

```

package MoareAI.Class;

import net.minecraft.src.*;

//Created by MoareAI

public class GuiDigitalMisc extends GuiScreen
{
    public GuiDigitalMisc(TileEntityDigitalMisc tileentity)
    {
        super();
        xSize = 124;
        ySize = 63;
        Function = tileentity.Type;
        if (Function == 3)
        {
            Value = tileentity.ClockTime;
            CurrentValue = tileentity.ClockTime;
        }
        if (Function == 4 || Function == 2)
        {
            State = tileentity.State[2];
            CurrentState = tileentity.State[2];
        }
        if (Function == 6)
        {
            Value = tileentity.PulseLength;
            CurrentValue = tileentity.PulseLength;
        }
        if (Function == 7)
        {
            Value = tileentity.Count;
            CurrentValue = tileentity.State2;
        }
        EntityDigital = tileentity;
        FunctionName = ItemDigitalMisc.FunctionName[Function];
    }

    //Mouse click
    protected void mouseClicked(int i, int j, int k)
    {
        super.mouseClicked(i, j, k);
        if(Mouse.isButtonDown(0))
        {
            if (Keyboard.isKeyDown(42) || Keyboard.isKeyDown(54))
                Value = Value+mod_DigitalFunctions.GUIStep;
            else
                Value++;
            State = !State;
        }
        else
            if(Mouse.isButtonDown(1))
            {
                if (Keyboard.isKeyDown(42) || Keyboard.isKeyDown(54))
                {
                    if (Value > mod_DigitalFunctions.GUIStep-1)
                    {
                        Value = Value-mod_DigitalFunctions.GUIStep;
                    }
                }
                else
                    if (Value > 0)
                    {
                        Value--;
                    }
            }
        }
    }

```

```

    }
    State = !State;
}

//Display
public void drawScreen(int j, int l, float f)
{
    int i = mc.renderEngine.getTexture("/MoareAI/Gui/Neutral.png");
    GL11.glColor4f(1.0F, 1.0F, 1.0F, 1.0F);
    mc.renderEngine.bindTexture(i);
    int j1 = width - xSize >> 1;
    int k = height - ySize >> 1;
    drawTexturedModalRect(j1-14, k-8, 0, 0, 176, 4);
    drawTexturedModalRect(j1+162, k-8, 229, 0, 4, 37);
    drawTexturedModalRect(j1-14, k-4, 0, 6, 4, 37);
    drawTexturedModalRect(j1-11, k+29, 56, 39, 177, 4);
    drawTexturedModalRect(j1-10, k-4, 4, 4, 172, 33);
    fontRenderer.drawString((new StringBuilder()).append("Parameters for
").append(FunctionName).toString(), j1-7, k, 0x404040);
    if (Function != 4 && Function != 2)
    {
        fontRenderer.drawString((new
StringBuilder()).append(FunctionDesc[Function]).append(": ").append(Value).toString(),
j1-7, k+9, 0x404040);
        fontRenderer.drawString((new StringBuilder()).append("Current value:
").append(CurrentValue).toString(), j1-7, k+18, 0x404040);
    }
    else
    {
        fontRenderer.drawString((new
StringBuilder()).append(FunctionDesc[Function]).append(": ").append(State).toString(),
j1-7, k+9, 0x404040);
        fontRenderer.drawString((new StringBuilder()).append("Currently Mirrored:
").append(CurrentState).toString(), j1-7, k+18, 0x404040);
    }
}

//Pause the game while doing this
public boolean doesGuiPauseGame()
{
    return true;
}

//Keyboard
protected void keyTyped(char c, int i)
{
    super.keyTyped(c, i);
    if (i==mc.gameSettings.keyBindRight.keyCode ||
i==mc.gameSettings.keyBindForward.keyCode || Keyboard.getEventKey() == 78)
    {
        if (Keyboard.isKeyDown(42) || Keyboard.isKeyDown(54))
        {
            Value = Value+mod_DigitalFunctions.GUIStep;
        }
        else
        {
            Value++;
        }
        State = !State;
    }
    else
    if (i==mc.gameSettings.keyBindLeft.keyCode ||
i==mc.gameSettings.keyBindBack.keyCode || Keyboard.getEventKey() == 74)

```

```

{
    if (Keyboard.isKeyDown(42) || Keyboard.isKeyDown(54))
    {
        if (Value > mod_DigitalFunctions.GUIStep-1)
        {
            Value = Value-mod_DigitalFunctions.GUIStep;
        }
    }
    else
    if (Value > 0)
    {
        Value--;
    }
    State = !State;
}
else
if (Keyboard.getEventKey() == 19)
{
    if (Keyboard.isKeyDown(42) || Keyboard.isKeyDown(54))
    {
        Value = CurrentValue;
        State = CurrentState;
    }
    else
    {
        if (Function == 3)
            Value = mod_DigitalFunctions.ClockTime;
        if (Function == 6)
            Value = mod_DigitalFunctions.PulseGenLength;
        if (Function == 7)
            Value = mod_DigitalFunctions.DefaultCount;
        if (Function == 4 || Function == 2)
            State = false;
    }
}
else
if (Keyboard.isKeyDown(42) || Keyboard.isKeyDown(54))
{
}
else
{
    if (Function == 3)
        EntityDigital.ClockTime = Value;
    if (Function == 6)
        EntityDigital.PulseLength = Value;
    if (Function == 7)
        EntityDigital.Count = Value;
    if (Function == 4 || Function == 2)
        EntityDigital.State[2] = State;
    mc.thePlayer.closeScreen();
}
}

private int CurrentValue;
private int Value;
private boolean State;
private boolean CurrentState;
private int Function;
private String FunctionDesc[] = {null, null, "Mirrored", "Pulsewidth", "Mirrored",
null, "Pulselength", "Counter value"};
private String FunctionName;
private TileEntityDigitalMisc EntityDigital;
private int xSize;

```

GuiDigitalMisc.java

```
private int ySize;
public static final int BORDER = 4;
public static final int GRIDX = 5;
public static final int GRIDY = 6;
public static final int CRAFTX = 99;
public static final int CRAFTY = 24;
public static final int IMGWIDTH = 176;
public static final int IMGHEIGHT = 166;
public static final int IMGIDX = 29;
public static final int IMGIDY = 15;
public static final int MIDWIDTH = 116;
public static final int MIDHEIGHT = 55;
}
```