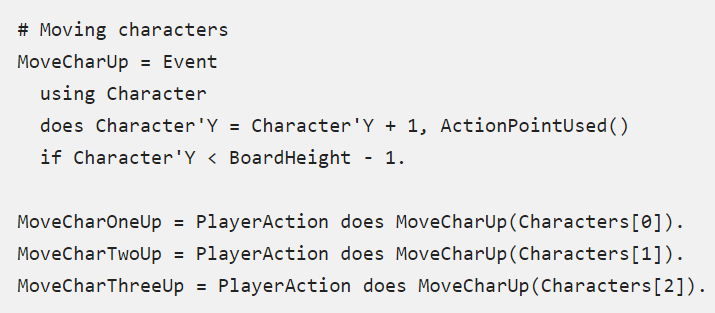
What to change in 0.0.5

# Big Problems

1. Duplication of behavior.
   * 
   * Would be nice to be able to say, characters can move. And only specify how characters move, not need to specify how they move for each character.
   * But still need that option, some characters might move uniquely.
   * Not sure how to solve this yet.
2. Cannot intuitively add or delete data within a PlayerAction / Event.
   * Such as with Yahtz, how to add another coin in runtime?
   * Solved this one by adding RemoveAt, Add and Length members to all arrays.

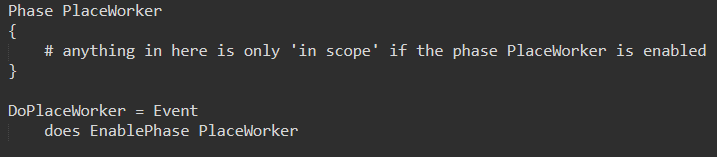
# Potential Solutions

1. Duplication of behavior.
2. Adding / Removing data.

These two maybe can be solved together, by one thing maybe?

1. Constraints / Rules
   1. What are they?
      1. Things which are always evaluated every PlayerAction.
      2. Kinda like rules.
      3. Evaluated at the end of the PlayerAction. Top to bottom in text file. If order really matters, then don’t use a constraint, bake the constraint into the PlayerAction.
      4. Might be good since it’s possibly closer to how designers think.
   2. This would reorganize and compress the behavior, but won’t reduce duplication, only reduces duplication of boilerplate code. For example above, would still need to specify the movement for each character
2. Can do operations on arrays as a whole.
   1. Specify an operation, and it will be done on each index in the array.
   2. This doesn’t really work cleanly. What about operations on more than one array? If the operations are always combinatorial then what about when the indexes are the same? Does the index operate on itself? I can’t think of an intuitive way to do this yet.
   3. Recursion might be the only solution.
   4. Always force elementwise? Is that even useful?
      1. Operations must be on two arrays of the same size.
3. Simply need a syntax for creating and deleting things from arrays.

# Things that might be nice

1. Create a more oop like system. So the behavior and state changes are created together.
   1. Maybe this is already possible. Just need to think about the code differently. Might not need to change the language at all.
2. Phase control
   1. Don’t need this in 0.0.5. But potentially will later. If there are a lot of phase control variables, (things such as DoingMoveWorker = true, or DoingLevelUp = true) like I suspect there might be, then this would be necessary.
   2. Can enable and disable as many phases as desired.
   3. 
   4. Can also query if a state is enabled or not. IsPhaseEnabled.