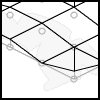
## Terrain Modification in Grid Based Games – Part 1: Isometric Landscape



Target of this tutorial is the development of a scrolling isometric tile map where the terrain can be manipulated. Inspiration for a test like this were Populous and SimCity with their respective methods of terrain manipulation. While Populous lets the player change the height of a single node (forming a pyramid), SimCity always pushes the whole tile. The challenge here is that node’s heights can only be altered in dependency of their adjacent nodes.  
This part deals with the creation of a tile map displaying the isometric map with terrain.  
  
**Part 1.1: The basic node grid**  
Some basic variable defining the grid and tile size as well as the array holding the nodes and its position pendant.

var nodeArray:Array = new Array();

var posArray:Array = new Array();

var nodeRows:int = 9;

var nodeCols:int = 9;

var tileSizeH:int = 32; //horizontal size

var tileSizeV:int = 16; //vertical size

//the tile size represents a 45° rotated square with backwards tilting

This function will create nodes in a nested loop. u stands for columns, v for the respective rows. Nodes are created as Objects, holding information without graphical representation. When a node is created the object is pushed into an array. At the same time a string is created and pushed into a second array with the same index as the object in the first array. That way a node can be found by its coordinates

function makeGrid()

{

for ( var u = 0; u <= nodeCols; u++ )

{

for ( var v = 0; v <= nodeRows; v++ )

{

var node:Object = new Object();

node.u = u;

node.v = v;

node.nodePos = u + "." + v;

nodeArray.push(node);

posArray.push(node.nodePos);

}

}

}

Depending on row and column numbers the array holds the respective number of nodes. By using u and v the way shown below for placement the coordinate system is rotated by 45°.

function placeNodes()

{

for each ( var node in nodeArray )

{

node.xPos = ( node.u - node.v ) \* tileSizeH;

node.yPos = ( node.v + node.u ) \* tileSizeV;

}

}

The next function places the nodes on a tilted plane called nodeMap.

function drawNodes()

{

var map:Sprite = new Sprite();

//the map size

var mapSizeH = ( nodeCols - 1 + nodeRows - 1 ) \* tileSizeH;

var mapSizeV = ( nodeRows - 1 + nodeCols - 1 ) \* tileSizeV;

//centering the map on the stage

map.x = stage.stageWidth \* 0.5;

map.y = stage.stageHeight \* 0.5 - mapSizeV \* 0.5;

//graphic options, nodes will be represented by small black circles

map.graphics.lineStyle(1, 0x000000);

for each ( var node in nodeArray )

{

map.graphics.drawCircle(node.xPos, node.yPos, 3);

}

addChild(map);

}

Up to this moment nothing happened. We need to call all three functions to make it work.

makeGrid();

placeNodes();

drawNodes();

Here is what it should look like:

**Part 1.2: The basic node grid with heights**  
Adding height to the nodes is rather simple. Altered code is marked.

var nodeArray:Array = new Array();

var posArray:Array = new Array();

var nodeRows:int = 9;

var nodeCols:int = 9;

var tileSizeH:int = 32; //horizontal

var tileSizeV:int = 16; //vertical

var tileSizeM:int = 8; //mountain

function makeGrid()

{

for ( var u = 0; u <= nodeCols; u++ )

{

for ( var v = 0; v <= nodeRows; v++ )

{

var node:Object = new Object();

node.u = u;

node.v = v;

node.nodePos = u + "." + v;

nodeArray.push(node);

posArray.push(node.nodePos);

}

}

}

function placeNodes()

{

for each ( var node in nodeArray )

{

node.xPos = ( node.u - node.v ) \* tileSizeH;

node.yPos = ( node.v + node.u ) \* tileSizeV;

node.zPos = Math.round(Math.random()) \* tileSizeM;

}

}

function drawNodes()

{

var nodeMap:Sprite = new Sprite();

var nodeMapSizeH = ( nodeCols - 1 + nodeRows - 1 ) \* tileSizeH;

var nodeMapSizeV = ( nodeRows - 1 + nodeCols - 1 ) \* tileSizeV;

nodeMap.x = stage.stageWidth \* 0.5;

nodeMap.y = stage.stageHeight \* 0.5 - nodeMapSizeV \* 0.5;

nodeMap.graphics.lineStyle(1, 0x000000);

for each ( var node in nodeArray )

{

nodeMap.graphics.drawCircle(node.xPos, node.yPos - node.zPos, 3);

}

addChild(nodeMap);

}

makeGrid();

placeNodes();

drawNodes();

Right now, we can see that nodes have different heights though it is hardly visible.

