

```
1
2 #include <iostream>
3 #include "Player.h"
4 #include "Game.h"
5
6
7 /// <summary>
8 /// @author Peter Lowe
9 /// @version 1.0
10 /// @date may 2016
11 ///
12 /// </summary>
13
14 Player::Player()
15 {
16 }
17 //pete was hgere
18
19 Player::~~Player()
20 {
21 }
22 /// <summary>
23 /// @brief load & setup player sprite and roadHeight
24 /// </summary>
25 void Player::initialise()
26 {
27     if (!m_playerTexture.loadFromFile("ASSETS/IMAGES/mario.png"))
28     {
29         std::cout << "error with mario file";
30     }
31     m_playerSprite.setTexture(m_playerTexture);
32     m_roadHeight = 230;
33     m_playerSprite.setPosition(Game::s_screenWidth / 2, m_roadHeight);
34 }
35 /// <summary>
36 /// @brief jump up 100 pixels.
37 ///
38 /// reacting to a key press event raise the position by 100
39 /// </summary>
40 void Player::jump()
41 {
42     m_playerSprite.setPosition(m_playerSprite.getPosition().x,
43                               m_playerSprite.getPosition().y - 100);
44 }
45 /// <summary>
46 /// @brief apply gravity.
47 ///
48 /// move the player position down one so long as he is above raod
49 /// </summary>
50 void Player::fall()
51 {
```

```
52     if (m_playerSprite.getPosition().y < m_roadHeight)
53     {
54         m_playerSprite.setPosition(m_playerSprite.getPosition().x,
55                                     m_playerSprite.getPosition().y + 1);
56     }
57
58
59     /// <summary>
60     /// @brief move the player left.
61     ///
62     /// move the player position one pixel to left
63     /// if horizontal scale is +1 (facing right) the move player 188 to flip
64     /// sprite but keep player head in same location and set scale
65     /// to minus 1
66     /// </summary>
67     void Player::left()
68     {
69         m_playerSprite.setPosition(m_playerSprite.getPosition().x - 1,
70                                     m_playerSprite.getPosition().y);
71         if (m_playerSprite.getScale().x == 1)
72         {
73             m_playerSprite.setPosition(m_playerSprite.getPosition().x + 118,
74                                         m_playerSprite.getPosition().y);
75         }
76         m_playerSprite.setScale(-1, 1);
77     }
78
79     /// <summary>
80     /// @brief checks if game is over.
81     ///
82     /// check if player position is off screen if so the return true
83     /// </summary>
84     /// <returns>game over or not</returns>
85     bool Player::departed()
86     {
87         if (m_playerSprite.getPosition().x < 0 || m_playerSprite.getPosition().x >
88             Game::s_screenWidth)
89         {
90             return true;
91         }
92         else
93         {
94             return false;
95         }
96     }
97
98     void Player::resetPosition()
99     {
100         m_playerSprite.setPosition(Game::s_screenWidth / 2, m_roadHeight);
101     }
102
103     /// <summary>
```

```
100 /// @brief move the player right.
101 ///
102 /// move the player position one pixel to right
103 /// if horizontal scale is -1 (facing left) the move player 188 to flip
104 /// sprite but keep player head in same location and set scale
105 /// to plus 1
106 /// </summary>
107 void Player::right()
108 {
109     m_playerSprite.setPosition(m_playerSprite.getPosition().x + 1,
110                                m_playerSprite.getPosition().y);
111     if (m_playerSprite.getScale().x == -1)
112     {
113         m_playerSprite.setPosition(m_playerSprite.getPosition().x - 188,
114                                    m_playerSprite.getPosition().y);
115     }
116     m_playerSprite.setScale(1, 1);
117 }
118 /// <summary>
119 /// @brief update world on player.
120 ///
121 /// movement is initiated from the gameplay class
122 /// </summary>
123 /// <param name="time">update delta time</param>
124 void Player::update(sf::Time t_deltaTime)
125 {
126     fall();
127 }
128 /// <summary>
129 /// @brief draw player sprite.
130 /// </summary>
131 /// <param name="window">reference to main render window</param>
132 void Player::render(sf::RenderWindow& t_window)
133 {
134     t_window.draw(m_playerSprite);
135 }
```