To Investigate Players' Behavior When Price Variable Is Put In The Beer Game

Showing Young, Ph.D.
Associate Professor, Dep. of Business, Management, National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C.
E-mail: *young@cm.nsysu.edu.tw*

Shyh-Jane Li
Ph.D. Candidate, Dep. of Business Management, National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C.
E-mail: *shyhjane@stoll.twbbs.org*

Chun-Fu Chen
Master Student, Dep. of Business Management, National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C.
E-mail: *cfchen@edoors.com*

Yu-Ing, Huang
Ph.D. Candidate, Dep. of Business Management, National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C.
E-mail: *artimas@bm.nsysu.edu.tw*

Abstract

Beer Game was developed by Sloan School of Management which didn't add price variable into the game. The reasons were: (1) As long as there is time delay, it will induce dynamic complexity. (2)If there is price variable, maybe we can't observe patterns and structure in the Beer Game. In this research, we try to add the price variable into the Beer Game to verify the statement made by Sloan School of Management. Besides, we can explore whether it will influence players' decision or not. The learning effects would be what we concern about. After modeling and some real tests, we found that bullwhip effect still exists, and it's much apparent then before. Besides, players are affected by price variables. From discussions after the game, we can find players lack of systems thinking, they have misperceptions of feedback and get used to put blame on others. Those learning effects are almost the same as traditional Beer Game induced. However, the Beer Game with price variable can conclude many behaviors made by players. Compare with the Basic Beer Game, adding the price variable seems too hard for the players to experience the structure produce behavior.

Keywords: Beer Game、Price、Bullwhip Effect、System Dynamics

Introduction

In the trade market, the powers of supply and demand always have a great influence. These two powers determine the price in the market and people make their decision of production and expenditure by the relate prices between stuffs. In other words, in the economic market, price is the guide of making decision, there is a price mechanism, as known as price system.

Beer Game was developed by Sloan School of Management which is a virtual game; however, it is developed in the supply and demand market. The four players in the Beer Game: Retailer, Wholesaler, Distributor, and Manufacturer, make their order and deliver decisions by the supply and demand market. But in the Beer Game didn't add price variable into the game. The reasons were: (1) As long as there is time delay, it will induce dynamic complexity. (2) If there is price variable, it may interfere with the observations of players which we hope they observe.

The researcher is a member of the Systems Thinking and Organizational Learning Laboratory, according to the experience of teaching the Beer Game, the players gave us a suggestion that we should put the price variable into the Beer Game to simulate more closely to the real world.

Base on these reasons, the research try to put the price variable into the Beer Game to see if the variable indeed has a great influence on the Beer Game or not. There are three purposes of this research. Firstly, to verify the saying of the Sloan School of Management is correct. That is we observe if there is a great influence on the basic structure of the Beer Game when price variable is put into the game. Second, we try to see the effect of the decision making behavior of the players when price variable is put into the Beer Game. Last, we try to figure out if the players have new experiences when the price variable is added into the Beer Game.

Price Variables may have two effects on players. First, when the players expect the price would rise up, it usually causes a serious delay. This is because each player (No matter what character he is) would stock their goods for the higher price. And cut down their production then they would earn more in the future (Chang, Ching-hsi, 2000). Second, if the price would keep rising up, players would like store their goods, so they order more goods from their vendors that they could sell at the high price to earn the revenues. They would sell out at the highest price they expect. However, the action of stock goods or make big order would cause the whip effect;

this is what we want to know further.

Introduction to Beer Distribution Game

The Beer Distribution Game dates to the earliest days of system dynamics. It was developed by Jay Forrester on Industrial Dynamics in 1961. He used system dynamics model to simulate the whole produce-distribution system. He found that when the demand increases 10% from the customers, the demand of the supplier would dramatically swings and it take a while to adjust back to the level of 10%. This is because when the information passes on, each production of the supply chain would increase the demand. This situation happens more obviously and violence at the upper player. To continue Forrester's research, Sterman use the concept of system dynamics to simulate the situation to show the whip effect happened in the supply chain, this is famous game called "Beer Distribution Game." There are beer retailer, wholesaler, distributor, and manufacturer. The whole simulate process went on Sterman's classes. His students play as a role of four production stage and process order and deliver work. In the whole process each player didn't communicate with each other, they made their decision only by the order from the customers. In the linear cost structure, after several simulations we found that the variables of orders were obviously been expanded. The simulation result could improve that the existence of the whip effect. Sterman (1989) called it as system irrational behavior or misperception of feedback.

There are several defects in the traditional physical Beer Distribution Game: (1) Hard to management (2) Hard to learn for some people (3) Easily misunderstand or waste time when count or pass the coins (4) For those only need to make the order decision players, the complex process make them hard to focus on the order (5) Players need to draw the backlog chart. To eliminate these defects, therefore, came out the electronic version of the Beer Distribution Game.

The electronic Beer Distribution Game firstly released in the late eighties. These games can be separated into two groups (Goodman et al.1993): First group was used to replace the traditional physical beer game. Players use the computers to play the game in the internet as past. This version made it more vividly and beautifully and saved all orders and backlog information and needn't draw the backlog charts by oneself. The other version was designed as one or a group of people play the game. It helped facilitators explain more easily after playing the traditional physical Beer Distribution Game. It also gave those who want to know more in dynamic complexity of the game but couldn't find others to play the physical beer game with a chance.

In the February of 1988, Pecos River Learning Center developed the first version of electronic Beer Distribution Game. In the late eighties, Digital Equipment Corporation developed the Beer Distribution Game controlled by VAX VMS which was designed to replace the traditional physical beer game. Innovation Associates used Microsoft Excel to develop the Beer Distribution Game. The advantage of this version was all interaction could be done by computers, so the player could try it by oneself. MIT System Dynamics Group designed the Beer Distribution Game as Management Fight Simulator for its interface. Players could choose the information, the length of time delay and whatever that they want as the basic design of The People Express Simulator.

