Final Presentation

Yunzhou Li, Hongyu Sun, Zixuan Fang, Zhiwei Xie
Overview

● Background

● Game Logic & UI Design
  ○ Menu Diagram
  ○ Game Diagram

● Hardware
  ○ Memory
  ○ Input Device

● Software
  ○ AI Algorithm
Background

Gomoku is a board game played by two players on a 15x15 grid board. Players take turns placing their pieces, and the first player to form a continuous line of five pieces of the same color (horizontally, vertically, or diagonally) wins.
Game Logic & UI Design

- Menu

```
GOMOKU
START PVP
START PVE
CREATE ROOM
JOIN ROOM
EXIT
```
Game Logic & UI Design

- In Game
# Memory

<table>
<thead>
<tr>
<th>Name</th>
<th>Graphic</th>
<th>Size (pixels)</th>
<th>Bits per Pixel</th>
<th>Total Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Board</td>
<td><img src="image" alt="Board Graphic" /></td>
<td>500 x 480</td>
<td>4</td>
<td>120,000</td>
</tr>
<tr>
<td>Black Piece</td>
<td><img src="image" alt="Black Piece Graphic" /></td>
<td>33 x 31</td>
<td>8</td>
<td>1023</td>
</tr>
<tr>
<td>White Piece</td>
<td><img src="image" alt="White Piece Graphic" /></td>
<td>33 x 31</td>
<td>8</td>
<td>1023</td>
</tr>
<tr>
<td>Fonts</td>
<td><img src="image" alt="Fonts Graphic" /></td>
<td>55 x 8 x 5</td>
<td>1</td>
<td>225</td>
</tr>
<tr>
<td>Memory Budget (bits)</td>
<td></td>
<td></td>
<td></td>
<td>978,168</td>
</tr>
<tr>
<td>Address</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>---------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>Selected mark Y location</td>
<td>Selected mark X Location</td>
<td>Last Piece Y Location</td>
<td>Last Piece X Position</td>
</tr>
<tr>
<td>06</td>
<td>Message group index</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>08</td>
<td>Message Index, 0 means no message is selected.</td>
<td>Selected Message Info</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Touchpad mark X position</td>
<td>Touchpad mark Info</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Touchpad mark Y position</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Address 06:
- We catagorize messages into groups, so that we can set the visibility of a group of message simultaneously
- 0~9 - Visibility status of each message in one group. So there are at most 10 messages in a group.
- 11~15 - Message group index. At most 64 groups
- So we could have at most 640 messages to show using this interface

Address 00, offset:
- 0 - Whether to show menu page or board page
- 1 - Whether to show touchpad cursor
- 2 - Player UI, player piece information. Whether player 1 takes black piece
- 3 - Player UI, current player mark. Whether it is player 1’s turn.
- 4 - Player UI, player 2 profile image. Whether player 2’s profile image is cat or dog
- 5 - Message box UI. Whether to show the popup message
- 8 - Victory Sound
- 9 - Defeat sound
- 10 - Show menu sound
- 11 - Piece move sound
Input Devices

- Xbox One controller
- XP Touchpad
- Both use libusb
Input Device

**Base class**: BaseInputDevice

Methods:

open_device, close_device, create_handling_thread, stop_handling_thread

and more …

Xbox: inherits from BaseInputDevice

class XboxController
### Transfer protocol

<table>
<thead>
<tr>
<th>data[2]</th>
<th>Direction Button</th>
</tr>
</thead>
<tbody>
<tr>
<td>data[3]</td>
<td>Function Button(Y/B/A/X/L B/RB)</td>
</tr>
<tr>
<td>data[4]</td>
<td>Button LT</td>
</tr>
<tr>
<td>data[5]</td>
<td>Button RT</td>
</tr>
<tr>
<td>data[6:9]</td>
<td>Movement of the left rotate</td>
</tr>
<tr>
<td>data[10:13]</td>
<td>Movement of the right rotate</td>
</tr>
</tbody>
</table>
**Touchpad** : inherits from `BaseInputDevice`

```cpp
class TouchPad
```

### Different status

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO PEN</td>
<td>No Pen detected</td>
</tr>
<tr>
<td>PEN</td>
<td>Pen is detected</td>
</tr>
<tr>
<td>LEFT</td>
<td>Left button clicked</td>
</tr>
<tr>
<td>RIGHT</td>
<td>Right button clicked</td>
</tr>
<tr>
<td>LEFT&amp;RIGHT</td>
<td>Both are clicked</td>
</tr>
</tbody>
</table>

```cpp

class Touchpad : public BaseInputDevice{
public:
    Touchpad()
    {
        device_name_ = "Touchpad";
        vendor_id_ = TOUCHPAD_VENDOR_ID;
        product_id_ = TOUCHPAD_PRODUCT_ID;
        interface_ = TOUCHPAD_INTERFACE;
        endpoint_ = TOUCHPAD_ENDPOINT;
    }
}
```
Software

- Basic game logic — Implemented in C++
- AI algorithm — MiniMax with Alpha-Beta Pruning

on branch tsuki — Test it and have fun!
Thank you