CMP105 Games Programming

Tile-based game

This week

- Tile based game
  - Tile Sets
  - Tile Maps
  - How they work together
- Example
  - Building a small section of a level
  - Adding collision

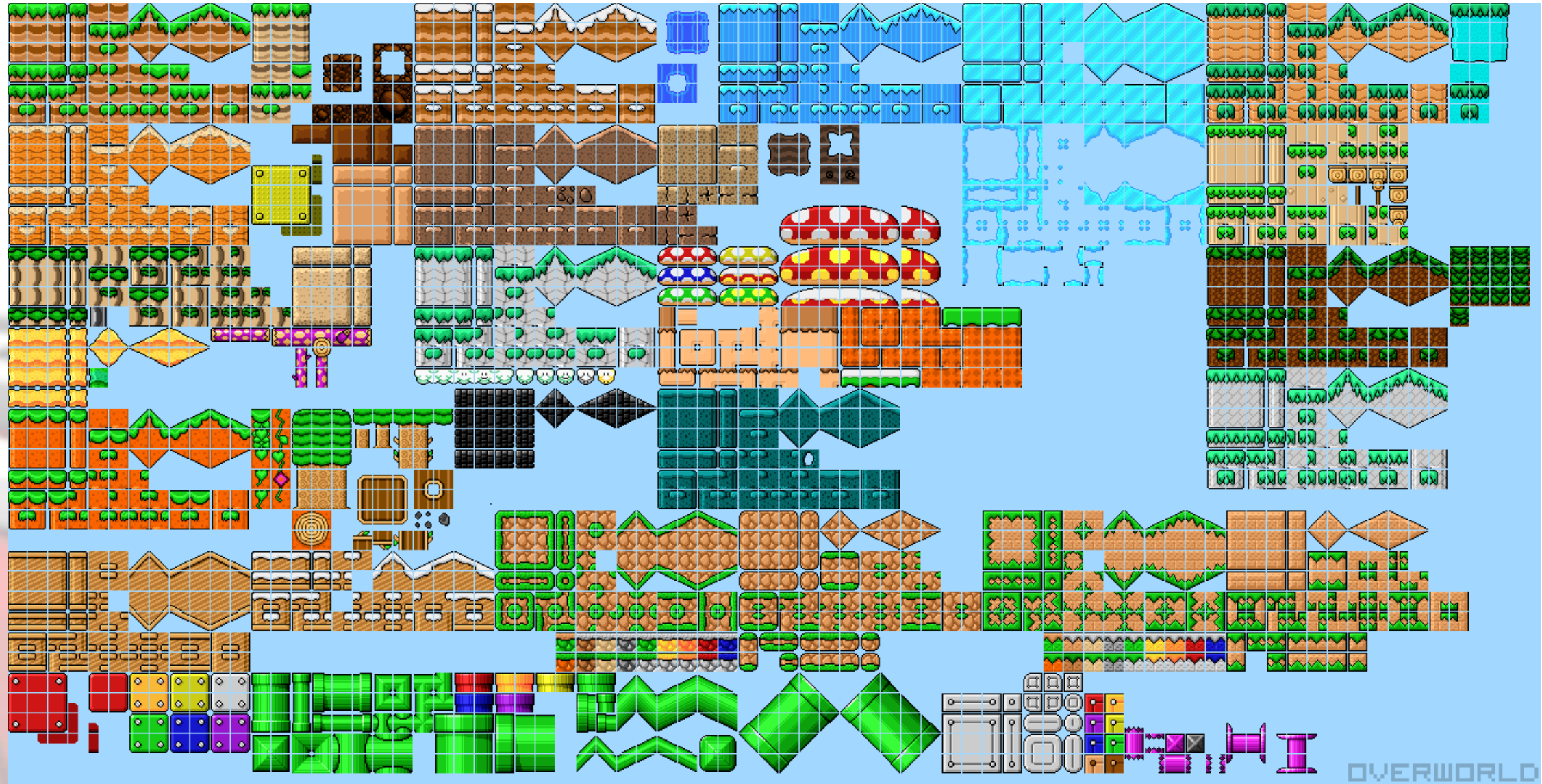
Tile-based

- A tile-based game lays out tiles in order to create each level
- Components
  - A tile
    - Small image acts like a puzzle piece of art for building larger images
  - A map
    - Groupings of tiles put together to create a level, section or area

