

# Web ページを用いた教育用ゲーム開発事例

片 山 益 男

## 要 旨

JavaScript, CSS 技術を用いた, ブラウザーで実行できるゲームを開発した。ここで考案したシステム構成法は広く適用可能である。教育ツールとして利用する場合の限界についても考察した。

## 1. はじめに

本テーマに取り組むことになった主な動機は, web ページで, どの程度の対話処理 (プログラムによるコントロール) が可能かということを経験的に確かめてみたかったためである。

web ページはハイパーリンクによる関連情報の選択的提供に有効であるという認識は定着している。しかし利用者がそれ以上の積極的な働きかけをする例としては, オンライン・ショッピングで商品を購入する時の利用法のようなイメージが中心で, そのような利用にはサーバー処理が関与するので敷居が高いという固定観念を持っている場合が多いのではなかろうか。

筆者もそのような一人であったが, JavaScript, CSS (Cascading Style Sheet) などの初歩を勉強するうちに, もっと利用できるのではないかという疑問がわき, 試行錯誤の結果がこの発表となった。

対象として取り上げたのは, 筆者が20年あまり前に開発した「レコード店のマネージメント・ゲーム」である[1,2]。このゲームは学生の支持があり, 現在まで利用を継続している。

本稿ではまず, web ページ利用システムでのゲームの流れ, システム構成モジュールの設計方針と相互関連づけについて説明する。次いで, 本システムを教育目的に利用することに対する考察を行い, ゲームを教育ツールとして用いることの限界について言及したい。

## 2 ゲームの構成

### 2.1 ゲームの流れ

ゲームの規則, 条件は手計算の場合と全く同一とした。すなわち, 1枚1,500円で仕入れ, 2,500円で販売するという単純なもので, 経営の意思決定項目として「販売方法」「広告の有無」「陳列台の台数」「仕入数量」「借入金の増減」を設定し, 利益追求などを競わせるというものである。

web ページを利用したゲームにおいて、どのような画面をどのような順序で表示させているかを次に説明する。それによりゲームの流れを把握することができる。

web ページを読み込むと図1のオープニング画面が表示される。

「データ入力」ボタンを押すと図2が表示され、ゲーム開始時の初期値を入力する。次に図3で基本需要や広告効果のタイプを指定する。これらは手計算のゲームでは環境からの与件であり、ゲームの進行係がその都度指定するのであるが、全く同じ方式を取ると、その都度サーバーへのアクセスを発生させることになるので、このステップで事前に設定することとした。意思決定項目での指定内容がどのような影響を与えるかの理解がゲーム利用の主目的なので、このような設定方法が学習に悪影響を与えることはない。

基本需要のパターンや広告効果の指定は、図3に示すように、デフォルト値からの選択、または数値指定ができるように設計した。

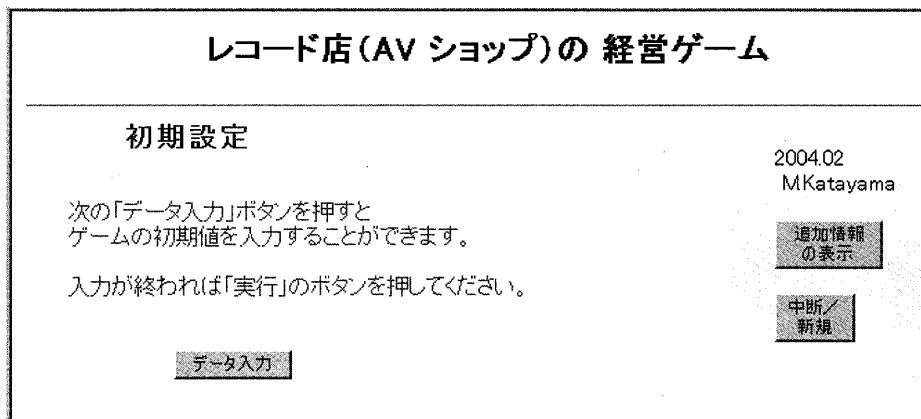


図1 オープニング画面

### レコード店(AV ショップ)の 経営ゲーム

---

**初期設定** 2004.02  
MKatayama

ゲーム開始の初期値(期首貸借対照表の値)を入力してください。

追加情報  
の表示

中断/  
新規

現金預金	<input type="text" value="2200"/>
売掛金	<input type="text" value="1700"/>
受取手形	<input type="text" value="0"/>
商品	<input type="text" value="2700"/>
買掛金	<input type="text" value="1200"/>
借入金	<input type="text" value="6000"/>
納税引当金	<input type="text" value="280"/>
什器備品	<input type="text" value="1600"/>
構築物	<input type="text" value="1800"/>
資本金	<input type="text" value="2000"/>
繰越利益金	<input type="text" value="240"/>
税引純利益	<input type="text" value="280"/>

図 2 B/S 初期値入力

### レコード店(AV ショップ)の 経営ゲーム

---

**初期設定** 2004.02  
MKatayama

需要のパターンと広告効果についての設定

追加情報  
の表示

中断/  
新規

既設のパターンから選択

- A (横ばい)
- B (上昇)
- C (下降)
- D (変動大)
- パターンの作成(以下で設定)

---

[需要パターン]

1月目の基本需要枚数  枚

月あたりの変化: 横ばいなら 0 を、減少ならマイナスの値を記入

枚数  枚 または、対前月比(%)

不規則変動の幅 (対基本枚数の +/-%)

[広告効果の大きさ] (5~30%)

効果の平均(%)

