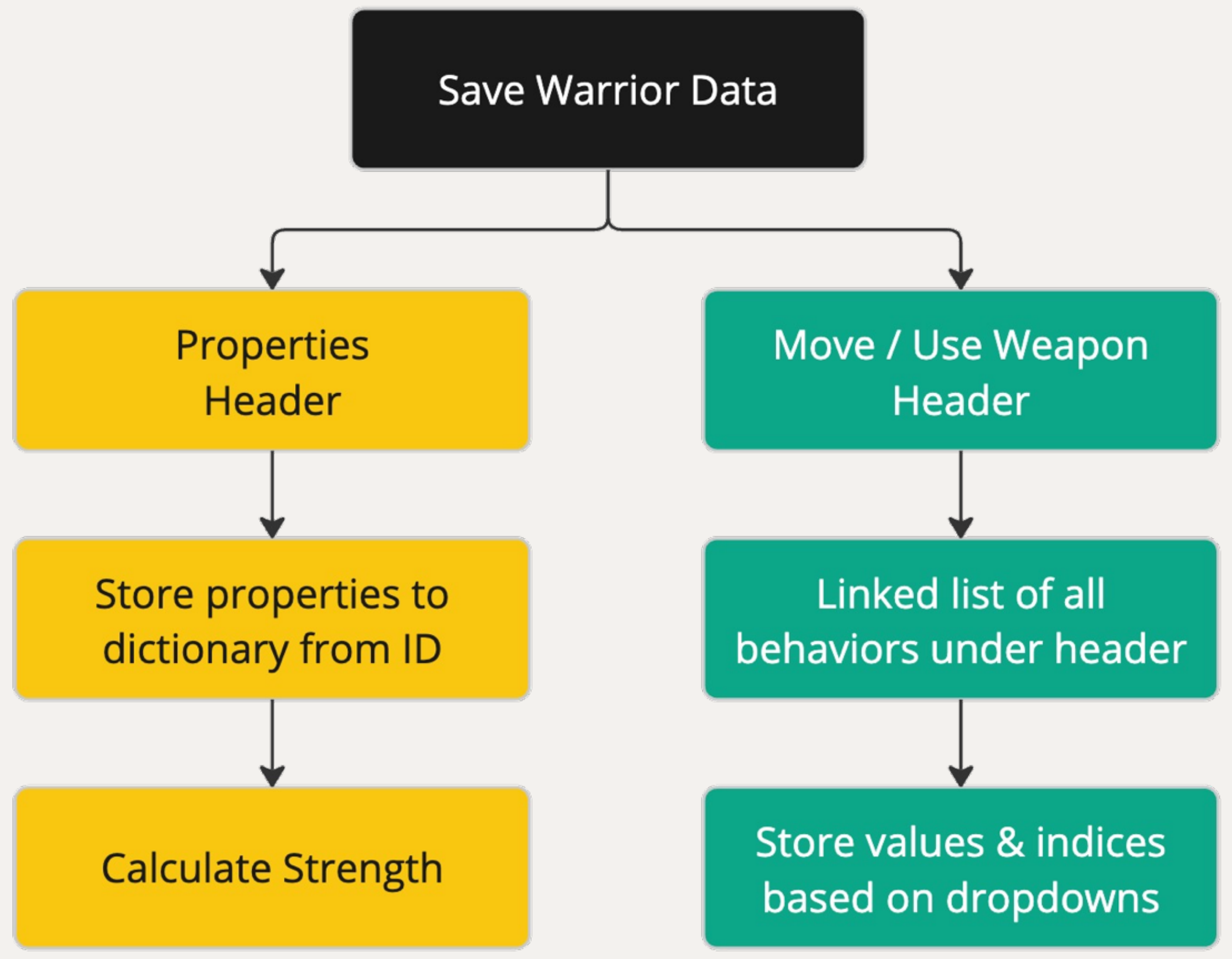


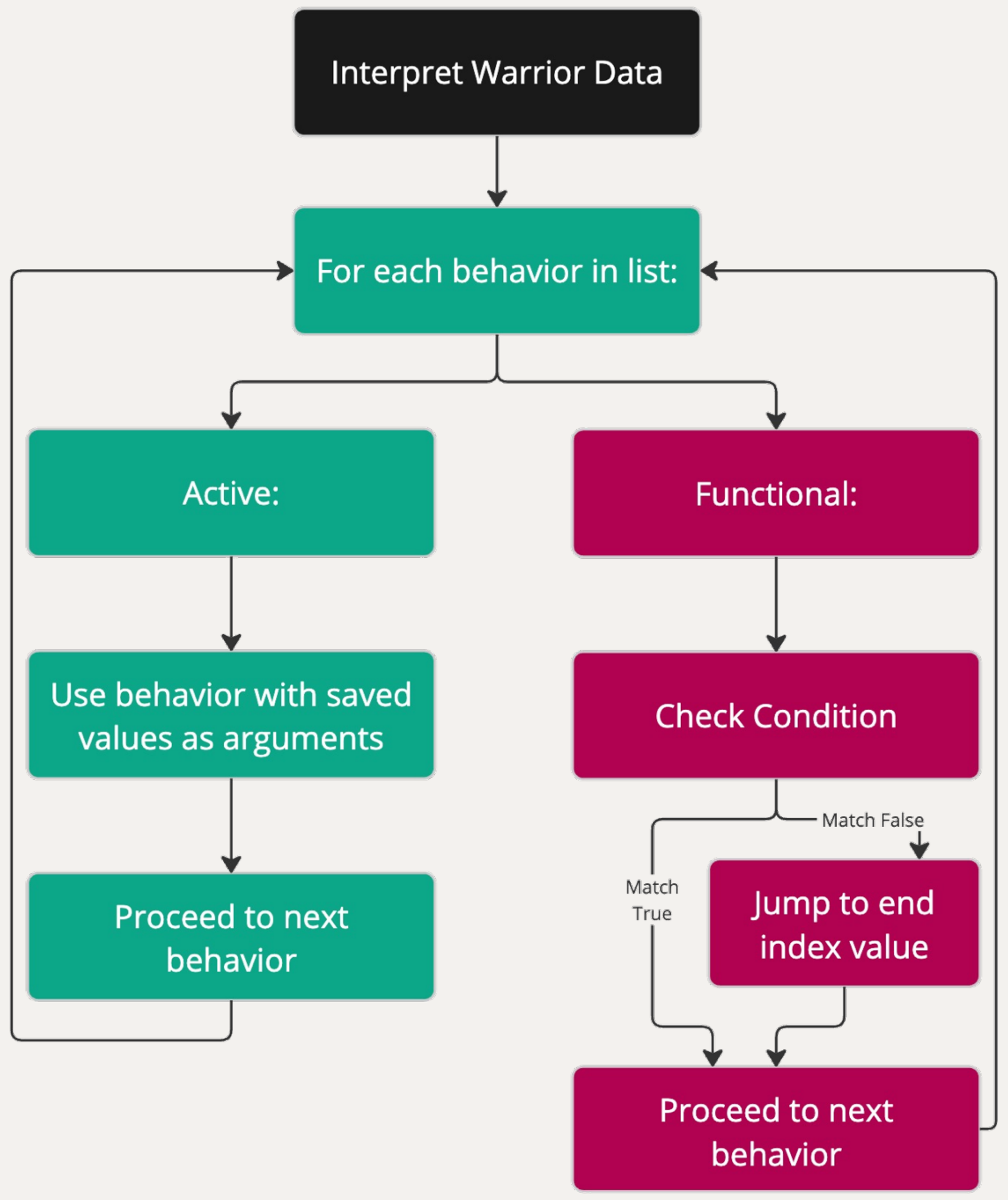
Figure 9. A more complicated warrior

Code Interpretation

Each **code block** holds a property or behavior ID and a list of values corresponding to its arguments.

- **Properties:** Slider or input field stored.
- **Behaviors:** Dropdown data stored. For conditionals, jump indices are stored as well.


Save Warrior Data
Properties Header
Store properties to dictionary from ID
Calculate Strength
Move / Use Weapon Header
Linked list of all behaviors under header
Store values & indices based on dropdowns



Game-Based Learning

- **Tutorials** introduce players to new mechanics in each level. Success in a given level comes from effective application of new mechanics to growing knowledge.
- **Level Structure:**
  - 1-5: Properties and Behaviors
  - 6-10: Movement
  - 11-15: Conditionals
  - 16-20: Looping
- To emphazise learning and experimentation, **difficulty is low**, and players are not punished for failure.
- **The Sandbox** allows players to experiment without limits.

Evaluation & Results

- Testers took the same **coding knowledge quiz** before and after playing the game.
- **Qualitative** data gathered from anonymous surveys afterward.

Results

LG 1: Gain familiarity with foundational programming concepts.

- **7.6%** average increase in quiz scores from before playing to after, with 7/12 players showing notable improvement.
- Occasional difficulty starting, but strong **pacing** afterwards.

LG 2: Build confidence in programming skill and feel less intimidated by the idea of learning how to code.

- **13.8%** increase in reported confidence. All players either increased or remained the same.
- Successfully **simplified complex concepts**, experimental environment.

On average, players rated their enjoyment of *Magicode* **6/7**. Players enjoyed the simplicity, strategy, and approachability that encouraged them to keep playing. Based on feedback, future work would implement greater mechanical and instructional clarity through optional hints, examples, a glossary reference, and more time with each mechanic.