```cpp
// Copyright 2023 Thomas O’Connor
#ifndef CHECKERS_HPP
#define CHECKERS_HPP

#include <algorithm>
#include <fstream>
#include <iostream>
#include <string>
#include <vector>
#include <SFML/Audio.hpp>
#include <SFML/Graphics.hpp>
#include <SFML/System.hpp>
#include <SFML/Window.hpp>

using std::cout;
using std::endl;
using std::vector;
using sf::Vector2f;

#define TILE_SIZE 64
#define BOARD_DIMENSIONS 8
#define BOARD_OFFSET 32

class Checkers : public sf::Drawable {
 public:
  // Constructors
  Checkers() { initializeBase(); }

  // Getters
  bool nothingSelected(void) { return !stillPlaying; }
  bool isWon(void);
  bool getWinner(void);
  sf::Vector2i getSelectedPawn(void);

  // Interactors
  void selectPiece(sf::Vector2i mouseLocation);
  void movePiece(sf::Vector2i mouseLocation);
  void deselectPiece(void);
  void switchTurn(void) { playerTurn = !playerTurn; }
  void restart(bool& winCondition);

  // Performers
  void playSound(void);
  void visualMoveAssist(sf::RenderTarget& target);
  void drawStar(sf::RenderTarget& target, sf::Texture star, int yc, int xc);

 private:
  // Draw game in SFML
  virtual void draw(sf::RenderTarget& target, sf::RenderStates states) const;
  // Initialize game storage vectors
  void initializeBase(void);
  // Automatically king pawns at respective finish lines
  void finishLine(void);

 private:
  // 2D array that stores the current game state
  vector<vector<char>> currentGameState;
  // player turn - 0 is black 1 is red
  bool playerTurn = 0;
  // if still playing, don’t end the game prematurely
  bool stillPlaying = 0;
  // if unable to move final piece, set win to true
  bool setWinTrue = 0;
};

// Helper functions
bool mouseInGameBounds(sf::Vector2i mouseLocation);
void drawBackingRectangle(sf::RenderTarget& target, int x, int y);

#endif
```