効果変動の有無  なし  あり

図 3 需要パターン, 広告効果の入力

以上で初期設定は終わり、「実行」ボタンでゲームの実行画面（図4～7）へと進む。図4～7の各画面の中央には、その時点で意思決定するのに直接必要となる情報が表示される（各画面で表示内容が変化）。意思決定にあたり、どのような情報提供が必要か、それらを見やすい様式で表示するにはどうすれば良いかという課題は重要なテーマの一つであるが、その方法は常識に委ねられたままのようである。ここではそのような問題意識を持ちつつ試行錯誤を重ね、一応図4～7の様式に落ち着いた。さらに、右側に配置されている「追加情報の表示」ボタンで、過去の決定内容（図8）をいつでも表示できる。手計算方式では実行経過を記したワークシートをいつでも参照できるので問題が表面化しないが、コンピュータ利用方式では意識的に考慮しなければならない。

各画面は意思決定結果を入力する時に表示され、それ以外の処理はすべて自動的に実行される。

実行途中で支払資金が不足すると「倒産」状態となる。そのときは図9の画面が表示される。なお、ゲームを中断し最初からやり直したいときには、図の右側にある「中断／新規」ボタンを押す。

### レコード店(AVショップ)の経営ゲーム

---

**1期 1月目**

販売制約(先月)	
在庫制約(先月)	
販売枚数(先月)	
在庫数量	1800
仕入支払後現金残高	1000
入金予定額(今月)	1700
現金在高	2200
仕入金額(翌月払)	
借入残高	6000

販売方法

広告

陳列台数(先月)     4

陳列台数

仕入可能枚数

販売枚数

仕入枚数

売上高

借入／返済額

2004.02

MKatayama

追加情報の表示

中断／新規

販売方法と、広告を出すか否かを指定してください。

<p>販売方法</p> <p><input type="radio"/> 現金売</p> <p><input checked="" type="radio"/> 掛売</p> <p><input type="radio"/> 手形売</p>	<p>広告</p> <p><input checked="" type="radio"/> 出さない</p> <p><input type="radio"/> 出す</p>
--------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------

図4 販売方法などの指定

### レコード店(AV ショップ)の 経営ゲーム

---

<b>1 期 1 月目</b>		2004.02 MKatayama
販売制約(先月)	販売方法	掛売
在庫制約(先月)	広告	出さない
販売枚数(先月)	陳列台数(先月)	4
在庫数量 1800	陳列台数	4
仕入支払後現金残高	仕入可能枚数	600
入金予定額(今月)	販売枚数	
現金在高 2700	仕入枚数	
仕入金額(翌月払)	売上高	
借入残高 6000	借入/返済額	

陳列台の台数を入力してください。

図 5 陳列台台数の指定

### レコード店(AV ショップ)の 経営ゲーム

---

<b>1 期 1 月目</b>		2004.02 MKatayama
販売制約(先月)	販売方法	掛売
在庫制約(先月)	広告	出さない
販売枚数(先月)	陳列台数(先月)	4
在庫数量 1800	陳列台数	4
仕入支払後現金残高	仕入可能枚数	600
入金予定額(今月)	販売枚数	
現金在高 2700	仕入枚数	
仕入金額(翌月払)	売上高	
借入残高 6000	借入/返済額	

仕入枚数を入力してください。(販売枚数の決定、経費の支払いまで自動的に処理します。)

図 6 仕入枚数の指定

## レコード店(AV ショップ)の経営ゲーム

1 期 1 月 目			2004.02 MKatayama
販売制約(先月)		販売方法	掛売
在庫制約(先月)		広告	出さない
販売枚数(先月)		陳列台数(先月)	4
在庫数量	1720	陳列台数	4
仕入支払後現金残高		仕入可能枚数	600
入金予定額(翌月)	1700	販売枚数	680
現金在高	2110	仕入枚数	600
仕入金額(翌月払)	900	売上高	1700
借入残高	6000	借入/返済額	

借入金の追加/返済金額(返済ならマイナス, 変化無しなら 0)を入力してください。

0

図7 借入金の増減を指定

第1期 Fri Feb 06 19:24:58 2004

需要パターンA(横ばい)

今期の記録

項目名	1月	2月	3月	4月	5月	6月
販売方法	1	0.9	1.1	1.1	1.1	0
広告の有無	0	0	0	1	0	0
陳列台数	4	4	4	5	5	0
販売法補正	680	612	748	897	748	0
在庫制約	720	720	719	803	862	0
実際販売枚数	680	612	719	803	748	0
広告代	0	0	0	100	0	0
陳列台費用	0	0	0	10	10	0
人件費等	530	530	530	530	530	0
借入利息	60	60	60	60	80	0

図8 追加情報の表示

以上で1サイクルのゲームが終了し、図10が表示される。この画面で、そのサイクル(月)のゲームを確定するかどうかを選択できる。これは誤って途中で間違っただけの入力をしたとき、部分的なやり直しで済むようにという配慮である。

### レコード店(AV ショップ)の 経営ゲーム

---

**2 期 1 月目**

販売制約(先月)	680	販売方法	掛売
在庫制約(先月)	720	広告	出さない
販売枚数(先月)	680	陳列台数(先月)	4
在庫数量	1720	陳列台数	
仕入支払後現金残高	-100	仕入可能枚数	
入金予定額(今月)	1700	販売枚数	
現金在高	920	仕入枚数	
仕入金額(翌月払)		売上高	
借入残高	4500	借入/返済額	

