Pufferfish Games: Grid System

Sean Miles and Yuto Akutsu

Missing Functionality: Grid System

- Grid Based Movement/Placement of Objects
- Developers
  - Can create a grid of a certain size
  - Can lock objects to a specified grid
- Players
  - Move objects on the grid
  - Cannot move grid-locked objects in between or out of grid cells

Grid System API

Grid Class Overview

- `initialize`
- `update`
- `draw`
- `getObjCell`

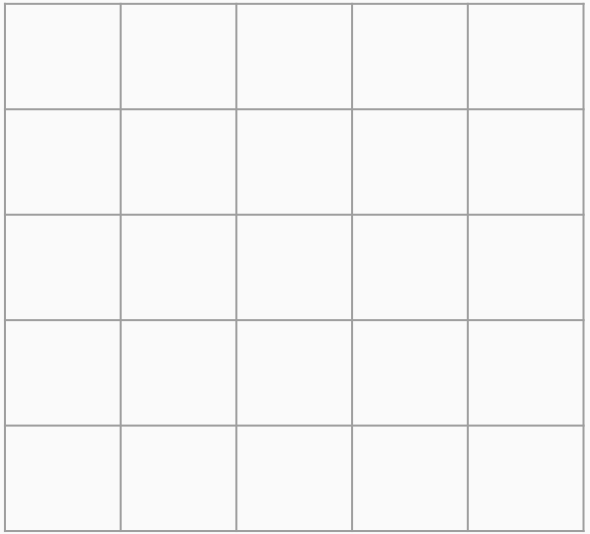
Grid Class

- `initialize(row, col)`
  - Initialize the grid:
    - number of columns (width) and number of rows (height)
    - create array of grid objects
    - cell size of map is canvas size / row or col
- `update()`
  - Updates all changes in each state of GridObjects in the system

Grid Class

- `draw(camera.getVPMatrix())`
  - draw entire grid
- `getObjCell(cellX, cellY)`
  - returns GridObject in a specified cell if occupied
  - if unoccupied, return NULL

A cell is occupied if any part of a GridObject exists in it



GridObject Class Overview

- `initialize`
- `getSize`
- `setSize`
- `getPos`
- `setPos`
- `lockObject`
- `unlockObject`

GridObject Class

- `initialize(renderable, grid, positionX, positionY cellSizeX, cellSizeY, locked)`
  - initialize an object with:
    - a Renderable
    - parent grid this objects exists on
    - position on the grid
    - size in width and height (in amount of cells)
    - If the object starts gridlocked or not

GridObject Class

- `getSize`
  - get an object's width and height in amount of cells
- `setSize(cellSizeX, cellSizeY)`
  - set/resize an object's width and height in amount of cells
  - check other slots are unoccupied before resizing larger (resize to the right and down)
  - must be an integer, as an object cannot be taking up half of a grid cell

GridObject Class

- `getPos`
  - get an object's position in the current grid
- `setPos(cellX, cellY)`
  - set an object's position in the current grid if unoccupied
  - uses Grid function `getObjCell` to check if unoccupied

GridObject Class

- `lockObject`
  - Locking a GridObject adds it to an array in the grid that limits movement
- `unlockObject`
  - Unlock a GridObject to remove it from the array, no longer limiting movement