The information delay of the distribution system is the most obvious and famous problem in the beer game. Therefore, after the electronic beer game was developed, there came out another beer game which use EDI to reduce the information delay of the beer game(Jose A.D. Machuca, Rafael del Pozo Barajas, 1997). There were two types using EDI electronic beer game. One was sequential, another was simultaneous. The difference was at the point that each player took part in the game. In simultaneous beer game, four players (retailer, wholesaler, distributor, and manufacturer) act at the same time, therefore it caused a period of time delay. However in the sequential beer game, the players made their decision one by one. That was to say that from the retailers to the manufacturers can make their own decision separately. But it would take more time.

Besides, after the electronic beer game appeared, to test artificial agents would have a better performance than human, and to eliminate the whip effect or discover an effect business strategy, therefore developed the Stationary Beer Game or called Columbia Beer Game (Chen F., 2000). The differences between the Stationary Beer Game and the traditional Beer Game were the assumptions in the game: (1) four players know the consumers' demand. (2) Consumers' demand is a random number. (3) each player would have the backlog cost, but the shortage cost would happen on the retailer.

This Beer Game is worth learning at these points: (1) players should have some common concept of the consumers' demand, they should know some. (2) With theory benchmark we can evaluate our performance (3) students interested in how to use the demand allocation to reset the new strategy (4) Response to production knowledge (5) Using computer make it easier to play the game.(as Figure 1. shows)

The other beer games were developed in order to close the reality. Take supply chain as a cost center and the consumers' needs were deliver to suppliers as two scopes, we can develop four types of beer games. (as Figure 2. shows) Compare Game I with Game II the difference is the lower cost of the whole supply chain. And compare Game III with Game Ⅳ the difference is which players' accounting cost is lower. (Fangruo Chen & Rungson Samroengraja ,2000)

The insight of electronic and traditional beer game are the same, although it overcame the traditional beer game's defects, there are several defects in itself: (1) The computer system might break down. (2) you couldn't correct after mistakes. And these defects could be improved and correct in the future.

Figure 1. The comparisons of the MIT and Columbia Beer Game

	MIT Beer Game	Columbia Beer Game
Consumers' Demand	Deterministic	Stochastic
Information Delay	Four players are the same	Four players have different information delay
Physical Delay	Fixed number	Fixed number
Shortage Cost	Four players both have the same shortage cost	Only retailer has shortage cost
Backlog Cost	Four players both have the same backlog cost	The upper supplier has less backlog cost
Consumers' Demand Information	Players both don't know the consumers' demand information	Players know consumers' demand

Source: Steven O. Kimbrough, D.J. Wu, Fang Zhong, 2001

Figure 2. Four types of Beer Game developed from the Columbia Beer Game

	Consumers' demand didn't pass to suppliers	Consumers' demand passed to suppliers
Four players as a group	Game I	Game II
Each player play as an independent cost center	Game III	Game IV

Data From: Fangruo Chen and Rungson Samroengraja, 2000

The development of the Beer Distribution Game

The earliest System Dynamics model about supply chain management was developed by Jay Forrester in Industrial Dynamics in 1961. He used system dynamics model to model the whole produce-distribution system. He found that when the demand increases 10% from the customers, the demand of the supplier would dramatically swings and it take a while to adjust back to the level of 10%. This is because when the information passes on, each production of the supply chain would increase the demand. This situation happens more obviously and violence at the upper player. To continue Forrest’s research, Sterman use the concept of system dynamics to simulate the situation to show the whip effect happened in the supply chain, this is famous game called “Beer Distribution Game.” There are beer retailer, wholesaler, distributor, and manufacturer. The whole simulate process went on Sterman’s classes. His students play as a role of four production stage and process order and deliver work. In the whole process each player didn’t communicate with each other, they made their decision only by the order from the customers. In the linear cost structure, after several simulations we found that the variables of orders were obviously been expanded. The simulation result could improve that the existence of the whip effect. Sterman (1989) called it as system irrational behavior or misperception of feedback.

There are several defects in the traditional physical Beer Distribution Game: (1) Hard to management. (2) Hard to learn for some people. (3) Easily misunderstand or waste time when count or pass the coins. (4) About those which only need to make the order decision players, the complex process makes them hard to focus on the order. (5) Players need to draw the backlog chart. To eliminate these defects, therefore, came out the electronic version of the Beer Distribution Game.

The electronic Beer Distribution Game was first released in the late eighties. These games can be separated into two groups (Goodman et al.1993): First group was used to replace the traditional physical beer game. Players use the computers to play the game in the internet as past. This version made it more vividly and beautifully and saved all orders and backlog information and needn’t draw the backlog charts by oneself. The other version was designed as one or a group of people play the game. It helped facilitators explain more easily after playing the traditional physical Beer Distribution Game. It also gave those who want to know more in dynamic complexity of the game but couldn’t find others to play the physical beer game with a chance.

In the February of 1988, Pecos River Learning Center developed the first version of electronic Beer Distribution Game. In the late eighties, Digital Equipment Corporation developed the Beer Distribution Game controlled by VAX VMS which was designed to replace the traditional physical beer game. Innovation Associates used Microsoft Excel to develop the Beer Distribution Game. The advantage of this version was all interaction could be done by computers, so the player could try it by oneself. MIT System Dynamics Group designed the Beer Distribution Game as Management Fight Simulator for its interface. Players could choose the information, the length of time delay and whatever that they want as the basic design of The People Express Simulator.

The information delay of the distribution system is the most obvious and well-known problem in the beer game. Therefore, after the electronic beer game was developed, there came out another beer game which use EDI to reduce the information delay of the beer game (Jose A.D. Machuca, Rafael del Pozo Barajas, 1997). There were two types using EDI electronic beer game. One was sequential, another was simultaneous. The difference was at the point that each player took part in the game. In simultaneous beer game, four players (retailer, wholesaler, distributor, and manufacturer) act at the same time, therefore it caused a period of time delay. However in the sequential beer game, the players made their decision one by one. That was to say that from the retailers to the manufacturers can make their own decision separately. But it would take more time.

