**Changed the switchStates method from an instance method to a static method and also changed the StarterStage contructor to only take in the buildState variable located in the StarterStage class.**

**package** sonar.gamestates.states;

**import** java.awt.Graphics;

**import** sonar.GSM;

**import** sonar.GameState;

**import** sonar.Screen;

**import** sonar.StateBuilder;

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

**import** sonar.gamestates.states.levels.stages.entities.SpriteManager;

**import** sonar.gamestates.states.levels.stages.entities.animations.tiles.TileManager;

**public** **class** StarterStage **extends** GameState

{

//This is where the player first starts.

**private** LM lm;

**public** StarterStage(StateBuilder buildState)

{

**super**(buildState, path, identity, *gsm*);

lm = **new** LM("Starter", **this**);

}

**protected** **void** update()

{

**if**(getKey() == **null**)

{

initKey();

setSmanage(**new** SpriteManager(getIdentity()));

setTmanage(**new** TileManager(getSmanage()));

}

getKey().update();

lm.update();

**if**(getKey().start)

{

resetKeyboard();

resetSmanage();

resetTmanage();

//Reset the player input method here

lm.getCurrentLevel().getMmanager().starterMob.setPlayerInput(**null**);

GSM.*switchStates*(GSM.*getPastState*(), GSM.*getCurrentState*());

}

}

**protected** **void** render(**int** xScroll, **int** yScroll, Screen screen, Graphics g)

{

lm.render(xScroll, yScroll, screen);

}

}