

Parallel Collections and Actor Threads

2-3-2012

Opening Discussion

- Minute essay comments
 - Using threads in projects.
 - Units and threads is a great place to use today's topic.
 - Drawing is not parallel. Graphics is a bottleneck.
 - Networking will automatically add an extra thread because of there is a blocking call that waits for new users.

GUIs are Multithreaded

- All events, including repainting, happen in an event thread.
- If you block the event thread, the GUI becomes non-responsive.
- Anything that will take much time should go in a separate thread.

Parallel Collections

- Scala 2.9 introduced `scala.collection.parallel`.
- The methods of these collections do their work in parallel.
- Convert from regular collections to parallel ones by calling the “`par`” method.
- Convert back with “`seq`”.
- Not all collections convert efficiently.

Actor Threads and Futures

- The `scala.actors` library provides an alternative threading model we will explore in depth later on.
- For now there are two methods that simplify launching threads.
 - `Actor.actor(body: => Unit):Actor`
 - `Futures.future[T](body: => T):Future[T]`
- Use the first to launch code in a thread. Use the second if you want a return value.

Code

- Let's write some.

Minute Essay

- Questions about parallel before we move on to streams?