# GRAPHLER with OOP

kobayashikorio@gmail.com
2016/10/31

**usage**

Board area,

Click makes text-box

with Option-key Click shows menu for data save and load

Text-box,

Drag moves text-box around

with Shift-key Click move into the title input mode, and Return-key move out to the graphics mode

with Command-key Click deletes text-box

with Option-key Drag the mouse into another text-box make connection with an arrow

Arrow,

with Command-key Click deletes arrow

```
(* board *)
Module[{boardGraphics,
  boardRatio=1.4,
  boardImageScale=400,
  boardDisable=False,
  myStyle={Medium,FontFamily->"Helvetica"}},
 (* graphics-box-class *)
 box[nam_]:=Module[{
    textInputMode=False,
    pos,
    boxSize,
    title=Null,
    arrowLine={},(* arrow leader *)
    arrowPointList={},
    targetArrowList={}},(* arrows graphics *)
   (* graphics-arrow-class *)
   arrowImage[nam[pointName_]]^:=Module[{
      delta,
      bsize,
      tpos,
      arrowDrawback=0},
     EventHandler[(
       delta=Abs[pos-(tpos=(boxPosGet[pointName]))];
       bsize=0.5*boxSizeGet[pointName]/
(boardRatio*boardImageScale);

arrowDrawback=Norm[delta]*If[Apply[ArcTan,delta]>Apply[Ar
cTan,bsize],
          Abs[bsize[[2]]/delta[[2]]],
          Abs[bsize[[1]]/delta[[1]]]];
       ;
```

```
{Arrowheads[Medium],Arrow[{pos,boxPosGet[pointName]},
{0,arrowDrawback}]}),

{"MouseClicked":>If[CurrentValue["CommandKey"],boxArrowPo
intDelete[nam[pointName]]]
       },PassEventsUp->False]
     ];(* end of arrow class *)

boxArrowPointList[nam[pointName_]]^:=AppendTo[arrowPointL
ist,pointName];
   boxArrowPointDelete[nam[pointName_]]^:=(

arrowPointList=Delete[arrowPointList,Position[arrowPointL
ist,pointName][[1]]]
     );
   boxPosSet[nam[x_]]^:=(pos=x);
   boxPosGet[nam]^:=pos;
   boxTitleSet[nam[x_]]^:=title=x;
   boxTitleGet[nam]^:=title;
   boxAPListSet[nam[x_]]^:=arrowPointList=x;
   boxAPListGet[nam]^:=arrowPointList;
   boxSizeSet[nam[x_]]^:=boxSize=x;
   boxSizeGet[nam]^:=boxSize;
   boxBoth[nam]^:=
    Which[
     textInputMode==True,
     (* graphics of text-input-mode *)
     {arrowLine,targetArrowList,

Text[EventHandler[InputField[Dynamic[title],String,FieldS
ize->Small
         ,BaseStyle->myStyle],
        {"MouseEntered":>None,

"ReturnKeyDown":>(boardDisable=False;textInputMode=False;
boxSizeSet[nam[ImageDimensions@Rasterize@Text@Framed[titl
e,FrameMargins->5,BaseStyle->myStyle]]])
         }],pos]},

     textInputMode==False,(* graphics-mode *)
     {Dynamic[{arrowLine,targetArrowList}],
      EventHandler[Text[Framed[title,FrameMargins-
>5,BaseStyle->myStyle],Dynamic[pos],Background-
>LightGray],
       {"MouseDragged":>Switch[CurrentValue["OptionKey"],
          True,
(arrowLine=Line[{boxPosGet[nam],MousePosition["Graphics"]
}]),
          False,(pos=MousePosition["Graphics"])
          ],(* end of if-current value *)
```

```
"MouseClicked":>Which[probe[$x[{nam,boardDisable}]];
          CurrentValue["CommandKey"],removeInst[nam],

CurrentValue["ShiftKey"],textInputMode=True;boardDisable=
True],