Besides, after the electronic beer game appeared, to test artificial agents would have a better performance than human, and to eliminate the whip effect or discover an effect business strategy, therefore developed the Stationary Beer Game or called Columbia Beer Game (Chen F., 2000). The differences between the Stationary Beer Game and the traditional Beer Game were the assumptions in the game. (1) Four players know the consumers’ demand. (2) Consumers’ demand is a random number. (3) each player would have the backlog cost, but the shortage cost would happen on the retailer.

This Beer Game is worth learning at these points: (1) players should have some common concept of the consumers’ demand, they should know some. (2) With theory benchmark we can evaluate our performance. (3) Students interested in how to use the demand allocation to reset the new strategy. (4) Response to production knowledge (5) Using computer make it easier to play the game.(as Figure 1. shows)

The other beer games were developed in order to close the reality. Take supply chain as a cost center and the consumers' needs were deliver to suppliers as two scopes, we can develop four types of beer games. (as Figure 2. shows) Compare Game I with Game II the difference is the lower cost of the whole supply chain. And compare Game III with Game IV the difference is which players' accounting cost is lower.(Fangruo Chen & Rungson Samroengraja ,2000)

The insight of electronic and traditional beer game are the same, although it overcame the traditional beer games' defects, there are several defects in itself: (1) The computer system might break down (2) you couldn't correct after mistakes. And these defects could be improved and correct in the future.

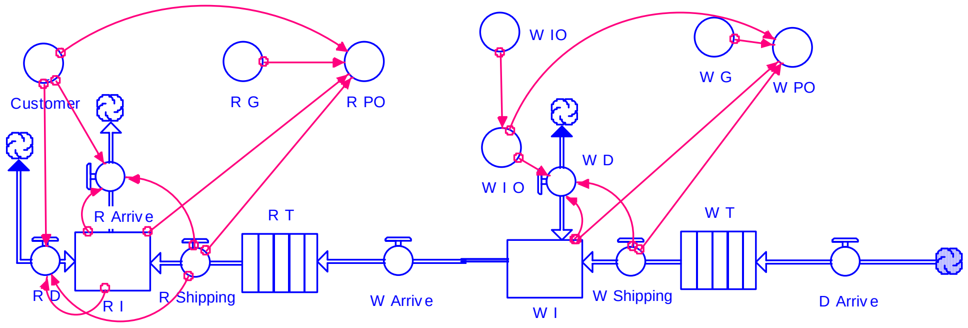
Research Process

Through adding price variable into beer game, we can observe how this variable has any influences on players' behavior and try to give them a new experience. This research uses action research, in redesigning the Beer Distribution Game, we modify the game perfect through observing players' responses, after game experience sharing, and after action reflection.

Add price variable into the Beer Distribution Game

Modeling the Base model of the Beer Game

The original Beer Game was developed by Sloan School of Management. It is similar to Richman Game on the desktop, but easier. The player only makes a decision—order decision. The last physical desktop version was modified by the Systems Thinking and Organizational Learning Laboratory in National Sun Yat-Sen University in Taiwan. And the following basic ithink model is made according to the physical game. The researcher modeled a basic Beer Game model which as Figure 3 shows.



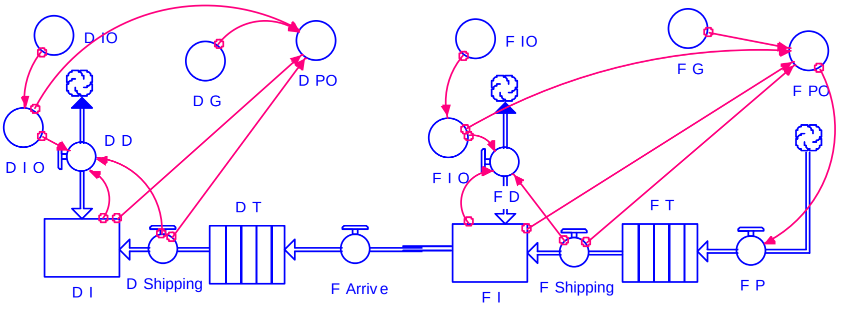


Figure 3. Basic model of the Beer Game

In this model, we set each players' average inventory goal is 12 boxes and model for 50 weeks. The patterns of these players' order which as Figure 4 shows. About this Figure, we know that the players' order patterns show the bullwhip effect clearly, as we know before.

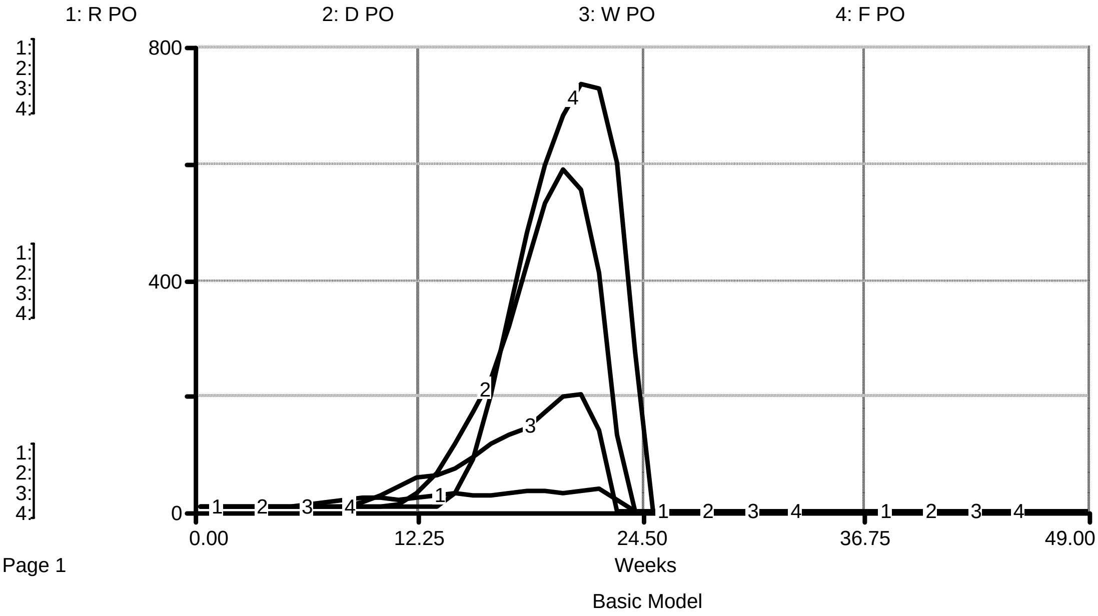
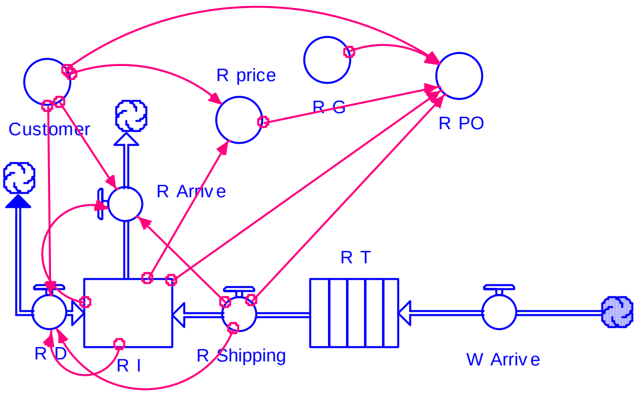


