```cpp

#include "GamePlay.h"
#include "Game.h"
#include <iostream>



GamePlay::GamePlay() :
    m_jumpKeyPressed{false}
{
}


GamePlay::~GamePlay()
{
}

/// <summary>
/// @brief load texture and initialise player.
///
/// load and set scale on texture then initialise player
/// </summary>
void GamePlay::initialise()
{
    if (!m_backgroundTexture.loadFromFile("ASSETS/IMAGES/background.jpg"))
    {
        std::cout << "error with  background file";
    }

    m_backgroundSprite.setTexture(m_backgroundTexture);
    float scalex = Game::s_screenWidth / m_backgroundTexture.getSize().x;
    float scaley = Game::s_screenHeight / m_backgroundTexture.getSize().y;
    m_backgroundSprite.setScale(scalex, scaley);
    m_backgroundSprite.setPosition(0, 0);

    m_player.initialise();

}
/// <summary>
/// @brief check for for event prior to update.
///
/// check for up arrow key press before main loop calls update
/// </summary>
/// <param name="event">sf event from os</param>
void GamePlay::processInput(sf::Event t_event)
{
    if ( sf::Event::KeyPressed == t_event.type)
    {
        if (sf::Keyboard::Up == t_event.key.code )
        {
            m_jumpKeyPressed = true;
```

```
53          }
54      }
55  }
56  /// <summary>
57  /// @brief main gameplay loop.
58  ///
59  /// check for player input and call player methods
60  /// then update player and check if game over
61  /// </summary>
62  /// <param name="time">update delta time</param>
63  void GamePlay::update(sf::Time t_deltaTime)
64  {
65      if (m_jumpKeyPressed)
66      {
67          m_player.jump();
68      }
69
70      if (sf::Keyboard::isKeyPressed(sf::Keyboard::Left))
71      {
72          m_player.left();
73      }
74      if (sf::Keyboard::isKeyPressed(sf::Keyboard::Right))
75      {
76          m_player.right();
77      }
78      if (m_player.departed())
79      {
80          Game::currentState = GameState::MainMenu;
81          m_player.resetPosition();
82      }
83      m_player.update(t_deltaTime);
84      m_jumpKeyPressed = false;
85  }
86  /// <summary>
87  /// @brief render the game scene.
88  ///
89  /// draw the background then the player
90  ///
91  /// </summary>
92  /// <param name="window">refrence to main render window</param>
93  void GamePlay::render(sf::RenderWindow& t_window)
94  {
95      t_window.draw(m_backgroundSprite);
96      m_player.render(t_window);
97  }
```