main.cpp

calls GameManager::start()

**CLASSES**

**GameManager**

gameManger.h

gameManager.cpp



public:

const static int SCREEN\_WIDTH = 1600;

const static int SCREEN\_HEIGHT = 1200;

static void start();

static void gameLoop();

enum State {Uninitialized, SplashScreen, State1, State2,ExitGame};

static void setState(State s);

static GameState \*getState();

private:

static State \_state;

static GameState\* \_currentState;

static GameState\* \_splashscreen;

static GameState \* \_stateOne;

static GameState \* \_stateTwo;

static sf::RenderWindow \_window;

static sf::Clock \_clock;

**GameState**

gameState.h

gameState.cpp

public:

virtual ~GameState();

virtual void init() = 0;

virtual void handleInput(sf::Event\* event) = 0;

virtual void update(float timeElapsed) = 0;

virtual void draw(sf::RenderWindow\* window) = 0;

virtual void endGameLogic() {};

**StateOne**

stateOne.h

stateOne.cpp



class StateOne : public GameState

{

void init() override;

void handleInput(sf::Event\* event) override;

void update(float timeElapsed) override;

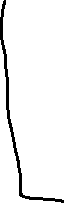
void draw(sf::RenderWindow\* window) override;

void endGameLogic() override;

};



**StateTwo**



stateTwo.h

stateTwo.cpp

class StateTwo : public GameState

{

void init() override;

void handleInput(sf::Event\* event) override;

void update(float timeElapsed) override;



void draw(sf::RenderWindow\* window) override;

void endGameLogic() override;

};



**SplashScreen**

splashscreen.h

splashscreen.cpp

class Splashscreen : public GameState

{

void init() override;

void handleInput(sf::Event\* event) override;

void update(float timeElapsed) override;

void draw(sf::RenderWindow\* window) override;

void endGameLogic() override;

};