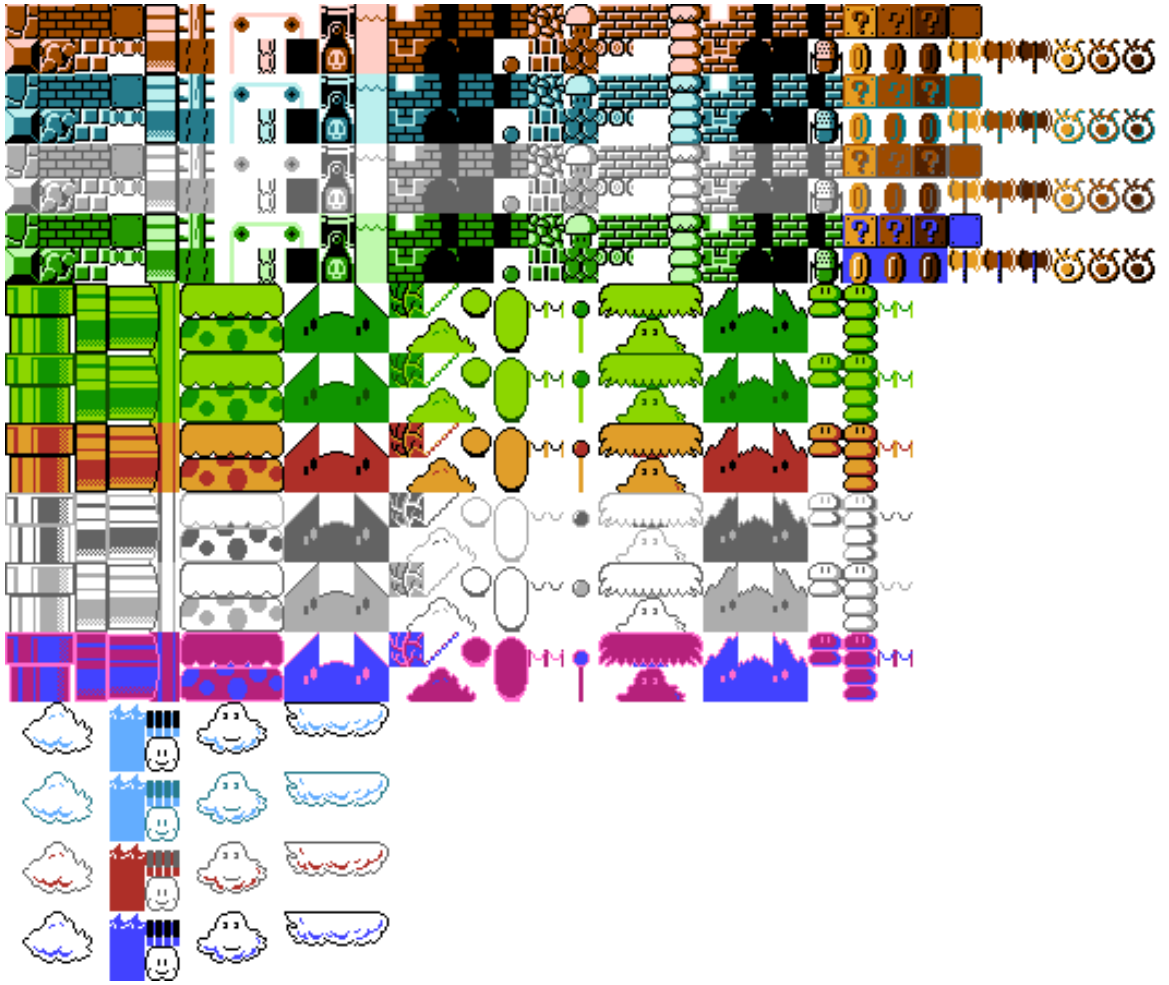
Tile-based

- Main benefit of tile based
  - No need to create large images by hand for each level
  - Instead of 50 large images for 50 levels
  - One image of 100 tiles and some maps
- Some example Tile sheets