2004.02  
MKatayama

追加情報  
の表示

中断/  
新規

同じ初期値を用いてやり直す

初期値入力からやり直す

図9 倒産発生時の表示

### レコード店(AV ショップ)の 経営ゲーム

---

**1 期 1 月目**

販売制約(先月)		販売方法	掛売
在庫制約(先月)		広告	出さない
販売枚数(先月)		陳列台数(先月)	4
在庫数量	1720	陳列台数	4
仕入支払後現金残高		仕入可能枚数	600
入金予定額(翌月)	1700	販売枚数	680
現金在高	2110	仕入枚数	600
仕入金額(翌月払)	900	売上高	1700
借入残高	6000	借入/返済額	0

2004.02  
MKatayama

追加情報  
の表示

中断/  
新規

今月のゲームを確定します。それとも今月のゲームを再度やり直しますか？

確定
やり直し

図10 1サイクル終了時

「確定」ボタンを押すと次のサイクルが始まる（図11）。「やり直し」ボタンを押すと、現在のサイクルの初期状態が表示される。

図11は図4と同じステップの画面であるが、より多くの項目に数値が入っている。

1期（6ヶ月）が終了したときは、図12が表示され、次のアクションが選択できる。「ワークシートの表示」または「決算書の表示」ボタンを押すと、別のウィンドウにそれぞれ図13、14が表示される。どちらも手計算ゲームとほぼ同じ様式にしてある。これらは閉じずに置き、後刻、別の結果と比較することができる。

「継続」ボタンを押すと、次期の開始画面として図15が表示される。「終了」ボタンを押すと図16が表示され、それ以降を継続することはできない。

## 2.2 ゲーム・システムの構成法

### 1) JavaScript と CSS による画面連結

上に紹介した、数多くの画面を持つ本システムには何個くらいの HTML 文書が用いられているのか？ 実はたった1個のファイルだけである。複数の HTML 文書（ページ）を用いると、ページ間での直接的な情報伝達ができないため、サーバーを経由せざるを得なくなり、非実際的な方法となる。複数のウィンドウを表示させるのに、メイン・ウィンドウからサブ・ウィンドウを表示させるという方法もあり、本システムでも一部分用いているが、サブ・ウィンドウからメイン・ウィンドウへ情報を伝達させるのも制約が多くて、自由自在に使いこなすのは難しい。

試行錯誤の結果、ここでは JavaScript と CSS とを用いて、画面を連結するという方法で解決した。画面といってもフル画面ではなく、画面全体の一部分であっても良いので「画面部品」と呼ぶことにする。まず複数の画面部品を定義し、それぞれに名前を付ける。本システムで実際に用いている定義から2つを次に例示する（例1では「gpirt4」、例2では「gpirt8」という名前が付けられている）。例示の中で記述されているように、その画面部品の表示位置を指定することができる。異なる画面部品の表示位置が同じであってもよい。

#### 例 1

```
<div id="gpirt4" style="position: absolute; top: 120; left: 30; color:blue">
<form name="f2" ><br>
  次の「データ入力」ボタンを押すと <br>
  ゲームの初期値を入力することができます。<br><br>
  入力が終われば「実行」のボタンを押してください。<br><br><br>
    <button type="button" onClick = "initdata ()"> データ入力 </button>
</form>
</div>
```

#### 例 2

```
<div id="gpirt8" style="position: absolute; top: 455;left: 100; color:blue"width:300px">
```



### レコード店(AV ショップ)の 経営ゲーム

---

**1 期 2 月目**

販売制約(先月)	680	販売方法	2004.02
在庫制約(先月)	720	広告	MKatayama
販売枚数(先月)	680	陳列台数(先月)	4
在庫数量	1720	陳列台数	<input type="button" value="追加情報の表示"/>
仕入支払後現金残高	1210	仕入可能枚数	<input type="button" value="中断/新規"/>
入金予定額(今月)	1700	販売枚数	
現金在高	2110	仕入枚数	
仕入金額(翌月払)		売上高	
借入残高	6000	借入/返済額	

販売方法と、広告を出すか否かを決めてください。

<p>販売方法</p> <p><input type="radio"/> 現金売</p> <p><input checked="" type="radio"/> 掛売</p> <p><input type="radio"/> 手形売</p>	<p>広告</p> <p><input checked="" type="radio"/> 出さない</p> <p><input type="radio"/> 出す</p>
--------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------

図11 次のサイクルの開始

### レコード店(AV ショップ)の 経営ゲーム

---

**1 期 6 月目**

2004.02  
MKatayama

1期が終了したので決算書項目を計算しました。

次の期へ、ゲームを継続しますか？

継続する場合は、ワークシートの内容がクリアされますから、

「継続」ボタンを押す前に、ワークシートを表示させ、それをプリントアウトするか、

その内容をコピーして、Excel のワークシートに貼り付けて保存してください。

(ワークシートのページを HTML 文書として保存できません。)