**Part 1.3: The basic map with drawn tiles**  
Now we will draw some edges between the nodes. The basic nodes are again drawn without height and act like a zero level now.

var nodeArray:Array = new Array();

var posArray:Array = new Array();

var nodeRows:int = 9;

var nodeCols:int = 9;

var tileSizeH:int = 32; //horizontal

var tileSizeV:int = 16; //vertical

var tileSizeM:int = 8; //mountain

function makeGrid()

{

for ( var u = 0; u <= nodeCols; u++ )

{

for ( var v = 0; v <= nodeRows; v++ )

{

var node:Object = new Object();

node.u = u;

node.v = v;

node.nodePos = u + "." + v;

nodeArray.push(node);

posArray.push(node.nodePos);

}

}

}

function placeNodes()

{

for each ( var node in nodeArray )

{

node.xPos = ( node.u - node.v ) \* tileSizeH;

node.yPos = ( node.v + node.u ) \* tileSizeV;

node.zPos = Math.round(Math.random()) \* tileSizeM;

}

}

function drawNodes()

{

var nodeMap:Sprite = new Sprite();

var nodeMapSizeH = ( nodeCols - 1 + nodeRows - 1 ) \* tileSizeH;

var nodeMapSizeV = ( nodeRows - 1 + nodeCols - 1 ) \* tileSizeV;

nodeMap.x = stage.stageWidth \* 0.5;

nodeMap.y = stage.stageHeight \* 0.5 - nodeMapSizeV \* 0.5;

nodeMap.graphics.lineStyle(1, 0xaaaaaa);

for each ( var node in nodeArray )

{

nodeMap.graphics.drawCircle(node.xPos, node.yPos, 3);

}

addChild(nodeMap);

}

makeGrid();

placeNodes();

drawNodes();

With the next function nodes are found in the nodeArray via its position string in the posArray. Remember, node object and position string have the same index in the respective array. That way more difficult map types (hexagonal, triangles) can be used.

function getNodeByCoords( u, v )

{

var posString:String = u + "." + v;

var nodePos:int = posArray.indexOf(posString);

if ( nodePos >= 0 )

{

return nodeArray[nodePos];

}

else

{

return null;

}

}

Tiles are also created as objects at this point because manipulation is not yet implemented redrawing the whole map right now does't take much time. So, tiles are created with the north, east, south and west node.

var tileArray:Array = new Array();

function makeTiles()

{

for ( var u = 0; u < nodeCols; u++ )

{

for ( var v = 0; v < nodeRows; v++ )

{

var tile:Object = new Object();

tile.n = getNodeByCoords(u, v);

tile.e = getNodeByCoords(u+1, v);

tile.s = getNodeByCoords(u+1, v+1);

tile.w = getNodeByCoords(u, v+1);

tileArray.push(tile);

}

}

}

The northern node will act as reference point here. In many cases the reference point of isometric graphics will be the lower, southern point but those are easily interchangeable.

function drawTiles()

{

var tileMap:Sprite = new Sprite();

var tileMapSizeH = ( nodeCols - 1 + nodeRows - 1 ) \* tileSizeH;

var tileMapSizeV = ( nodeRows - 1 + nodeCols - 1 ) \* tileSizeV;

tileMap.x = stage.stageWidth \* 0.5;

tileMap.y = stage.stageHeight \* 0.5 - tileMapSizeV \* 0.5;

tileMap.graphics.lineStyle(1, 0x000000);

for each ( var tile in tileArray )

{

with( tileMap.graphics )

{

moveTo(tile.n.xPos, tile.n.yPos - tile.n.zPos);

lineTo(tile.e.xPos, tile.e.yPos - tile.e.zPos);

lineTo(tile.s.xPos, tile.s.yPos - tile.s.zPos);

lineTo(tile.w.xPos, tile.w.yPos - tile.w.zPos);

lineTo(tile.n.xPos, tile.n.yPos - tile.n.zPos);

}

}

addChild(tileMap);

}

makeTiles();

drawTiles();

Tile edges are drawn from node to node:

**Part 1.4: The map wire mesh**  
By changing the tile drawing function additional edges are shown.

function drawTiles()

{

var tileMap:Sprite = new Sprite();

var tileMapSizeH = ( nodeCols - 1 + nodeRows - 1 ) \* tileSizeH;

var tileMapSizeV = ( nodeRows - 1 + nodeCols - 1 ) \* tileSizeV;

tileMap.x = stage.stageWidth \* 0.5;

tileMap.y = stage.stageHeight \* 0.5 - tileMapSizeV \* 0.5;

tileMap.graphics.lineStyle(1, 0x000000);

for each ( var tile in tileArray )

