

経営システム科学I'97 #7

久野 靖

1998.1.31

1 本日の内容

白井先生、藤森先生、および皆様の要望に応じて、ゲーム生成システムにいくつか機能を追加しました。「とりあえず」追加したので、あまり仕様がこなれていないのですが、大改造するとトラブルの元なので今期はこれくらいで行こうと思います。以下に追加機能について説明します。なお、これに伴いコマンドも「gg3」にしました。説明していない部分についてはこれまでと変わりありません。

ついでに、ゲーム記述について間違いやすい点と間違いを見つけるヒントなどについても説明しておきます。

2 入力変数の0期以前の値について

ivar 変数の0期以前の値を直接初期設定したいという要望が多かったので、そのための命令を追加しました。

```
ipre 変数名 [初期値 ...]
```

ipre 命令は ivar 変数に初期値を設定する命令である。指定する変数は入力変数でなければならない。指定した値は順にラウンド0、ラウンド-1、ラウンド-2、...におけるその変数の値になる(1期以降はプレイヤーが入力するので値は設定できない)。

3 メッセージ機能について

ゲームの進行につれて、何らかの「イベント」(例: 生産能力が増えた/減った、事故が発生した等)が起きたとき、これをプレイヤーに通知したいという要望がありましたので、そのための機能を追加しました。

```
tmsg("チーム宛メッセージ");  
gmsg("全体宛メッセージ");
```

これらの機能はいずれも、tlet 命令の中でのみ使えます。ただし制約として、1行につき1回しか書けません。tlet 命令は継続行がいくつでも書けますから、複数使いたい場合は適宜行を分けてください。

tmsg(...) は、当該チーム宛のメッセージを生成します(他のチームにはそのメッセージは見られません)。これに対し、gmsg(...) は全チーム宛のメッセージを生成します。いずれも、1つの期の処理で同じメッセージが複数回出る場合には(見やすくするため)重複を削除し1回だけ出力します。

メッセージの中に「\${変数指定}」という形のもを書くと、その部分是对应する変数の値に置き換えられて表示されます(末尾の例を見ください)。

メッセージを見る時には入口ページ/コントローラページの「message」ボタンを使用します。「どの期の」メッセージを見たいかは前もって「Round」ドロップダウンリストで選択する必要があります。なお、現在は N 期の計算に最して生成されたメッセージは N 期のメッセージとして表示されますので注意してください(たとえば第4期のデータを入力するときには第3期のメッセージを参照します)。

本当は第4期の入力中に見るのは「第4期のメッセージ」であって欲しいような気もするので、この部分は将来変更するかも知れません。今学期はこのままにしておきます。

4 ゲーム記述の際の注意点

よくある間違いですが、ゲーム記述で `tlet` の中に `if` を書くとき、`tlet` を複数に分けてはいけません。たとえば次のは間違いです。

```
tlet if(注文数 > 在庫数) {
  tlet   販売数 = 在庫数;
  tlet   注文残 = 注文数 - 在庫数;
  tlet }
tlet ...
```

あくまで `if` 等の制御構造は1つの `tlet` の中にまるまる含まれていなければなりません。ですから次のようにしてください。

```
tlet if(注文数 > 在庫数) {
    販売数 = 在庫数;
    注文残 = 注文数 - 在庫数;
  }
tlet ...
```

ゲーム記述ファイルでは、「1文字目が空白で始まる行はすべて前の行の続き(継続行)として扱われる」ことになっているのに注意。いっそのこと、「連続する `tlet` は使わず、できるだけ継続行にしてしまう」というのも可能です。ただし `prop` などの命令がはさまるとそこではどのみち `tlet` を分けなければなりません。

5 間違いさがしのヒント

ゲーム生成システムはできるだけ自分でエラーを見つけてユーザに警告するようにしていますが、`tlet` の中だけの変数名やメッセージ機能を変換するだけでそのままCコンパイラに渡します。従って、ここにエラーがあるとCコンパイラのエラーメッセージが出ます。よくあるエラーを挙げておきましょう。

- 関数名などのスペルミス
- かつこの対応が合っていない
- {と}の対応が合っていない
- 文末の「;」を忘れている
- `tlet` の継続行を使うべきなのに2つ以上の `tlet` の連続になっている(上述)。

Cコンパイラのエラーメッセージには、誤りのあったファイル名と誤りのあった行の番号が書かれています。その先頭のことをメモしてください。

該当箇所のソースをそのまま見ても行番号がないし変数が変換されているため見にくいと思います。次のコマンドを使ってください。

```
ggmore ソースファイル | cat -n | less
```

しかし基本的には、エラーがあったら元のソースの最後に修正した付近をよく見るのが一番速いかも知れません。

また、モデルは動くが変数の値が思い通りでない場合は、コントローラがすべての変数を見えるようにしておけば、変数の値を検査することができます(これまでに示した例題はすべてそうしてあります)。中間結果が怪しい場合には、Cのprintfを使ってその怪しい値を表示させても構いません。その場合は、表示はコントローラ画面で「モデル実行」させた時にブラウザ画面に表示されます。

