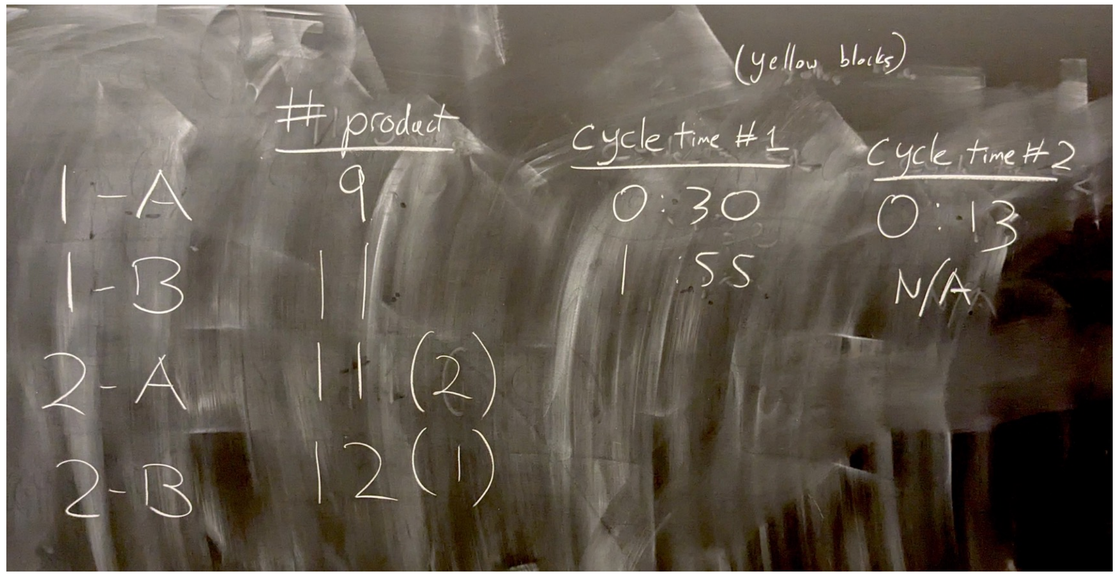
Manufacturing Systems III

System Design and Simulation #2: Lean Game

LEGO Demo

	A. Buffer = 2	B. Buffer = 10
Scenario 1 unreliable machines	✅	✅
Scenario 2 detecting defects	✅	✅

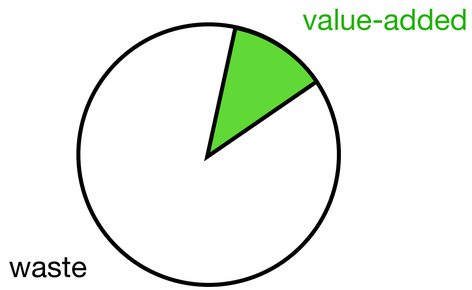


Rules

- assemble at a **consistent rate**
- no pre-building
- M1 stops when buffer is full
- M2 can only build with parts from the buffer

Different Types of Efficiency/Productivity

People Students



engaged in producing/assembling learning

watching machine running

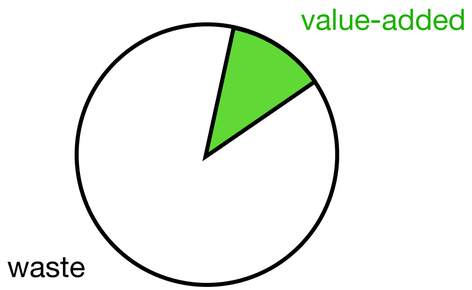
producing defects

looking for tools

fixing machine breakdowns

producing unnecessary items

Materials



being transformed into parts

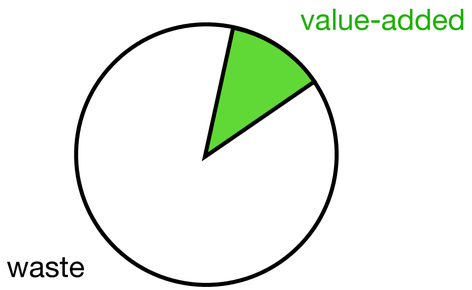
being transported

sitting in storage/buffers

being inspected

getting reworked

Machines



transforming parts

unnecessary movements

being set up

being broken down

producing defecting parts

producing parts that are not needed

plant manager: tries to maximize utilization of resources (Toyota: squeezing water out of dry towels)

Production Rate

increases if:

- disruptions are reduced
- increase rate of slowest machine (bottleneck)
- introduce buffers appropriately

Negative Effects of Buffers (Inventory/Work in Process)

- storage costs money (space, management, etc)
- cycle time (time in system) increases
  - takes longer to detect defects
- products can become obsolete/shrink

Typical Design Guidelines

Line Balancing (Case IV)

Short Cycle Time (Case II, V)

- minimum disruptions (Case II, V)
- optimum WIP (Case VI)
- minimum material handling (Case I)
- batch vs one piece

Fast Feedback

Minimum Demand Lumpiness

	# Machines	Reliability	Operation Times	Buffers
Case I	1	✅	same	❌
Case II	1	❌	same	❌
Case III	2+	✅	same	❌
Case IV	2+	✅	different	❌
Case V	2+	❌	same	❌
Case VI	2+	❌	same	✅

Typical Design Guidelines

Line Balancing (Case IV)

Short Cycle Time (Case II, V)

- minimum disruptions (Case II, V)
- optimum WIP (Case VI)
- minimum material handling (Case I)
- batch vs one piece

Fast Feedback

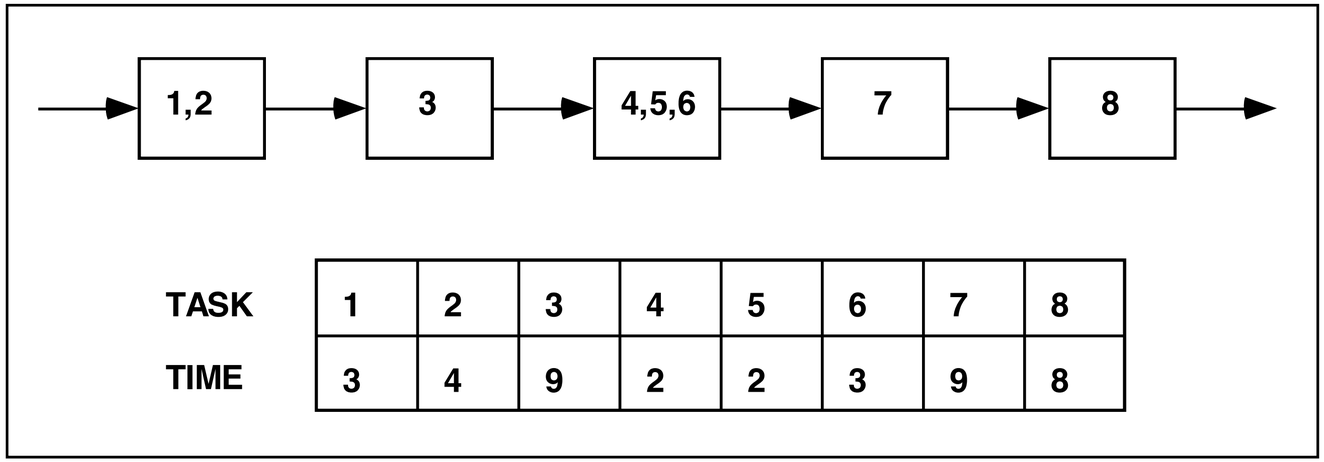
Minimum Demand Lumpiness

	# Machines	Reliability	Operation Times	Buffers
Case I	1	✅	same	❌
Case II	1	❌	same	❌
Case III	2+	✅	same	❌
Case IV	2+	✅	different	❌
Case V	2+	❌	same	❌
Case VI	2+	❌	same	✅