Figure 4. Players Order Pattern in Basic Beer Game

Add price variable

Now we try to add price variable into the Basic Beer Game model to see if it influence the basic structure of the Beer Game. We add the price variables into this model which affected by each players' inventory and the orders from their lower player. And we suppose that each player's order has an influence by the price. The function works as the following rule: We define the price of each box range is from $150 to $250.If the price goes higher, that means the demand of beer need more, so the player would order more boxes and deliver more to satisfy the market needs. (as Figure 5. shows)



Function: 200-(Inventory/Customer Demands*5)
Basic Price：185 (200-(12/4*5))
if 200-(Inventory/Customer Demands*5)>=250，price=250
if 150<200-(I/Demands*5) <250，price=200-(I/Demands*5)
if 200-(Inventory/Demands*5)<=150，price=150

Figure 5. Add price variables into the Beer Game: retail as example

After adding price variables into the Beer Game, we found that the patterns of each player’s order shake a lot. (as Figure 6.) However, the big picture of the patterns seem don’t change a lot.

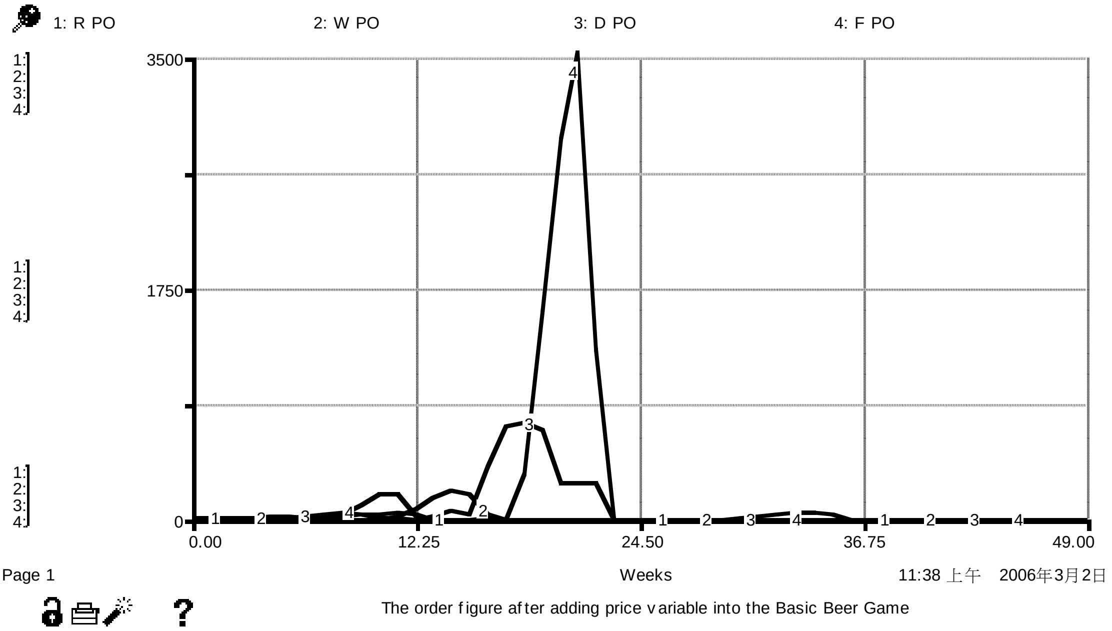


Figure 6. The order figure after adding price variable into the Beer Game

After modeling simulation, we can find adding price variable into the model the order would be extended. To investigate the players' thought, we have tested some groups as Figure 7. The orders shown very unstable, we can know the players were controlled by price. To fulfill the buyers needs and got max profits, they stored goods and caused serious problem. In the future, we consider adding more soft variables into the model.

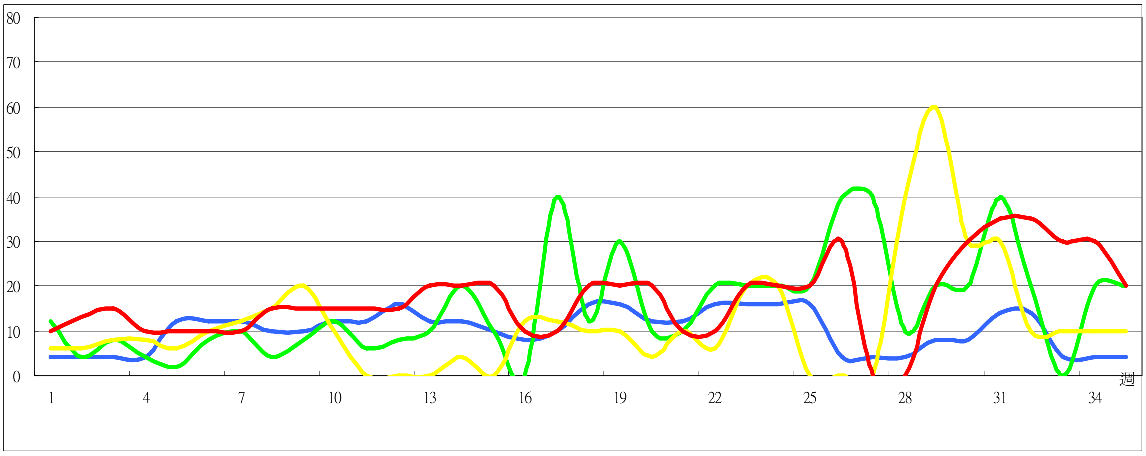


Figure 7. The pattern of orders when it was added price variable in Beer Distribution Game

