Gather round fam, come, have a seat
We have stories to tell, and this one's a treat.
Consider a minnow, it wants to survive,
But in shark-fested waters it can be hard to strive.
But these little minnows, eaten they weren't,
They're all here together, in a game that's concurrent.
So the problem here that we wish to address:
So many fishies, not even one less.
And all of these guys, all of them stored
Each trying to access the same very board.
An issue like this, two ways we can go:
The first is called Erlport; the second, Pyro.
But with Erlport you see,
Things get quite tricky,
This intermediate step
Which we cannot forget
We've got all these fishies, all swimming around
Need to move quick; let's not be sharkbound.
And these little fishies, they're in such a fervor
Don't wanna take time to talk with the server.
But fish to gen server to board is the sport,
If we decided to go with Erlport.
But now listen closely: for Pyro you'll hear,
Straight from fishies to board? The choice here is clear!
There can be no question, there can be no debate,
We've resolved this argument, it need not get late.
For Pyro is simple, it's nice, it will work,
Our fishies will swim, the sharks they can lurk,
Our implementation, Mark will admire.
The reason: well, it's cuz Pyro is fire.

For our project, we are building a concurrent version of the classic Sharks and Minnows game. Each user controls a different minnow on a single board. The goal is to not get eaten and thus stay alive the longest. The main issue we needed to tackle was how to allow multiple users to simultaneously access the same board. The two ways we considered addressing this were with Erlport and Pyro. Ultimately, we chose to use Pyro since we thought it would be simpler. Whereas Erlport would add in an extra layer to the implementation, since the users would have to communicate with the gen server, which in turn would have to communicate with the board, Pyro allows the various users to communicate with the board directly. In addition, Pyro will be fire.