{

if ( tile.n.zPos == tile.s.zPos )

{

tile.ver = true; //vertical heights are the same

}

else

{

tile.ver = false;

}

if ( tile.e.zPos == tile.w.zPos )

{

tile.hor = true; //horizontal heights are the same

}

else

{

tile.hor = false;

}

with(tileMap.graphics)

{

if ( tile.ver && tile.hor )

{

if ( tile.n.zPos > tile.e.zPos )

{

moveTo(tile.n.xPos, tile.n.yPos - tile.n.zPos);

lineTo(tile.s.xPos, tile.s.yPos - tile.s.zPos);

}

else if ( tile.n.zPos < tile.e.zPos )

{

moveTo(tile.e.xPos, tile.e.yPos - tile.e.zPos);

lineTo(tile.w.xPos, tile.w.yPos - tile.w.zPos);

}

//if both are the same, there is no line

}

else if ( tile.ver && !tile.hor )

{

moveTo(tile.e.xPos, tile.e.yPos - tile.e.zPos);

lineTo(tile.w.xPos, tile.w.yPos - tile.w.zPos);

}

else if ( !tile.ver && tile.hor )

{

moveTo(tile.n.xPos, tile.n.yPos - tile.n.zPos);

lineTo(tile.s.xPos, tile.s.yPos - tile.s.zPos);

}

moveTo(tile.n.xPos, tile.n.yPos - tile.n.zPos);

lineTo(tile.e.xPos, tile.e.yPos - tile.e.zPos);

lineTo(tile.s.xPos, tile.s.yPos - tile.s.zPos);

lineTo(tile.w.xPos, tile.w.yPos - tile.w.zPos);

lineTo(tile.n.xPos, tile.n.yPos - tile.n.zPos);

}

}

addChild(tileMap);

}

Alright, all terrain changes are included.

**Part 1.5: The map wire mesh with refined edges**  
Though that map does not need any backface culling right now (you can easily look over the mountains, so there are no hidden edges) let's give it some sides to emphasize the 3D look.

function drawTiles()

{

var tileMap:Sprite = new Sprite();

var tileMapSizeH = ( nodeCols - 1 + nodeRows - 1 ) \* tileSizeH;

var tileMapSizeV = ( nodeRows - 1 + nodeCols - 1 ) \* tileSizeV;

tileMap.x = stage.stageWidth \* 0.5;

tileMap.y = stage.stageHeight \* 0.5 - tileMapSizeV \* 0.5;

//tileMap.graphics.lineStyle(1, 0x000000);

for each ( var tile in tileArray )

{

tile.n.zPos == tile.s.zPos ? tile.ver = true : tile.ver = false;

tile.e.zPos == tile.w.zPos ? tile.hor = true : tile.hor = false;

with(tileMap.graphics)

{

lineStyle(1, 0xaaaaaa);

if ( tile.s.u == nodeCols )

{

moveTo(tile.e.xPos, tile.e.yPos);

lineTo(tile.e.xPos, tile.e.yPos - tile.e.zPos);

moveTo(tile.s.xPos, tile.s.yPos - tile.s.zPos);

lineTo(tile.s.xPos, tile.s.yPos);

lineTo(tile.e.xPos, tile.e.yPos);

}

if ( tile.s.v == nodeRows )

{

moveTo(tile.w.xPos, tile.w.yPos);

lineTo(tile.w.xPos, tile.w.yPos - tile.w.zPos);

moveTo(tile.s.xPos, tile.s.yPos - tile.s.zPos);

lineTo(tile.s.xPos, tile.s.yPos);

lineTo(tile.w.xPos, tile.w.yPos);

}

lineStyle(1, 0x000000);

if ( tile.ver && tile.hor )

{

if ( tile.n.zPos > tile.e.zPos )

{

moveTo(tile.n.xPos, tile.n.yPos - tile.n.zPos);

lineTo(tile.s.xPos, tile.s.yPos - tile.s.zPos);

}

else if ( tile.n.zPos < tile.e.zPos )

{

moveTo(tile.e.xPos, tile.e.yPos - tile.e.zPos);

lineTo(tile.w.xPos, tile.w.yPos - tile.w.zPos);

}

//if both are the same, there is no line

}

else if ( tile.ver && !tile.hor )