Conclusion

After a few experiment, although bullwhip effect still exists, it’s not much apparent then before. Besides, players are affected by price variables. From discussions after the game, we can find players lack systems thinking, they have misperceptions of feedback and get used to put blame on others. Those learning effects are the same as traditional beer game induced. However, beer game with price variable can conclude many behaviors made by players. To contrast with the panic buying and stock conditions, such as respirators panic buying in 2003, toilet paper price rise caused by pulp cost increased in 2004, players can simulate the real world through playing the game. They can reflect and learn, then bring this experience into real life.

Appendix

The functions of the Basic Beer Game model and the Add Price Variable Beer Game model.

The Basic Beer Game Model:

```
    Distributor
    D_I(t) = D_I(t - dt) + (D_Shipping + D_D - D_Arrive) * dt
    INIT D_I = 12

    INFLOWS:
    D_Shipping = CONVEYOR OUTFLOW
    D_D =
if(D_I>=0)then(if(D_Shipping+D_I>=D_I_O)then(0)else(D_Shipping+D_I-D_I_O)) else
(if(D_Shipping>=-D_I)then(if (D_Shipping>= D_I_O-D_I)then(D_Shipping-(D_I_O-D_I))
else (D_Shipping-D_I_O))else(D_Shipping-D_I_O))
    OUTFLOWS:
    D_Arrive    (Not in a sector)
    D_T(t) = D_T(t - dt) + (F_Arrive - D_Shipping) * dt
    INIT D_T = 8
        TRANSIT TIME = 2
        INFLOW LIMIT = INF
        CAPACITY = INF
    INFLOWS:
    F_Arrive    (Not in a sector)
    OUTFLOWS:
    D_Shipping = CONVEYOR OUTFLOW
    D_G = 12
    D_IO = delay(W_PO,1,4)
    D_I_O = delay(D_IO,1,4)
    D_PO =
if((if(D_I>=0)then(if(D_Shipping>=D_I_O)then(D_G-D_I+D_I_O-(D_Shipping-
D_I_O))else(D_G-D_I+(D_I_O-D_Shipping)))else(if(D_Shipping>=D_I_O-D_I)then(D_G-(
D_Shipping-D_I_O+D_I))else(D_G+(D_I_O-D_I-D_Shipping))))>=0)then(if(D_I>=0)then(if
(D_Shipping>=D_I_O)then(D_G-D_I+D_I_O-(D_Shipping-
D_I_O))else(D_G-D_I+(D_I_O-D_Shipping)))else(if(D_Shipping>=D_I_O-D_I)then(D_G-(
D_Shipping-D_I_O+D_I))else(D_G+(D_I_O-D_I-D_Shipping))))else(0)
    Factory
    F_I(t) = F_I(t - dt) + (F_Shipping + F_D - F_Arrive) * dt
    INIT F_I = 12
    INFLOWS:
    F_Shipping = CONVEYOR OUTFLOW
    F_D = if(F_I>=0)then(if(F_Shipping+F_I>=F_I_O)then(0)else(F_Shipping+F_I-F_I_O))
else (if(F_Shipping>=-F_I)then(if (F_Shipping>= F_I_O-F_I)then(F_Shipping-(F_I_O-F_I))
else (F_Shipping-F_I_O))else(F_Shipping-F_I_O))
    OUTFLOWS:
    F_Arrive    (Not in a sector)
    F_T(t) = F_T(t - dt) + (F_P - F_Shipping) * dt
    INIT F_T = 8
        TRANSIT TIME = 2
        INFLOW LIMIT = INF
        CAPACITY = INF
    INFLOWS:
    F_P = delay(F_PO,1,4)
    OUTFLOWS:
    F_Shipping = CONVEYOR OUTFLOW
    F_G = 12
    F_IO = delay(D_PO,1,4)
```

F_I_O = delay(F_IO,1,4)

F_PO = if((if(F_I>=0)then(if(F_Shipping>=F_I_O)then(F_G-F_I+F_I_O-(F_Shipping-F_I_O))else(F_G-F_I+(F_I_O-F_Shipping)))else(if(F_Shipping>=F_I_O-F_I)then(F_G-(F_Shipping-F_I_O+F_I))else(F_G+(F_I_O-F_I-F_Shipping))))>=0)then(if(F_I>=0)then(if(F_Shipping>=F_I_O)then(F_G-F_I+F_I_O-(F_Shipping-F_I_O))else(F_G-F_I+(F_I_O-F_Shipping)))else(if(F_Shipping>=F_I_O-F_I)then(F_G-(F_Shipping-F_I_O+F_I))else(F_G+(F_I_O-F_I-F_Shipping))))else(0)

Retailer

R_I(t) = R_I(t - dt) + (R_Shipping + R_D - R_Arrive) * dt

INIT R_I = 12

INFLOWS:

R_Shipping = CONVEYOR OUTFLOW

R_D = if(R_I>=0)then(if(R_Shipping+R_I>=Customer)then(0)else(R_Shipping+R_I-Customer)) else (if(R_Shipping>=-R_I)then(if (R_Shipping>= Customer-R_I)then(R_Shipping-(Customer-R_I)) else (R_Shipping-Customer))else(R_Shipping-Customer))