Super Mario world



Zelda and more Mario

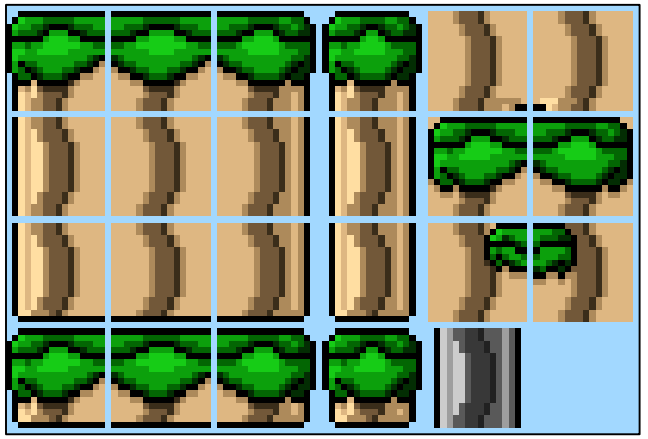


Pokemans

Tiles und Original-Tielset: **Wesley FG** ... Aufbereitung für Mappainter: The Libertine / **www.pokefans.net**

Tile Set

- A Tile
  - An element of a sprite/tile sheet
  - Can be combined to make different sections
- A Tile set
  - A list of tiles
  - Index value identifies tile
  - Based on order of storage



1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	0

Tile Map

- A data structure describing what and where tiles should be placed
- We need to know the dimension
  - 6 x 5s
  - 10 x 6
  - 100 x 100
  - Etc
- Store Tile index in an array
  - Easier to work with a single dimension array (vector)
  - Requires that we store dimensions of the map separately

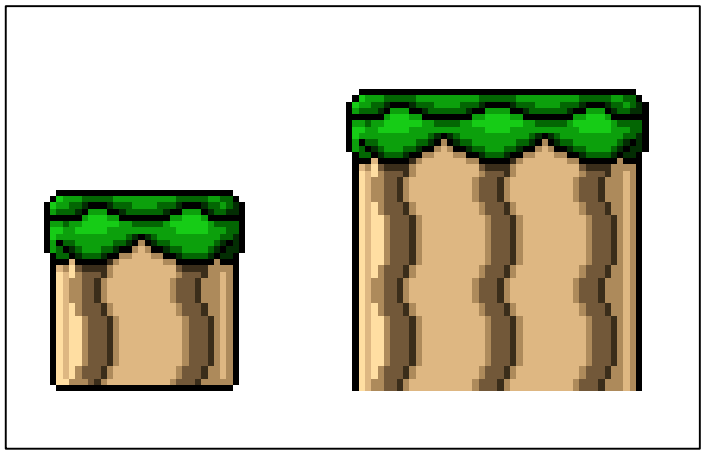
```
std::vector<int> map = {
        0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0,
        0, 0, 0, 1, 2, 3,
        1, 2, 0, 4, 5, 6,
        7, 9, 0, 4, 5, 6,
};
```

Working together

Tile Set

```
std::vector<int> map = {
        0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0,
        0, 0, 0, 1, 2, 3,
        1, 2, 0, 4, 5, 6,
        7, 9, 0, 4, 5, 6,
};
```

Tile Map



Level

Example

- Build a classes to represent
  - A Tile
  - A Tile Map (simple array)
  - A Level (Tile + Map)
- Bring it all together to render a level
  - Added bonus of collisions