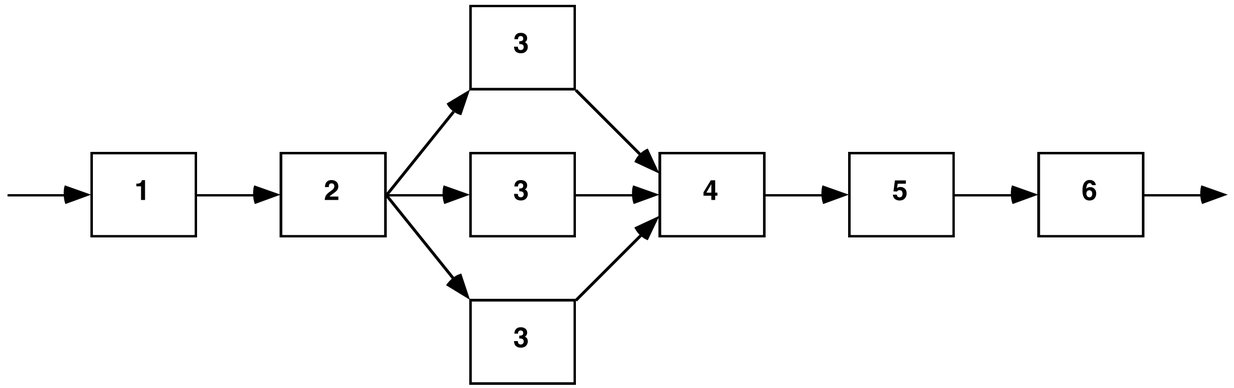
Line Balancing

what can you do to balance the line?

- break slow steps into multiple serial steps
  - evenly distribute the workload
- break slow steps into parallel stations



TASK	1	2	3	4	5	6	7	8
TIME	3	4	9	2	2	3	9	8



TASK	1	2	3	4	5	6
TIME	3	4	9	4	2	3

Typical Design Guidelines

Line Balancing (Case IV)

Short Cycle Time

- minimum disruptions (Case II, V, VI)
- optimum WIP (Case VI)
- minimum material handling
- batch vs one piece

Fast Feedback

Minimum Demand Lumpiness

	# Machines	Reliability	Operation Times	Buffers
Case I	1	✅	same	❌
Case II	1	❌	same	❌
Case III	2+	✅	same	❌
Case IV	2+	✅	different	❌
Case V	2+	❌	same	❌
Case VI	2+	❌	same	✅

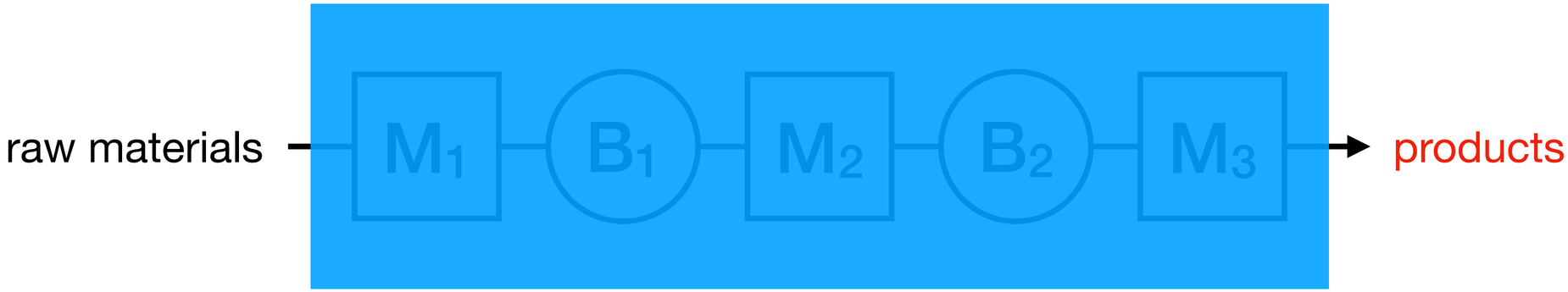
Cycle Time

“the time a part spends in the system”

- used in multiple different ways in different places: you have to ask!

Little’s Law

$$L = \lambda w$$



$$w = \frac{L}{\lambda}$$

L: total inventory levels
λ: system production rate
w: total waiting time (time for one part to get through the full system)

Typical Design Guidelines

Line Balancing (Case IV)

Short Cycle Time

- minimum disruptions (Case II, V, VI)
- optimum WIP (Case VI)
- minimum material handling
- batch vs one piece