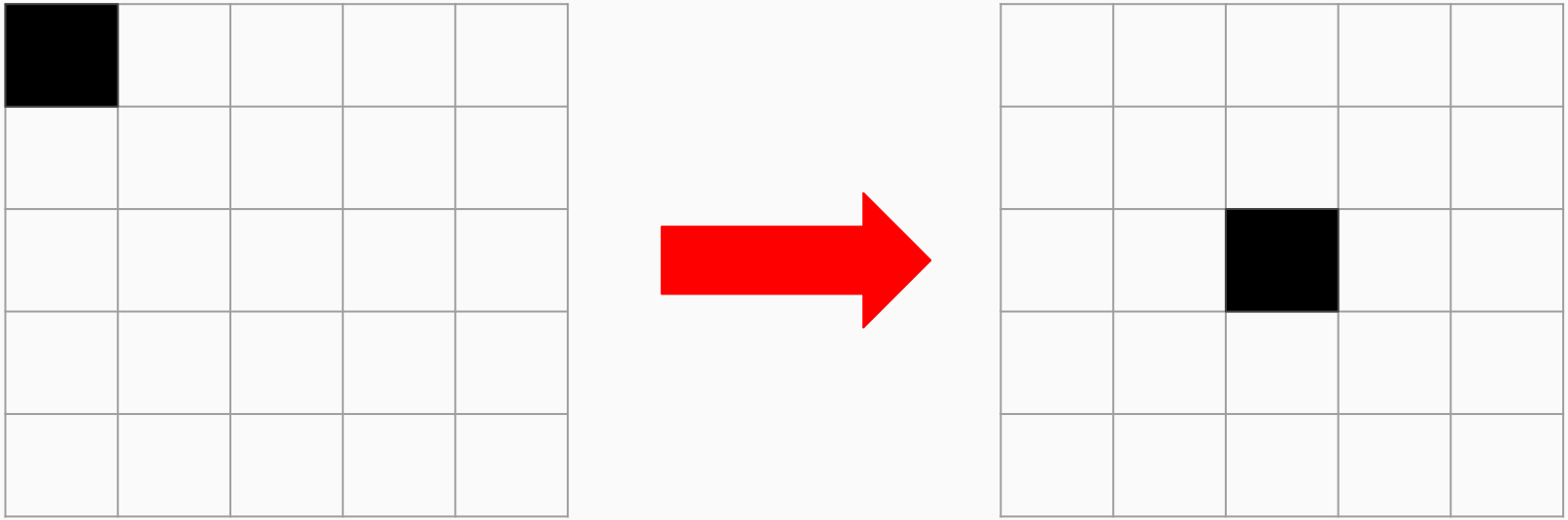
Example API Calls

```
Grid map = new Grid(5, 5)
GridObject obj = new GridObject(square, map, 1, 1, 1, 1, false)
```

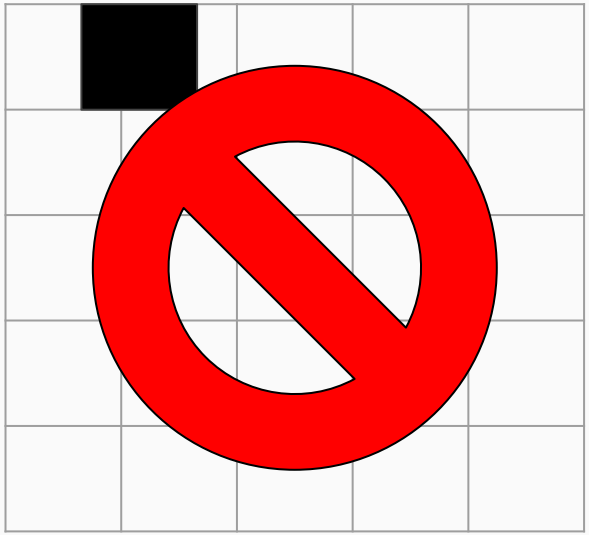
- map has 5 rows and 5 columns
- cell size of map is canvas size / row or col

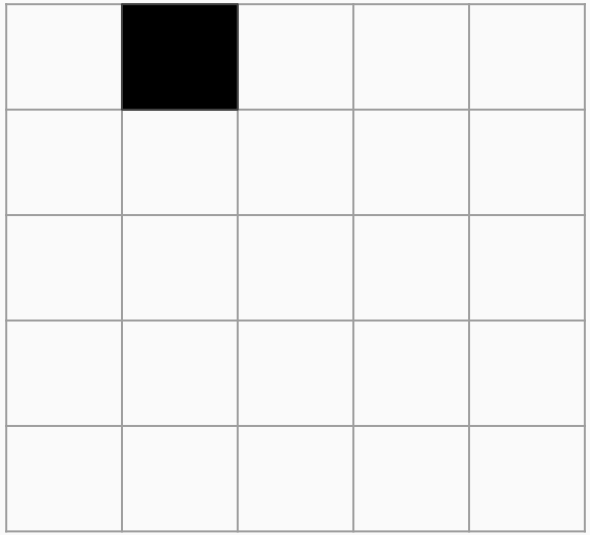
- obj has a reference to the renderable (square) it is related to
- obj’s parent grid is map
- obj is located at (1, 1)
- obj has a cell size of 1x1
- obj is not gridlocked

```
Grid map = new Grid(5, 5)
GridObject obj = new GridObject(square, map, 1, 1, 1, 1, false)

obj.getPos()                // returns 1, 1
obj.setPos(3, 3)            // returns true, slot unoccupied
obj.getPos(obj)             // returns 3, 3
map.getObjCell(3, 3)        // returns reference of obj
map.getObjCell(1, 1)        // returns NULL
```

