

# 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

# Proposed Functionalities

- Create grids with a specified width and height
- Create grid objects that fill a certain width and height in the grid
- Grid objects can be locked onto the grid, limiting movement

# Completed Functionalities

- Grid class with width and height and cells with width and height
- GridObject class with a specified GameObject, position, and width and height using parent/child relationship
- GridObjects can be locked to a Grid, limiting movement to cells only
- + GridObjects can be resized or have their GameObject reassigned
- + GridObjects have a specific bounding box of the full cell sizes, separate from the GameObject's bounding box

# Examples

# Demo 1: Large/Resizable Hero and Patrol

- Gridlocked movement
- GridObject of 1x1
- GridObject of 2x2
- Movement of GridObjects larger than 1x1 with occupied cell checks
- Resizing a GridObject to be smaller/larger with occupied cell checks
- Drawing of Grid and occupied cells: green for origin/parent, red for child

## Demo 2: Non-gridlocked Hero and Patrol

- Non-gridlocked movement
- 2 GridObjects of 1x1
- Grid position still stored, checks for occupied cells for the actual GridObject cells but does not limit movement of GameObject
- Drawing of Grid and occupied cells: green for origin/parent, red for child



```
MyGame.prototype.loadScene = function ()  
{  
    gEngine.Textures.loadTexture(this.kMinionSprite);  
};
```

```
MyGame.prototype.draw = function ()  
{  
    // Step A: clear the canvas  
    gEngine.Core.clearCanvas([0.9, 0.9, 0.9, 1.0]); // clear to light gray  
  
    this.mCamera.setupViewProjection();  
    this.mMsg.draw(this.mCamera);    // only draw status in the main camera  
    this.mGrid.draw(this.mCamera);  
};
```

```

MyGame.prototype.initialize = function ()
{
    // Step A: set up the cameras
    this.mCamera = new Camera(
        vec2.fromValues(0, 0), // position of the camera
        200,                    // width of camera
        [0, 0, 640, 480]       // viewport (orgX, orgY, width, height)
    );
    this.mCamera.setBackgroundColor([0.8, 0.8, 0.8, 1]);
    // sets the background to gray

    this.mMsg = new FontRenderable("Status Message");
    this.mMsg.setColor([0, 0, 0, 1]);
    this.mMsg.getXform().setPosition(-96, -70);
    this.mMsg.setTextHeight(4);

    this.mGrid = new Grid(5, 5, 25, 25);
    this.mGrid.setDraw(true);

    this.mHero = new Hero(this.kMinionSprite, 35, 50);
    this.mHero = new GridObject(this.mHero, this.mGrid,
                                0, 0,
                                2, 2, true);
    this.mHero.getGameObject().getXform().incSizeBy(10);

    this.mPatrol = new Patrol(this.kMinionSprite, 30, 30);
    this.mPatrol = new GridObject(this.mPatrol, this.mGrid,
                                3, 3,
                                1, 1, true);

    this.mGrid.addObject(this.mHero);
    this.mGrid.addObject(this.mPatrol);
};

```

```
MyGame.prototype.update = function ()
{
    var msg = "Status: ";
    var echo = "";

    // Input

    this.mGrid.update();

    echo += "Grid Size: " + this.mGrid.getNumCols() + "x" + this.mGrid.getNumRows() + " with ";
    echo += "Cell Size: " + this.mGrid.getCellWidth() + "x" + this.mGrid.getCellHeight() + " ";
    echo += "Hero: " + this.mHero.getPos() + " ";
    echo += "Patrol: " + this.mPatrol.getPos() + " ";
    echo += "Objects: " + this.mGrid.getNumObjects() + " ";

    msg += echo;
    this.mMsg.setText(msg);
};
```

# Challenges

- Checking of occupied cells when resizing/moving with GridObjects larger than 1x1
- Parent/child relationship for GridObjects larger than 1x1 so that `getObjFromCell()` works regardless of where it is called in the GridObject
- Making `getClosestCell()` works regardless of cell size of GridObjects

# Future Changes

- If given more time on the project, we would like to:
  - Optimize `getClosestCell()`
  - Expand on the gridlock functionality - allow movement from `GameObject` even if `GridObject` is locked, but maintain Grid position
  - Make each cell be able to store more than one `GridObject`

Questions?