**Initialized the walking variable to true if movement is detected in the updateMovement method located in the CharacterMobBuilder class and also initialized the walking variable to false otherwise in the updateMovement method.**

**package** sonar.gamestates.states.levels.stages.entities.animations.mobs;

**import** sonar.gamestates.Keyboard;

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

**import** sonar.gamestates.states.levels.stages.entities.animations.DynamicAnimation;

**public** **class** MobManager

{

**public** Mob starterMob;

**public** MobManager(SpriteManager manage, String level)

{

**if**(level.equals("StarterLevel1")) buildStarterLevel1(manage);

**if**(level.equals("StarterLevel2")) buildStarterLevel2(manage);

**if**(level.equals("StarterLevel3")) buildStarterLevel3(manage);

**if**(level.equals("StarterLevel4")) buildStarterLevel4(manage);

}

**private** **void** buildStarterLevel1(SpriteManager manage)

{

//SonarBat Mob

starterMob = **new** SonarBat(**new** CharacterMobBuilder(40, 40));

starterMob.setUp(**new** DynamicAnimation(manage.sonarbatMobUp, manage.sonarbatMobUp1, manage.sonarbatMobUp2));

starterMob.setDown(**new** DynamicAnimation(manage.sonarbatMobDown, manage.sonarbatMobDown1, manage.sonarbatMobDown2));

starterMob.setLeft(**new** DynamicAnimation(manage.sonarbatMobLeft, manage.sonarbatMobLeft1, manage.sonarbatMobLeft2));

starterMob.setRight(**new** DynamicAnimation(manage.sonarbatMobRight, manage.sonarbatMobRight1, manage.sonarbatMobRight2));

starterMob.setCurAnim(starterMob.getUp());

starterMob.setPlayer(**true**);

}

**private** **void** buildStarterLevel2(SpriteManager manage)

{

}

**private** **void** buildStarterLevel3(SpriteManager manage)

{

}

**private** **void** buildStarterLevel4(SpriteManager manage)

{

}

}

**interface** MobBuilder

{

**int** getX();

**int** getY();

String getType();

**int** getMobDirection();

}

**class** CharacterMobBuilder **implements** MobBuilder

{

**private** **int** x, y;

**private** **boolean** player;

**private** DynamicAnimation up, down, left, right;

**private** **int** mobDirection;

**private** Keyboard playerInput;

**private** **boolean** walking;

CharacterMobBuilder(**int** x, **int** y)

{

**this**.x = x;

**this**.y = y;

player = **false**;

mobDirection = 0;

}

**void** setUp(DynamicAnimation animation){up = animation;}

**void** setDown(DynamicAnimation animation){down = animation;}

**void** setLeft(DynamicAnimation animation){left = animation;}

**void** setRight(DynamicAnimation animation){right = animation;}

**void** setPlayer(**boolean** value){player = value;}

DynamicAnimation getUp(){**return** up;}

DynamicAnimation getDown(){**return** down;}

DynamicAnimation getLeft(){**return** left;}

DynamicAnimation getRight(){**return** right;}

**boolean** getPlayer(){**return** player;}

**public** **int** getX(){**return** x;}

**public** **int** getY(){**return** y;}

**public** String getType(){**return** "Character";}

**public** **int** getMobDirection(){**return** mobDirection;}

**void** setMobDirection(**int** xa, **int** ya, **boolean** onLadder)

{

**if**(xa < 0) mobDirection = 3;

**if**(xa > 0) mobDirection = 1;

**if**(onLadder)

{

**if**(ya < 0) mobDirection = 0;

**if**(ya > 0) mobDirection = 2;

}

}

**void** move(**int** xa, **int** ya)

{

setMobDirection(xa, ya, **false**);

}

**void** setPlayerInput(Keyboard input)

{

**if**(getPlayer()) playerInput = input;

}

Keyboard getPlayerInput()

{

Keyboard input = **null**;

**if**(getPlayer()) input = playerInput;

**return** input;

}

**byte**[] playerControls(**int** xa, **int** ya, **boolean** onLadder)

{

**byte**[] movement = **null**;

**if**(getPlayer())

{

**if**(playerInput.left) xa--;

**if**(playerInput.right) xa++;

**if**(onLadder)

{

**if**(playerInput.up) ya--;

**if**(playerInput.down) ya++;

}

movement = **new** **byte**[2];

movement[0] = (**byte**) xa;

movement[1] = (**byte**) ya;

}

**return** movement;

}

**void** updateMovement(**int** xa, **int** ya)

{

**if**(xa != 0 || ya != 0)

{

move(xa, ya);

walking = **true**;

}

**else** walking = **false**;

}

}