

# Torque Game Builder – Whack-A-Mole Tutorial Part 5

## Part 5 - If I had a Hammer...

### *Drag in your hammer*

The last thing that is missing is the hammer. Drag the hammer animation from Animated Sprites tab into your level. Click on it and set it's Name property to moleHammer (this is found under the Edit->Scripting tab).

Then put the following function in moleLevel.cs:

```
function SceneWindow2D::onMouseMove(%this, %mod, %worldPosition)
{
    moleHammer.setPosition(%worldPosition);
}
```

*Code Sample 5.1*

This will make the hammer sprite follow the mouse cursor. Let's also hide the regular cursor. Add this line to startGame() in game.cs:

```
function startGame(%level)
{
    //exec game scripts
    exec("./mole.cs");
    exec("./moleLevel.cs");

    //exec our gui files
    exec("~/gui/endScreen.gui");

    // Set The GUI.
    Canvas.setContent(mainScreenGui);
    Canvas.setCursor(DefaultCursor);

    // Hide the cursor
    Canvas.hideCursor();

    moveMap.push();

    sceneWindow2D.setUseObjectMouseEvents( true );

    if( isFile( %level ) || isFile( %level @ ".dso" ))
    {
        sceneWindow2D.loadLevel(%level);
        $levelForRestart = %level;
    }
}
```

*Code Sample 5.2*

# Torque Game Builder – Whack-A-Mole Tutorial Part 5

## ***Display the end screen***

For the end-screen-menu we want it to be shown again... replace your old function in moleLevel.cs with this

```
function moleLevel::onUpdateScene(%this)
{
    // Calculate how much time is left
    %timeLeft = $MOLE_LEVEL::timePerGame - %this.getSceneTime();

    if( %timeLeft < 0 )
    {
        %this.setScenePause( true );
        Canvas.pushDialog( MoleEndScreenGui );
        Canvas.showCursor();
    }

    // Round it to full seconds
    %timeLeft = mFloor( %timeLeft + 0.5 );

    // Update the GuiTextCtrl
    MoleGuiTimer.setText( %timeLeft );
}
```

*Code Sample 5.3*

...and when the game ends replace your old endGame function in game.cs with this one:

```
function endGame()
{
    Canvas.showCursor();
    sceneWindow2D.endLevel();
    moveMap.pop();
}
```

*Code Sample 5.4*

And lastly we want the hammer to strike if a mole is hit. Modify mole::onMouseDown():

```
function mole::onMouseDown(%this, %modifier, %worldPosition, %mouseClicks)
{
    if(%this.getAnimationName() $= ("animMoleComeOut" @ %this.moleColor) )
    {
        %this.playAnimation( "animMoleWhacked" @ %this.moleColor );

        //let the hammer strike, with animation auto-restore
        moleHammer.playAnimation( "animHammer", true );
    }
}
```

*Code Sample 5.5*

And now, at long last, we are finally finished! Test it out and enjoy!