

GraphMaker ver.0

Mathematica v.8からグラフパッケージに改訂があり、組織図や遷移図などの一般的なマップを分析する手段が提供された。一方、マップそのものを新規に構成することは必ずしも容易ではない。そこでマップを構成するために、グラフパッケージに頂点の名称や座標や辺を与えるツールを開発した。

ver.7, ver.8, ver.9で動作するが、ver.8で開発したのでver.9の動作に問題のある可能性は否定できない。

使用法

用意されているパレットは左からobject→"□", arrow→"⇔", textbox→"☒", flush→"☐"である。各パレットのモードにおいて、object：ボードをクリックするとボックスが作成される。ボックスはドラッグできる。Commandキーを押しながらクリックすると消去される

arrow：ボックスをクリック・ドラッグしてボックス間に矢印をひく。Commandキーを押しながらドラッグすると消去される

textbox：ボックスをクリックすると入力ウィンドウが表示されるので、ボックスのラベルを入力する

flush：ボックスの全てを消去する

```
In[1]:= ClearAll["Global`*"];
worksize = 400; targetsize = 40;
slevel = targetsize / worksize;
refsize = targetsize / (worksize + 20.);
srec = {refsize, refsize};

xclear := (q = {}; r = {}; s = {}; arrowVec = {{}, {}, {}}; delArrowVec = {{}, {}, {}});
grec[grecp_, zn_] := {EdgeForm[Black],
  White, Rectangle[grecp - srec / 2, grecp + srec / 2], If[(tt = s[[zn]]) == Null, tt = ""];
  Text[Style[tt, Black, FontFamily -> "Osaka"], grecp]};
garrow[xr_, xq_] := (
  n = Length[r];
  Table[makearrow[r[[i]]], {i, n}]
);
makearrow[nr_] := Arrow[Flatten[{Take[q, nr[[1]]], Take[q, nr[[2]]], 1}, srec / 2];

xappendTo[cp_] := Switch[
  xd = Map[Norm, Map[Subtract[#, cp] &, q]];
  xn = Flatten@Position[Map[Norm, xd], x_ /; x < slevel];
  mode,
  object,
  If[xn == {}, AppendTo[q, cp]; AppendTo[s, Null], If[CurrentValue["CommandKey"],
    q = Delete[q, xn]; r = xrdelete[r, xn]; s = Delete[s, xn]],
  textbox,
  If[xn != {},
    vname = s[[xn[[1]]]];
    CreateDialog[{contents, InputField[Dynamic@vname, String],
      DefaultButton[DialogReturn[s[[xn[[1]]]] = vname]}]];
  ];
];
xrdelete[rr_, rxn_] := (
  rrr = Delete[rr, Take[Position[rr, rxn], All, 1]];
  Replace[rrr, x_? (# > rxn[[1]] &) -> x - 1, {3}]
);
xdrag[cp_] := Switch[
  yd = Map[Norm, Map[Subtract[#, cp] &, q]];
  yn = Flatten@Position[Map[Norm, yd], x_ /; x < slevel];
  mode,
```

```

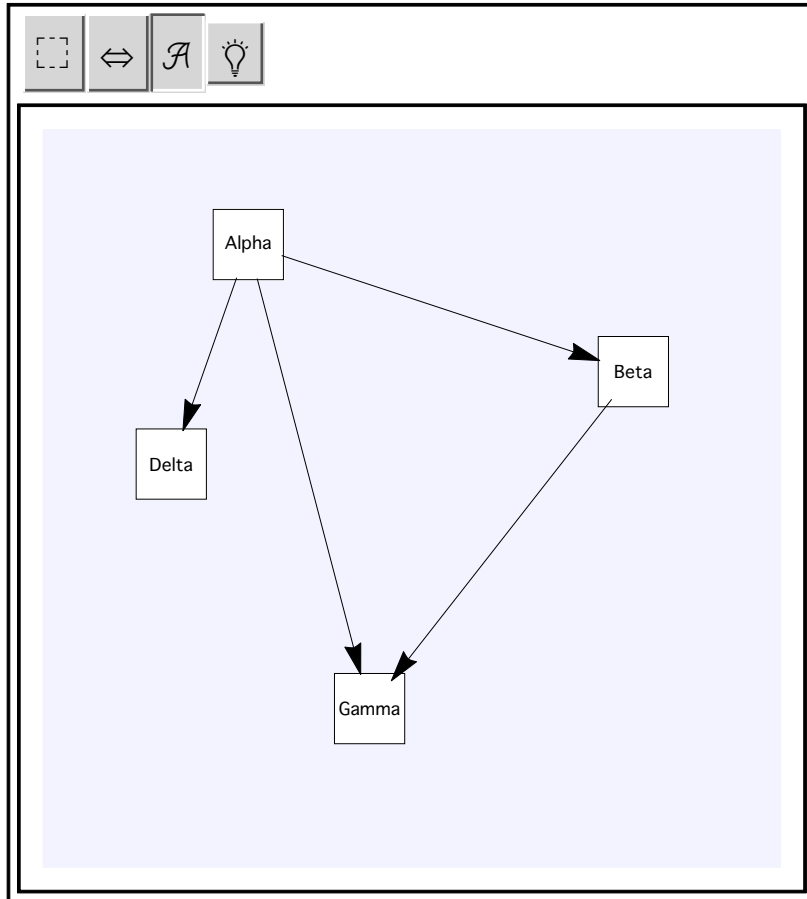
object,
q[[yn]] = cp,
arrow,
If[CurrentValue["CommandKey"],
  (delArrowVec[[2]] = cp; If[(yn != {}) && (yn ≠ deArres), deArre = yn];
  If[delArrowVec[[1]] == {}, delArrowVec[[1]] = q[[yn[[1]]]]; deArres = yn);),
  (arrowVec[[2]] = cp; If[(yn != {}) && (yn != arres), arre = yn];
  If[arrowVec[[1]] == {}, arrowVec[[1]] = q[[yn[[1]]]]; arres = yn);)
];
];
xarrow := (
  arrowVec = {{}, {}, {}};
  If[arres ≠ arre, r = AppendTo[r, {arres, arre}]];
  arres = arre = {};
);
xdearrow := (
  delArrowVec = {{}, {}, {}};
  If[deArres ≠ deArre, r = DeleteCases[r, {deArres, deArre}]];
  deArres = deArre = {};
);
Unprotect[Arrow];
Arrow[{{}, {}, {}]} = {};
Protect[Arrow];

xclear;
board = {Blue, Opacity[0.05], Rectangle[{0, 0}, {1, 1}]}];
menu = Row[{SetterBar[Dynamic[mode], {object → "☐", arrow → "↔", textbox → "A"},
  Appearance -> "Palette", BaseStyle → {Large}, BaselinePosition → Center],
  Button["☐", xclear, Appearance -> "Palette", ImageSize → {30, 33},
  BaseStyle → {Large}, BaselinePosition → Center]}];
targetpain = Framed@Deploy@Graphics[Dynamic@{g}, ImageSize → {worksize, worksize}];

g = EventHandler[{board,
  EventHandler[Dynamic@{{Blue, Arrow[arrowVec]}, {Red, Arrow[delArrowVec]},
    {Table[grec[q[[i]], i], {i, Length@q}}, {garrow[r, q]}},
    "MouseDown" => (xDrag[MousePosition["Graphics"]]),
    "MouseUp" => (xarrow; xdearrow)]
}, "MouseClicked" => (xappendTo[MousePosition["Graphics"]])];

Framed@Column[{menu, targetpain}]