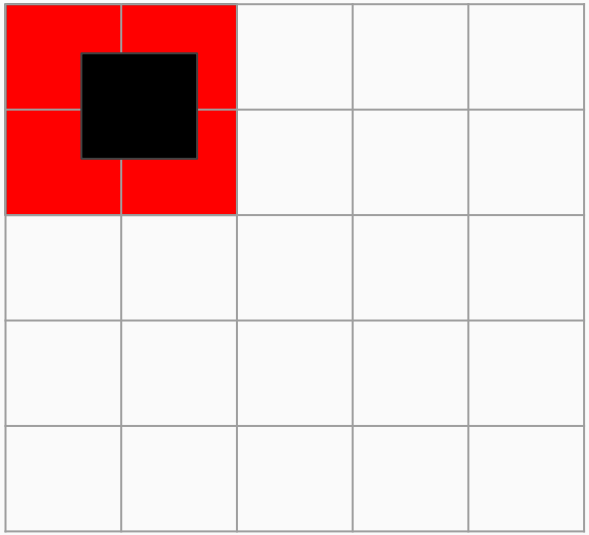


```
Grid map = new Grid(5, 5)
GridObject obj = new GridObject(square, map, 1, 1, 1, 1, false)

obj.lockObject()
// any movement by obj will be snapped to the closest cell
```

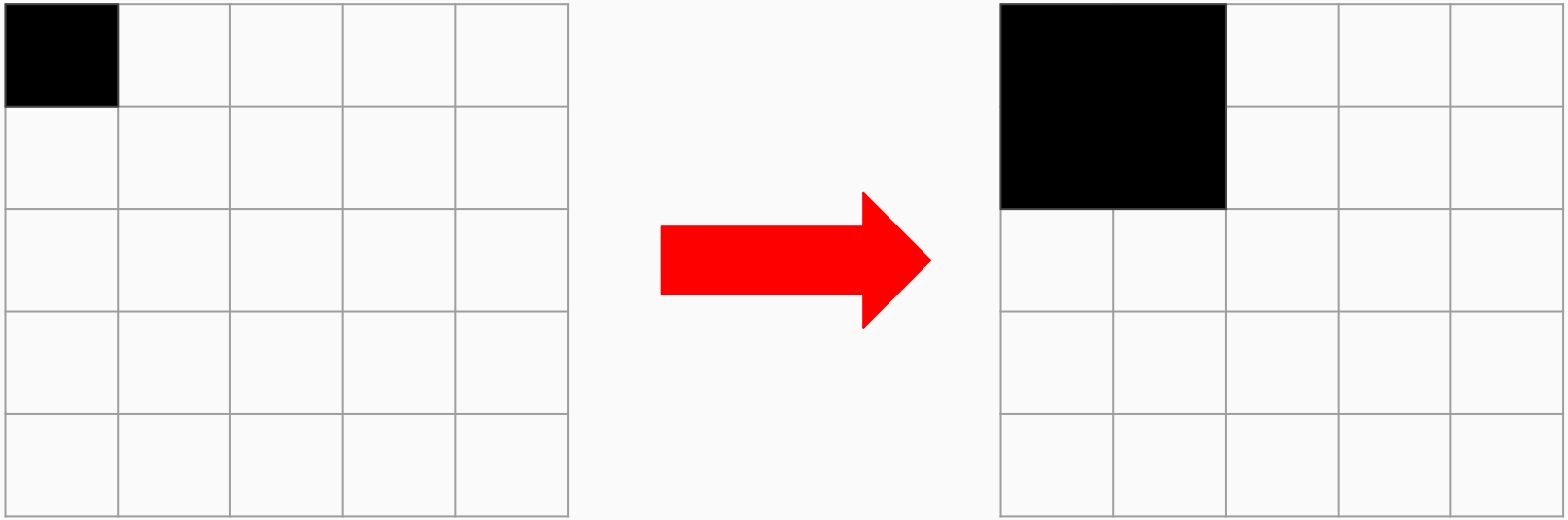




```
Grid map = new Grid(5, 5)
GridObject obj = new GridObject(square, map, 1, 1, 1, 1, false)

obj.unlockObject()
// any movement by obj is smooth, can move between cells
// more cells occupied if any part of obj is in a cell
```



```
Grid map = new Grid(5, 5)
GridObject obj = new GridObject(square, map, 1, 1, 1, 1, false)

obj.getSize()                    // returns 1, 1
obj.setSize(2, 2)                    // returns true, slots unoccupied
obj.getSize()                    // returns 2, 2
```



Questions?