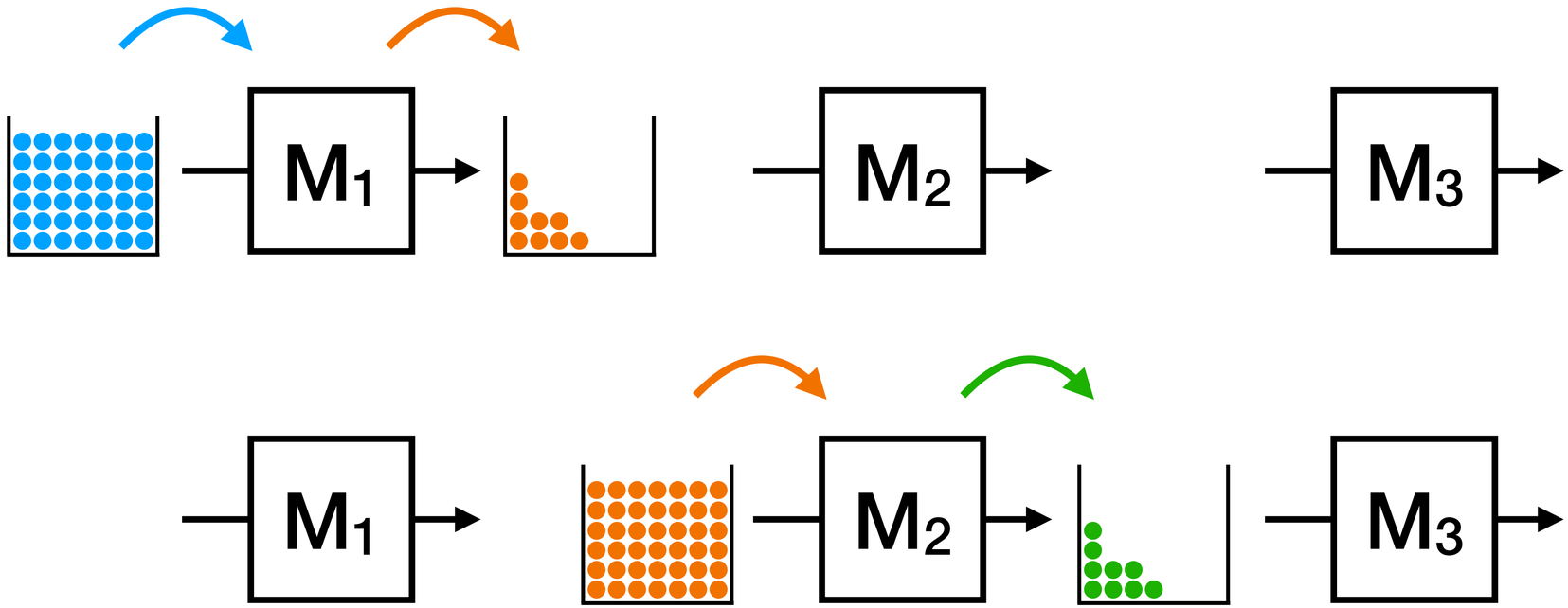
Fast Feedback

Minimum Demand Lumpiness

	# Machines	Reliability	Operation Times	Buffers
Case I	1	✅	same	❌
Case II	1	❌	same	❌
Case III	2+	✅	same	❌
Case IV	2+	✅	different	❌
Case V	2+	❌	same	❌
Case VI	2+	❌	same	✅

Batching

full batch has to finish at M1 before moving (together) to M2



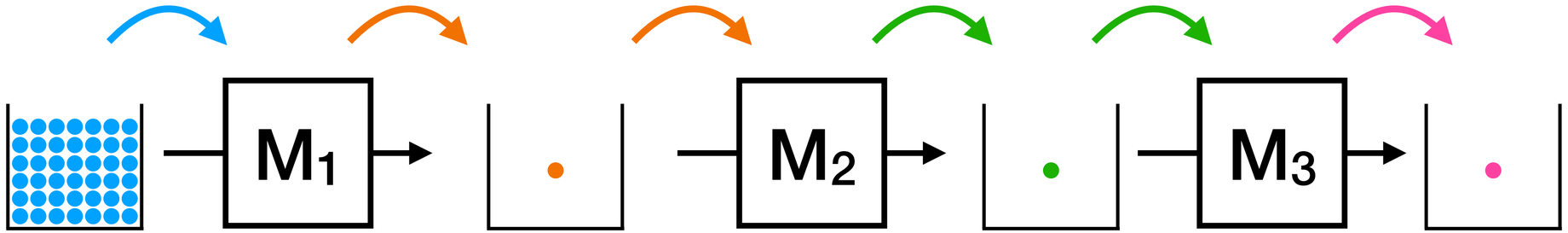
Operation time, τ : 3 minutes

Batch (lot) size: 1,000

Cycle time: 1,000 * 3min + 1,000 * 3min + 1,000*3min = 9,000min

Single Piece Flow

parts wait in the first queue, then process by themselves

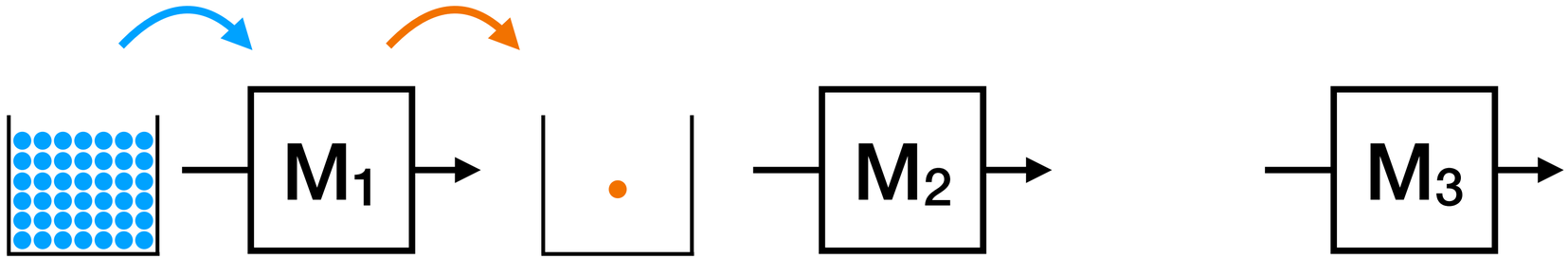


Operation time, τ : 3 minutes

Batch (lot) size: 1

Cycle time: 1,000 * 3min + 2 * 3min = 3,006min

Single Piece Flow



Operation time, τ : 3 minutes

Batch (lot) size: 1

Cycle time: 1,000 * 3min + 2 * 3min = 3,006min

Cycle Time vs **Operation Time** vs **Takt Time** vs **Lead Time**

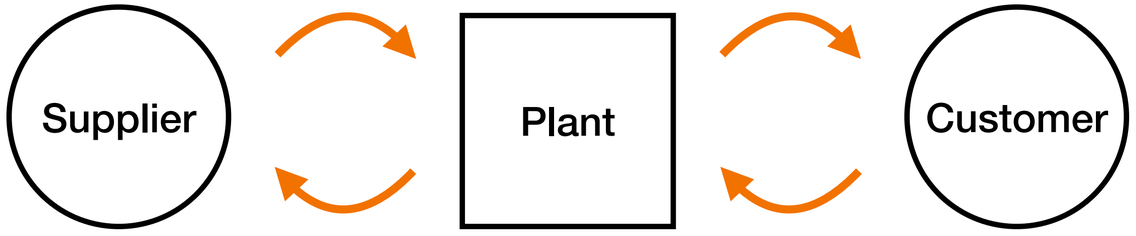
cycle time: the time a part spends in the whole system

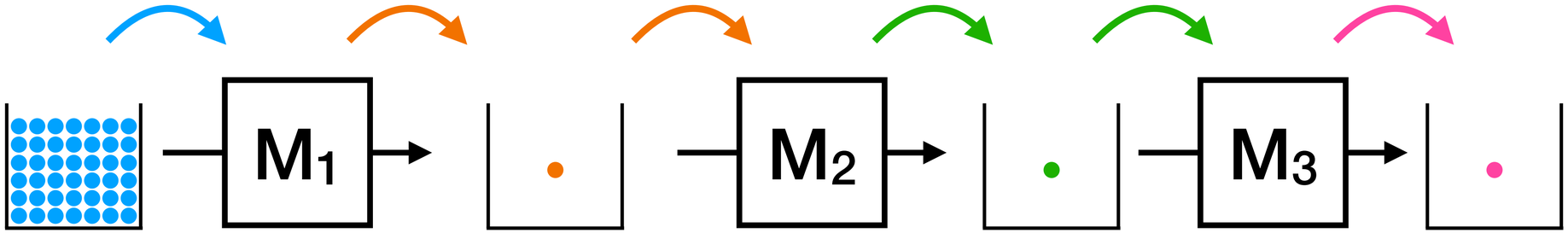
operation time: the time a part spends at a station being transformed (process time)