```



Out[23]=

このツールから、頂点の情報 q 、辺の情報 r 、頂点のラベルが得られる

$\{q, r, s\}$

```

{{{0.228488, 0.81467}, {0.390351, 0.318638},
 {0.625313, 0.678914}, {0.808062, 0.360409}},
 {{1}, {3}}, {{3}, {2}}, {{3}, {4}}, {Alpha, Gamma, Beta, Delta}}

```

上記のデータからウィンドウにマップを再現することができる。

```

{q, r, s} = {{{0.22848788638262324`, 0.8146701388888888`},
 {0.3903508771929825`, 0.3186383928571428`}, {0.62531328320802`,
 0.6789140820802004`}, {0.808061821219716`, 0.36040948725981614`}},
 {{1}, {3}}, {{3}, {2}}, {{3}, {4}}, {"Alpha", "Gamma", "Beta", "Delta"}}

```

```

{{{0.228488, 0.81467}, {0.390351, 0.318638},
 {0.625313, 0.678914}, {0.808062, 0.360409}},
 {{1}, {3}}, {{3}, {2}}, {{3}, {4}}, {Alpha, Gamma, Beta, Delta}}

```

$\{q,r,s\}$ からGraph関数を使ってマップを描くことができる (Ver.8 以降)。

```
In[25]:= Graph[Map[Rule[#[[1]], #[[2]]] &, Flatten[r, {1, 3}]],  
  VertexShape -> Graphics[{White, EdgeForm[Black], Rectangle[]}], VertexSize -> 0.2,  
  VertexCoordinates -> q, VertexLabels -> (n = 1; Map[Rule[n++, #] &, s]), ImagePadding -> 60]
```

Out[25]=

