

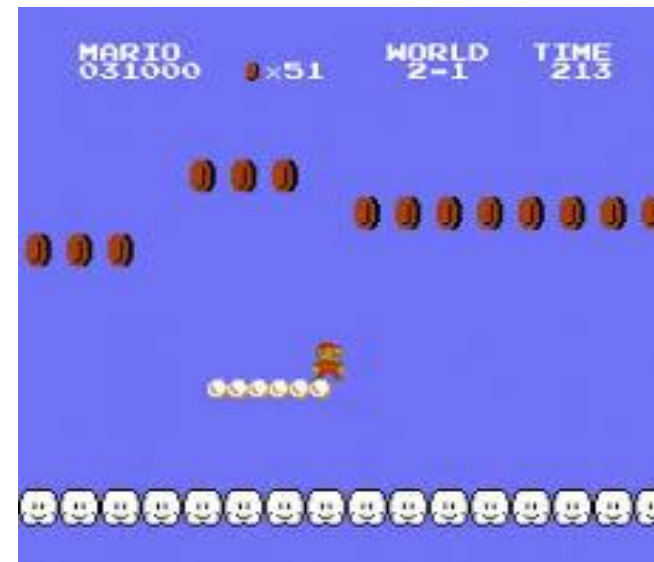
The slide features four decorative green geometric shapes in the corners: a tall thin triangle at top-left, a large complex polygon at top-right, a small triangle at bottom-left, and a large complex polygon at bottom-right. All are in various shades of green.

# HTML5 and JavaScript Games Programming Week 6

[Mario.Soflano@uws.ac.uk](mailto:Mario.Soflano@uws.ac.uk)

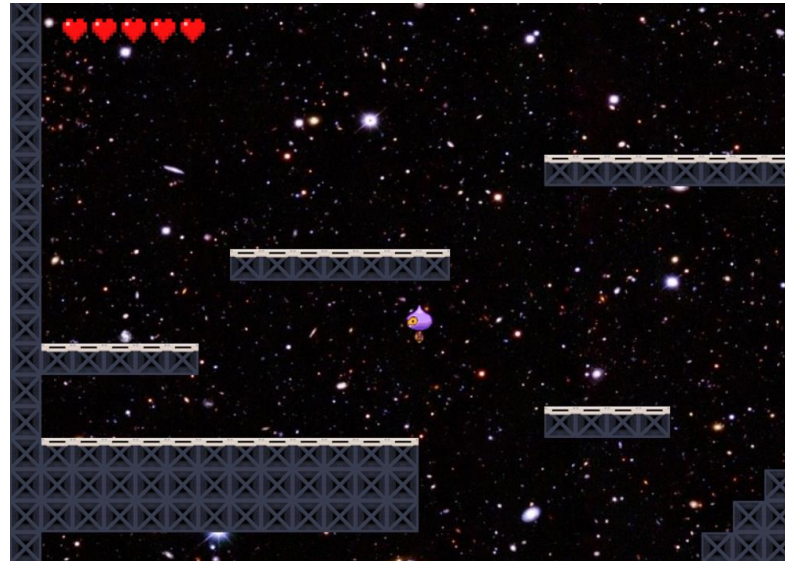
# Environment Design

- ▶ Use environment to control the pace and rhythm of the game
- ▶ Do not create a blank space for the sake of it



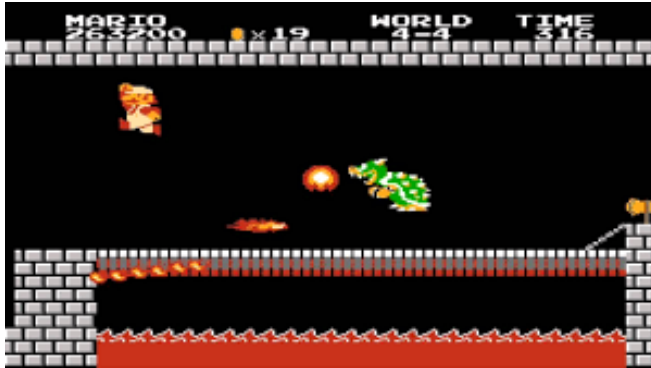
# Environment Design

- ▶ Study other games and their environments
- ▶ If similar environment design has been applied in other games, differentiate your design while maintaining the challenge and value of your environment



# Environment Design

- ▶ Environments should grow as the player progress through them
- ▶ Quality is better than quantity



# Environment Design Principles

## ► Overlap





# Environment Design Principles

## ► Base-Height (Vertical perspective)



# Environment Design Principles

- Scale - the greater the distance an object is from the view point, the smaller it appears