Takt time: the time between finished parts needed to meet customer demand

lead time: the time between ordering and delivery of parts/materials/supplies

$$Takt\ time = \frac{daily\ available\ time}{daily\ average\ demand}$$





Cycle Time vs **Operation Time** vs **Takt Time** vs **Lead Time**

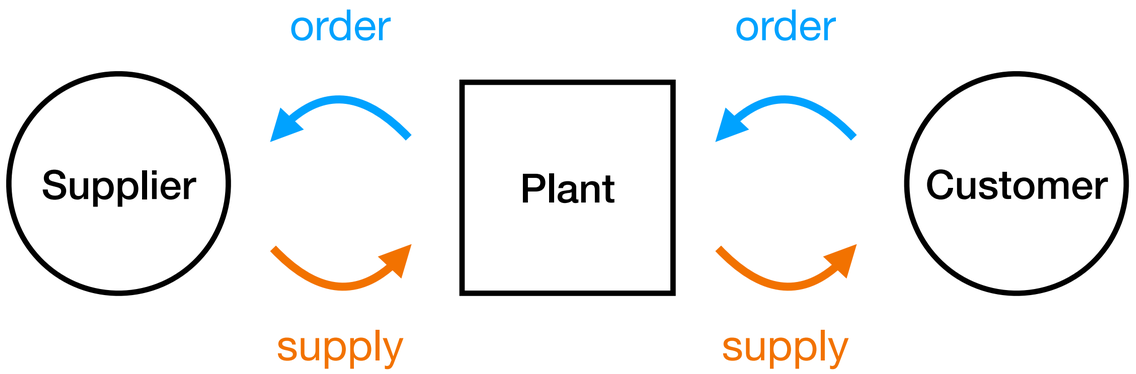
cycle time: the time a part spends in the whole system

operation time: the time a part spends at a station being transformed (process time)

Takt time: the time between finished parts needed to meet customer demand

lead time: the time between ordering and delivery of parts/materials/supplies

$$Takt\ time = \frac{daily\ available\ time}{daily\ average\ demand}$$



Typical Design Guidelines

Line Balancing (Case IV)

Short Cycle Time

- minimum disruptions (Case II, V, VI)
- optimum WIP (Case VI)
- minimum material handling
- batch vs one piece

Fast Feedback

Minimum Demand Lumpiness

"I didn't have any visibility on future demand."

"I ordered more to ensure a safe level of stock."

"My supplier didn't deliver so I increased my orders."

"I was afraid that I wouldn't have enough stock."

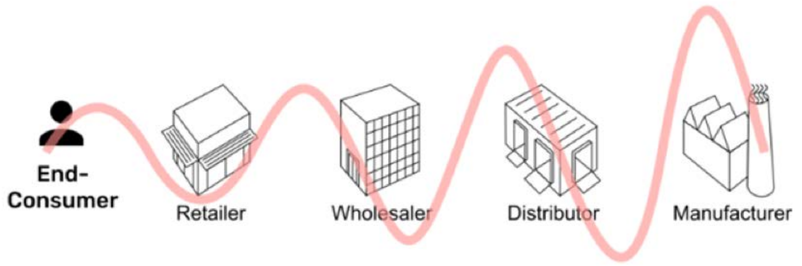
"I started to have backorders so I increased my orders to be safe."

"My customer suddenly increased his orders."

"The Lead Time didn't allow me to react quickly enough."

"I tried to optimize my own stock."

"I lost count of how much I had ordered before."



what was needed to prevent propagation?

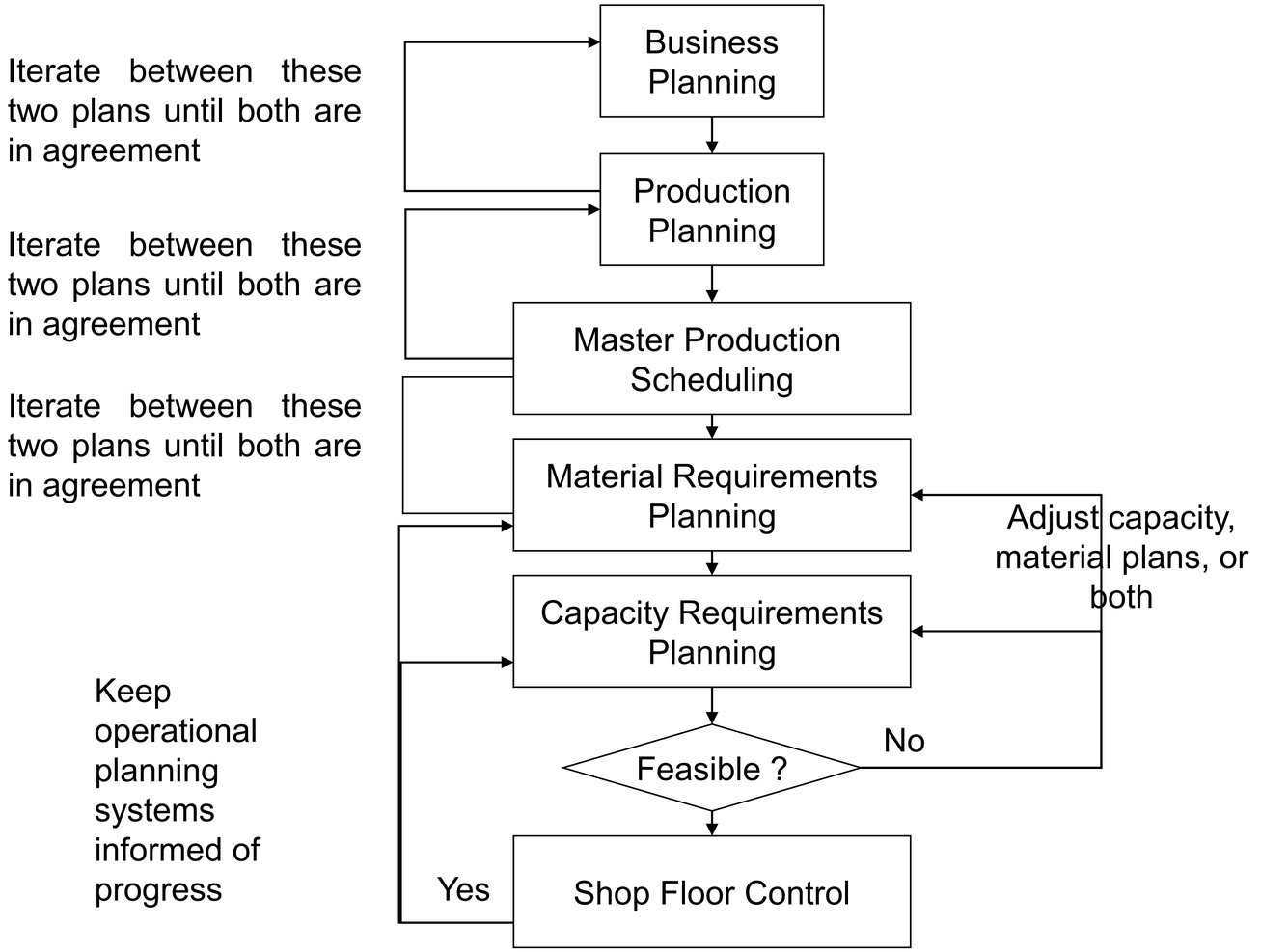
Resource Planning + Material Flow

many different planning methodologies

- same aim: improve communication

issue: forecasting is not always accurate

- leads to backorders or inventory (investment)