OUTFLOWS:

R_Arrive = if(R_I>=0)then(if(R_Shipping+R_I>=Customer)then(Customer)else(R_Shipping+R_I)) else (if(R_Shipping>=-R_I)then(if (R_Shipping>= Customer-R_I)then(Customer-R_I) else (R_Shipping))else(R_Shipping))

R_T(t) = R_T(t - dt) + (W_Arrive - R_Shipping) * dt

INIT R_T = 8

TRANSIT TIME = 2

INFLOW LIMIT = INF

CAPACITY = INF

INFLOWS:

W_Arrive   (Not in a sector)

OUTFLOWS:

R_Shipping = CONVEYOR OUTFLOW

R_G = 12

R_PO = if((if(R_I>=0)then(if(R_Shipping>=Customer)then(R_G-R_I+Customer-(R_Shipping-Customer))else(R_G-R_I+(Customer-R_Shipping)))else(if(R_Shipping>=Customer-R_I)then (R_G-(R_Shipping-Customer+R_I))else(R_G+(Customer-R_I-R_Shipping))))>=0)then(if(R_I>=0)then(if(R_Shipping>=Customer)then(R_G-R_I+Customer-(R_Shipping-Customer))else(R_G-R_I+(Customer-R_Shipping)))else(if(R_Shipping>=Customer-R_I)then (R_G-(R_Shipping-Customer+R_I))else(R_G+(Customer-R_I-R_Shipping))))else(0)

Customer = GRAPH(time)

(1.00, 4.00), (2.00, 4.00), (3.00, 4.00), (4.00, 4.00), (5.00, 8.00), (6.00, 8.00), (7.00, 8.00), (8.00, 8.00), (9.00, 8.00), (10.0, 8.00), (11.0, 8.00), (12.0, 8.00), (13.0, 8.00), (14.0, 8.00), (15.0, 8.00), (16.0, 8.00), (17.0, 8.00), (18.0, 8.00), (19.0, 8.00), (20.0, 8.00), (21.0, 8.00), (22.0, 8.00), (23.0, 8.00), (24.0, 8.00), (25.0, 8.00), (26.0, 8.00), (27.0, 8.00), (28.0, 8.00), (29.0, 8.00), (30.0, 8.00), (31.0, 8.00), (32.0, 8.00), (33.0, 8.00), (34.0, 8.00), (35.0, 8.00), (36.0, 8.00), (37.0, 8.00), (38.0, 8.00), (39.0, 8.00), (40.0, 8.00), (41.0, 8.00), (42.0, 8.00), (43.0, 8.00), (44.0, 8.00), (45.0, 8.00), (46.0, 8.00), (47.0, 8.00), (48.0, 8.00), (49.0, 8.00), (50.0, 8.00)

Wholesaler

W_I(t) = W_I(t - dt) + (W_Shipping + W_D - W_Arrive) * dt

INIT W_I = 12

INFLOWS:

W_Shipping = CONVEYOR OUTFLOW

W_D =

```
if(W_I>=0)then(if(W_Shipping+W_I>=W_I_O)then(0)else(W_Shipping+W_I-W_I_O)) else
(if(W_Shipping>=-W_I)then(if (W_Shipping>=
W_I_O-W_I)then(W_Shipping-(W_I_O-W_I)) else
(W_Shipping-W_I_O))else(W_Shipping-W_I_O))
    OUTFLOWS:
    W_Arrive    (Not in a sector)
    W_T(t) = W_T(t - dt) + (D_Arrive - W_Shipping) * dt
    INIT W_T = 8
        TRANSIT TIME = 2
        INFLOW LIMIT = INF
        CAPACITY = INF
    INFLOWS:
    D_Arrive    (Not in a sector)
    OUTFLOWS:
    W_Shipping = CONVEYOR OUTFLOW
    W_G = 12
    W_IO = delay(R_PO,1,4)
    W_I_O = delay(W_IO,1,4)
    W_PO =
if((if(W_I>=0)then(if(W_Shipping>=W_I_O)then(W_G-W_I+W_I_O-(W_Shipping-
W_I_O))else(W_G-W_I+(W_I_O-W_Shipping)))else(if(W_Shipping>=W_I_O-W_I)then(W
_G-(W_Shipping-W_I_O+W_I))else(W_G+(W_I_O-W_I-W_Shipping))))>=0)then(if(W_I>
=0)then(if(W_Shipping>=W_I_O)then(W_G-W_I+W_I_O-(W_Shipping-
W_I_O))else(W_G-W_I+(W_I_O-W_Shipping)))else(if(W_Shipping>=W_I_O-W_I)then(W
_G-(W_Shipping-W_I_O+W_I))else(W_G+(W_I_O-W_I-W_Shipping))))else(0)

    Not in a sector
    D_Arrive =
if(D_I>=0)then(if(D_Shipping+D_I>=D_I_O)then(D_I_O)else(D_Shipping+D_I)) else
(if(D_Shipping>=-D_I)then(if (D_Shipping>= D_I_O-D_I)then(D_I_O-D_I) else
(D_Shipping))else(D_Shipping))
    OUTFLOW FROM:    D_I (IN SECTOR:    Distributor)
    INFLOW TO:    W_T        (IN SECTOR:    Wholesaler)
    F_Arrive =
if(F_I>=0)then(if(F_Shipping+F_I>=F_I_O)then(F_I_O)else(F_Shipping+F_I)) else
(if(F_Shipping>=-F_I)then(if (F_Shipping>= F_I_O-F_I)then(F_I_O-F_I) else
(F_Shipping))else(F_Shipping))
    OUTFLOW FROM:    F_I (IN SECTOR:    Factory)
    INFLOW TO:    D_T (IN SECTOR:    Distributor)
    W_Arrive =
if(W_I>=0)then(if(W_Shipping+W_I>=W_I_O)then(W_I_O)else(W_Shipping+W_I)) else
(if(W_Shipping>=-W_I)then(if (W_Shipping>= W_I_O-W_I)then(W_I_O-W_I) else
(W_Shipping))else(W_Shipping))
    OUTFLOW FROM:    W_I        (IN SECTOR:    Wholesaler)
    INFLOW TO:    R_T (IN SECTOR:    Retailer)
```