Lynn and the Spirit of Inao  
(Studio Ghibli)

# Environment Design Principles

## ► Atmosphere





# Environment Design Principles

- Focus - blur the objects in the far



# Environment Design Principles

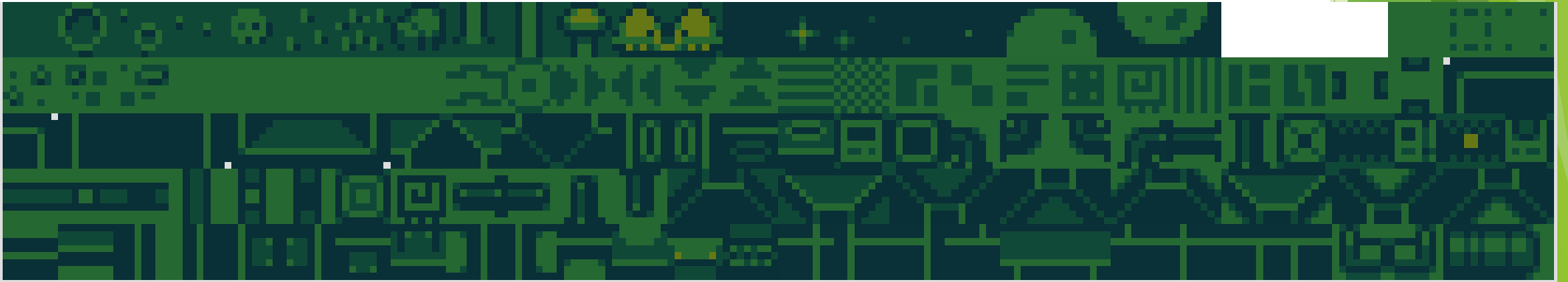
- ▶ Parallax (create layers of objects that moves in different speed)
- ▶ <https://www.youtube.com/watch?v=DfRAOvg3YRg>

# Phaser - Game Environment Data Format

- ▶ Game environment in Phaser is represented as tiles
- ▶ Since Phaser is a library, there is no graphical game environment design available
- ▶ Instead, Phaser uses CSV and JSON which contains a series of variables and values which determine the image and the position of the image
- ▶ The image itself is a PNG file containing tilesheet of images created by tools such as Piskel

# Comma-Separated Values (CSV)

Determine the size of the tile in correspond to the image





# Comma-Separated Values (CSV)

- ▶ Create a sequence of numbers which determine the image to be displayed on the tiles. The first number will show the first tile at the top left of the game window
- ▶ New row in the CSV file means create a new row of tiles in the game environment

	A	B	C	D	E
1	0	5	39		
2					
3	0	6			
4					
5					



# JavaScript Object Notation (JSON)

- ▶ **JSON** (JavaScript Object Notation) is a lightweight data-interchange format which is are easy for humans to read and write and for machines to parse and generate (<http://json.org>)
- ▶ **JSON** is ideal for data interchange between languages
- ▶ **JSON** consists of two elements:
  - ▶ A collection of name/value pairs. The structure is:  
`{ string1 : value1, string 2 : value2, ... }`
  - ▶ An ordered list of values. The structure is:  
`[ value1, value2, ... ]`
- ▶ So one name can have multiple values:  
`{ string1 : [ value1, value2, ... ], string 2 : [ value11, value12, ... ], ... }`



# JSON In Phaser

Layers: Data, Height, Name, Type, Visible, Width, X, Y

Orientation: Orthogonal

Tileheight:

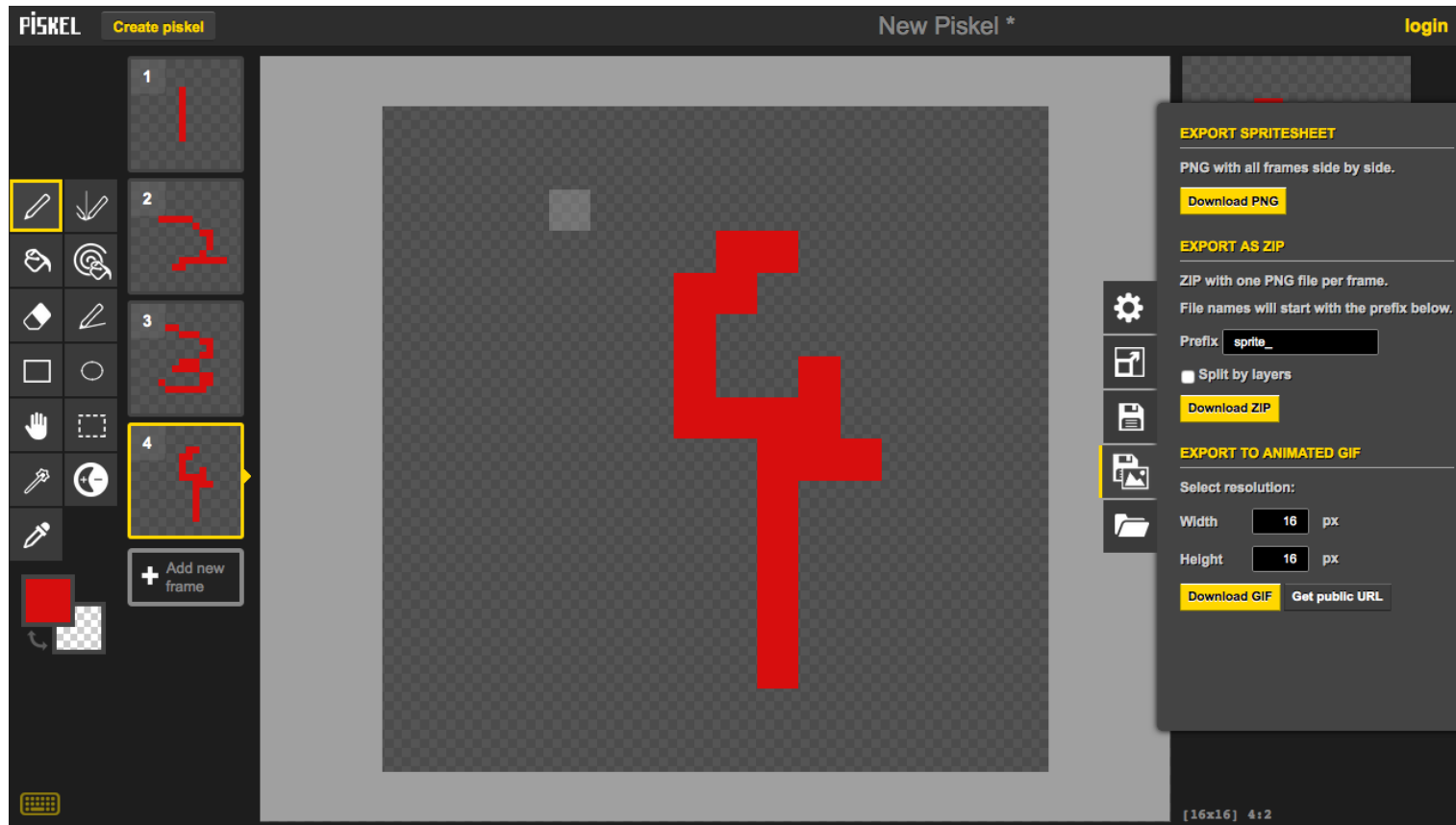
Tilewidth:

Tilessets: Firstgrid, Image, Imagewidth, Imageheight, Spacing,  
Tilecount, Tileheight, Tilewidth