図12 1期(6サイクル)終了時の画面

第1期 Fri Feb 06 19:29:15 2004

## ワークシート

項目名	1月	2月	3月	4月	5月	6月	計/残	
販売方法	1	0.9	1.1	1.1	1.1	1.1	0	0
広告の有無	0	0	0	1	0	1	0	0
現金残高1	2200	2110	3850	2240	2625	2302	2089	0
買掛金	1200	900	1020	915	1500	1500	1305	0
現金残高2	1000	1210	2830	1325	1125	802	0	0
売掛金	1700	1700	0	0	0	0	0	0
受取手形	0	0	0	0	1797	2007	1870	2242
現金残高3	2700	2910	2830	1325	2922	2809	0	0
陳列台数	4	4	4	5	5	5	0	0
在庫数1	1800	1720	1788	1679	1876	2128	2101	0
仕入枚数	600	680	610	1000	1000	870	0	0
在庫数2	2400	2400	2398	2679	2876	2998	0	0
基本需要枚数	680	680	680	680	680	680	0	0
広告効果%	0.2	0.2	0.2	0.2	0.2	0.2	0	0
広告効果補正	680	680	680	816	680	816	0	0
販売法補正	680	612	748	897	748	897	0	0
在庫制約	720	720	719	803	862	899	0	0
実際販売枚数	680	612	719	803	748	897	0	0
在庫数3	1720	1788	1679	1876	2128	2101	0	0
仕入金額	900	1020	915	1500	1500	1305	0	0
売上高	1700	1530	1797	2007	1870	2242	0	0
現金売上	0	1530	0	0	0	0	0	0
現金残高4	2700	4440	2830	1325	2922	2809	0	0
広告代	0	0	0	100	0	100	0	0
陳列台費用	0	0	0	10	10	10	0	0
人件費等	530	530	530	530	530	530	0	0
借入利息	60	60	60	60	80	80	0	0
現金残高5	2110	3850	2240	625	2302	2089	0	0
借入/返済額	0	0	0	2000	0	0	0	0
現金残高6	2110	3850	2240	2625	2302	2089	0	0
借入残高	6000	6000	6000	6000	8000	8000	8000	0

図13 ワークシートの表示

第 2 期 Fri Feb 06 19:35:06 2004

## 決算書項目

## 貸借対照表

科目名	開始時	1期末	2期末
現金預金	2200	1809	4795
売掛金	1700	0	1700
受取手形	0	4112	0
商品	2700	3151	2578
買掛金	1200	1305	1020
借入金	6000	8000	8000
納税引当金	280	173	79
什器備品	1600	1400	1200
構築物	1800	1700	1600
資本金	2000	2000	2000
繰越利益金	240	520	694
税引純利益	280	174	80
総資産		12172	11873
総資本		12172	11873

## 損益計算書

科目名	開始時	1期	2期
売上高		11146	10877
期首商品在高		2700	3151
商品仕入高		7140	5955
期末商品在高		3151	2578
売上総利益		4457	4349
販売管理費		3710	3710
人件費等		3180	3180
減価償却費		300	300
広告費		200	200
陳列台費用		30	30
営業利益		747	639
支払利息		400	480
経常利益		347	159

図14 決算書の表示 (2期継続時の例)

### レコード店(AV ショップ)の 経営ゲーム

---

**2 期 1 月 目**

販売制約(先月)	897
在庫制約(先月)	899
販売枚数(先月)	897
在庫数量	2100
仕入支払後現金残高	504
入金予定額(今月)	1870
現金在高	1809
仕入金額(翌月払)	
借入残高	8000

販売方法	
広告	
陳列台数(先月)	5
陳列台数	
仕入可能枚数	
販売枚数	
仕入枚数	
売上高	
借入/返済額	

2004.02  
MKatayama

追加情報  
の表示

中断/  
新規

販売方法と、広告を出すか否かを指定してください。

販売方法	広告
<input type="radio"/> 現金売	<input type="radio"/> 出さない
<input type="radio"/> 掛 売	<input checked="" type="radio"/> 出 す
<input checked="" type="radio"/> 手形売	

OK

図15 次の期の開始

### レコード店(AV ショップ)の 経営ゲーム

---

**2 期 6 月 目**

ゲームを終了します。

決算書項目やワークシートの内容を保存したい場合は、  
それらを表示させて、プリントアウトするか、  
その内容をコピーして、Excelのワークシートに貼り付けて  
保存してください。  
(ページ内容を HTML 文書として保存できません。)

ワークシートの表示

決算書の表示

2004.02  
MKatayama

追加情報  
の表示

中断/  
新規

図16 ゲーム終了時の画面

```
<form name="f8" ><br>
今月のゲームを確定します。それとも今月のゲームを再度やり直しますか? <br><br><br>
    <button type="button" onClick = "nextm ()"> 確定 </button>
    <button type="button" onClick = "redo ()"> やり直し </button>
</form>
</div>
```

各画面部品の表示, 非表示の指定は Java Script の中で, 次のような記述をすることにより行える。「gpirt4」を表示させ「gpirt8」を非表示にするという例を次に示す。

### 例 3

```
document.all ("gpirt4") .style.visibility = "visible";
document.all ("gpirt8") .style.visibility = "hidden";
```

複数画面を連結するには, 1つの画面にボタンを配置し, そのボタンに実行する関数を対応づける(予め実行する内容をすべて関数群として定義しておく必要がある)。対応づけた関数の中に, 次に表示させる画面部品を指定しておき, その画面でまたボタンと関数の対応づけを準備しておけば, 希望する順序で処理が進行する。

ボタンと関数の対応づけの仕方は, 上に示した例 1 では次のようになっている。

```
<button type="button" onClick = "initdata () "> データ入力 </button>
```

すなわち「データ入力」という名前のついたボタンを押せば「initdata ()」という関数が実行される。ちなみに initdata () という関数の内容は次のように定義されている。

### 例 4

```
// 開始時決算データの入力画面を表示
function initdata () {
document.all ("gpirt4") .style.visibility = "hidden";
document.all ("gpirt9") .style.visibility = "visible";
}
```

複数の画面部品で, ボタンの位置が同じであっても, そのボタンをクリックしたとき実行されるのは, 表示画面部品での定義であり, その他の非表示画面部品での定義をアクティブ化させることはない。複数の画面部品がアクティブなとき, 表示されているすべての部分(ボタンやテキストボックス)に対してアクセスできる。