6 例題: 投資による製造能力/コストの変化

ipre とメッセージ機能を含んだ簡単な例題として、投資を行うと製造能力がアップし、また製造コストも減るというモデルを作ってみました。

```
# Case: Develop1
# ・商品の値付けと製造数指示と研究開発投資を入力。
# ・価格が安いほどよく売れる。
# ・商品は在庫範囲内で売れる
# ・ある期に製造を指示すると翌期に製造量だけ在庫が増える
# ・製造能力/製造コストは研究開発投資に応じて変動
#
# ゲームの規模
#
def game-name Invest on R&D
def max-team 3
def max-round 10
password ctr t1 t2 t3
#
ooption fmt %1.0lf
#
# シリーズ定数
#
scon 商品需要 497 1195 2447 4037 5406 6626 8177 9451 9945 10713
#
# 広域定数
#
gcon 能力指数上限 1200
gcon 製造単価下限 50
gcon 製造単価上限 200
gcon 最低有効価格 50
#
```

```

# 入力変数と入力ページ
#
ipage decision パラメタの入力
  <H1>価格入力</H1>
  <P>販売価格、製造指示、R&D 投資金額を入力してください。</P>
ivar 販売価格 range 0 1000 120
ivar 製造指示 range 0 10000 100
ivar 投資金額 range 0 100000 0
ipre 投資金額 5000
#
# チーム毎モデル変数と初期値
#
tvar 能力指数 500
tvar 能力段階 5
tvar 製造数
tvar 製品在庫 200
tvar 販売可能数
tvar 受注数
tvar 販売数
tvar 売上金額
tvar 製造単価
tvar 製造能力
tvar 製造コスト
tvar 差引損益
tvar 現金貯金 150000
#
# 計算モデル
#
tlet 能力指数 = 0.7 * 能力指数@1 + 0.3 * 0.1 * 投資金額@1;
tlet if(能力指数 > 能力指数上限) 能力指数 = 能力指数上限;
tlet 能力段階 = rint(0.01 * 能力指数);
tlet if(能力段階 > 能力段階@1) tmsg("能力が${能力段階}にアップしました。");
tlet if(能力段階 < 能力段階@1) tmsg("能力が${能力段階}にダウンしました。");
tlet if(能力段階 >= 8 && 能力段階 > 能力段階@1) {
    gmsg("チーム${チーム}の能力レベルが${能力段階}に到達しました。"); }
tlet 製造単価 = rint(製造単価上限 - 0.3*能力指数);
tlet if(製造単価 < 製造単価下限) 製造単価 = 製造単価下限;
tlet 製造能力 = rint(能力指数);
tlet 製造数 = min2(製造指示, 製造能力);
tlet 製造コスト = 製造数 * 製造単価;
pinv 受注数 = 商品需要 by 販売価格-最低有効価格;
tlet 受注数 = rint(受注数);
tlet 販売可能数 = 製品在庫@1 + 製造数@1;
tlet 販売数 = min2(受注数, 販売可能数);
tlet 製品在庫 = 販売可能数 - 販売数;
tlet 売上金額 = 販売数 * 販売価格;

```

```

tlet 差引損益 = 売上金額 - 製造コスト - 投資金額;
tlet 現金貯金 = 現金貯金@1 + 差引損益;
#
# 出力指定
#
opage sales 販売状況 public
  <H1>販売状況</H1>
  <P>第${ラウンド}期: 需要: ${商品需要}</P>
beginnable
out teams
out teams-vars 販売価格::%5.3lf 販売数 売上金額 差引損益
endtable
#
opage stocks 製造/在庫 teamspec
  <H1>製造・在庫状況</H1>
  <P>第${ラウンド}期、チーム: ${チーム}、総需要: ${商品需要}</P>
beginnable
out rounds
out rounds-vars 製造指示 製造数 製品在庫 販売可能数 受注数 販売数
endtable
#
opage balance 収支の状況 teamspec
  <H1>収支の状況</H1>
  <P>第${ラウンド}期、チーム: ${チーム}、総需要: ${商品需要}</P>
beginnable
out values '項目' '収入' '支出'
out values '前期繰越 現金貯金@1 -'
out values '収入金額 売上金額 -'
out values '支出金額 製造コスト+投資金額'
out values '差引損益 - 差引損益'
out values '今期繰越 - 現金貯金'
endtable
#
opage allvteam 全変数チーム横断 control
  <H1>第${ラウンド}ラウンド: 全変数チーム横断</H1>
beginnable
out teams
out teams-allvars
endtable
#
opage allvround 全変数ラウンド横断 control
  <H1>チーム${チーム}: 全変数ラウンド横断</H1>
beginnable
out rounds
out rounds-allvars
endtable

```

```
#  
# end
```