Project: SFML Snake

Github: https://github.com/excisionhd/CS256/blob/master/SFMLGames/SFMLProject.cpp

```
/****************
* FILENAME : SFMLProject.cpp
* DESCRIPTION : Game development project using SFML
* library for the game snake.
* AUTHOR : Amir Sotoodeh
* START DATE : 5/27/18
#include "stdafx.h"
#include <SFML/Graphics.hpp>
#include <time.h>
using namespace sf;
//global variables
int num vertBox = 30, num horzBox = 20;
int size = 16; //number of pixels
int w = size * num horzBox; //background number of pixels in width
int h = size * num vertBox; //background number of pixels in height
int direction, snake length = 4;
int player2 direction, player2 snake length = 4;
float timer = 0, delay = 0.1;
//struct of snake, holds 100 max length
struct Snake
     int x, y;
}s[100], s2[100];
struct Fruit
     int x, y;
}food;
//render every frame when called
void Tick() {
     //moves the rest of the snake based on the next's position
     for (int i = snake length; i > 0; --i)
```

```
s[i].x = s[i - 1].x;
      s[i].y = s[i - 1].y;
}
for (int i = player2_snake_length; i > 0; --i) {
      s2[i].x = s2[i - 1].x;
      s2[i].y = s2[i - 1].y;
}
//move direction of snake
//user up
if (direction == 3)
      s[0].y -= 1;
//user down
if (direction == 0)
      s[0].y += 1;
//user left
if (direction == 1)
      s[0].x -= 1;
//user up
if (direction == 2)
      s[0].x += 1;
if (player2 direction == 3)
      s2[0].y -= 1;
//user down
if (player2 direction == 0)
      s2[0].y += 1;
//user left
if (player2_direction == 1)
      s2[0].x -= 1;
//user up
if (player2 direction == 2)
      s2[0].x += 1;
//grow if snake eats food
if ((s[0].x) == food.x && (s[0].y) == food.y) {
      snake length++;
      delay *= 0.85;
      //randomly place food after eaten
      food.x = rand() % num horzBox;
      food.y = rand() % num vertBox;
}
if ((s2[0].x) == food.x && (s2[0].y) == food.y) {
      player2_snake_length++;
```

```
//randomly place food after eaten
            food.x = rand() % num_horzBox;
            food.y = rand() % num vertBox;
      }
      //check boundaries
      if (s[0].x > num horzBox)
            s[0].x = 0;
      if (s[0].x < 0)
            s[0].x = num horzBox;
      //top and bottom
      if (s[0].y > num vertBox)
            s[0].y = 0;
      if (s[0].y < 0)
            s[0].y = num vertBox;
      //check boundaries
      if (s2[0].x > num horzBox)
            s2[0].x = 0;
      if (s2[0].x < 0)
            s2[0].x = num_horzBox;
      //top and bottom
      if (s2[0].y > num_vertBox)
            s2[0].y = 0;
      if (s2[0].y < 0)
            s2[0].y = num vertBox;
      //do not allow snake to go over its body
      for (int i = 1; i < snake length; i++)</pre>
      {
            //halves the snake
            if (s[0].x == s[i].x && s[0].y == s[i].y)
                  snake length = i;
      }
      for (int i = 1; i < snake_length; i++)</pre>
            //halves player2 snake
            if (s2[0].x == s2[i].x \&\& s2[0].y == s2[i].y)
                  player2_snake_length = i;
      }
}
int main()
{
      srand(time(0));
```

delay *= 0.85;

```
RenderWindow window(VideoMode(w, h), "Snake Game!");
//load textures
Texture t1, t2, t3, t4;
t1.loadFromFile("Pictures/white.png");
t2.loadFromFile("Pictures/red.png");
t3.loadFromFile("Pictures/food.png");
//create sprites (with dimensions)
Sprite sprite1(t1);
Sprite sprite2(t2);
Sprite spritePlayer2(t2);
Sprite sprite3(t3);
//place food at 10,10
food.x = 10;
food.y = 10;
//set initial position for second player snake
for (int i = 0; i < player2_snake_length; i++) {</pre>
      s2[i].x = 19;
}
Clock clock;
while (window.isOpen())
{
      float time = clock.getElapsedTime().asSeconds();
      clock.restart();
      timer += time;
      Event e;
      //close window
      while (window.pollEvent(e))
      {
            if (e.type == Event::Closed) {
                  window.close();
            }
      }
      //define user inputs for direction
      if (Keyboard::isKeyPressed(Keyboard::Up)) direction = 3;
      if (Keyboard::isKeyPressed(Keyboard::Down)) direction = 0;
      if (Keyboard::isKeyPressed(Keyboard::Left)) direction = 1;
```

```
if (Keyboard::isKeyPressed(Keyboard::Right)) direction = 2;
            if (Keyboard::isKeyPressed(Keyboard::W)) player2_direction = 3;
            if (Keyboard::isKeyPressed(Keyboard::S)) player2 direction = 0;
            if (Keyboard::isKeyPressed(Keyboard::A)) player2 direction = 1;
            if (Keyboard::isKeyPressed(Keyboard::D)) player2 direction = 2;
            if (timer > delay) {
                  timer = 0;
                  Tick();
            }
            window.clear();
            //Draw the background with sprites
            for (int i = 0; i < num horzBox; i++)</pre>
                  for (int j = 0; j < num vertBox; j++)</pre>
                  {
                         sprite1.setPosition(i*size, j*size);
                         window.draw(sprite1);
                  }
            //Draw snake
            for (int i = 0; i < snake length; i++) {</pre>
                  sprite2.setPosition(s[i].x*size, s[i].y*size);
                  window.draw(sprite2);
            }
            //Draw snake2
            for (int i = 0; i < player2_snake_length; i++) {</pre>
                  spritePlayer2.setPosition((s2[i].x*size), (s2[i].y*size));
                  window.draw(spritePlayer2);
            }
            //Draw fruit
            sprite3.setPosition(food.x*size, food.y*size);
            window.draw(sprite3);
            window.display();
      }
      return 0;
}
```