#15.2 Tilemaps in SFML

Um eine Tilemap in SFML umzusetzen, brauchen wir als erstes eine neue Entity, die wir Tilemap nennen.

class Tilemap

{

Screen screen;

// Konstruktor

public Tilemap(Screen parentScreen)

{

}

// Loop Methode

public void loop()

{

}

public void draw()

{

}

}

Eine Tilemap braucht aufjedenfall immer ein Tileset. Deswegen fügen wir als Parameter im Konstruktor eine Textur hinzu. Die speichern wir auch als Variable in der Klasse.

Texture tileset;

// Konstruktor

public Tilemap(Screen parentScreen, Texture set)

{

tileset = set;

}

Außerdem müssen wir zusätzlich die Abmessungen der Tiles wissen, um das Tileset unterteilen zu können. Dazu fügen wir noch die Parameter tileWidth und tileHeight hinzu und speichern auch sie in der Klasse.

Texture tileset;

int tileWidth;

int tileHeight;

// Konstruktor

public Tilemap(Screen parentScreen, Texture set, int width, int height)

{

//Speicher alle Parameter in der Klasse

screen = parentScreen;

tileset = set;

tileWidth = width;

tileHeight = height;

}

Jetzt müssen wir das Tileset im Konstruktor in seine Tiles aufteilen. Dazu erstellen wir uns einen Array von Sprites. Jedes Sprite soll ein Tile sein, also brauchen wir soviele Sprites wie Tiles. Da wir die Maße eines Tiles kennen und die Höhe und Breite der Textur, können wir ausrechnen wie viele Tiles wir haben.

public Tilemap(Screen parentScreen, Texture set, int width, int height)

{

//Speicher alle Parameter in der Klasse

screen = parentScreen;

tileset = set;

tileWidth = width;

tileHeight = height;

//Anzahl der Tiles in einer Reihe

int w = (int)(tileset.Size.X / tileWidth);

//Anzahl der Tiles in einer Spalte

int h = (int)(tileset.Size.Y / tileHeight);

//Anzahl der Tiles im Tileset

int amount = w \* h;

}

Jetzt brauchen wir einen Sprite-Array, der die Tiles speichern kann.

Texture tileset;

int tileWidth;

int tileHeight;

Sprite[] tiles;

// Konstruktor

public Tilemap(Screen parentScreen, Texture set, int width, int height)

{

//Speicher alle Parameter in der Klasse

screen = parentScreen;

tileset = set;

tileWidth = width;

tileHeight = height;

//Anzahl der Tiles in einer Reihe

int w = (int)(tileset.Size.X / tileWidth);

//Anzahl der Tiles in einer Spalte

int h = (int)(tileset.Size.Y / tileHeight);

//Anzahl der Tiles im Tileset

int amount = w \* h;

//Erstelle Array zum Speichern der Tiles

tiles = new Sprite[amount];

}

Jetzt müssen wir mit einer Schleife das Array füllen, dazu nehmen wir uns immer ein Stück aus dem Tileset und gehen dann zum nächsten Tile.

// Konstruktor

public Tilemap(Screen parentScreen, Texture set, int width, int height)

{

[...]

//Erstelle Array zum Speichern der Tiles

tiles = new Sprite[amount];

//Fülle das Array mit allen Tiles

for (int i = 0; i < amount; i++)

{

tiles[i] = new Sprite(tileset,new IntRect( (i%w) \* tileWidth,

(i/w) \* tileHeight, tileWidth, tileHeight));

}

}

Jetzt haben wir unser Array befüllt. Was noch fehlt ist jetzt die eigentliche map, die wir zeichnen wollen. Die map ist ein 2D Feld aus ganzzahlen, deswegen benutzen wir einen 2-Dimensionalen-Integer Array. Ein 2 Dimensionaler Array ist einfach ein Array, der 2 Indexe hat. Diese kann man wie x/y koordinaten sehen um die Werte in einem 2D Feld anzuordnen, wie in einer Tilemap. Die Map kriegen wir auch als Parameter im Konstruktor.

int[][] map;

// Konstruktor

public Tilemap(Screen parentScreen, Texture set, int width, int height,

int[][] tilemap)

{

//Speicher alle Parameter in der Klasse

screen = parentScreen;

tileset = set;

tileWidth = width;

tileHeight = height;

map = tilemap;;

[...]

Als letztes müssen wir in der Draw-Methode unsere Tiles nurnoch wie in der Map zeichnen. Dazu laufen wir durch das 2-Dimensionale Array (dafür brauch man 2 for-schleifen),und bewegen und zeichnen das Tile, das in der Map steht.

public void draw()

{

//Durchlaufe jede Reihe der Map

for (int y = 0; y < map.Length; y++)

{

//Durchlaufe jedes Tile der Reihe

for (int x = 0; x < map[y].Length; x++)

{

//Bewege das Tile an die richtige Stelle

tiles[map[y][x]].Position = new Vector2f(tileWidth\*x, tileHeight\*y);

//Zeiche das Tile

screen.draw(tiles[map[y][x]]);

}

}

}

Jetzt ist unsere TileMap Klasse fertig. Wir können sie jetzt in unserem Mainscreen benutzen. Dazu brauchen wir eine Tileset Textur, deren Tilemaße und eine Map. Das Tileset findet man im Ressourcen Ordner(die Maße stehen im Dateinamen), eine Beispiel Map ist hier gegeben.

Tilemap tilemap;

int[][] map = new int[][] {

new int[]{1,1,1,1,1,1},

new int[]{1,3,2,2,3,1},

new int[]{1,2,2,2,2,1},

new int[]{1,2,2,2,2,1},

new int[]{1,2,2,2,2,1},

new int[]{1,3,2,2,3,1},

new int[]{1,1,1,1,1,1}

};

// Setup, wird immer einmal zu Beginn eines Screens aufgerufen

// Hier Startwerte setzen!

public override void setup()

{

Texture tileset = new Texture("assets/tileset32x32.png");

tilemap = new Tilemap(this, tileset, 32, 32, map);

}

// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen

public override void loop()

{

tilemap.loop();

tilemap.draw();

}