A multimodal interface for chess

How we made people gesticulate and scream at their computers

SAPIENZA
UNIVERSITÀ DI ROMA

Giuseppina Iannotti, 1938436
Davide Marincione, 1927757

Sapienza, University of Rome

A. Y. 2023 - 2024

The Idea

Remember this?

Figure: Wizard's Chess, Harry Potter and the Philosopher's Stone

The Idea 2

Know this feeling?

Figure: Some stock image of an hand holding a chess piece.

Before that

We have to get from here...

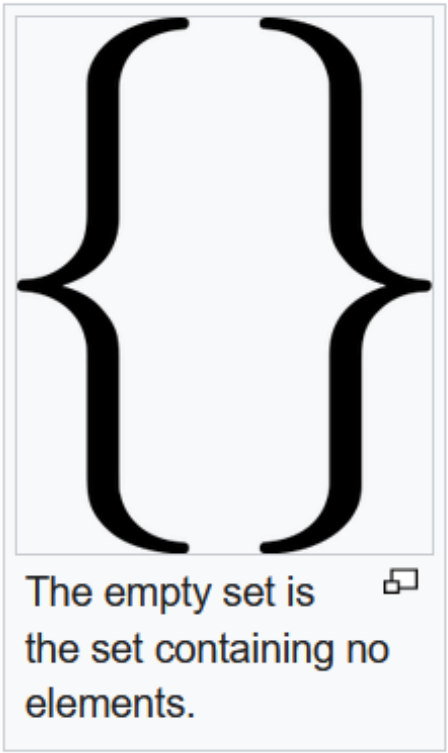

Figure: What we have.

To here!

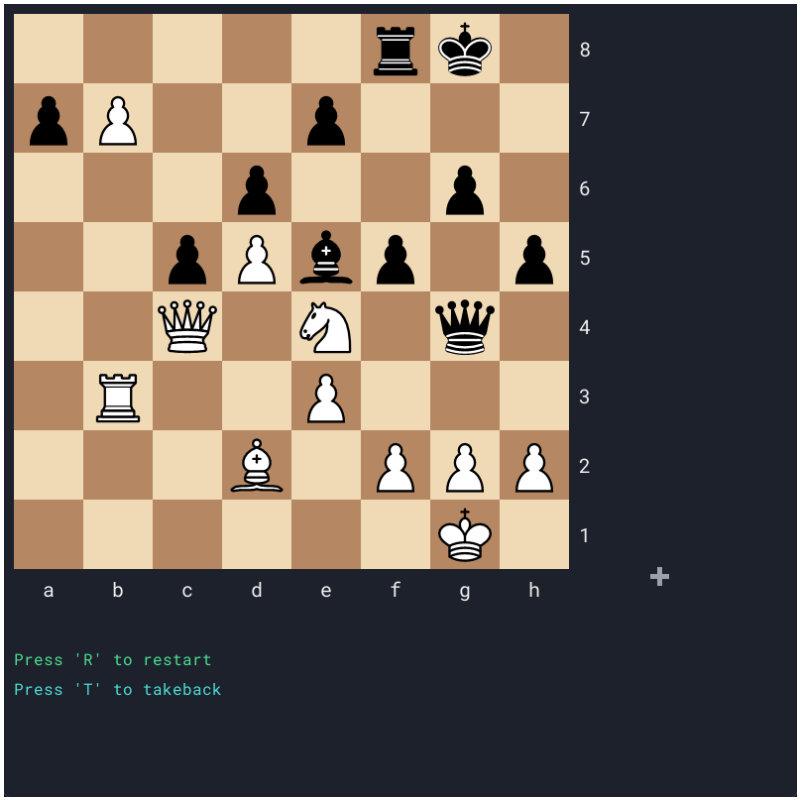

Figure: What we want.