このような対応づけを用いて作成した本システムの画面と処理（関数）の関連図を図17に示す（倒産時の処理のような、例外的な処理部分は複雑さを避けるため省略してある）。その前段として、本システムで定義した関数、画面部品のリストを表1, 2に示す。さらに、各関数の中で、どの画面部品を表示／非表示させているかの一覧を表3に示す。

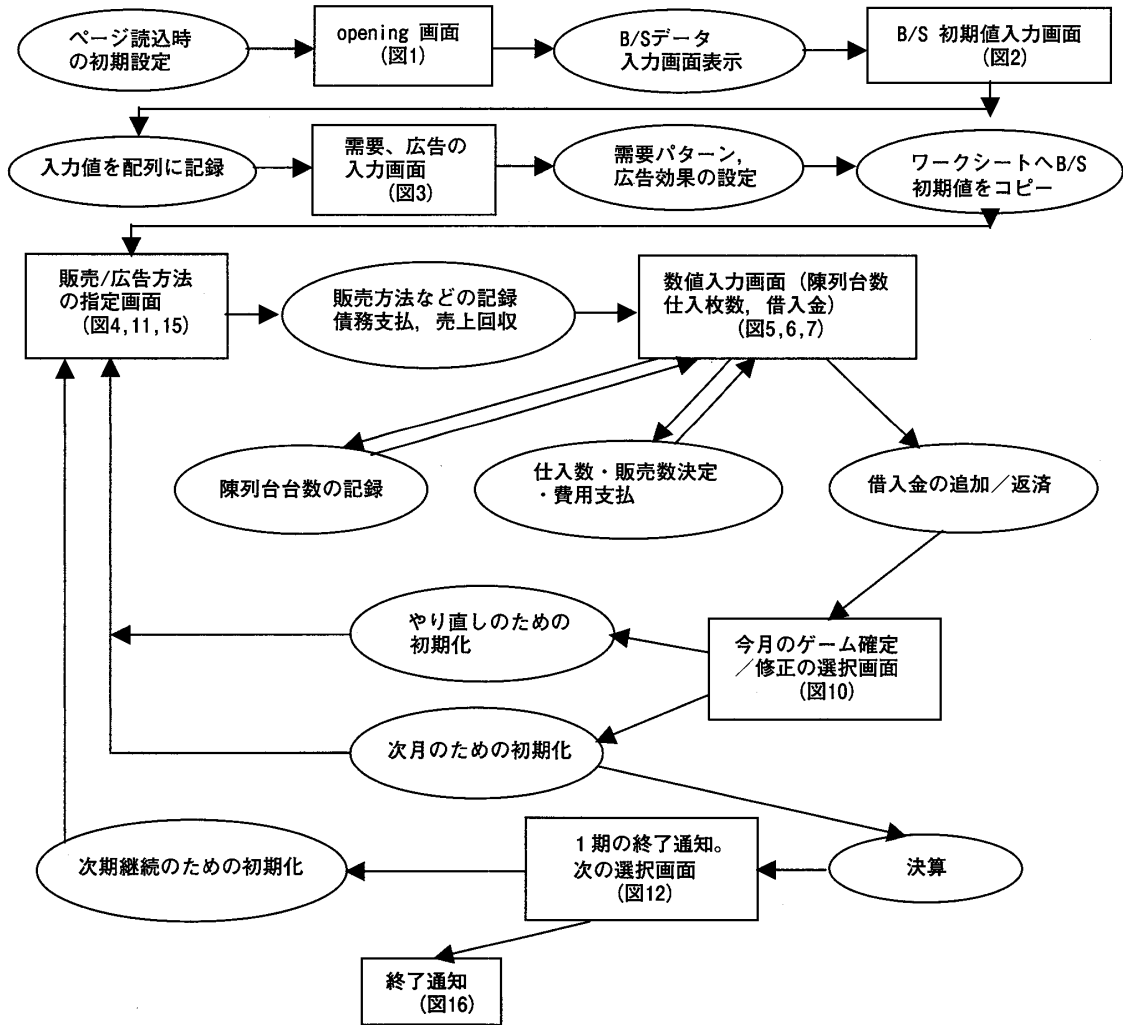


図17 処理（関数）と画面の関係

表1 画面部品のリスト

名称	利 用 目 的	ボタンで呼び出す function
area1	現ステップの案内 (x 期 x 月)	
gp1	決定に必要な諸情報の表示	
gp2	何を入力するか案内 (各ステップで内容は変化)	
gp3	数値の入力 (陳列台数, 仕入枚数, 借入金)	go1
gp4	opening 画面	initdata
gp5	販売 / 広告方法の指定	howsales
gp6	倒産の通知	tosan1,2
gp7	右側に常時表示	show2
gp8	今月のゲーム確定 / 修正の選択	nextm,redo
gp9	開始時の B/S 初期値入力画面	init2
gp9a	需要パターンと広告効果の設定	juyoplan
gp10	1 期の終了, 次の選択	keizoku,done1,showwkt
gp11	ゲーム終了の通知と次の選択	showbspl,showwkt

