public class JumpClass2 extends ApplicationAdapter {

SpriteBatch batch; // Used to place textures on screen

Texture runSheet; //Texture used to hold run sprite sheet

Texture flipSheet; //Texture used to hold flip sprite sheet

float h = 0; // Stores height of device screen

private static final int FRAME\_COLS = 5;

private static final int FRAME\_ROWS = 5;

Animation<TextureRegion> runAnimation;

Animation<TextureRegion> jumpAnimation;

Animation<TextureRegion> flipRunAnimation;

Animation<TextureRegion> flipJumpAnimation;

TextureRegion[] runFrames;

TextureRegion[] jumpFrames;

TextureRegion[] flipRunFrames;

TextureRegion[] flipJumpFrames;

TextureRegion currentFrame;

TextureRegion jumpCurrentFrame;

TextureRegion flipRunCurrentFrame;

TextureRegion flipJumpCurrentFrame;

float stateTime;

Integer frameIndex;

Integer jumpFrameIndex;

Integer flipRunFrameIndex;

Integer flipJumpFrameIndex;

boolean isFliped = false;

float deltaTime=0;

int playerState = 0;

@Override

public void create () {

// SpriteBatch used to draw Sprite onto screen in render() method

batch = new SpriteBatch();

//run animation (Needs to be mostly in the render method)

runSheet = new Texture(Gdx.files.internal("sprite\_sheet.png"));

flipSheet = new Texture(Gdx.files.internal("flip\_sprite\_sheet.png"));

TextureRegion[][] tmp = TextureRegion.split(runSheet, runSheet.getWidth() / FRAME\_COLS, runSheet.getHeight() / FRAME\_ROWS);

TextureRegion[][] temp = TextureRegion.split(flipSheet, flipSheet.getWidth() / 5, flipSheet.getHeight() / 5);

runFrames = new TextureRegion[9];

jumpFrames = new TextureRegion[12];

flipRunFrames = new TextureRegion[10];

flipJumpFrames = new TextureRegion[13];

int index = 0;

int jumpIndex = 0;

for(int i=0; i<FRAME\_ROWS; i++){

for(int j=0; j<FRAME\_COLS; j++){

if(index < 9){

runFrames[index] = tmp[i][j];

}

if(index > 9 && index < 22 ){

jumpFrames[jumpIndex++] = tmp[i][j];

}

index++;

}

}

int flipIndex = 0;

int flipJumpIndex = 0;

for(int i=0; i<FRAME\_ROWS; i++){

for(int j=0; j<FRAME\_COLS; j++){

if(flipIndex < 10){

flipRunFrames[flipIndex] = temp[i][j];

}

if(flipIndex >= 10 && flipIndex < 23 ){

flipJumpFrames[flipJumpIndex++] = temp[i][j];

}

flipIndex++;

}

}

runAnimation = new Animation(1.5f, runFrames);

jumpAnimation = new Animation(1.5f, jumpFrames);

flipRunAnimation = new Animation(1.5f, flipRunFrames);

flipJumpAnimation = new Animation(1.5f, flipJumpFrames);

stateTime = 0f;

h = Gdx.graphics.getHeight();

}

@Override

public void render () {

stateTime += Gdx.graphics.getDeltaTime() \* 30;

currentFrame = runAnimation.getKeyFrame(stateTime, true);

jumpCurrentFrame = jumpAnimation.getKeyFrame(stateTime, true);

flipRunCurrentFrame = flipRunAnimation.getKeyFrame(stateTime,true);

flipJumpCurrentFrame = flipJumpAnimation.getKeyFrame(stateTime, true);

frameIndex = runAnimation.getKeyFrameIndex(stateTime);

jumpFrameIndex = jumpAnimation.getKeyFrameIndex(stateTime);

flipRunFrameIndex = flipRunAnimation.getKeyFrameIndex(stateTime);

flipJumpFrameIndex = flipJumpAnimation.getKeyFrameIndex(stateTime);

// Gdx.app.log("current time",Float.toString(stateTime));

// Gdx.app.log("current frame index",Integer.toString(frameIndex));

batch.begin();

if(checkTouch) {

if (touchSpace.overlaps(flipButtonRect)) {

//Do flip animation, move the hitbox of the character to the new location \*instantly\*

if(isFliped == false){

isFliped = true;

playerState = 1;

}else if(isFliped == true){

isFliped = false;

playerState = 0;

}

} else if (touchSpace.overlaps(jumpButtonRect) ) {

//Do jump animation, move the hitbox of the character up then down

playerState = 2;

}

}

//player state, switch between run, flip and jump

if(playerState == 0){

batch.draw(currentFrame, 20, h/2 + (platform.getHeight() / 2));

}else if(playerState == 1){

batch.draw(flipRunCurrentFrame, 20, h/2 - platform.getHeight());

}else if(playerState == 2 && isFliped == false){

batch.draw(jumpCurrentFrame, 20, h/2 + (platform.getHeight() / 2));

//play animation only once

if(jumpCurrentFrame.equals(jumpAnimation.getKeyFrame(jumpAnimation.getAnimationDuration()))){

playerState = 0;

}

}else if(playerState == 2 && isFliped == true){

batch.draw(flipJumpCurrentFrame, 20, h/2 - platform.getHeight());

if(jumpCurrentFrame.equals(jumpAnimation.getKeyFrame(jumpAnimation.getAnimationDuration()))){

playerState = 1;

}

}

batch.end();

}

@Override

public void dispose () {

batch.dispose();

background.dispose();

}

}