

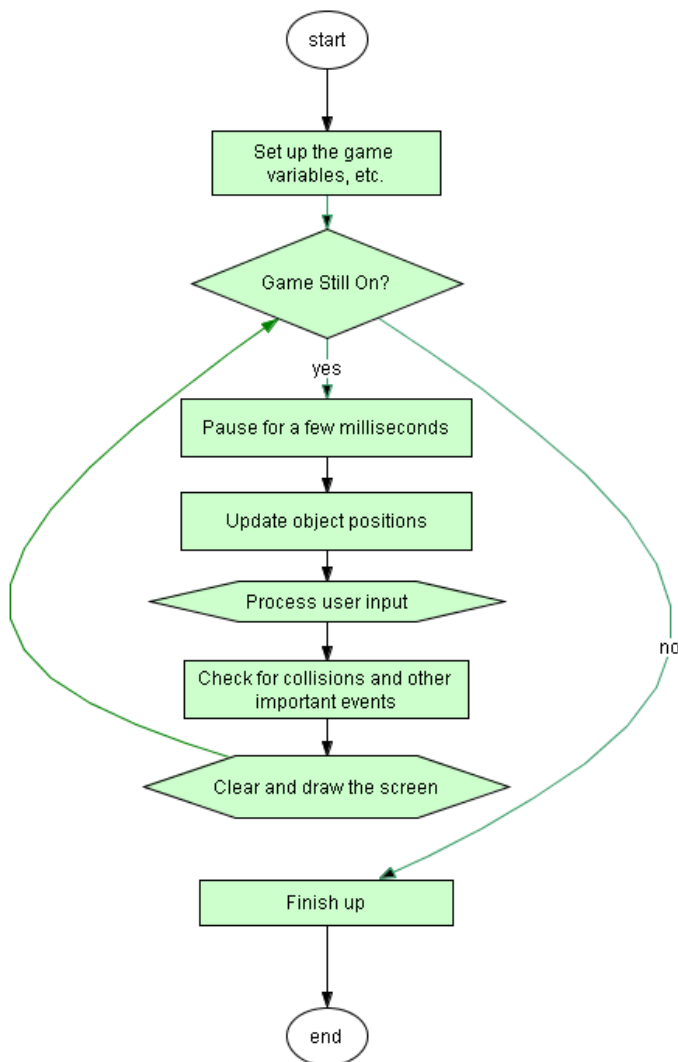
The GameEngine Design Pattern

So you want to program a live action game? You need three things - better input methods, control over screen refresh, and a new design pattern.

1. The GameEngine Design Pattern

A game engine is a loop. You should loop about 60 times per second. Inside the loop, you pause, update everything, respond to input, check for collisions and other special events, then redraw everything and loop back.

Whenever you can, you should put parts of the game loop code into methods (e.g. a method to detect collisions, a method to draw each object, etc.)



See **CrappyGame.java** for an example of this design pattern in action.

The basic idea is simple, but getting the details right can be very tricky!

2. Control Over Screen Refresh

By default, the **hsafx** window is refreshed 60 times per second. But this refresh can be out of synch with your game, and this can cause the screen to flicker.

c.autoRefreshOff()

Call this at the start of the game to turn off the auto refresh. With auto refresh turned off, nothing you draw will show up until you refresh the console.

c.refresh()

Call this whenever you're finished drawing a picture on the window. It will cause everything you have drawn since the last refresh to show up.

c.autoRefreshOn()

You probably won't need this, but if you ever use **c.nextInt**, **c.nextLine**, or other input methods, you'll have to turn the auto refresh back on to see the flashing cursor. Don't forget to turn it back off when the game resumes.

3. Some Better Input Methods

If you're doing **keyboard input**, you need methods that tell you what keys are being pressed right now without pausing the action and waiting for the user.

If you're doing **mouse input**, you need methods that tell you where the mouse is right now and what buttons are being held down.

See the code in **GameInputExamples.zip** for examples of all the required methods in action.

In BlueJ, double-click the **hsafx** package, then open **Console.java** in the editor, and change "Source Code" to "Documentation" to see all the methods you can use.