**Added the lm.getCurrentLevel().getMmanager().starterMob.setPlayerInput(null) method call to the getKey.start if statement to allow the playerInput to be reset whenever we change directly to the inventory state.**

**package** sonar.gamestates.states;

**import** java.awt.Graphics;

**import** sonar.gamestates.GSM;

**import** sonar.gamestates.GameState;

**import** sonar.gamestates.Screen;

**import** sonar.gamestates.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, String path, String identity, GSM gsm)

{

**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**);

getGsm().switchStates(getGsm().getPastState(), getGsm().getCurrentState());

System.***out***.println("Entered the Inventory!");

}

}

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

{

lm.render(xScroll, yScroll, screen);

}

}