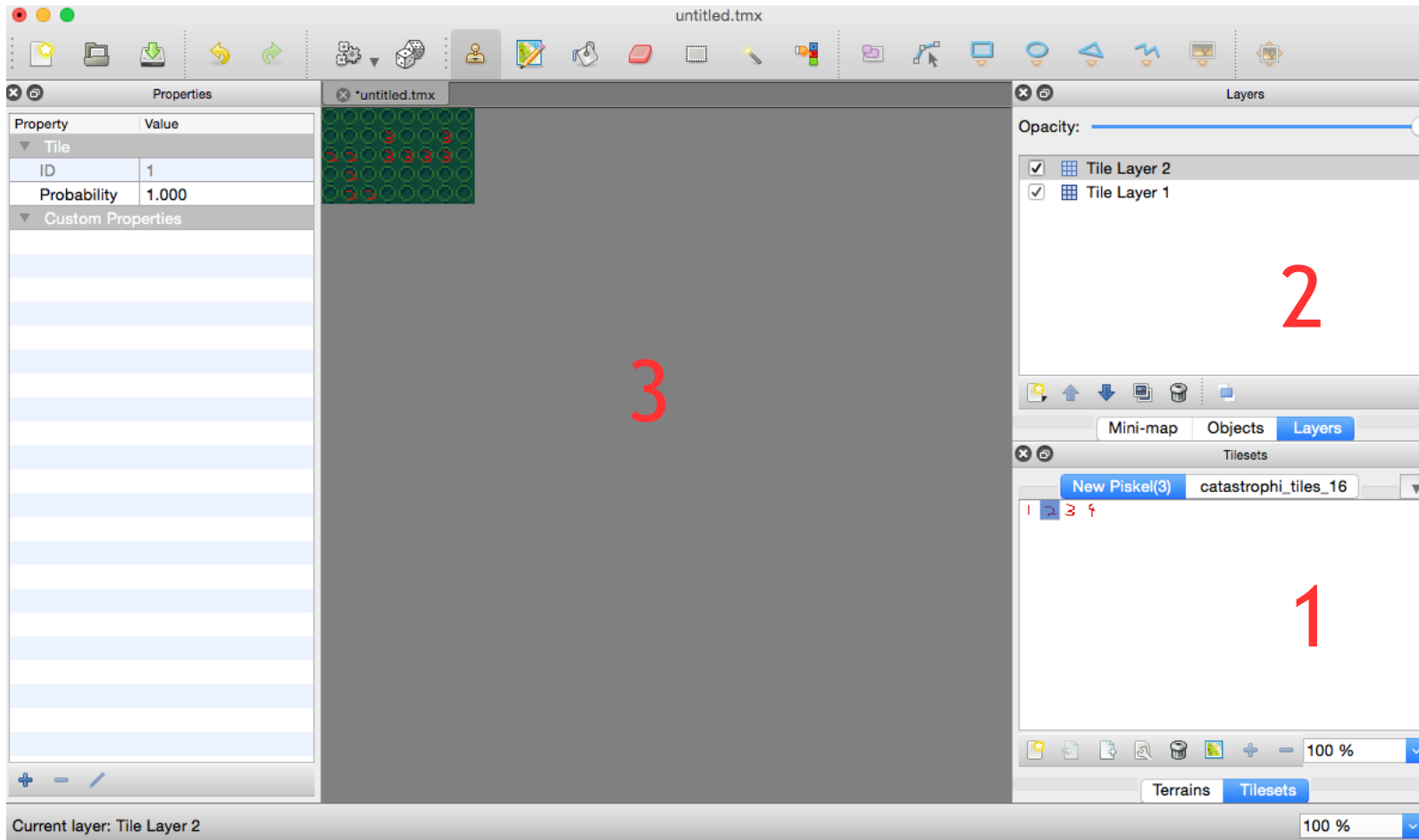
# Resources for Game Environment

Piskel: to create spritesheet



# Resources for Game Environment

**Tiled:** to generate tilemap (CSV or JSON)



# Resources for Game Environment

1. Tilesets window: Add / remove tilesets resources
2. Layer window: Add / remove layer
3. Tilemap window: design the game environment

## Steps:

1. Add tilesets
2. Drag and drop particular tile from the tilesets window to the tilemap window
3. If multiple layers intended, create layer(s) as required in layer window before adding tiles to the intended layer

# Implementing Game Environment

- ▶ Load the tilemap (normally in preload state)
  - `load.tilemap('tilemapVarNameJSON', 'tilemapJSON.json', null, Phaser.Tilemap.TILED_JSON);`
    - `load.tilemap('tilemapVarNameCSV', 'tilemapCSV.csv', null, Phaser.Tilemap.CSV);`

Load the spritesheet (normally in preload state)

- `load.image('tilesetVarName', 'tileImage.png');`



# Implement Game Environment

Display the CSV resources (normally in create state)

```
// add.tilemap(variablename, width,height)
add.tilemap('tilemapVarNameCSV', 16, 16);
// add the tileset
addTilesetImage('tilesetVarName');
```

# Implement Game Environment

Display the JSON resources (normally in create state)

```
// add.tilemap(variablename, width,height)
add.tilemap('tilemapVarNameJSON');

// add the tileset: addTilesetImage(Name of Image
specified in JSON generated by Tiled',variable name
when loading spritesheet');
addTilesetImage('New Piskel(3)','tiles');
```

# Implement Game Environment

To create multilayer game environment:

1. Load the tilemaps and tilesets in preload state
2. Create a layer array variable (`var layer = {};`)
3. Add tilemap and tilesets
4. Add layers specified in JSON to the array

```
layer[0] = map.createLayer('Tile Layer 1');
```

```
layer[0].resizeWorld();
```

```
layer[1] = map.createLayer('Tile Layer 2');
```

```
layer[1].resizeWorld();
```

# References

- Pile Jr, J. 2D Graphics Programming for Games (2013). CRC Press
- Oxland, K. Gameplay and Design. Addison-Wesley
- [www.piskelapp.com/](http://www.piskelapp.com/)
- <http://www.mapeditor.org/>