**Final changes made to the RC candidate so it is a bit more stable. Simple is better. BufferStrategy kept at 2.**

**package** sonar;

**import** java.awt.Canvas;

**import** java.awt.Graphics;

**import** java.awt.image.BufferStrategy;

**import** javax.swing.JFrame;

**import** sonar.gamestates.GSM;

**public** **class** Game **extends** Canvas

{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** **short** width, height;

**private** **byte** scale;

**private** JFrame frame;

**private** **boolean** running;

**private** GSM gsm;

Game(**short** width, **short** height, **byte** scale)

{

**this**.width = width;

**this**.height = height;

**this**.scale = scale;

frame = **new** JFrame();

}

**void** start()

{

**if**(running) **return**;

running = **true**;

gsm = **new** GSM(**this**);

run("SonarBat");

}

//Observers

**private** **void** run(String title)

{

**double** delta = 0;

**byte** updates = 0;

**short** frames = 0;

**byte** ticks = 60;

**double** ns = 1000000000 / ticks;

BufferStrategy bs = **null**;

**long** renderTime = System.*currentTimeMillis*();

**long** updateTime = System.*nanoTime*();

**do**

{

//Perform the gameWorld world functions

**long** now = System.*nanoTime*();

delta += (now - updateTime) / ns;

updateTime = now;

**if**(delta >= 1)

{

update();

delta--;

updates++;

}

render(bs);

frames++;

//Display the game's title every 1 second.

**if**(System.*currentTimeMillis*() - renderTime >= 1000)

{

frame.setTitle(title + " | ups: " + updates + " fps: " + frames);

updates = 0;

frames = 0;

renderTime += 1000;

}

**if**(bs == **null**) createBufferStrategy(2);

bs = getBufferStrategy();

}

**while**(running);

}

**private** **void** update()

{

gsm.update();

}

**private** **void** render(BufferStrategy bs)

{

**if**(bs == **null**) **return**;

Graphics g = bs.getDrawGraphics();

gsm.render(g);

g.dispose();

bs.show();

}

JFrame getFrame(){**return** frame;}

**public** **short** getWindowWidth(){**return** width;}

**public** **short** getWindowHeight(){**return** height;}

**byte** getWindowScale(){**return** scale;}

}