Tile class

- A basic class that inherits from Sprite
  - Doesn’t do anything special
  - Essentially a static sprite
- Will store a subsection of the Tile sheet
- Could be Animated sprite

Map class

- Will represent our level
  - Major function is to build a level/section/map
  - From a provided TileSet and TileMap
- Also handles rendering of the section

```cpp
#pragma once
#include <math.h>
#include "Tile.h"

class Map
{
    public:
        Map();
        ~Map();

        void loadTexture(char* filename);
        void setTileSet(std::vector<Tile> ts);
        void setTileMap(std::vector<int> tm, sf::Vector2u mapDimensions);
        void buildLevel();

        void render(sf::RenderWindow* window);
        std::vector<Tile>* getLevel(){ return &level; };
        void setPosition(sf::Vector2f pos) { position = pos; };
    protected:
        std::vector<Tile> tileSet;
        std::vector<int> tileMap;
        std::vector<Tile> level;
        sf::Texture texture;
        sf::Vector2u mapSize;
};
```

Map.cpp

```
void Map::render(sf::RenderWindow* window)
{
    for (int i = 0; i < (int)level.size(); i++)
    {
    window->draw(level[i]);
    }
}

void Map::loadTexture(char* filename)
{
        texture.loadFromFile(filename);
}

void Map::setTileSet(std::vector<Tile> ts)
{
        tileSet = ts;
}
```

```
void Map::setTileMap(std::vector<int> tm, sf::Vector2u mapDimensions)
{
    tileMap = tm;
    mapSize = mapDimensions;
}

void Map::buildLevel()
{
    if (tileSet.size() > 0 && tileMap.size() > 0)
    {
        int x, y = 0;
        sf::Vector2f tileSize(tileSet[0].getSize().x, tileSet[0].getSize().y);
        for (int i = 0; i < (int)tileMap.size(); i++)
        {
                x = i % mapSize.x;
                y = (int)floor(i / mapSize.x);
                tileSet[tileMap[i]].setPosition(x * tileSize.x, y * tileSize.y);
                level.push_back(tileSet[tileMap[i]]);
                level[i].setTexture(&texture);
                level[i].updateAABB();
        }
    }
}
```

Using the Map class

- Create a new Map variable
  - Here mine is called level
- The setup
  - Load texture
  - Set Tile Set
  - Set Tile Map
  - Build level

Using the Map class

- Load texture

```
level.loadTexture("gfx/marioTiles.png");
```

- Set Tile Set

```
Tile tile;
std::vector<Tile> tiles;
for (int i = 0; i < 7; i++)
{
    tile.setSize(sf::Vector2f(32, 32));
    tile.setAlive(true);
    tiles.push_back(tile);
}
```

