**Added the screen.clear method before the gamestate render method to clean the screen before rendering.**

**package** sonar.gamestates;

**import** java.awt.Graphics;

**import** java.awt.image.BufferedImage;

**import** java.awt.image.DataBufferInt;

**import** sonar.Game;

**import** sonar.gamestates.states.Inventory;

**import** sonar.gamestates.states.MenuState;

**import** sonar.gamestates.states.PasswordState;

**import** sonar.gamestates.states.StarterStage;

**import** sonar.gamestates.states.levels.Screen;

**public** **class** GSM

{

//The GSM class that allows us to switch between different gamestates.

**private** Game game;

**private** GameState currentState, pastState;

**private** BufferedImage image;

**private** Screen screen;

**public** GSM(Game currentGame)

{

game = currentGame;

screen = **new** Screen(game.getWindowWidth(), game.getWindowHeight());

image = **new** BufferedImage(screen.getWidth(), screen.getHeight(), BufferedImage.***TYPE\_INT\_RGB***);

setState(StateHolder.***menuState***);

}

**public** **void** switchStates(GameState active, GameState passive)

{

**if**(pastState != **null**)

{

currentState = active;

pastState = passive;

}

}

//A gsm is in charge of loading and setting states

**private** **void** loadState(**int** state)

{

currentState = **null**;

**if**(state == StateHolder.***menuState***) currentState = **new** MenuState(**new** SingleStateBuilder(), "/textures/states/menu/Menu.png", "Menu", **this**);

**if**(state == StateHolder.***passwordState***) currentState = **new** PasswordState(**new** SingleStateBuilder(), "/textures/states/password/Password.png", "Password", **this**);

**if**(state == StateHolder.***starterStage***) currentState = **new** StarterStage(**new** DualStateBuilder(), "/textures/states/starter/Starter.png", "Starter", **this**);

**if**(currentState.getBuildState().stateType().equals("Dual")){pastState = **new** Inventory(**new** DualStateBuilder(), "/textures/states/inventory/Inventory.png", "Inventory", **this**);}

}

**public** **void** setState(**int** state)

{

loadState(state);

}

**public** **void** update()

{

currentState.update();

}

**public** **void** render(Graphics g)

{

**int** xScroll = 0;

**int** yScroll = 0;

screen.clear();

currentState.render(xScroll, yScroll, screen, g);

**int**[] pixels = ((DataBufferInt) image.getRaster().getDataBuffer()).getData();

**for**(**int** i = 0; i < pixels.length; i++)

{

pixels[i] = screen.getPixels()[i];

}

g.drawImage(image, 0, 0, game.getWidth(), game.getHeight(), **null**);

}

**public** Game getGame(){**return** game;}

**public** GameState getCurrentState(){**return** currentState;}

**public** GameState getPastState(){**return** pastState;}

}