{

moveTo(tile.e.xPos, tile.e.yPos - tile.e.zPos);

lineTo(tile.w.xPos, tile.w.yPos - tile.w.zPos);

}

else if ( !tile.ver && tile.hor )

{

moveTo(tile.n.xPos, tile.n.yPos - tile.n.zPos);

lineTo(tile.s.xPos, tile.s.yPos - tile.s.zPos);

}

moveTo(tile.n.xPos, tile.n.yPos - tile.n.zPos);

lineTo(tile.e.xPos, tile.e.yPos - tile.e.zPos);

lineTo(tile.s.xPos, tile.s.yPos - tile.s.zPos);

lineTo(tile.w.xPos, tile.w.yPos - tile.w.zPos);

lineTo(tile.n.xPos, tile.n.yPos - tile.n.zPos);

}

}

addChild(tileMap);

}

I also did change the horizontal and vertical check to inline conditionals because the script is shorter that way. Two different colors are used to easily differentiate sides and surface. The function now checks if the southern node of a tile is in the last node row or column and if that is the case it draws the remaining visible side of the tile (two in case of the middle one).

Reload the page for different random terrains.

Yoho!

### More articles in Terrain Modification:

* Terrain Modification in Grid Based Games - Part 1: Isometric Landscape
* [Terrain Modification in Grid Based Games - Appendix A: Dynamic Tiles](http://www.yarrcade.com/2010/09/24/terrain-modification-in-grid-based-games-appendix-a-dynamic-tiles/)
* [Terrain Modification in Grid Based Games – Part 2: Moving Nodes](http://www.yarrcade.com/2010/09/27/terrain-modification-in-grid-based-games%e2%80%93part-2-moving-nodes/)
* [Terrain Modification in Grid Based Games – Part 3: Active Landscaping](http://www.yarrcade.com/2010/10/02/terrain-modification-in-grid-based-games-%e2%80%93-part-3-active-landscaping/)
* [Terrain Modification in Grid Based Games – Appendix B: Shading](http://www.yarrcade.com/2010/10/08/terrain-modification-in-grid-based-games-%e2%80%93-appendix-b-shading/)
* [Terrain Modification in Grid Based Games – Appendix C: Shading and Bitmaps](http://www.yarrcade.com/2010/10/24/terrain-modification-in-grid-based-games-%e2%80%93-appendix-c-shading-and-bitmaps/)
* [Terrain Modification in Grid Based Games - Exercise A: Minimap and Comets](http://www.yarrcade.com/2010/11/08/terrain-modification-in-grid-based-games-exercise-a-minimap-and-comets/)

This entry was posted in [Terrain Modification](http://www.yarrcade.com/category/tutorial-series/terrain-modification-tutorial-series/), [Tutorial](http://www.yarrcade.com/category/tutorial/), [as3](http://www.yarrcade.com/category/game-development/actionscript-3/), [flash](http://www.yarrcade.com/category/game-development/flash-game-development/), [grids](http://www.yarrcade.com/category/game-development/grids/), [mochiads](http://www.yarrcade.com/category/game-development/mochiads/) and tagged [dynamic tile drawing](http://www.yarrcade.com/tag/dynamic-tile-drawing/), [populous](http://www.yarrcade.com/tag/populous/), [simcity](http://www.yarrcade.com/tag/simcity/), [terrain manipulation](http://www.yarrcade.com/tag/terrain-manipulation/), [terrain modification](http://www.yarrcade.com/tag/terrain-modification/). Bookmark the [permalink](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/).

