

Halma 3.0

Derek Phanekham & Johna Rutz

Details

- A local opponent shall play the halma game using a graphic interface.
- The product shall display all valid moves on the graphic interface.
- The product shall display the JSON of a valid move regardless of whether the user is a web based or local opponent.

Functional Requirements

- C++
- Qt
- Local User Interface
- Communicate with web via JSON
- Accept URL to web based opponent

HTTP POST Format

```
{"boardSize": { "height": 9, "width": 18 },
  "yourBoard": { "pieces": [{ "x": 0, "y": 0 }, { "x": 0, "y": 1 } ],
    "destinations": [{ "x": 0, "y": 0 }, { "x": 0, "y": 3 },
      { "x": 0, "y": 1 } ] },
  "enemyBoard": { "pieces": [{ "x": 0, "y": 0 }, { "x": 0, "y": 1 } ],
    "destinations": [{ "x": 0, "y": 0 }, { "x": 0, "y": 1 }, { "x": 0, "y": 1 } ] }
}
```

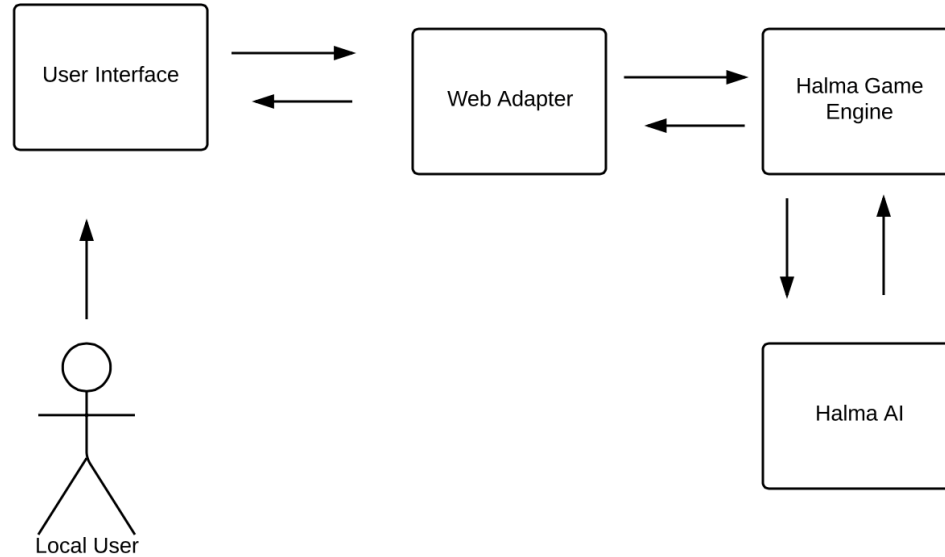
Move request format

```
{"from": {"x":2, "y":4}, "to": {"x":9, "y":10}}
```

Non Functional Requirements

- 30 second timer
- Dual User:
 - Human vs. Human
 - Human vs. AI
 - AI vs. AI
- New and exciting interference rules

Design



Source Code Control

Git repository