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

int main(int argc, char* argv[]) {
    // Command line arguments
    std::string inputFileName = argv[1];
    // Open file and check for failure
    std::ifstream myLevel;
    myLevel.open(inputFileName);
    if (!myLevel) exit(1);

    // Create game and input values from file
    Sokoban Game;
    Game >> myLevel;

    // Rednder the window using the appropriate game dimensions
    sf::RenderWindow window
    (sf::VideoMode(TILE_SIZE * Game.getWidth(),
    TILE_SIZE * Game.getHeight()), "Sokoban");
    window.setFramerateLimit(15);
    // Create clock
    sf::Clock clock;
    // Load win condition variable
    bool performWinConditionOnce = 1;

    while (window.isOpen()) {
        // Process events
        sf::Event event;
        while (window.pollEvent(event)) {
            // Close window: exit
            if (event.type == sf::Event::Closed) window.close();
        }
        // Clear screen
        window.clear();

        if (!Game.isWon()) {
            // Get Keyboard input and impliment correct movement
            if (Keyboard::isKeyPressed(Keyboard::W)) Game.movePlayer(UP);
            if (Keyboard::isKeyPressed(Keyboard::A)) Game.movePlayer(LEFT);
            if (Keyboard::isKeyPressed(Keyboard::S)) Game.movePlayer(DOWN);
            if (Keyboard::isKeyPressed(Keyboard::D)) Game.movePlayer(RIGHT);
            if (Keyboard::isKeyPressed(Keyboard::Up)) Game.movePlayer(UP);
            if (Keyboard::isKeyPressed(Keyboard::Left)) Game.movePlayer(LEFT);
            if (Keyboard::isKeyPressed(Keyboard::Down)) Game.movePlayer(DOWN);
            if (Keyboard::isKeyPressed(Keyboard::Right)) Game.movePlayer(RIGHT);
            // Draw the arena and clock
            window.draw(Game);
            Game.drawElapsingTime(window, clock);
            // Display all elements
            window.display();
        } else if (performWinConditionOnce) {
            // Draw the win state
            sf::Time elapsed = clock.restart();
            int minutes = elapsed.asSeconds() / 60;
            int seconds = static_cast<int>(elapsed.asSeconds()) % 60;
            std::string timeString = "     you win\nyour time: " + 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);
            timeText.setPosition(sf::Vector2f(
                (Game.getWidth() * TILE_SIZE / 2) - 120,
                (Game.getHeight() * TILE_SIZE / 2) - 60));
            window.draw(timeText);
            window.display();
            // Play win sound after screen display
            Game.playSound();
            performWinConditionOnce = 0;
        }
        // Special keyboard input for restarting game, undoing actions, and closing window
        if (Keyboard::isKeyPressed(Keyboard::R)) {
            Game.restart(clock);
            performWinConditionOnce = 1;
        }
        if (Keyboard::isKeyPressed(Keyboard::Z)) {
            Game.undo();
            performWinConditionOnce = 1;
        }
        if (Keyboard::isKeyPressed(Keyboard::X)) window.close();
    }
    return 0;
}
```