The Add Price Variable Beer Game Model:

```
    Distributor
    D_I(t) = D_I(t - dt) + (D_Shipping + D_D - D_Arrive) * dt
    INIT D_I = 12
    INFLOWS:
    D_Shipping = CONVEYOR OUTFLOW
    D_D =
if(D_I>=0)then(if(D_Shipping+D_I>=D_I_O)then(0)else(D_Shipping+D_I-D_I_O)) else
(if(D_Shipping>=-D_I)then(if (D_Shipping>= D_I_O-D_I)then(D_Shipping-(D_I_O-D_I))
else (D_Shipping-D_I_O))else(D_Shipping-D_I_O))
```

```
    OUTFLOWS:
    D_Arrive    (Not in a sector)
    D_T(t) = D_T(t - dt) + (F_Arrive - D_Shipping) * dt
    INIT D_T = 8
        TRANSIT TIME = 2
        INFLOW LIMIT = INF
        CAPACITY = INF
    INFLOWS:
    F_Arrive    (Not in a sector)
    OUTFLOWS:
    D_Shipping = CONVEYOR OUTFLOW
    D_G = 12
    D_IO = delay(W_PO,1,4)
    D_I_O = delay(D_IO,1,4)
    D_PO = int((2-(250-D_price)/185)*(if((if(D_I>=0)then(if(D_Shipping>=D_I_O)then(D_G-D_I+D_I_O-(D_Shipping-D_I_O))else(D_G-D_I+(D_I_O-D_Shipping)))else(if(D_Shipping>=D_I_O-D_I)then(D_G-(D_Shipping-D_I_O+D_I))else(D_G+(D_I_O-D_I-D_Shipping)))))>=0)then(if(D_I>=0)then(if(D_Shipping>=D_I_O)then(D_G-D_I+D_I_O-(D_Shipping-D_I_O))else(D_G-D_I+(D_I_O-D_Shipping)))else(if(D_Shipping>=D_I_O-D_I)then(D_G-(D_Shipping-D_I_O+D_I))else(D_G+(D_I_O-D_I-D_Shipping))))else(0)))
    D_price = IF((200-(D_I/D_I_O*5))>=250)then 250 else(IF(200-(D_I/D_I_O*5))<=150then 150 else(200-(D_I/D_I_O*5)))
    Factory
    F_I(t) = F_I(t - dt) + (F_Shipping + F_D - F_Arrive) * dt
    INIT F_I = 12
    INFLOWS:
    F_Shipping = CONVEYOR OUTFLOW
    F_D = if(F_I>=0)then(if(F_Shipping+F_I>=F_I_O)then(0)else(F_Shipping+F_I-F_I_O)) else (if(F_Shipping>=-F_I)then(if (F_Shipping>= F_I_O-F_I)then(F_Shipping-(F_I_O-F_I)) else (F_Shipping-F_I_O))else(F_Shipping-F_I_O))
    OUTFLOWS:
    F_Arrive    (Not in a sector)
    F_T(t) = F_T(t - dt) + (F_P - F_Shipping) * dt
    INIT F_T = 8
        TRANSIT TIME = 2
        INFLOW LIMIT = INF
        CAPACITY = INF
    INFLOWS:
    F_P = delay(F_PO,1,4)
    OUTFLOWS:
    F_Shipping = CONVEYOR OUTFLOW
    F_G = 12
    F_IO = delay(D_PO,1,4)
    F_I_O = delay(F_IO,1,4)
    F_PO = int((2-(250-F_price)/185)*(if((if(F_I>=0)then(if(F_Shipping>=F_I_O)then(F_G-F_I+F_I_O-(F_Shipping-F_I_O))else(F_G-F_I+(F_I_O-F_Shipping)))else(if(F_Shipping>=F_I_O-F_I)then(F_G-(F_Shipping-F_I_O+F_I))else(F_G+(F_I_O-F_I-F_Shipping)))))>=0)then(if(F_I>=0)then(if(F_Shipping>=F_I_O)then(F_G-F_I+F_I_O-(F_Shipping-F_I_O))else(F_G-F_I+(F_I_O-F_Shipping)))else(if(F_Shipping>=F_I_O-F_I)then(F_G-(F_Shipping-F_I_O+F_I))else(F_G+(F_I_O-F_I-F_Shipping))))else(0)))
```

```
        F_price = IF((200-(F_I/F_I_O*5))>=250)then 250
        else(IF(200-(F_I/F_I_O*5))<=150then 150 else(200-(F_I/F_I_O*5)))

        Retailer
        R_I(t) = R_I(t - dt) + (R_Shipping + R_D - R_Arrive) * dt
        INIT R_I = 12

        INFLOWS:
        R_Shipping = CONVEYOR OUTFLOW
        R_D =
if(R_I>=0)then(if(R_Shipping+R_I>=Customer)then(0)else(R_Shipping+R_I-Customer))
else (if(R_Shipping>=-R_I)then(if (R_Shipping>=
Customer-R_I)then(R_Shipping-(Customer-R_I)) else
(R_Shipping-Customer))else(R_Shipping-Customer))
        OUTFLOWS:
        R_Arrive =
if(R_I>=0)then(if(R_Shipping+R_I>=Customer)then(Customer)else(R_Shipping+R_I)) else
(if(R_Shipping>=-R_I)then(if (R_Shipping>= Customer-R_I)then(Customer-R_I) else
(R_Shipping))else(R_Shipping))
        R_T(t) = R_T(t - dt) + (W_Arrive - R_Shipping) * dt
        INIT R_T = 8
                TRANSIT TIME = 2
                INFLOW LIMIT = INF
                CAPACITY = INF
        INFLOWS:
        W_Arrive    (Not in a sector)
        OUTFLOWS:
        R_Shipping = CONVEYOR OUTFLOW
        R_G = 12
        R_PO =
int((2-(250-R_price)/185)*(if((if(R_I>=0)then(if(R_Shipping>=Customer)then(R_G-
