CS 7B - Fall '19 - Project 2: Dots and Boxes Game with SFML

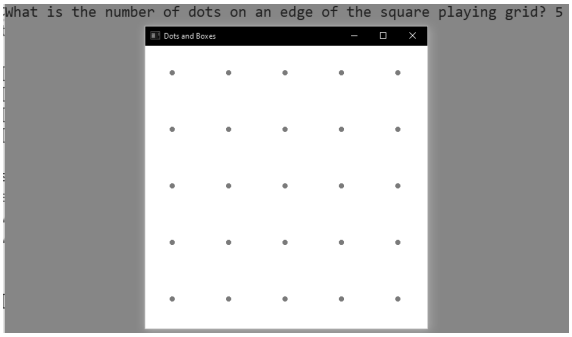
This assignment is designed to develop your skills in using the SFML library to display images in the graph window. You will need to make extensive research into the library documentation at https://www.sfml-dev.org/learn.php.

Due Wednesday, October 16 by the start of lecture.
Working in groups is not permitted

You should also read chapter 16 of *Winning Ways*, a authored by John Conway, for an analysis of Dots and Boxes strategies.

The ultimate goal of this project is to get a working two-player version of the game Dots and Boxes where one player is the user and the other is automatic; "An AI opponent," as it were.

1. Get the information from the user about the number of dots on an edge of the square grid, so that, if the user asks for, say, a 4by4 grid, then the play-space comes out neatly displayed with a margin, something like that shown to the right.



Something like this will get you started:

```
const int wWidth = 440, wHeight = 440;
int edge = 4;
sf::RenderWindow window(sf::VideoMode(wWidth, wHeight), "Dots and Boxes");
//sf::CircleShape shape(100.f);
//shape.setFillColor(sf::Color::Green);
cout << "What is the number of dots on an edge of the square playing grid? ";
cin >> edge;
std::vector<sf::CircleShape> dots(edge*edge);
for (int i = 0; i < edge*edge; ++i) {
  dots[i].setFillColor(sf::Color(200, 100, 0));
  dots[i].setRadius(4);
  dots[i].setPosition(sf::Vector2f(40 + (i % edge) * wWidth / edge, 40 + (i / edge) *
    wHeight/edge));
  dots[i].setOrigin(2, 2);
}
```

2. Create a `Player` class to manage, at first, a single player: to keep track of what edges the player has chosen to draw. To keep track of this you might use a vertex array like this:

```
sf::Vertex line[] =  {
sf::Vertex(sf::Vector2f(42.f,42.f)),
sf::Vertex(sf::Vector2f(42.f, 42.f+wHeight/edge))
};
for (int i = 0; i < 2 * edges; ++i)
line[i].color = sf::Color(sf::Color::Red);
```

and then draw them like so:

```
window.draw(line, 2, sf::Lines);
```

3. Now add code to detect when all four edges of a box have been drawn and fill in the box with a color (use `sf::rectangleshape()`?) and mark the box with the players initials.
4. Add a second player that is a dumb computer that incorporates a simple greedy algorithm of drawing the first line it sees is available.
5. Modify the "AI" so that the dumb computer is a little smarter and will complete a box is possible.
6. Improve the "AI" so to incorporate some more sophisticated strategy suggested by Conway in the the referenced article.