We need some OOP

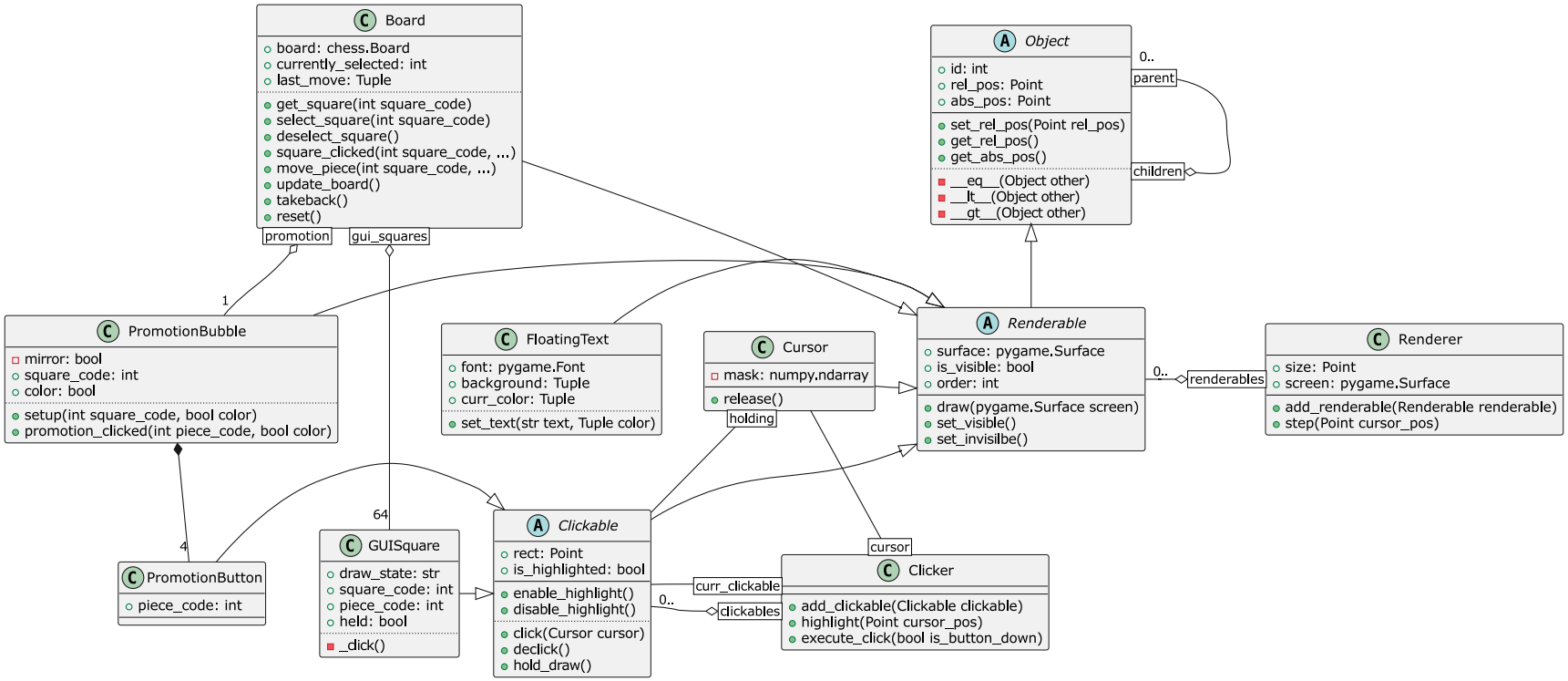

Figure: Class diagram of the game's elements

A bit in detail 1

The `Renderer`...draws `Renderables`!

1. Keeps track of them.
2. Draws them based on each object's `order` attribute.
3. Draws them only if they are set to visible.

The `Clicker`:

1. Keeps track of the `Clickables`.
2. Highlights the current `Clickable`, calls its click/declick method.
3. Drives hold/release with `Cursor`.

A bit in detail 2

Our `Cursor` is this neat thing:

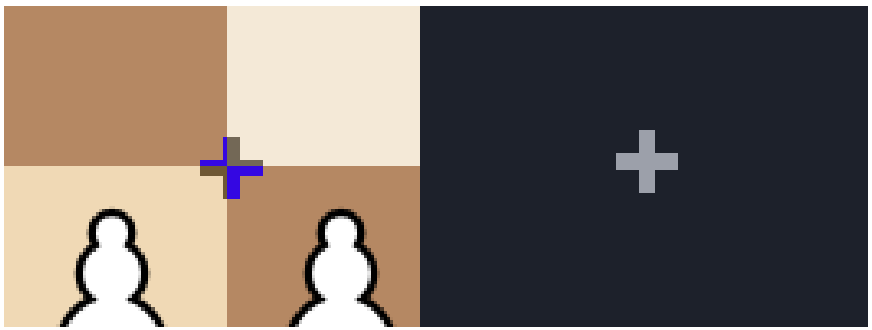

Figure: Our `Cursor`.

It is simple, but we are pretty happy about it:

1. It is extremely visible, because of the dynamic color

$$c^* = (c + 128) \bmod 256.$$

2. It can hold pieces.
3. Being stylistically different might have helped!

A bit in detail 3

The `Board`:

1. Wraps a `chess.Board` object (and all its complicated chess logic).
2. Handles the state of all the `GUISquare` and that of the `PromotionBubble`.
3. Plays audio when moves are done!

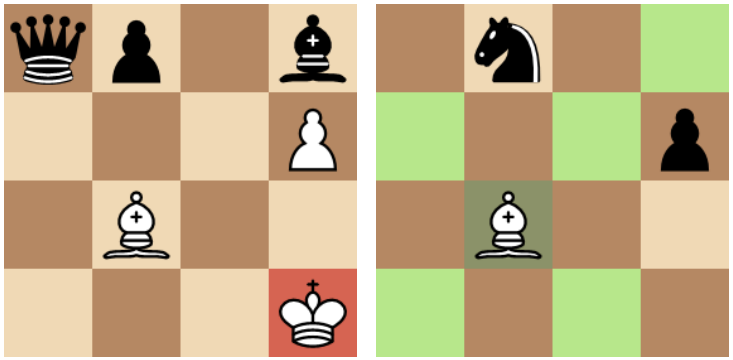

Figure: Examples of `GUISquare` states

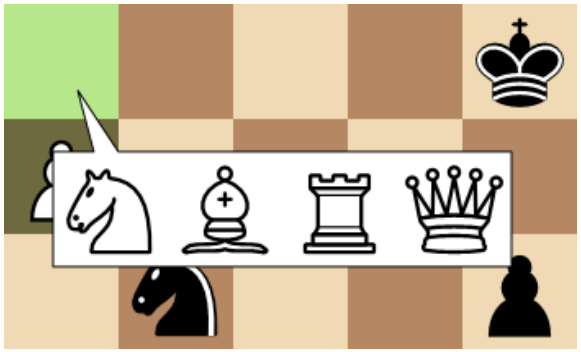

Figure: What `PromotionBubble` looks like

The main loop

All of this runs on the main thread, within the loop:

1. Update cursor with latest mouse or hand position.
2. `clicker.highlight(cursor_pos)`.
3. Resolve events, such as mouse clicks, key presses (quit game, takebacks), and moves done (for the AI).
4. Resolve voice commands.
5. `renderer.step()`.
6. Run metrics recorder.

Dragonfly? What's that?

Rules 1

Rules 2

Validating commands

Good ol' mediapipe

'Hand'made normalization

Gesture recognition

Hand2Cursor mapping

Recording users

Some metrics

Results 1

Results 2

Conclusions