* [Print](http://www.printfriendly.com/print?url=http%3A%2F%2Fwww.yarrcade.com%2F2010%2F09%2F19%2Fterrain-modificatio-in-grid-based-games-part-1-isometric-landscape%2F&partner=sociable)
* [Digg](http://digg.com/submit?phase=2&url=http%3A%2F%2Fwww.yarrcade.com%2F2010%2F09%2F19%2Fterrain-modificatio-in-grid-based-games-part-1-isometric-landscape%2F&title=Terrain%20Modification%20in%20Grid%20Based%20Games%20-%20Part%201%3A%20Isometric%20Landscape&bodytext=This%20part%20of%20the%20terrain%20manipulation%20tutorial%20deals%20with%20the%20creation%20of%20a%20tile%20map%20displaying%20the%20isometric%20map%20as%20a%20mesh.)
* [Sphinn](http://sphinn.com/index.php?c=post&m=submit&link=http%3A%2F%2Fwww.yarrcade.com%2F2010%2F09%2F19%2Fterrain-modificatio-in-grid-based-games-part-1-isometric-landscape%2F)
* [del.icio.us](http://delicious.com/post?url=http%3A%2F%2Fwww.yarrcade.com%2F2010%2F09%2F19%2Fterrain-modificatio-in-grid-based-games-part-1-isometric-landscape%2F&title=Terrain%20Modification%20in%20Grid%20Based%20Games%20-%20Part%201%3A%20Isometric%20Landscape&notes=This%20part%20of%20the%20terrain%20manipulation%20tutorial%20deals%20with%20the%20creation%20of%20a%20tile%20map%20displaying%20the%20isometric%20map%20as%20a%20mesh.)
* [Facebook](http://www.facebook.com/share.php?u=http%3A%2F%2Fwww.yarrcade.com%2F2010%2F09%2F19%2Fterrain-modificatio-in-grid-based-games-part-1-isometric-landscape%2F&t=Terrain%20Modification%20in%20Grid%20Based%20Games%20-%20Part%201%3A%20Isometric%20Landscape)
* [Google Bookmarks](http://www.google.com/bookmarks/mark?op=edit&bkmk=http%3A%2F%2Fwww.yarrcade.com%2F2010%2F09%2F19%2Fterrain-modificatio-in-grid-based-games-part-1-isometric-landscape%2F&title=Terrain%20Modification%20in%20Grid%20Based%20Games%20-%20Part%201%3A%20Isometric%20Landscape&annotation=This%20part%20of%20the%20terrain%20manipulation%20tutorial%20deals%20with%20the%20creation%20of%20a%20tile%20map%20displaying%20the%20isometric%20map%20as%20a%20mesh.)
* [email](mailto:?subject=Terrain%20Modification%20in%20Grid%20Based%20Games%20-%20Part%201%3A%20Isometric%20Landscape&body=http%3A%2F%2Fwww.yarrcade.com%2F2010%2F09%2F19%2Fterrain-modificatio-in-grid-based-games-part-1-isometric-landscape%2F)
* [Reddit](http://reddit.com/submit?url=http%3A%2F%2Fwww.yarrcade.com%2F2010%2F09%2F19%2Fterrain-modificatio-in-grid-based-games-part-1-isometric-landscape%2F&title=Terrain%20Modification%20in%20Grid%20Based%20Games%20-%20Part%201%3A%20Isometric%20Landscape)
* [StumbleUpon](http://www.stumbleupon.com/submit?url=http%3A%2F%2Fwww.yarrcade.com%2F2010%2F09%2F19%2Fterrain-modificatio-in-grid-based-games-part-1-isometric-landscape%2F&title=Terrain%20Modification%20in%20Grid%20Based%20Games%20-%20Part%201%3A%20Isometric%20Landscape)
* [Tumblr](http://www.tumblr.com/share?v=3&u=http%3A%2F%2Fwww.yarrcade.com%2F2010%2F09%2F19%2Fterrain-modificatio-in-grid-based-games-part-1-isometric-landscape%2F&t=Terrain%20Modification%20in%20Grid%20Based%20Games%20-%20Part%201%3A%20Isometric%20Landscape&s=This%20part%20of%20the%20terrain%20manipulation%20tutorial%20deals%20with%20the%20creation%20of%20a%20tile%20map%20displaying%20the%20isometric%20map%20as%20a%20mesh.)
* [Twitter](http://twitter.com/home?status=Terrain%20Modification%20in%20Grid%20Based%20Games%20-%20Part%201%3A%20Isometric%20Landscape%20-%20http%3A%2F%2Fwww.yarrcade.com%2F2010%2F09%2F19%2Fterrain-modificatio-in-grid-based-games-part-1-isometric-landscape%2F)

[http://www.natan.info/yarrcade/wp-content/themes/yarrcade/images/badge_468x60.gif](https://www.mochimedia.com/r/df814474072a5b17)

[Yarrcade and its Relocation  
←](http://www.yarrcade.com/2010/08/28/yarrcade-and-its-relocation/)

[Terrain Modification in Grid Based Games – Appendix A: Dynamic Tiles   
→](http://www.yarrcade.com/2010/09/24/terrain-modification-in-grid-based-games-appendix-a-dynamic-tiles/)

### 9 Responses to Terrain Modification in Grid Based Games – Part 1: Isometric Landscape

1. http://0.gravatar.com/avatar/a45beafe25c7914c10d5b449c550895e?s=40&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D40&r=R[*andi*](http://www.cafebun.com) says:

[October 9, 2010 at 5:11 am](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/#comment-123)

Hi.. this is nice tutorial., awesome!

but how to control character based on grid maps?