表2 関数のリスト

関数名	処 理 内 容	chain function
init1	ページ読込時の初期設定	init1a
init1a	ワークシートの初期化	
initdata	開始時決算データの入力画面を表示	
init2	テキストボックスの値を配列にコピー	
juyoplan	需要パターン, 広告効果の設定	mkjuyot1,2,3; init3
mkjuyot1	基本需要枚数表の作成 1	
mkjuyot2	基本需要枚数表の作成 2	
mkjuyot3	基本需要枚数表の作成 3	mkyuyot1,2
init3	ワークシートへ B/S 初期値をコピー	
howsales	販売方法などの記録, 債務支払, 売上回収	tosan
go1	数値入力時の選択的実行	step4,5,8
step4	陳列台台数の指定	
step5	仕入枚数決定・販売数決定・費用支払	tosan
step8	借入金の追加 / 返済	tosan
nextm	一ヶ月終了, 次月のための初期化	kessan
redo	その月の処理をやり直すための初期化	
tosan	倒産時の処理	
tosan1	同じ初期値で再実行のための初期化	init3
tosan2	初期値入力から再実行のための処理	init1
kessan	決算	tosan
keizoku	次の期を続けるための初期化	mkjuyot1,2; init1a,3
done1	終了	
showbspl	決算書の表示	
showwkt	ワークシートの表示	
show2	追加情報の表示	

表3 各関数からの画面部品表示制御

関数名	画面部品名 (数字部分のみ)												
	a1	1	2	3	4	5	6	7	8	9	9a	10	11
init1a		x	x	x	o	x	x	o	x	x	x	x	x
initdata					x					o			
init2										x	o		
init3		o	o	x	x	o					x		
howsales				o		x							
step8			x	x					o				
nextm		o	o	x	x	o	x		x				
redo		o	o	x	x	o	x		x				
tousan		o	x	x	x	x	o	x	x	x	x		
tosan1		x	x	x	o	x	x	o	x	x	x	x	x
kessan		x	x	x	x	x	x	x	x	x	x	o	
keizoku												x	
done1												x	o

O: 表示, x: 非表示

## 2) 入力データの記録

本ゲームでは、初期値として決算書の数値、進行途中では意思決定結果としての諸数値（仕入枚数、借入金増減額など）をそのつど入力していく必要がある。それらを変数や配列要素に、どのように記録すればよいのか、初歩的なマニュアルでは説明がなく、発見するまでに多少の時間、労力を費やした。採択した1つはeval関数を用いる方法である。例3に、「din」という名称のフォームの「d1」という名前のテキストボックスに入力した数値を、配列要素に代入する例を示す（複数の画面部品にそれぞれフォームを定義するとき、異なる画面部品間であっても、同一のフォーム名称を付けてはならない）。

### 例5

```
bst [1][1] =eval (document.din.d1.value) ;
```

今ひとつの方法として、図4、11に表示されている販売方法は選択方式となっているが、その記録は次のように処理した。まず画面部品を次のように定義した。

### 例6

```
<div id="gp5">
<div style="position: absolute; top:420; left: 80; background-color="#c4ffff"; width:150px; height:160px">
<form name="f3"><br>
  販売方法 <br><br>
  <input type="radio" name="msales"> 現金売 <br>
```



```

<input type="radio" name="msales" checked> 掛 売 <br>
<input type="radio" name="msales"> 手形売 <br>
      <button type="button" onClick = "howsales () ">OK</button>
</form>
</div>

```

そして、その読取りを次のように処理した。

```

// 販売方法の補正係数をワークシートの項目に記録
for (i=0; i<=2; i++) if (document.f3.elements [i].checked) { a=i+1; }
switch (a) {
case 1: wkt [n][1]=0.9; break;
case 2: wkt [n][1]=1; break;
case 3: wkt [n][1]=1.1; break;
}

```

### 3) 変数, 配列に記録された内容の画面への表示

通常のマニュアルでは、フォームのテキストボックスに表示させる方法が紹介されている。しかし、ワークシートや決算書のような多数のデータを含む内容を表示させるのにテキストボックスを使うのは非実際的である。

本システムで用いた2つの方法を紹介する。まず図13に例示したワークシートのような、配列データ全体を表示させるためには、サブ・ウィンドウを使う(新しいウィンドウを作成して表示)という方法をとった。ワークシートを表示させるために作成した関数を次に紹介する。document.write という関数を用い、配列 wkt [i][j] に記録されているワークシートのデータをサブ・ウィンドウに送り出すのであるが、タグ記号まで含めてHTML 表記に必要なすべての文字を送り出す必要がある。

#### 例 7

```

// ワークシートの表示
function showwkt () {
var subw2 = window.open ("","","width=500, height=700,scrollbars=yes, resizable=yes");

subw2.document.write ("第 " + termn + " 期 " + Date () + "<br>");
subw2.document.write ("<center>");
subw2.document.write ("<br> ワークシート <br><br>");

subw2.document.write ("<table border="1" cellpadding="0" cellspacing="2" >");

for (j = 0; j <= 31; j++) {
subw2.document.write ("<tr align = "right">");
for (i = 0; i <= 8; i++) {

```

```

subw2.document.write("<td>" + wkt [i][j] + "</td>");
}
subw2.document.write("</tr>");
}
subw2.document.write("</table>");
subw2.document.write("</center>");
subw2.document.close ();
}

```

今ひとつの工夫は、図4～7に例示したような、ステップ毎に内容が変化する意思決定支援情報の表示を、テキストボックスよりスマートな様式とするため、画面部品を使った方法である。仕入枚数が入力されると、その数値を画面に表示させる例を次に示す。まず画面部品「area207a」を定義する（3行目が空白になっているが、スペースが入っている）。

#### 例8

```

<div id="area207a" style="position: absolute; top: 260; left: 410; text-align:right; background-color: blanc
hediamond;width:60px; height:23px">

