GL Board Game: 3-D Game Support

Version 9.0.0.11

December 22, 2025

```
(require games/gl-board-game)       package: games
```

```
gl-board% : class?
  superclass: canvas%
```

```
 (new gl-board%
     [min-x min-x]
     [max-x max-x]
     [min-y min-y]
     [max-y max-y]
     [lift lift]
    [[move move]
     [who who]]
     ...superclass-args...) → (is-a?/c gl-board%)
  min-x : real?
  max-x : real?
  min-y : real?
  max-y : real?
  lift : real?
  move : (any/c gl-vector? . -> . any) = void
  who : string? = "this game"
```

The *min-x*, *max-x*, *min-y*, and *max-y* arguments specify the dimensions of the board plane to be visible in the window by default.

The *lift* argument specifies how many units a piece moves vertically when the user clicks on it.

The *move* function is called when a piece is moved to a space (possibly it's current space), when a space is clicked on, and when a space is dragged to another space. The *move* function is given the information of the piece or space selected and the coordinates to which it is moved.

The *who* argument is used for reporting an error to the user when GL is unavailable at run time.

```
(send a-gl-board add-space draw info) → void?
  draw : (-> any)
  info : any/c
```

Adds a space to the board. The *draw* thunk should draw the space (using GL commands) when called. The *info* value is given to the *move* function (supplied to the constructor) when the space is selected.

```
(send a-gl-board add-piece x y z draw info) → void?
  x : real?
  y : real?
  z : real?
  draw : ([shadow? boolean?] . -> . any)
  info : any/c
```

Adds a piece to the board. The *draw* thunk should draw the piece (using GL commands) when called. The *info* argument is given to the *move* function (supplied to the constructor) when the piece is moved. The piece is translated by *x*, *y*, and *z* before drawing.

```
(send a-gl-board remove-piece info) → void?
  info : any/c
```

Removes all pieces previously added with representative *info*.

```
(send a-gl-board add-heads-up w h draw info) → void?
  w : real?
  h : real?
  draw : (-> any)
  info : any/c
```

Add a “heads-up” display element whose size is *w* by *h* units with the given *draw* thunk and *info* reprsentative.

```
(send a-gl-board remove-heads-up info) → void?
  info : any/c
```

Removes all “heads-up” displays elements previous added with representative *info*.

```
(send a-gl-board set-space-draw info draw) → void?
  info : any/c
  draw : (-> any)
```

Sets the drawing function of all spaces added with representative `info`.

```
(send a-gl-board set-piece-draw info draw) → void?
  info : any/c
  draw : ([shadow? boolean?] . -> . any)
```

Sets the drawing function of all pieces added with representative `info`.

```
(send a-gl-board enable-piece info
                                can-move?) → void?
  info : any/c
  can-move? : any/c
```

Enables or disables moving of all pieces added with representative `info`.

```
(send a-gl-board enabled? info) → boolean?
  info : any/c
```

reports whether the first piece with representative `info` is enabled.

```
(send a-gl-board get-pieces) → list?
(send a-gl-board get-spaces) → list?
(send a-gl-board get-heads-up) → list?
```

Returns values for various kinds of content currently on the board. The result corresponds to `info` values given to `add-piece`, etc.