        "MouseMoved":>(pointName=nam),(* get the name of
another instance *)
        "MouseUp":>Which[
          CurrentValue["OptionKey"],
          (arrowLine={};
           boxArrowPointList[nam[pointName]];(* append
point to the list *)
           (* overide arrow instances *)

targetArrowList:=Map[arrowImage[nam[#]]&,arrowPointList])
,
          True,arrowLine={}(* safety eraser *)
          ],(* end of if-shift-key *)
        PassEventsUp->False}](* end of event-handler *)
      }(* end of graphics *)
     ];(* end of text-input-mode *)

buildUpArrowList[nam]^:=targetArrowList:=Map[arrowImage[n
am[#]]&,arrowPointList]
   ]; (* end of box-class *)

 (* function to make box-instance with unique name *)
 makeInst[newPos_]:=(
   newName=Unique[];
   box[newName];(* make new box instance *)
   boxPosSet[newName[newPos]];(* pop-up the instance on
the board *)

boxSizeSet[newName[ImageDimensions@Rasterize@Text@Framed[
Null,BaseStyle->myStyle]]];
   AppendTo[namList,newName];
   targetList:=Map[boxBoth,namList](* makeup graphics *)
   );
 (* function to remove box-instance *)
 removeInst[boxName_]:=
  (namList=Delete[namList,Position[namList,boxName]
[[1]]];
   targetList=Map[boxBoth,namList]);

 boardGraphics=Graphics[Dynamic[targetList],PlotRange-
>{{0,boardRatio},{0,1.}},ImageSize->{boardRatio,1}
*boardImageScale];

 delegate:=(
   Which[boardDisable==True,boardGraphics,
    boardDisable==False,EventHandler[boardGraphics,
```

```
     "MouseClicked":>Which[

CurrentValue["OptionKey"]==True,CreateDialog[Pane[Column[
{Button["Save",Put[Map[Map[#,namList]&,
{List,boxAPListGet,boxTitleGet,boxPosGet,boxSizeGet}],"gr
aph"];DialogReturn[]],Button["Load",buildUp[Get["graph"]]
;DialogReturn[]]}
          ],ImageMargins->5],Modal->True,WindowTitle-
>"objects"];boardDisable==False,
       True,makeInst[MousePosition["Graphics"]]
       ]](* end of EventHandler *)
    ]);(* end of delegate graphics *)

 (* build up graphics from file *)

buildUp[{namListX_,boxAPList_,boxTitle_,boxPos_,boxSize_}
]:=(
   Apply[Clear,namList=Flatten@namListX];
   targetList={};
   Map[box,namList];
   MapThread[boxPosSet[#1[#2]]&,{namList,boxPos}];
   MapThread[boxSizeSet[#1[#2]]&,{namList,boxSize}];
   MapThread[boxTitleSet[#1[#2]]&,{namList,boxTitle}];
   MapThread[boxAPListSet[#1[#2]]&,{namList,boxAPList}];

Map[buildUpArrowList,namList];Flatten@ReleaseHold[MapThre
ad[f[#1,#2]&,{namList,boxAPList}]/.f[nam_,p_]-
>Hold@Map[boxArrowPointList[nam,#]&,p]];
   targetList=Map[boxBoth,namList];
   boardDisable=False
   );
 ](* end of Module *)
```

Run program,

```
targetList={};namList={};
SetDirectory[NotebookDirectory[]];
Framed@Deploy@Dynamic@delegate
```

Saved graph data file can be convert to *Mathematica* graph.

```
SetDirectory[NotebookDirectory[]];
g=Get["graph"];
n=Length@g[[1]];
g1=g[[2]]/.Map[Apply[Rule,#]&,Transpose@{Flatten@g[[1]],R
ange[n]}];
g2=Flatten[Map[Thread,Transpose@{g1,Range[n]}],1]/.
{x_,y_}->Rule[y,x];
Graph[g2,VertexLabels-
>Map[Apply[Rule,#]&,Transpose@{Range[n],g[[3]]}],ImagePad
ding->20]
```