**Added an if statement to check the mob to see if it is a Character type in the getPlayerInput method located in the mob class and also added the character local variable to the getPlayerInput method.**

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

**import** sonar.gamestates.Keyboard;

**import** sonar.gamestates.Screen;

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

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

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

**public** **abstract** **class** Mob **extends** Entity

{

**private** MobBuilder buildMob;

**private** DynamicAnimation curAnim;

Mob(MobBuilder buildMob)

{

**super**("Mob");

**this**.buildMob = buildMob;

}

**public** **abstract** **void** update();

**public** **void** render(Screen screen){screen.render(**this**, getX(), getY(), getWidth(), getHeight());}

**public** **int** getX(){**return** buildMob.getX();}

**public** **int** getY(){**return** buildMob.getY();}

**public** Sprite getSprite(){**return** curAnim.getSprite();}

**public** **int** getWidth(){**return** curAnim.getSprite().getWidth();}

**public** **int** getHeight(){**return** curAnim.getSprite().getHeight();}

String getMobType(){**return** buildMob.getType();}

**void** setCurAnim(DynamicAnimation curAnim){**this**.curAnim = curAnim;}

DynamicAnimation getCurAnim(){**return** curAnim;}

MobBuilder getBuildMob(){**return** buildMob;}

**private** DynamicAnimation direction(DynamicAnimation animation, String direction, String methodType)

{

**if**(buildMob.getType().equals("Character"))

{

CharacterMobBuilder character = (CharacterMobBuilder) buildMob;

**if**(methodType.equals("Set"))

{

**switch**(direction)

{

**case** "Left":

{

character.setLeft(animation);

**break**;

}

**case** "Right":

{

character.setRight(animation);

**break**;

}

**case** "Up":

{

character.setUp(animation);

**break**;

}

**case** "Down":

{

character.setDown(animation);

**break**;

}

}

animation = **null**;

}

**else**

{

**switch**(direction)

{

**case** "Left":

{

animation = character.getLeft();

**break**;

}

**case** "Right":

{

animation = character.getRight();

**break**;

}

**case** "Up":

{

animation = character.getUp();

**break**;

}

**case** "Down":

{

animation = character.getDown();

**break**;

}

}

}

}

**return** animation;

}

**private** **boolean** player(**boolean** value, String methodType)

{

**if**(buildMob.getType().equals("Character"))

{

CharacterMobBuilder character = (CharacterMobBuilder) buildMob;

**if**(methodType.equals("Set")) character.setPlayer(value);

**else** value = character.getPlayer();

}

**return** value;

}

**void** setUp(DynamicAnimation animation){direction(animation, "Up", "Set");}

**void** setDown(DynamicAnimation animation){direction(animation, "Down", "Set");}

**void** setLeft(DynamicAnimation animation){direction(animation, "Left", "Set");}

**void** setRight(DynamicAnimation animation){direction(animation, "Right", "Set");}

**void** setPlayer(**boolean** value){player(value, "Set");}

DynamicAnimation getUp()

{

DynamicAnimation animation = **null**;

**return** direction(animation, "Up", "Get");

}

DynamicAnimation getDown()

{

DynamicAnimation animation = **null**;

**return** direction(animation, "Down", "Get");

}

DynamicAnimation getLeft()

{

DynamicAnimation animation = **null**;

**return** direction(animation, "Left", "Get");

}

DynamicAnimation getRight()

{

DynamicAnimation animation = **null**;

**return** direction(animation, "Right", "Get");

}

**boolean** getPlayer()

{

**boolean** value = **false**;

**return** player(value, "Get");

}

**int** getMobDirection(){**return** buildMob.getMobDirection();}

**void** setX(**int** value)

{

**if**(buildMob.getType().equals("Character"))

{

CharacterMobBuilder character = (CharacterMobBuilder) buildMob;

character.setX(value);

}

}

**void** setY(**int** value)

{

**if**(buildMob.getType().equals("Character"))

{

CharacterMobBuilder character = (CharacterMobBuilder) buildMob;

character.setY(value);

}

}

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

{

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

**if**(buildMob.getType().equals("Character"))

{

CharacterMobBuilder character = (CharacterMobBuilder) buildMob;

movement = character.playerControls(xa, ya, onLadder);

}

**return** movement;

}

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

{

**if**(buildMob.getType().equals("Character"))

{

CharacterMobBuilder character = (CharacterMobBuilder) buildMob;

character.updateMovement(xa, ya);

}

}

**void** updateAnimation(**int** mobDirection)

{

**switch**(mobDirection)

{

**case** 0:

{

curAnim = getUp();

**break**;

}

**case** 1:

{

curAnim = getRight();

**break**;

}

**case** 2:

{

curAnim = getDown();

**break**;

}

**case** 3:

{

curAnim = getLeft();

**break**;

}

}

}

**boolean** getWalking()

{

**boolean** walking = **false**;

**if**(buildMob.getType().equals("Character"))

{

CharacterMobBuilder character = (CharacterMobBuilder) buildMob;

walking = character.getWalking();

}

**return** walking;

}

Keyboard getPlayerInput()

{

**if**(buildMob.getType().equals("Character"))

{

CharacterMobBuilder character = (CharacterMobBuilder) buildMob;

}

}

}