

```

package net.minecraft.src;

import java.io.*;

//Created by MoareAI

public class GuiPulseClock extends GuiScreen
{
    public GuiPulseClock()
    {
        super();
        xSize = 124;
        ySize = 63;
    }

    //Mouse click
    protected void mouseClicked(int i, int j, int k)
    {
        super.mouseClicked(i, j, k);
        if(Mouse.isButtonDown(0))
        {
            mod_MiscDigital.ClockTime++;
        } else
        if(Mouse.isButtonDown(1))
        {
            if (mod_MiscDigital.ClockTime>1)
                mod_MiscDigital.ClockTime--;
        }
    }

    //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+25, k+16, 0, 0, 70, 4);
        drawTexturedModalRect(j1-25+120, k+16, 172, 0, 4, 28);
        drawTexturedModalRect(j1+25, k + 20, 0, 139, 4, 26);
        drawTexturedModalRect(j1+29, k + 43, 106, 162, 70, 4);
        drawTexturedModalRect(j1+29, k + 20, 29, 15, 66, 24);
        fontRenderer.drawString("Pulsewidth", j1+37, k+24, 0x404040);
        fontRenderer.drawString(new
StringBulder()).append(mod_MiscDigital.ClockTime).toString(), j1+56, k+33, 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)
        {
            mod_MiscDigital.ClockTime++;
        }
        else
    }

```

```

        if (i==mc.gameSettings.keyBindLeft.keyCode ||
i==mc.gameSettings.keyBindBack.keyCode)
        {
            if (mod_MiscDigital.ClockTime>1)
                mod_MiscDigital.ClockTime--;
        }
        else
        {
            cfgfile = mod_MiscDigital.cfgfile;
            try
            {
                cfgfile.getParentFile().mkdirs();
                if(cfgfile.exists() || cfgfile.createNewFile())
                {
                    Properties properties = new Properties();
                    if(cfgfile.canWrite())
                    {
                        FileOutputStream fileoutputstream = new
FileOutputStream(cfgfile);
                        properties.setProperty("PulseLength",
Integer.toString(mod_MiscDigital.PulseGenLength));
                        properties.setProperty("ClockPulseWidth",
Integer.toString(mod_MiscDigital.ClockTime));
                        properties.setProperty("BlockToggleID",
Integer.toString(mod_MiscDigital.BlockToggleID));
                        properties.setProperty("BlockRSLatchID",
Integer.toString(mod_MiscDigital.BlockRSID));
                        properties.setProperty("BlockPulseClockID",
Integer.toString(mod_MiscDigital.BlockClockID));
                        properties.setProperty("BlockHalfAdderID",
Integer.toString(mod_MiscDigital.BlockHalfAdderID));
                        properties.setProperty("BlockJKFlipFlopID",
Integer.toString(mod_MiscDigital.BlockJKFlipFlopID));
                        properties.setProperty("BlockPulseGeneratorID",
Integer.toString(mod_MiscDigital.BlockPulseGenID));
                        properties.setProperty("ItemToggleID",
Integer.toString(mod_MiscDigital.ItemToggleID));
                        properties.setProperty("ItemRSLatchID",
Integer.toString(mod_MiscDigital.ItemRSID));
                        properties.setProperty("ItemPulseClockID",
Integer.toString(mod_MiscDigital.ItemClockID));
                        properties.setProperty("ItemHalfAdderID",
Integer.toString(mod_MiscDigital.ItemHalfAdderID));
                        properties.setProperty("ItemJKFlipFlopID",
Integer.toString(mod_MiscDigital.ItemJKFlipFlopID));
                        properties.setProperty("ItemPulseGeneratorID",
Integer.toString(mod_MiscDigital.ItemPulseGenID));
                        properties.store(fileoutputstream, "MoareAI's Misc Digital
Functions Configurations");
                        fileoutputstream.close();
                    }
                }
            }
            catch(Throwable throwable) { }
            mc.thePlayer.func_20059_m();
        }
    }

    public File cfgfile;
    private int xSize;
    private int ySize;
    public static int time = 16;
    public static final int BORDER = 4;
    public static final int GRIDX = 5;

```

GuiPulseClock.java

```
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 IMGMDX = 29;
public static final int IMGMDY = 15;
public static final int MIDWIDTH = 116;
public static final int MIDHEIGHT = 55;
}
```