        R_I+Customer-(R_Shipping- Customer))else(R_G-R_I+(Customer-
        R_Shipping)))else(if(R_Shipping>=Customer-R_I)then(R_G-
        (R_Shipping-Customer+R_I))else(R_G+(Customer-R_I-
        R_Shipping))))>=0)then(if(R_I>=0)then(if(R_Shipping>=Customer)
        then(R_G-R_I+Customer-(R_Shipping- Customer))else(R_G-R_I+
        (Customer-R_Shipping)))else(if(R_Shipping>=Customer-R_I)then
        (R_G-(R_Shipping-Customer+R_I))else(R_G+(Customer-R_I-
        R_Shipping))))else(0)))
        R_price = IF((200-(R_I/Customer*5))>=250)then 250
        else(IF(200-(R_I/Customer*5))<=150then 150 else(200-(R_I/Customer*5)))
        Customer = GRAPH(time)
        (1.00, 4.00), (2.00, 4.00), (3.00, 4.00), (4.00, 4.00), (5.00, 8.00), (6.00, 8.00), (7.00,
8.00), (8.00, 8.00), (9.00, 8.00), (10.0, 8.00), (11.0, 8.00), (12.0, 8.00), (13.0, 8.00), (14.0,
8.00), (15.0, 8.00), (16.0, 8.00), (17.0, 8.00), (18.0, 8.00), (19.0, 8.00), (20.0, 8.00), (21.0,
8.00), (22.0, 8.00), (23.0, 8.00), (24.0, 8.00), (25.0, 8.00), (26.0, 8.00), (27.0, 8.00), (28.0,
8.00), (29.0, 8.00), (30.0, 8.00), (31.0, 8.00), (32.0, 8.00), (33.0, 8.00), (34.0, 8.00), (35.0,
8.00), (36.0, 8.00), (37.0, 8.00), (38.0, 8.00), (39.0, 8.00), (40.0, 8.00), (41.0, 8.00), (42.0,
8.00), (43.0, 8.00), (44.0, 8.00), (45.0, 8.00), (46.0, 8.00), (47.0, 8.00), (48.0, 8.00), (49.0,
8.00), (50.0, 8.00)
        Wholesaler
        W_I(t) = W_I(t - dt) + (W_Shipping + W_D - W_Arrive) * dt
        INIT W_I = 12
        INFLOWS:
        W_Shipping = CONVEYOR OUTFLOW
```

```
    W_D =
if(W_I>=0)then(if(W_Shipping+W_I>=W_I_O)then(0)else(W_Shipping+W_I-W_I_O))
else (if(W_Shipping>=-W_I)then(if (W_Shipping>=
W_I_O-W_I)then(W_Shipping-(W_I_O-W_I)) else
(W_Shipping-W_I_O))else(W_Shipping-W_I_O))
    OUTFLOWS:
    W_Arrive    (Not in a sector)
    W_T(t) = W_T(t - dt) + (D_Arrive - W_Shipping) * dt
    INIT W_T = 8
        TRANSIT TIME = 2
        INFLOW LIMIT = INF
        CAPACITY = INF
    INFLOWS:
    D_Arrive    (Not in a sector)
    OUTFLOWS:
    W_Shipping = CONVEYOR OUTFLOW
    W_G = 12
    W_IO = delay(R_PO,1,4)
    W_I_O = delay(W_IO,1,4)
    W_PO =
int((2-(250-W_price)/185)*(if((if(W_I>=0)then(if(W_Shipping>=W_I_O)then(W_G-W_I+
W_I_O-(W_Shipping-
W_I_O))else(W_G-W_I+(W_I_O-W_Shipping)))else(if(W_Shipping>=W_I_O-W_I)then(
W_G-(W_Shipping-W_I_O+W_I))else(W_G+(W_I_O-W_I-W_Shipping))))>=0)then(if(W
_I>=0)then(if(W_Shipping>=W_I_O)then(W_G-W_I+W_I_O-(W_Shipping-
W_I_O))else(W_G-W_I+(W_I_O-W_Shipping)))else(if(W_Shipping>=W_I_O-W_I)then(
W_G-(W_Shipping-W_I_O+W_I))else(W_G+(W_I_O-W_I-W_Shipping))))else(0)))
    W_price = IF((200-(W_I/W_I_O*5))>=250)then 250
    else(IF(200-(W_I/W_I_O*5))<=150then 150 else(200-(W_I/W_I_O*5)))
    Not in a sector
    D_Arrive =
if(D_I>=0)then(if(D_Shipping+D_I>=D_I_O)then(D_I_O)else(D_Shipping+D_I)) else
(if(D_Shipping>=-D_I)then(if (D_Shipping>= D_I_O-D_I)then(D_I_O-D_I) else
(D_Shipping))else(D_Shipping))
    OUTFLOW FROM:    D_I (IN SECTOR:    Distributor)
    INFLOW TO:    W_T        (IN SECTOR:    Wholesaler)
    F_Arrive =
if(F_I>=0)then(if(F_Shipping+F_I>=F_I_O)then(F_I_O)else(F_Shipping+F_I)) else
(if(F_Shipping>=-F_I)then(if (F_Shipping>= F_I_O-F_I)then(F_I_O-F_I) else
(F_Shipping))else(F_Shipping))
    OUTFLOW FROM:    F_I (IN SECTOR:    Factory)
    INFLOW TO:    D_T (IN SECTOR:    Distributor)
    W_Arrive =
if(W_I>=0)then(if(W_Shipping+W_I>=W_I_O)then(W_I_O)else(W_Shipping+W_I)) else
(if(W_Shipping>=-W_I)then(if (W_Shipping>= W_I_O-W_I)then(W_I_O-W_I) else
(W_Shipping))else(W_Shipping))
    OUTFLOW FROM:    W_I        (IN SECTOR:    Wholesaler)
    INFLOW TO:    R_T (IN SECTOR:    Retailer)
```