```
// Copyright 2023 Thomas O’Connor
#include "Sokoban.hpp"

// Extract the entire gameState into the Sokoban object
Sokoban& operator>>(Sokoban& game, std::ifstream& file) {
    file >> game._h; file >> game._w;
    // Given game dimensions, resize 2D vector and input from file
    game.gameState.resize(game._h);
    for (int i = 0; i < game._h; i++) {
        game.gameState[i].resize(game._w);
        for (int j = 0; j < game._w; j++) {
            file >> game.gameState[i][j];
        }
    }
    return game;
}

// Output the internal representation of the gameState to the terminal
void Sokoban::getGameState() const {
    for (int i = 0; i < _h; i++) {
        for (int j = 0; j < _w; j++) {
            cout << gameState[i][j];
        }
        cout << endl;
    }
}

// Overload the virtual draw function:
// Load required textures and display in window at prescribed locations
void Sokoban::draw(sf::RenderTarget& target, sf::RenderStates states) const {
    sf::Texture Wall, Box, Empty, Storage, Man;
    if (!Wall.loadFromFile("sokoban/block_06.png")) exit(1);
    if (!Box.loadFromFile("sokoban/crate_03.png")) exit(1);
    if (!Empty.loadFromFile("sokoban/ground_01.png")) exit(1);
    if (!Storage.loadFromFile("sokoban/ground_04.png")) exit(1);
    if (!Man.loadFromFile("sokoban/player_05.png")) exit(1);

    for (int i = 0; i < _h; i++) {
        for (int j = 0; j < _w; j++) {
            sf::Sprite tile, backTile;
            tile.setPosition(j * TILE_SIZE, i * TILE_SIZE);

            switch (gameState[i][j]) {
                case ’#’:
                    tile.setTexture(Wall);
                    break;
                case ’.’:
                    tile.setTexture(Empty);
                    break;
                case ’a’:
                    tile.setTexture(Storage);
                    break;
                case ’A’:
                    backTile.setPosition(j * TILE_SIZE, i * TILE_SIZE);
                    backTile.setTexture(Empty);
                    target.draw(backTile);
                    tile.setTexture(Box);
                    break;
                case ’@’:
                    backTile.setPosition(j * TILE_SIZE, i * TILE_SIZE);
                    backTile.setTexture(Empty);
                    target.draw(backTile);
                    tile.setTexture(Man);
                    break;
```

```cpp
            }
            target.draw(tile);
        }
    }
}

// Function that displays time in upper-left corner
void Sokoban::drawElapsingTime(sf::RenderWindow &window, sf::Clock &clock) {
    sf::Time elapsed = clock.getElapsedTime();
    int minutes = elapsed.asSeconds() / 60;
    int seconds = static_cast<int>(elapsed.asSeconds()) % 60;

    std::string timeString = std::to_string(minutes) + ":" +
    (seconds < 10 ? "0" : "") + std::to_string(seconds);

    sf::Font font;
    font.loadFromFile("sokoban/arial.ttf");
    sf::Text timeText(timeString, font, 30);
    timeText.setFillColor(sf::Color::White);

    window.draw(timeText);
}
```