```

```

</div>

```

そして、関数の中で次の文を実行させる。本例では変数 buy に入っている値が「area207a」で定義された場所に表示される。

```

area207a.innerHTML = buy;

```

#### 4) 1つのボタンで複数の処理を選択的に実行

図17の中ほどにある「数値入力画面」ではボタンが1つ置かれているだけだが、陳列台数、仕入枚数、借入金増減額の3種のデータ入力に用いている。その仕掛けは「数値入力画面」をアクティブにする関数の中に、次に実行する関数名を指定しておくという方法である。画面部品の種類をなるべく少なくする工夫として用いた。

#### 例9

```

// 数値入力ボタンが押されたとき
function go1 () {
switch (stepn) {
case 4: step4 (); break;
case 5: step5 (); break;
case 8: step8 (); break;
default:alert ("prog-error-1");
}
}

```

これらの方法を用いて、1つの文書内に必要なデータをすべて記録し、見かけ上多数の画面を有するシステムを構築することができた。完成した HTML 文書の総行数は1,000行前後である。ファイルサイズは46KB、実行時のメモリー追加使用量は1MB 以内と想定される。この程度の負荷なら、通常のパソコンで十分対応できる。

ただし、ブラウザの種類により処理のされ方に相違があり、すべてのブラウザで動作可能かどうかは確認していない。Windows PC と Mac では、画面部品の表示位置や、スペース文字の扱い方に相違があり、全く同じ画面としては表示されない。

今一つの制約は、読み込まれた HTML 文書のソースリストが可読であることである。JavaScript の部分を別のファイルに分離して読み込ませるという方法もあるようであるが、非公開部分をどうしても設定したいときの方策は不明である。

この開発から明らかになった1つは、ブラウザ・ソフトと、メモ帳のようなエディターさえあれば、プログラミング教育のほとんどが可能だということである。欠落しているのは外部データ (ファイル) 入出力機能だけである。

本システムは2002年夏に開発を始め、秋には完成させていた。雑用にまぎれて発表用原稿の作成には手がつかずにいたが、定年が急遽2年早まるという事態が発生し、その前には脱稿させておきたいという気持ちから作業を再開した。1年ぶりにソースリストを見て愕然としたのは、内容がまったく理解できないということであった。JavaScript の記述文法が記憶から消え去っていたのである。その愚を繰り返したくないので、備忘録としてスクリプトの記述構成を次に記す(関数は <head> ブロック内に、画面部品は <body> ブロック内に記述する)。

```
<html>
<head>
<meta http-equiv="content-type" content="text/html;charset=x-sjis">
<title> レコード店の経営ゲーム </title>
<SCRIPT LANGUAGE="JavaScript">
<!--
```

この部分に、すべての変数・配列・関数の定義を記述する。

```
window.onload = init1;// 読み込み時に実行する関数を指定
```

```
// JavaScript の終了
// -->
```

```
</SCRIPT>
</head>
```

次に HTML 文書本体を記述

```
<body>
<center>
<h2> レコード店 (AV ショップ) の経営ゲーム </h2><hr size=1><br>
</center>
```

画面部品の定義

```
<div id="xxx">
  <form name="xxx" ><br>
    <button type="button" onClick = "xxx () ">
      ....
  </form>
</div>
の繰り返し

</body>
</html>
```

### 3 教育目的から見た本システムの有効性

結論は、かなり限定的であろうという予想である。その予想の理由を次に説明する。

原型とした手計算によるゲームは冒頭に述べたように20年あまり前に開発し、たまたま学生の支持が継続するので現在まで利用してきた。開発の動機は、経営の論理的構造の部分を具体的な例を用いて理解を促進させたいという点にあった。適用の主対象としたのは関西の上位クラス大学の経営系学部での1科目であった。受講者数は毎年300~500名程度である。2002年に終了するまでの十数年間、このゲーム(手計算)をレポート問題の一つとして課した。問題内容の骨子は一貫して「同一の前提条件で、2つ以上のゲームを実行し、結果を比較せよ」というものである。

このレポートに対する典型的な感想は「初めは面倒な手計算をどうして行わないといけないのか、レポート作成のため仕方なく開始したが、意味が解ってくると面白くなった」「年数回のレポートのうち最も印象に残っている」というものであり、「後輩のため来年以降もこの課題を続けてください」という記述が年に数件ある。

本務校で、ゼミ(1クラス二十数人)の時間にも、このゲームを約10年間取り上げてきた。手計算ゲームに対する反応は同様で、過半数が興味を示すようになる。経営ゲームの開発を卒論のテーマとしたいという学生も数人出た。ゼミでは、手計算ゲーム終了後、このゲームをExcelにより計算を自動化するシートを作成するという演習を行わせる。使用するのは、計算式、if関数、

min 関数など、ごく初歩的な要素だけであるが、「Excel の実際的な使い方が体験できた」という効果がある。さて、Excel シートを用いると、意思決定項目への入力以外はすべて自動計算されるようになり、それを使って業績を競わせるという課題を出すのであるが、大部分の学生は積極的な関心を示さない。おざなりに結果を1つ提出しておしまいということになってしまう。

時間的な制約などから、それ以上のゲーム利用を続けることは行っていないが、利用を継続するには、単に「業績向上の余地を追求せよ」というような課題の与え方ではなく、学習者の取り組み意欲を高めるような課題の出し方や、着眼点に関するヒントを提供するような運営が必要になる。ゲーム開発の意図としては、販売方法と在庫方針の調和、どのような場合に広告が有効かの見定めなどについて追求するといった着眼を期待しているので、それらの事項を織り込んだ多段階の課題設定が有効ではないかと考えている。いずれにせよ、コンピュータ・ゲームの方が手計算よりも興味を継続させることが難しいという事例（一般的傾向？）があることを記しておきたい。