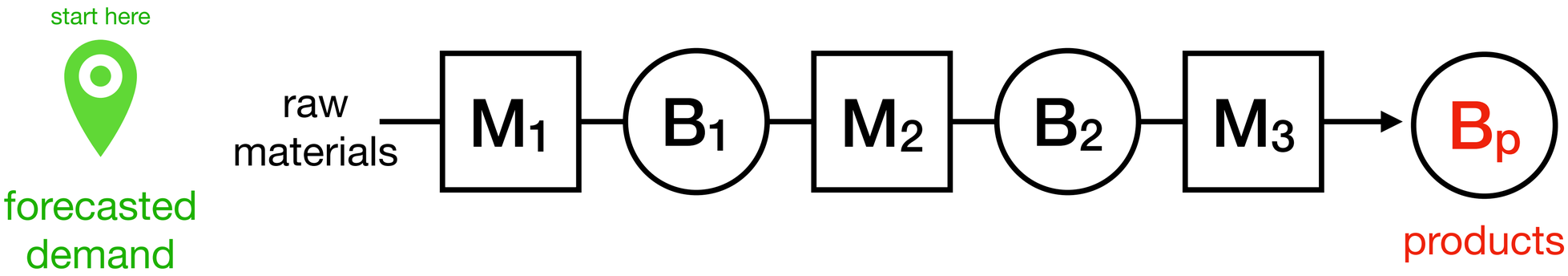


Push vs Pull System

which is best?

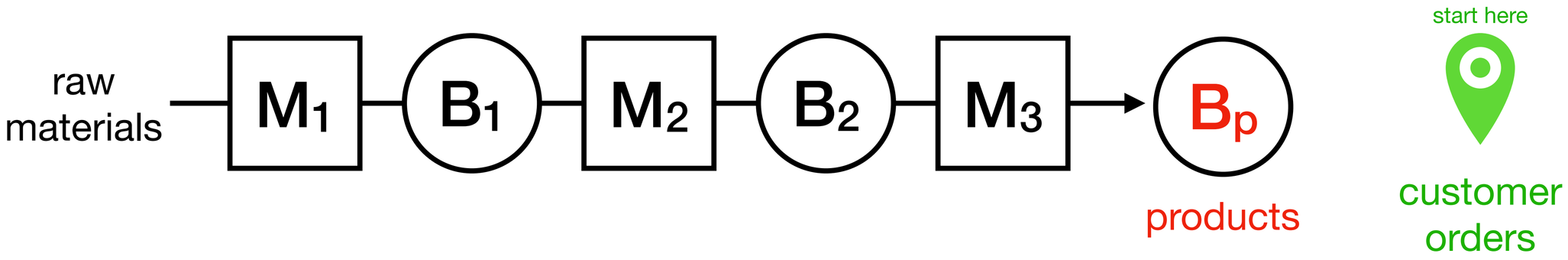
some of both

push: forecast demand, make what is predicted, store was is not purchased



---

pull: take orders from customer, send signal up chain, only produce what will be delivered (Kanban, lean, just-in-time)



Typical Design Guidelines

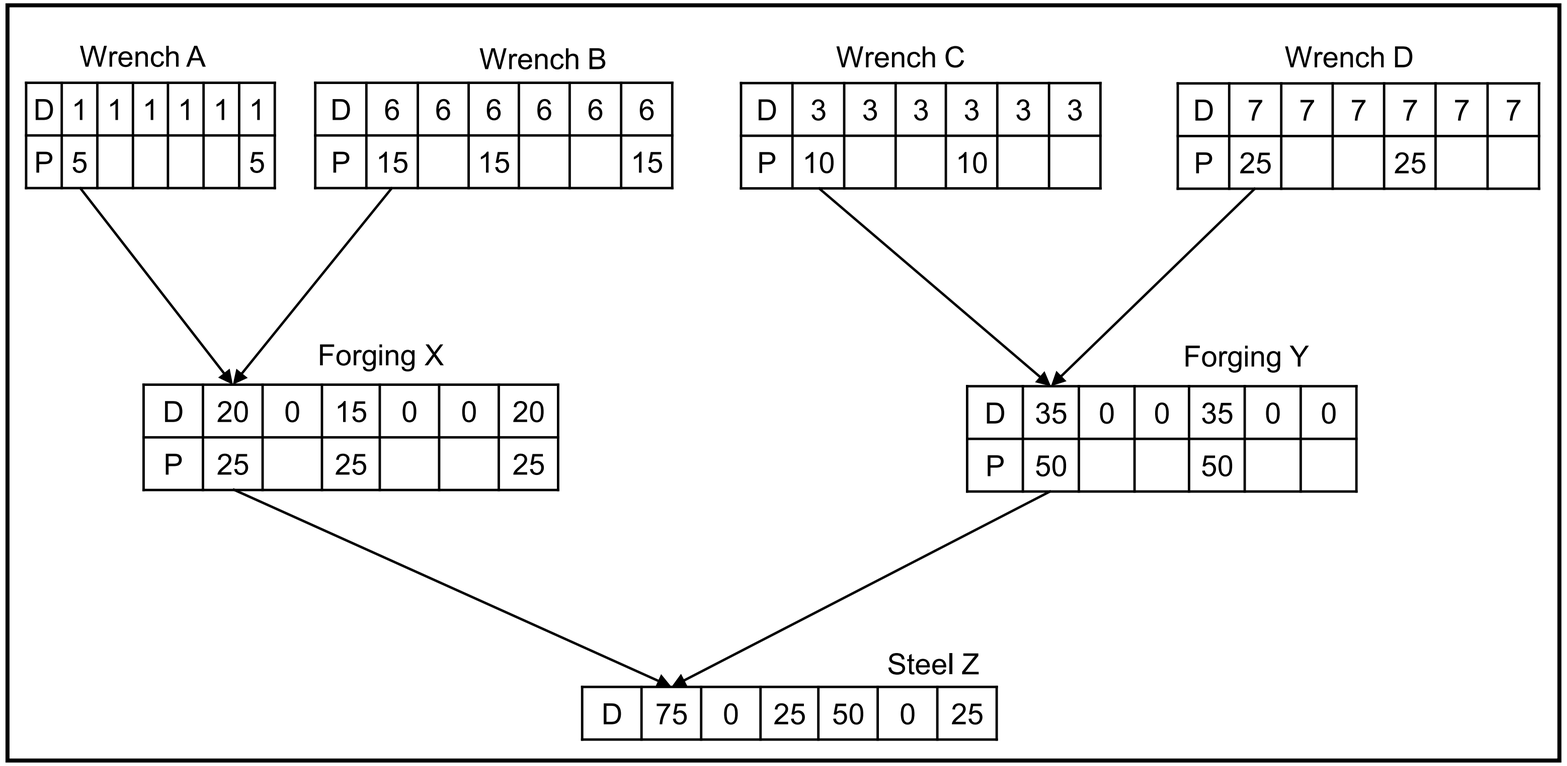
Line Balancing (Case IV)

