Game rules and functionality

General rules

- Rectangular playing field of variable size
- Rounds
  - At a start of every round the "Start of round" action (see below) of each object is executed in random order
  - At the end of every round the "End of round" action (see below) of each object is executed in random order
- Coordinates
  - Upper left: (0,0)
  - Lower right: (width-1, height-1)
- Objects
  - Can have subtypes
  - Either environment or buildings
  - Position defined by
    - upper left corner (most)
    - center (conveyors type 0-3, combiner)
    - center left or center top (conveyors type 4-7)
    - top left of thick part (miners)
  - Cannot overlap (except conveyors, which can cross)
- Buildings
  - Inputs: +
  - Outputs: -
  - Inert: o
- Points
  - Depends on the product produced
- Goal
  - Maximize points
  - Minimize needed rounds
- Program communication
  - Input and output use JSON (RFC 7159)
  - Terminates with a new line \n
  - Comments are not allowed
  - Output is a JSON list of objects, defining type, subtype (if applicable) and position E.g.: `{"type":"mine","x":3,"y":7,"subtype":1}`
  - Execution
    - Docker file (built and started under Ubuntu 22.04 LTS)
    - Resources available
      - CPUs: 2
      - Memory: 2G
      - Swap: 2G
    - The input defines the maximum calculation time in seconds t

Elements

Material

Element	Property	Description
Resource	Subtypes	8
Product	Subtypes	8
	Recipes	"resources" integer list, at index i is the needed quantity n of product i
	Points	"points", an integer
	Example JSON	`{"type":"product", "subtype":0..7, "resources": [0..,0..,0..,0..,0..,0..,0..,0..], "points":1..}`

Buildings

Element	Property	Description
Mine	Subtypes	4
	Size	3x4 or 4x3
	Connections	Inputs of Conveyors, Joiners or Factories
	Start of round	Resources at input are accepted
	End of round	Accepted resources are made available at output
	Example JSON	`{"type":"mine", "subtype":0..3, "x":0.., "y":0..}`
Conveyor	Subtypes	8
	Size	Types 0-3: 1x3 or 3x1; Types 4-7: 1x4 or 4x1
	Placement	Subtypes 4-7 can cross
	Start of round	Resources at input are accepted
	End of round	Accepted resources are made available at output
	Example JSON	`{"type":"conveyor", "subtype":0..7, "x":0.., "y":0..}`
Combiner	Subtypes	4
	Size	3x3
	Begin of round	Resources at input are accepted
	End of round	Accepted resources are made available at output
	Example JSON	`{"type":"combiner", "subtype":0..3, "x":0.., "y":0..}`
Factory	Subtypes	8
	Size	5x5

Element	Property	Description
	Start of round	Resources at input are accepted
	End of round	Produces as many products as needed resources were accepted
	Example JSON	`{"type":"factory", "subtype":0..7, "x":0.., "y":0..}`

Environment

Element	Property	Description
Deposit	Subtypes	8
	Connection	Mines only
	Quantity	width * height * 5
	End of round	Transfers 3 resources of type to each mine connected to it in random order
	Example JSON	`{"type":"deposit", "subtype":0..7, "x":0.., "y":0.., "width":1.., "height":1..}`
Obstacle	Subtypes	None
	Example JSON	`{"type":"obstacle", "x":0.., "y":0.., "width":1.., "height":1..}`

Remarks

- Transportation length does make a difference, more rounds are needed
- Outputs of mines, conveyors can only be connected to one other object
- Multiple mines can be attached to a deposit