**Player Direction**

1. Determine Player Orientation

Establish the rule that the character always faces the mouse cursor.

1. Divide the Screen into Regions

See file “pseudocode\_image 1.jpg” in processing folder. Using two lines intersecting at (400, 400)—specifically, y=xy = xy=x and y=−x+800y = -x + 800y=−x+800—divide the screen into four regions. The direction can be determined by comparing values, such as mouseY <= mouseX && mouseY >= -mouseX + 800.

1. Set Player Direction Vector

Use a PVector to record the player's direction. For instance, when the conditions for facing right are met (e.g., mouseY <= mouseX && mouseY >= -mouseX + 800), set playerDirectionPV.x = 1 and playerDirectionPV.y = 0.

**Player Animation**

1. Declare an Image for Player Sprite

Use a PImage variable to store the character's sprite, and update it dynamically as needed.

1. Trigger Melee Animation

When the player presses the right mouse button (mousePressed and mouseButton == RIGHT) and the melee attack is not already active (meleeAttackBoolean == false), trigger the melee animation.

1. Define Animations for Four Directions

Write four separate if conditions for each direction. For example, if the player is facing down (playerDirectionPV.y == 1) and a melee attack is triggered (meleeAttackBoolean == true), use frameCount to manage the animation sequence. For instance:

For frames 0–1: display the first image (if (0 <= frameCount - startMeleeAttack && frameCount - startMeleeAttack <= 1)).

Repeat this logic for subsequent frames to complete the animation.

**Building Mechanic**

1. Declare a Boolean for Build Mode

Use a boolean variable to track whether build mode is active.

1. Enable Build Mode

Set the boolean to true when conditions are met (e.g., materials ≥ 20) and the player presses the R key.

1. Restrict Building to Build Mode

Use an if statement to ensure that building functions only execute when build mode is active.

1. Add Towers

When the player presses the spacebar, trigger the building behavior by adding a new tower object to the towerArrayList.