Short Cycle Time

- minimum disruptions (Case II, V, VI)
- optimum WIP (Case VI)
- minimum material handling
- batch vs one piece

Fast Feedback

Minimum Demand Lumpiness

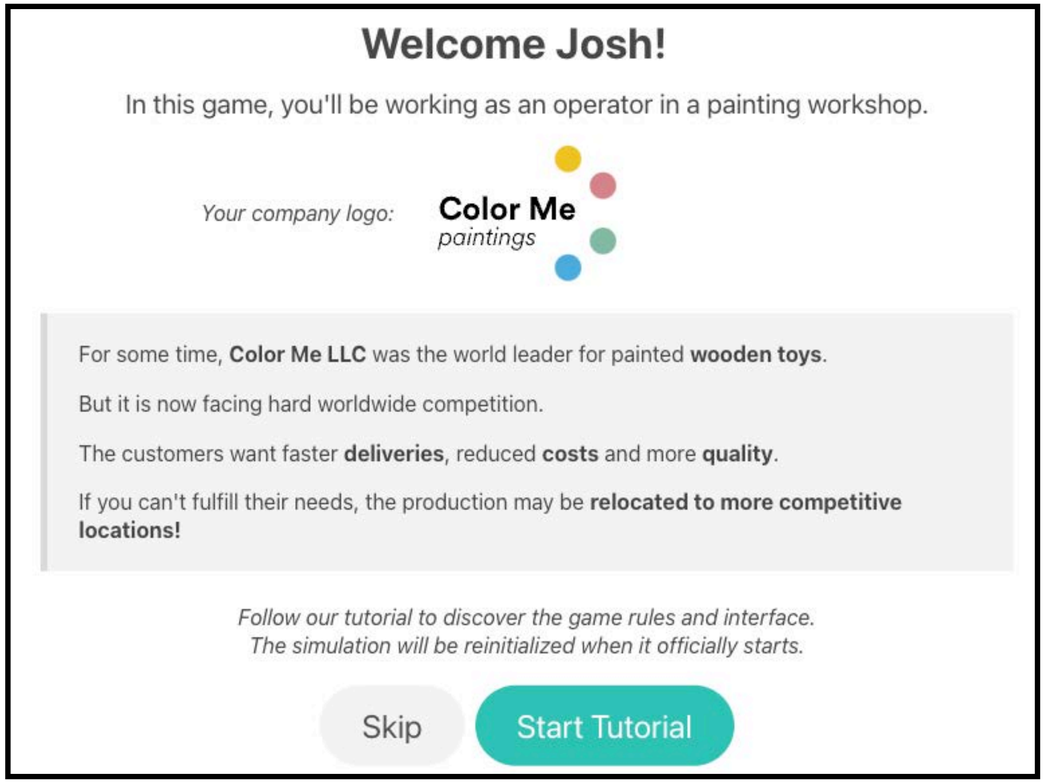


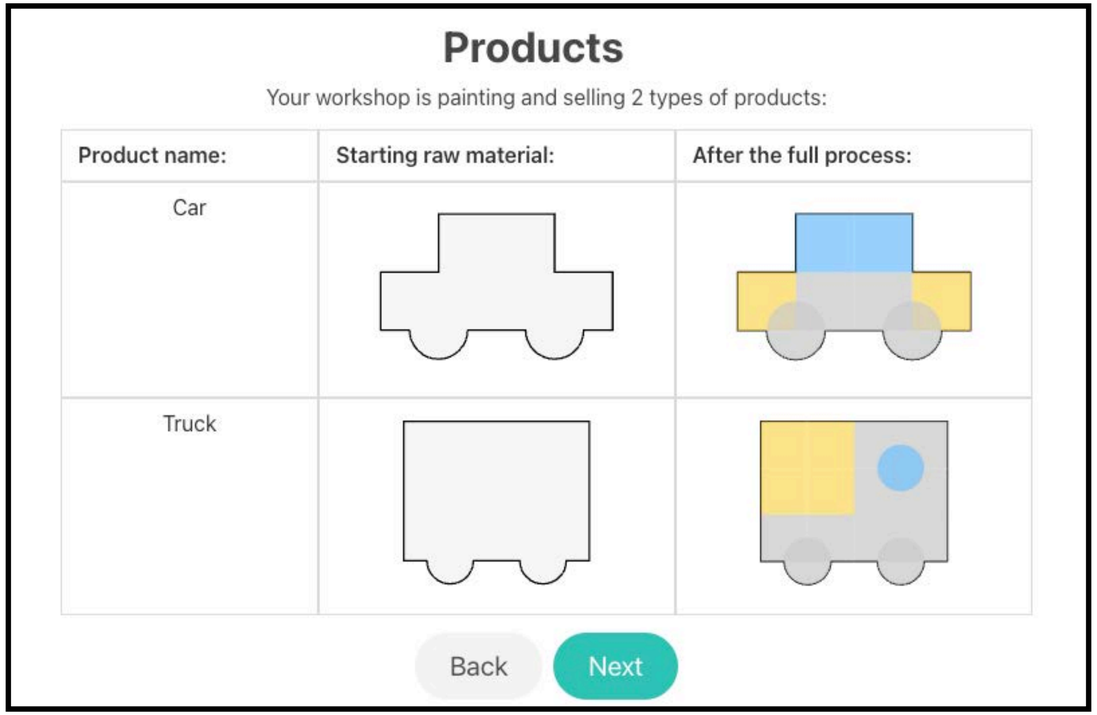
large variation in demand: high periods and zero periods