[Reply](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/?replytocom=123#respond)

* + http://1.gravatar.com/avatar/308e31225c5828c3b7e9559dcfa55013?s=40&d=http%3A%2F%2F1.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D40&r=Rkegogrog says:

[October 9, 2010 at 6:28 am](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/#comment-124)

Thank you.

The character control scheme depends on what you want to achieve. For unit movement in a tower defense game have a look at [this pathfinding tutorial](http://www.yarrcade.com/2010/04/10/grid-based-games-part-5-3-pathfinding/). While each node has a parent node there you can give the unit a start node and tell is in an ENTER\_FRAME event to move to that node’s parent node. The same goes for a dune type game whereas the start node would be the units actual node and the end node any node you clicked on. Send me a mail if you need something more on that.

[Reply](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/?replytocom=124#respond)

1. http://1.gravatar.com/avatar/70b5e8ae13969148c5a8516bd6d9b406?s=40&d=http%3A%2F%2F1.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D40&r=R[*webkos*](http://www.webkos.com.au) says:

[October 9, 2010 at 3:59 pm](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/#comment-125)

you solved my problem , thanks man

[Reply](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/?replytocom=125#respond)

1. http://0.gravatar.com/avatar/27f13066d79b98d7eff6300e412f077c?s=40&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D40&r=RJJ says:

[October 19, 2010 at 8:51 pm](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/#comment-133)

Hey, kegogrog thanks a lot for the tutorial. I found this after a lot of searching google on how to draw Isometric maps in AS3. I think a tutorial like this is rare.  
I have a question.

Consider this stage of your tutorial  
Part 1.3: The basic map with drawn tiles.  
At that stage, Could you show me a way to fill each tile of the map with a bitmap or some other graphic type.

I’m asking you at that stage because I notice, later on the tutorial gets more complicated.

[Reply](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/?replytocom=133#respond)

* + http://1.gravatar.com/avatar/308e31225c5828c3b7e9559dcfa55013?s=40&d=http%3A%2F%2F1.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D40&r=Rkegogrog says:

[October 21, 2010 at 3:25 pm](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/#comment-136)

What exactly do you want to achieve? Do you want to use graphics (imported or drawn in flash) for tiles? Or do you really want to fill the tile like texturing it?

[Reply](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/?replytocom=136#respond)

* + - http://0.gravatar.com/avatar/27f13066d79b98d7eff6300e412f077c?s=40&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D40&r=RJJ says:

[October 21, 2010 at 9:17 pm](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/#comment-137)

Thanks for taking time to reply kegogrog.

You asked  
“Do you want to use graphics (imported or drawn in flash), for tiles? Or do you really want to fill the tile like texturing it?”

I meant filling the tiles like texturing it.

* + - http://1.gravatar.com/avatar/308e31225c5828c3b7e9559dcfa55013?s=40&d=http%3A%2F%2F1.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D40&r=Rkegogrog says:

[October 24, 2010 at 10:19 pm](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/#comment-142)

How about [that link (click here)](http://www.yarrcade.com/2010/10/24/terrain-modification-in-grid-based-games-%e2%80%93-appendix-c-shading-and-bitmaps/)?  
<http://www.yarrcade.com/2010/10/24/terrain-modification-in-grid-based-games-%e2%80%93-appendix-c-shading-and-bitmaps/>

* + - http://0.gravatar.com/avatar/27f13066d79b98d7eff6300e412f077c?s=40&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D40&r=RJJ says:

[October 22, 2010 at 12:05 am](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/#comment-138)

Btw kegogrog, at your leisure please. You have already helped me understand a lot.

No urgency here :D. I am only learning.

1. http://0.gravatar.com/avatar/27f13066d79b98d7eff6300e412f077c?s=40&d=http%3A%2F%2F0.gravatar.com%2Favatar%2Fad516503a11cd5ca435acc9bb6523536%3Fs%3D40&r=RJJ says:

[October 19, 2010 at 9:03 pm](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/#comment-134)

Is a solution possible based on using beginBitmapFill() ?  
<http://www.adobe.com/livedocs/flash/9.0/ActionScriptLangRefV3/flash/display/Graphics.html#beginBitmapFill%28%29>

[R](http://www.yarrcade.com/2010/09/19/terrain-modificatio-in-grid-based-games-part-1-isometric-landscape/?replytocom=134#respond)