Traveller Terminal - Hidden Pages & QR Code Implementation

This README provides instructions on how to implement the QR code hidden pages and additional terminal logs for your Traveller RPG game.

Overview

This implementation adds several exciting features to your existing Traveller Terminal application:

1. **Hidden Pages** - Secret pages that can only be accessed via QR codes on physical handouts
2. **New Terminal Logs** - Additional terminals with lore-rich content that can be discovered by players
3. **QR Code Generator** - A tool for creating QR codes to print on your physical game handouts

Implementation Steps

1. Add New Components

Add these new files to your project:

- `src/components/HiddenPage.jsx` - The component that displays hidden page content
- `src/components/QRCodeGenerator.jsx` - A utility for creating QR codes for handouts

2. Update Routes

Update your `App.jsx` file to include the new routes:

jsx

```jsx
import React from "react";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import TravellerTerminal from "./components/TravellerTerminal";
import PoopTerminal from "./components/PoopTerminal";
import HiddenPage from "./components/HiddenPage";
import QRCodeGenerator from "./components/QRCodeGenerator";

function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<TravellerTerminal />} />
        <Route path="/poop" element={<PoopTerminal />} />
        <Route path="/hidden/:pageId" element={<HiddenPage />} />
        <Route path="/admin/qr-generator" element={<QRCodeGenerator />} />
      </Routes>
    </Router>
  );
}

export default App;
```

3. Add New Terminal Options

Update the `terminals` object in your `TravellerTerminal.jsx` component to include the new terminal options:

javascript

```javascript
const terminals = {
  "lysani01": { requiresRoll: 8, logs: "/logs/lysani01.json" },
  "s.elara01": { requiresRoll: false, logs: "/logs/s.elara01.json" },
  "slocombe875": { requiresRoll: 8, logs: "/logs/slocombe875.json" },
  "waferterm01": { requiresRoll: false, logs: "/logs/waferterm01.json" },
  "labpc81": { requiresRoll: 6, logs: "/logs/labpc81.json" },
  "vanagandr001": { requiresRoll: 7, logs: "/logs/vanagandr001.json" },
  // New secret terminals
  "blacksite-es1": { requiresRoll: 12, logs: "/logs/blacksite-es1.json" },
  "blacktalon": { requiresRoll: 10, logs: "/logs/blacktalon.json" },
  "vennik-personal": { requiresRoll: 10, logs: "/logs/vennik-personal.json" },
  "sayelle-logs": { requiresRoll: 9, logs: "/logs/sayelle-logs.json" },
  "fuwnet": { requiresRoll: 8, logs: "/logs/fuw-network.json" }
};
```

4. Add JSON Log Files

Create these JSON files in your `public/logs` directory:

- `blacksite-es1.json` - Eclipse Shard research facility logs
- `blacktalon.json` - Black Talon Cartel operations data
- `vennik-personal.json` - Rax Vennik's personal logs
- `sayelle-logs.json` - Sayelle's operative logs
- `fuw-network.json` - Free Union of Workers network data

5. Install Required Package

Install the QR code generator package:

bash

```bash
npm install qrcode.react
```

Creating QR Codes for Handouts

Once you've implemented the system, follow these steps to create QR codes for your physical handouts:

1. Navigate to `/admin/qr-generator` in your deployed application
2. Select the hidden page you want to link to from the dropdown
3. Enter the base URL where your application is hosted

4. Download the generated QR code
5. Include the QR code on your physical game handouts

Hidden Page IDs

The following hidden page IDs are available:

- `eclipseshard` - Data about the mysterious Eclipse Shard
- `blacktalonops` - Black Talon Cartel operation files
- `blacksite` - Location data for the secret Blacksite ES1
- `sayelle` - Enhancement records for the operative Sayelle
- `fuw` - Resistance intelligence from the Free Union of Workers

Game Usage Suggestions

Here are some ways to use these features in your Traveller game:

1. **Physical Handouts** - Create in-universe documents with QR codes hidden as barcodes, serial numbers, or design elements
2. **Discovery Rewards** - Let players earn terminal passwords by completing quests or succeeding at skill checks
3. **Progressive Revelation** - Gradually introduce new terminal access as the story progresses
4. **Split Information** - Divide critical information across multiple terminals to encourage exploration
5. **Red Herrings** - Include some false information to keep players guessing

Hidden Terminal Clues

You could drop subtle hints about secret terminals in your game:

- "A hastily scribbled note mentions accessing 'blacksite-es1' for critical information."
- "You notice 'fuwnet' scratched into the corner of a maintenance panel."
- "A corrupt security officer whispers about 'vennik-personal' before passing out."

Customization

Feel free to modify the terminal logs and hidden page content to better fit your campaign's specific storyline. The JSON structure makes it easy to add new entries or modify existing ones.

Troubleshooting

- If QR codes aren't working, ensure your app is properly deployed and the URLs are correct
- Check that all JSON files are properly formatted with no trailing commas
- Verify that the terminal passwords match exactly what's in your code