

```

import java.awt.*;
import java.awt.event.KeyEvent;

import game2D.*;

// GCTester demonstrates how we can override the GameCore class
// to create our own 'game'. We usually need to implement at
// least 'draw' and 'update' (not including any local event handling)
// to begin the process. You should also add code to the 'init'
// method that will initialise event handlers etc. By default GameCore
// will handle the 'Escape' key to quit the game but you should
// override this with your own event handler.

/**
 * @author David Cairns
 *
 */
public class GCTileMapDemo extends GameCore {

    // It is often useful to move the whole screen image off a bit
    // and it is helpful if store these values so they are available
    // to the rest of the class
    int xoffset = 32; // Move tile map 32 pixels to the right
    int yoffset = 64; // Move tile map 64 pixels down

    long total; // Total time elapsed
    TileMap tmap = new TileMap(); // Our tile map, note that we load it in init()

    /**
     * The obligatory main method that creates
     * an instance of our GCTester class and
     * starts it running
     *
     * @param args The list of parameters this program might use (ignored)
     */
    public static void main(String[] args) {

        GCTileMapDemo gct = new GCTileMapDemo();
        gct.init();
        // Start in windowed mode at 736x256 pixels
        gct.run(false, 736, 256);
    }

    /**
     * Initialise the class, e.g. set up variables, load images,
     * create animations, register event handlers etc.
     */
    public void init()
    {
        total = 0;

        // Load the tile map and print it out so we can check it is valid
        tmap.loadMap("maps", "example-map.txt");
        System.out.println(tmap);
    }

    /**
     * Draw the current state of the game
     */
    public void draw(Graphics2D g)
    {
        // Be careful about the order in which you draw objects - you
        // should draw the background first, then work your way 'forward'
        g.setColor(Color.white);
        g.fillRect(0, 0, xoffset + 704, yoffset + 192);

        // Draw the tile map
        tmap.draw(g, xoffset, yoffset);

        // Show the 'score'
        g.setColor(Color.black);
        g.drawString("Time Expired: " + total, xoffset, yoffset);
    }
}

```

```

}

/**
 * Update any sprites and check for collisions
 *
 * @param elapsed The elapsed time between this call and the previous call of elapsed
 */
public void update(long elapsed)
{
    total += elapsed;
    if (total > 10000) stop();
}

/**
 * Override of the keyPressed event defined in GameCore to catch our
 * own events
 *
 * @param e The event that has been generated
 */
public void keyPressed(KeyEvent e)
{
    // Did the user press the 'C' key?
    if (e.getKeyCode() == KeyEvent.VK_C)
    {
        // Change some tile map entries
        tmap.setTileChar('.', 3, 2);
        tmap.setTileChar('c', 4, 2);
    }

    if (e.getKeyCode() == KeyEvent.VK_M)
    {
        // Load a different tile map
        tmap.loadMap("maps", "level2-map.txt");
    }
    if (e.getKeyCode() == KeyEvent.VK_UP) yoffset -= 5;
    if (e.getKeyCode() == KeyEvent.VK_DOWN) yoffset += 5;
}
}

```