Using the Map class

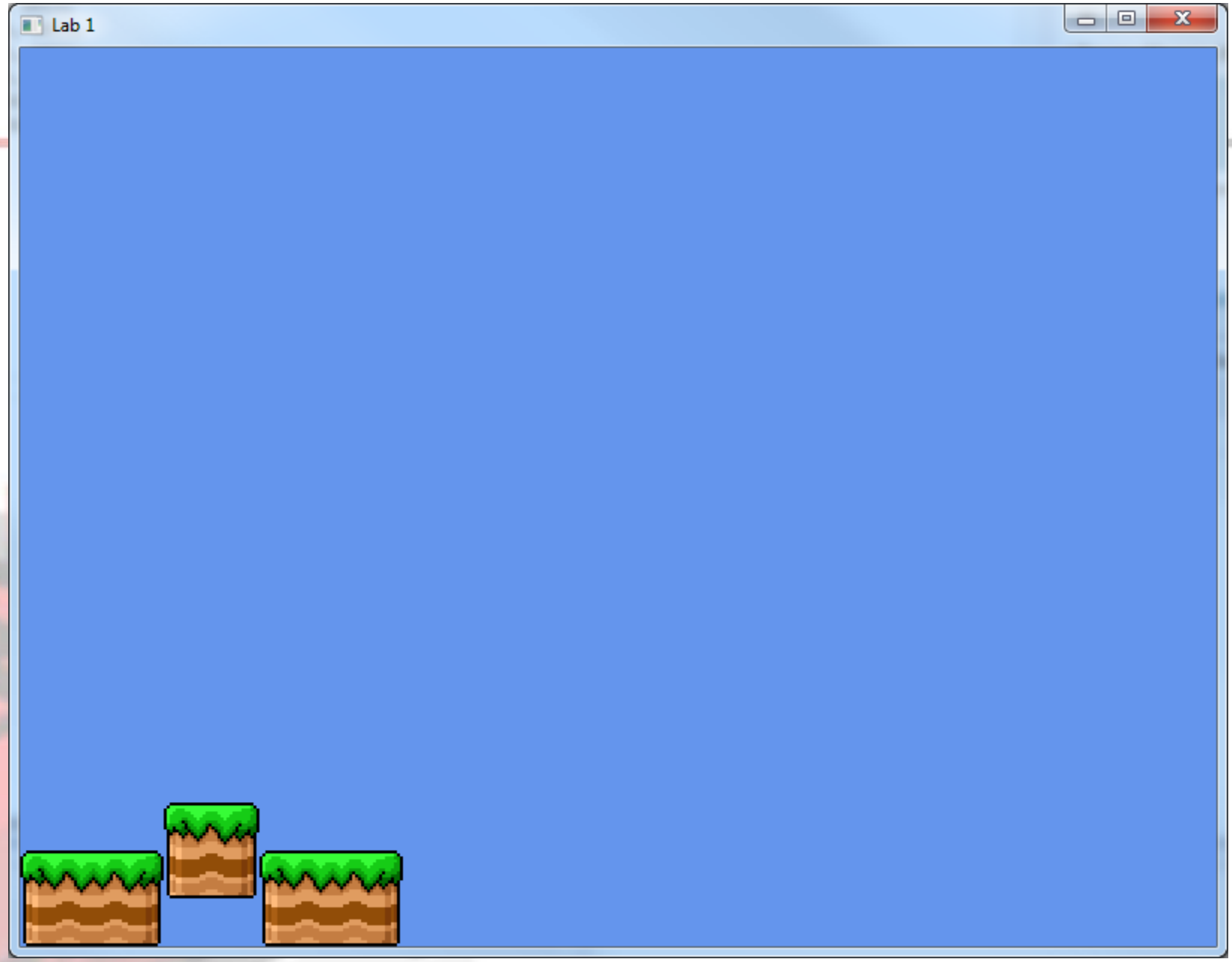
- Set tile set

```
tiles[0].setAlive(false);
tiles[0].setTextureRect(sf::IntRect(187, 51, 16, 16));
tiles[1].setTextureRect(sf::IntRect(0, 0, 16, 16));
tiles[2].setTextureRect(sf::IntRect(17, 0, 16, 16));
tiles[3].setTextureRect(sf::IntRect(34, 0, 16, 16));
tiles[4].setTextureRect(sf::IntRect(0, 34, 16, 16));
tiles[5].setTextureRect(sf::IntRect(17, 34, 16, 16));
tiles[6].setTextureRect(sf::IntRect(34, 34, 16, 16));

level.setTileSet(tiles);
```

Using the Map class

- Set Tile Map

```
// Map dimensions
sf::Vector2u mapSize(10, 6);
// build map
std::vector<int> map = {
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 1, 3, 0, 0, 0, 0, 0,
 1, 2, 3, 4, 6, 1, 2, 3, 0, 0,
 4, 5, 6, 0, 0, 4, 5, 6, 0, 0
};

level.setTileMap(map, mapSize);
level.buildLevel();
```


Lab 1

What about collision?

