**Element game breakdown**

Throughout this document I will be using the term ‘element’ to refer to any entity that has come from or was made from anything coming out of an Input. This can be a pure element (water) or a compound element (mud, etc.).

What does the player do, how does the game work?

The player has to place down modular assets to create a production line that can take in determined input to create a desired output.

Core game loop

* Player receives input from the wall
* Player uses modular tools to build a production line
* Player tests line, iterates and makes changes until the desired output is created

Optionally, player may choose to optimize their implementation.

Input

Within each level, there are four inputs.

* Air
* Water
* Earth
* Fire

Using these four elements, anything can be created.

Elements

Elements have their own purity levels. Purity level ranges from 1 – 0. When an element comes in via input, it has a purity level of 1.

Elements can conflict with each other, when an element comes too close to another element on the production line, the purity levels of these elements fall. If an elements purity level reaches 0, the element is destroyed.

The final purity level of the output could be a factor in determining the success of the implementation.

Tools

The player has a variety of different tools available to them:

Movement tools

* Energy path  
  Essentially conveyer belts. They move the element from A to B.

Combination / separation tools

* Combine tool  
  Combines two or more elements together. (E.g. water + earth -> mud).
* Separator tool  
  Separates an element into its ingredients. (E.g. mud -> water + earth).

Conversion tools

* Converter  
  Can be used to convert one element to another. Certain elements can only be converted into certain things and this can come at a cost (e.g. the purity level of the element).

Stats management tools

* Purifier  
  An element passing through a purifier will have their purity level raised. This should be costly.

Game challenges

**Space**

It is possible to create very inelegant solutions to a problem, they are large and not space efficient. You may come to later stages where you realise in order to build a part of a larger solution you can use a smaller system you’ve already built, however it’s too large and thus will require optimising.

There is also the issue of conflicting elements, you must be sure that conflicting elements do not route too close to each other or their purity levels will drop.

**Cycle efficiency / time**

This is another factor that can separate an efficient from an inefficient solution. You may find a solution that takes 15 cycles to complete whilst there is another solution that only requires 9. This challenge allows for replay-ability to try to find the most efficient solution.

Players must also ensure that their cycles sync up: two elements must enter a combiner on the same tick.

Bibliography

Infinifactory: <http://www.zachtronics.com/infinifactory/>

SpaceChem: <http://www.zachtronics.com/spacechem/>

Poly Bridge: <http://polybridge.drycactus.com/>

Little alchemy: <https://littlealchemy.com/>

Brown, M. (2015) *Puzzle Solving… or Problem Solving? | Game Maker’s Toolkit*. [Video] Available online: <https://www.youtube.com/watch?v=w1_zmx-wU0U> [Date of access: 05 October 2017]