今回の開発ゲームはまだ学生に実施させてはいないが、筆者自身の試行的実施の体験から、手計算ゲームのような教育効果を期待できないだろうと考えている。Excel シートを用いた場合と共通する現象であるが、コンピュータを使う場合、数値などのデータ入力の際に思考を働かせることが困難なのである。（課題達成への高い緊張感が存在しない場合）連打式パチンコと同様に、「狙い定めて」ではなく、「数撃ちゃ当たる」という心的状況を作り出してしまう傾向を持つためだと考えている。手計算ゲームの場合、入力データを用いて計算するときにゲームの規則を知ることが必要になるので、計算するという心的作業を通じて自然にゲームの規則理解が同時に進むことになり、ゲームに対する興味も高まることになる。コンピュータ利用の場合「計算」はコンピュータが肩代わりしてくれるので、人間の思考が介在する余地がなくなる。従って、コンピュータを利用した教育的ゲームには「思考を抑制する傾向を持つ」という共通的な性質があるのではなかろうか、というのが筆者の仮説である。

さて、何らかの方策で、学習者の課題達成への動機づけを高めることができた場合、学習ツールとして今回開発のような対話的ゲーム形式が有効なのであろうか？ ゲームの骨格理解のためには手計算方式の方が優れている。その次の段階での課題といえは、最適運用条件を探るという問題（外部条件が変化した場合の対処法を含む）か、ゲームの骨格（規則）そのものを変更すればどうなるかということの探索かのどちらかであることが多い。そのような問題を考える場合、対話的ゲーム形式はまどろっこしくて使う気にならないことが多い。ほとんどの場合、シミュレーション・モデルを作り、種々の条件下で実施した結果を比較するという方法を取るほうがずっと实际的である。今回対象としたゲームのような場合、Excel で簡単にゲーム規則を設定できるので、演習で作成した Excel シートを利用する方法の方がずっと実用度が高い。

それでは今回開発したゲームに出番はあるのだろうか？ まず課題達成への動機づけを高める（例えば成績評価の点数に加える）ことで、ある程度の利用は可能であろう。しかし対話的ゲームの特徴を生かせる最大の用途は、複数のプレイヤーが有限資源の争奪を競うような、本来の意

味でのゲーム（ネットワークゲーム）を実施する場合であろう。その場合、予め第一段階として手計算方式によるゲーム規則の理解、第二段階としてシミュレーション方式による運営ノウハウの学習を済ませておくことが望まれる。

ネットワークゲームは勿論、今回開発したものだけでは実施できないが、それを可能にするモジュールを追加すればできると予想している（条件が許せば取り組みたいと考えている）。

今一つの用途として、環境が動的に変化する場合の対応を学習する場合のツールとしての使い方がある。少し話が横道へそれるが、筆者は最近 PowerSimm という、システム・ダイナミックスのツールを少し勉強している。システム・ダイナミックスについては、すでに20年以上前に、それを用いた業界モデルの構築に参加したことがあり [3]、一応の基礎は理解しているが、今回のようなゲーム・モデルのシミュレータを作るのには不適當なものと思っていた。しかし最近、PowerSimm で離散現象がモデル化できるという話を聞いて再勉強する気になったのである。今回の例では、債務支払と売上債権回収のタイミングを変更すれば所要資金がどの程度変化するかというような問題をシミュレートできるからである。この点については多少の勉強の結果、記述可能であることを体験的に確認した。次におつかっている壁は、意思決定規則をどのようにしてシミュレーション・モデルに組み込むかという問題である。静的な規則なら簡単であるが、例えば売上傾向が下降から上昇に転じてきたとき、規則を変更しないと環境への適応が不十分となる。過去の売上データから傾向値を計算する統計モデルは存在するが、そのモデルに入力する多数のデータを PowerSimm のシミュレータ内に記録しておく方法が無いのである。（PowerSimm では、一定期間のシミュレーション実行結果を Excel シートに書き出し、Excel で必要な統計処理をして、その結果を再びシミュレータに取り込んで、次ステップのシミュレーションを実行するという方法で対処しているようである。）

そのような複雑な手数を掛けるなら、意思決定機能を人間が担当して、対話的ゲームでシミュレーションを実施するという方法の存在余地が出てくる。

#### 4 おわりに

web ページを JavaScript と CSS によりコントロールすることで、多画面とプログラム処理を含む複雑なシステムの構築ができることを例証した。

以前に開発した「we 利用自主学习システム」[4] に組み込むことで、サーバー負荷の負担を軽減する効用があることにも気づいた。

さらにサーバーとの連携部分を付加することによりネットワーク・ゲームに発展させる可能性についても一定の見通しを得た。

開発システムの利用に関して、コンピュータ利用ゲームというツールを教育目的に用いようとする場合、適用範囲がかなり限定されるのではないかという見解を示した。

### 引用文献

1. 拙稿, 「簡易マネジメント・ゲームの開発と教育事例」, 姫路短期大学研究報告, 第29号, pp.221-228, 1984
2. 拙著, 「経営システムと情報システム」付章1, 中央経済社, 2000
3. 拙稿, 「システム・ダイナミックスによるタオル業界の動態分析」, 産業能率, 1978年5月号
4. 拙稿, 「web 利用自主学习システムの一構成モデル」, 大阪産業大学経営論集, 第2巻, 第2号, pp.15-49, 2001