- All the Tiles are based on Sprite
  - We can re-use collision code
- Notice the use of the Alive variable
  - Not for spawning
  - But for if collision should be checked
    - Could be a background sprite we don't want to collide with
- Slight update to collision
  - For collision resolution, pass the colliding sprite
  - Further calculations can be done

What about collision?

```
std::vector<Tile>* world = level.getLevel();
for (int i = 0; i < (int)world->size(); i++)
{
   // if "alive" check collision
   if ((*world)[i].isAlive())
   {
       if (checkCollision(&player, &(*world)[i]))
       {
             player.collisionRespone(&(*world)[i]);
       }
   }
}
```

Collision response

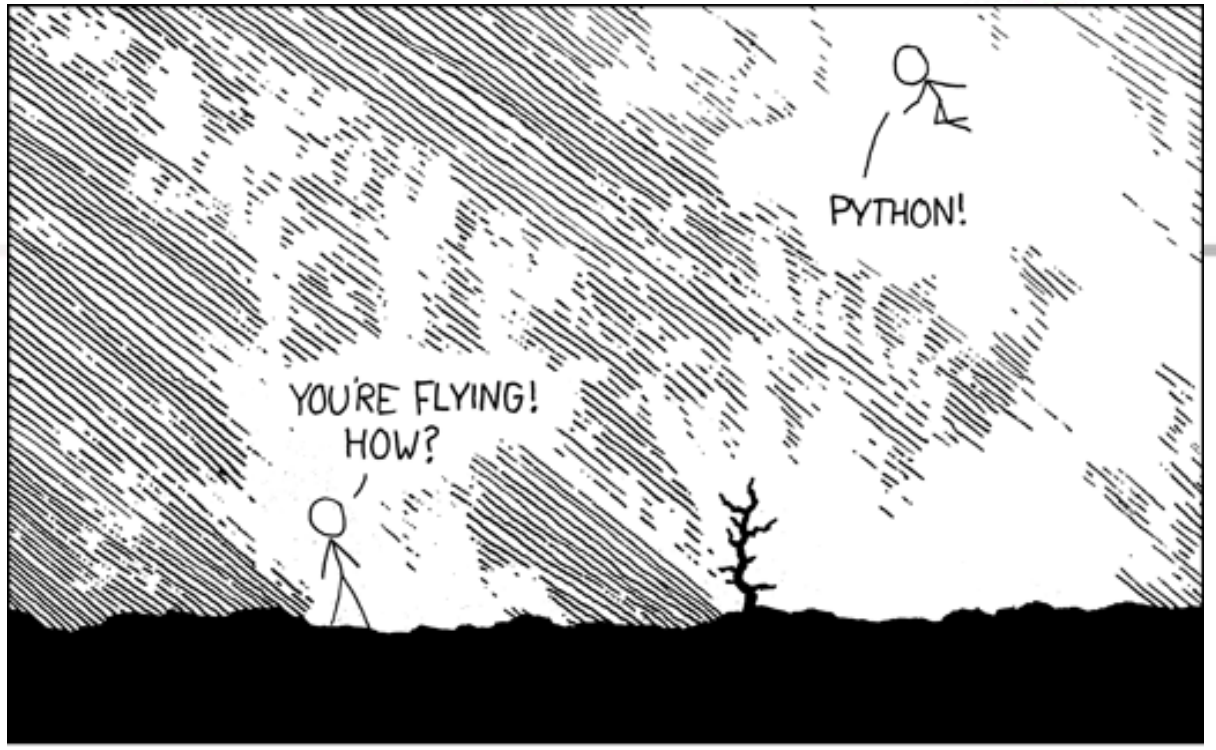
```
void Player::collisionRespone(Sprite* sp)
{
	velocity.y = 0;
	setPosition(getPosition().x, sp->getPosition().y-getSize().y);
}
```

Live demo

- With collisions

In the labs

- Building tile based level



https://xkcd.com/353/