- makes forecasting difficult ➡ expensive operations

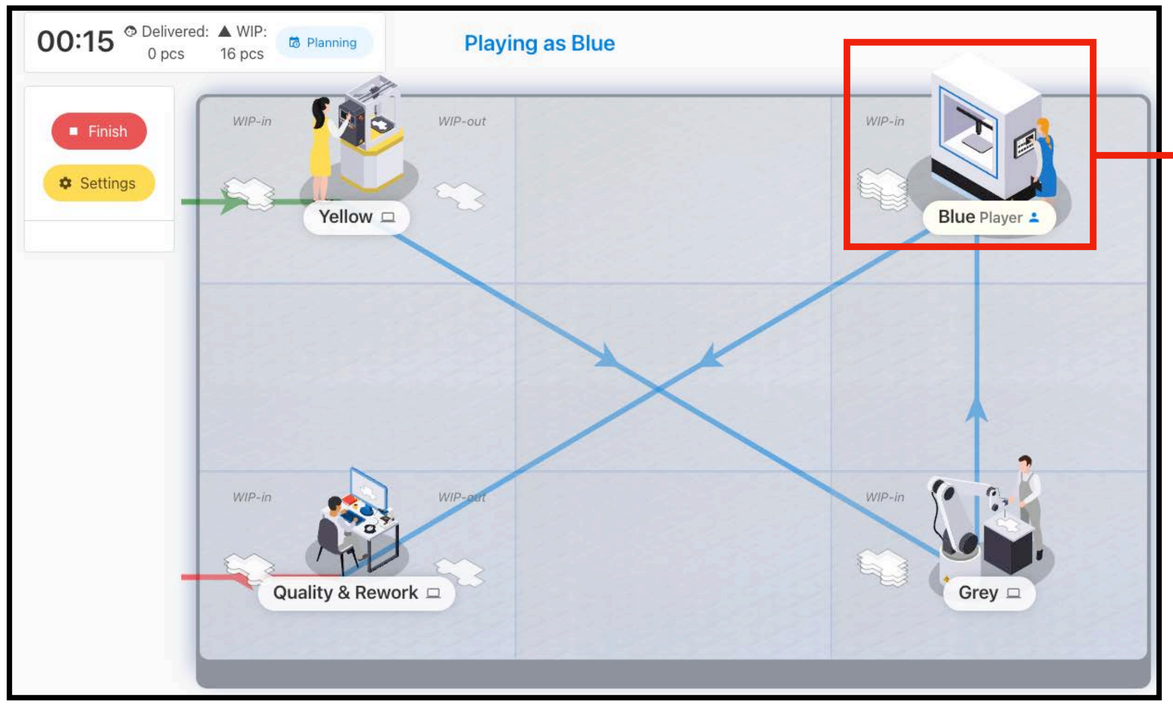
D: Demand
P: Production

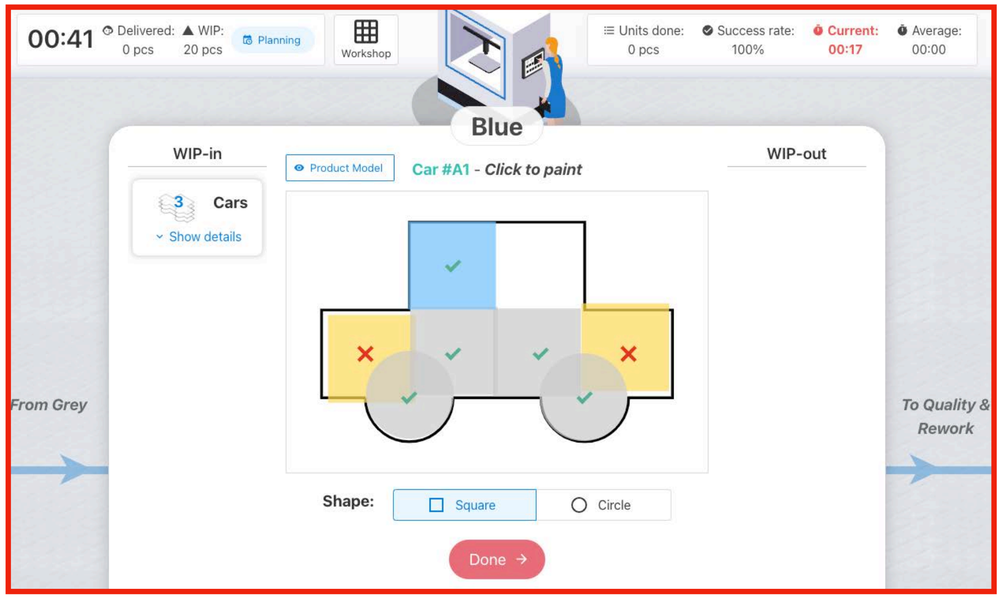
The Lean Game

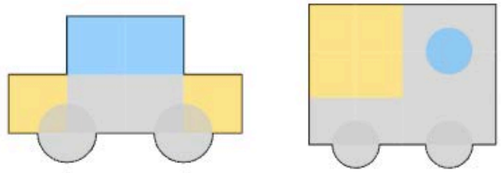


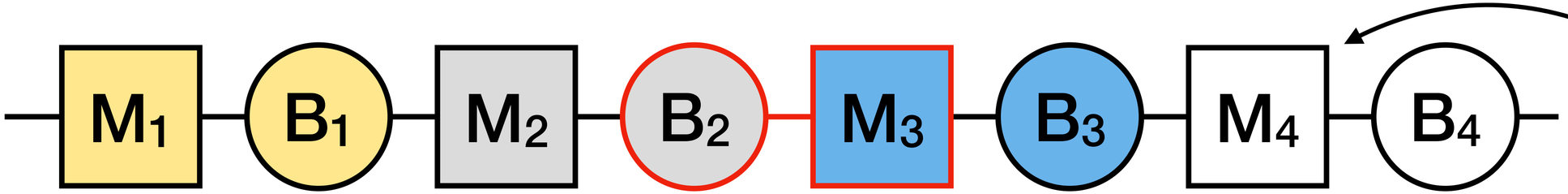


Game Overview



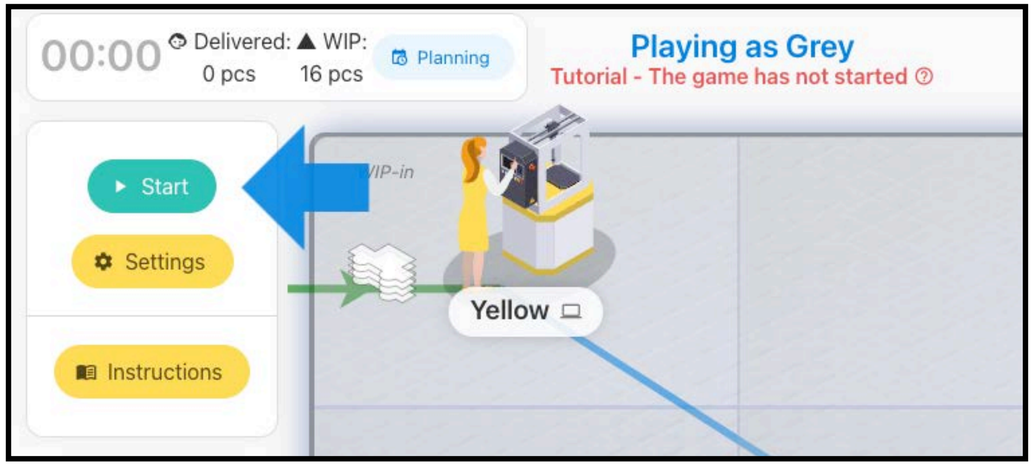






Quality & Rework: can change to any tool, any color, and can remove paint

Game Overview



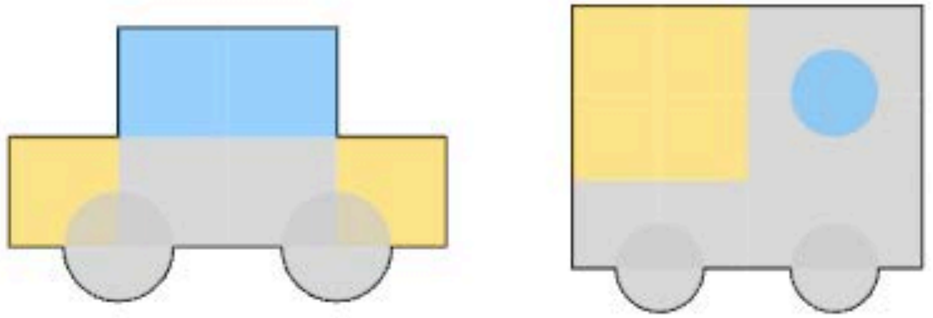
note that you get a tutorial first (for each game)

- if you hit “Start” the tutorial is over

games will last 6 minutes each

Agenda

1. play Round 1 - Basic Factory (communication encouraged!)
2. understand the results by following the Challenge packet
3. play Round 2 - four “Lean” improvements automatically implemented
4. play Round 3 - all (four more) “Lean” improvements automatically implemented

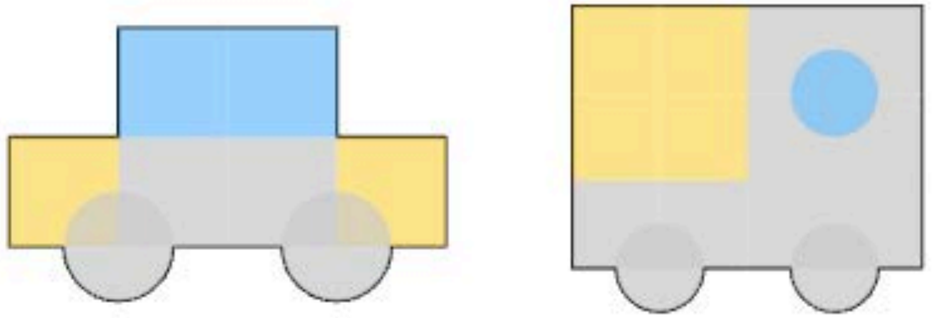


https://zensi.mu/1EpA

	Unit	Qty	Value
Total Revenue *All deliveries*	*+$100*	**24**	**+$2,400**
Labor cost *per station*	*-$50*	4	-$200
Rework operations *per ope* (per operation)	*-$5*	39	-$195
WIP Inventory *based on avg* (avg stock in WIP)	*-$60*	21	-$1,211
Late deliveries *per unit*	*-$30*	8	-$240
Defective deliveries *per unit*	*-$50*	0	$0
Total Expenses			**-$1,846**
Bottom Line			**+$554**

Agenda

1. ~~play Round 1 - Basic Factory (communication encouraged!)~~
2. ~~understand the results by following the Challenge packet~~
3. play Round 2 - four “Lean” improvements automatically implemented
4. play Round 3 - all (four more) “Lean” improvements automatically implemented

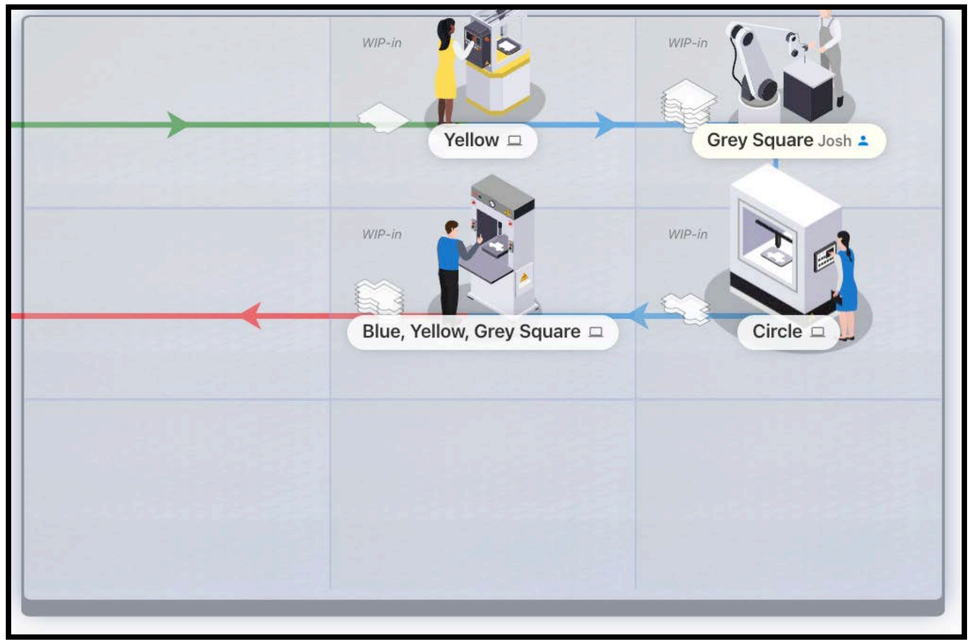


Move Stations: allows you to reorganize the station locations in the workshop

Reduce Batch Size: change the required number of units in a batch

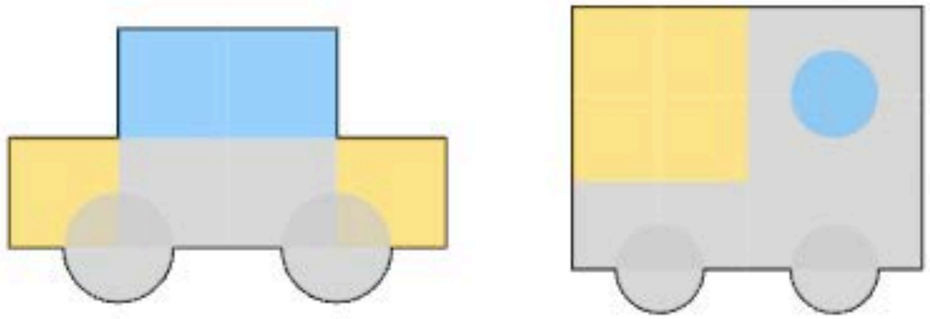
Auto-inspect Work (Jidoka): any/all stations will be able to remove the paint dots after placing them (click a painted square to remove)

Balance Workload: any/all stations can be configured to allow for tool changing, not just Quality & Rework



Agenda

1. ~~play Round 1 - Basic Factory (communication encouraged!)~~
2. ~~understand the results by following the Challenge packet~~
3. ~~play Round 2 - four “Lean” improvements automatically implemented~~
4. play Round 3 - all (four more) “Lean” improvements automatically implemented



Move Stations: allows you to reorganize the station locations in the workshop

Reduce Batch Size: change the required number of units in a batch

Auto-inspect Work (Jidoka): any/all stations will be able to remove the paint dots after placing them

Balance Workload: any/all stations can be configured to allow for tool changing, not just Quality & Rework

Level Production Plan (Hejunka): change the production schedule to make it more even

Add Click-guides to Reduce Defects (Poka-yoke): guides will help you choose where to click to make painting easier

Kanban (Pull System): pieces will automatically move to the next station

Reduce Tool-changeover Time (SMED): tool changeover times happen faster

Image Credits

Slide 1:

Slide 4:

2.008 Design and Manufacturing II

Spring 2025