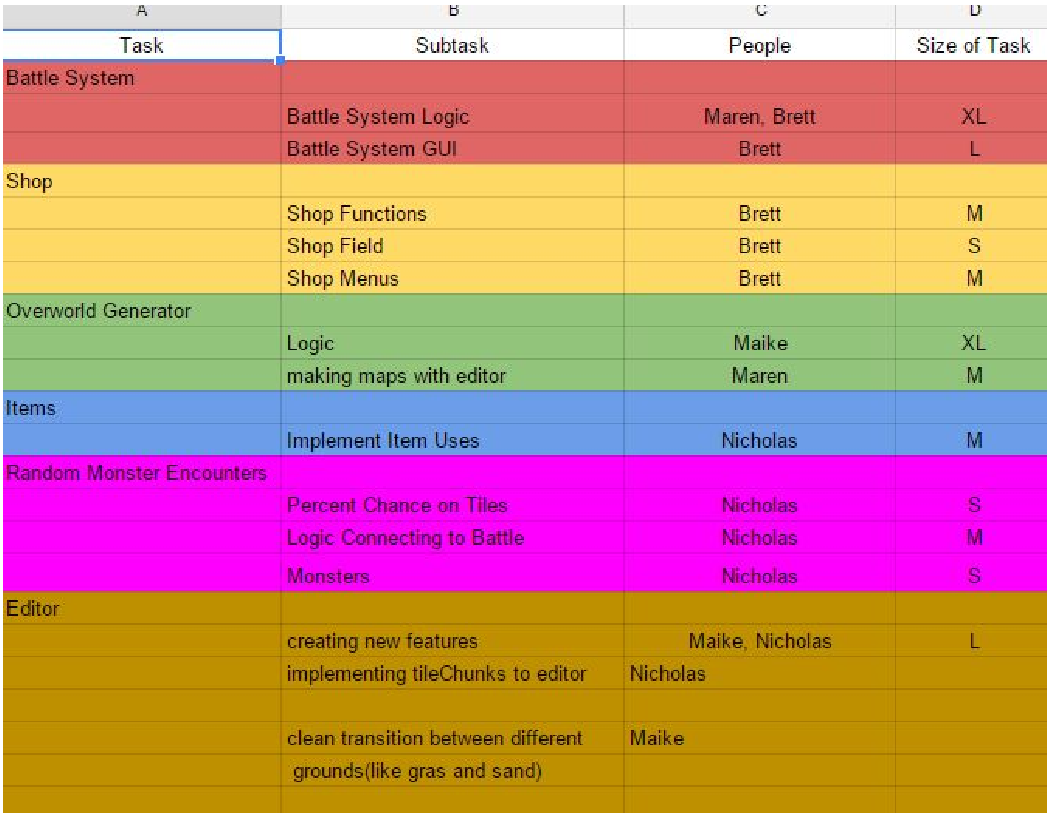
Sprint 3 Report

Brett Raible
Maren Sandner
Nicholas Paquette
Maike Keune

In the Sprint 3 report we will talk about our progress during the third Sprint of the computer science project.

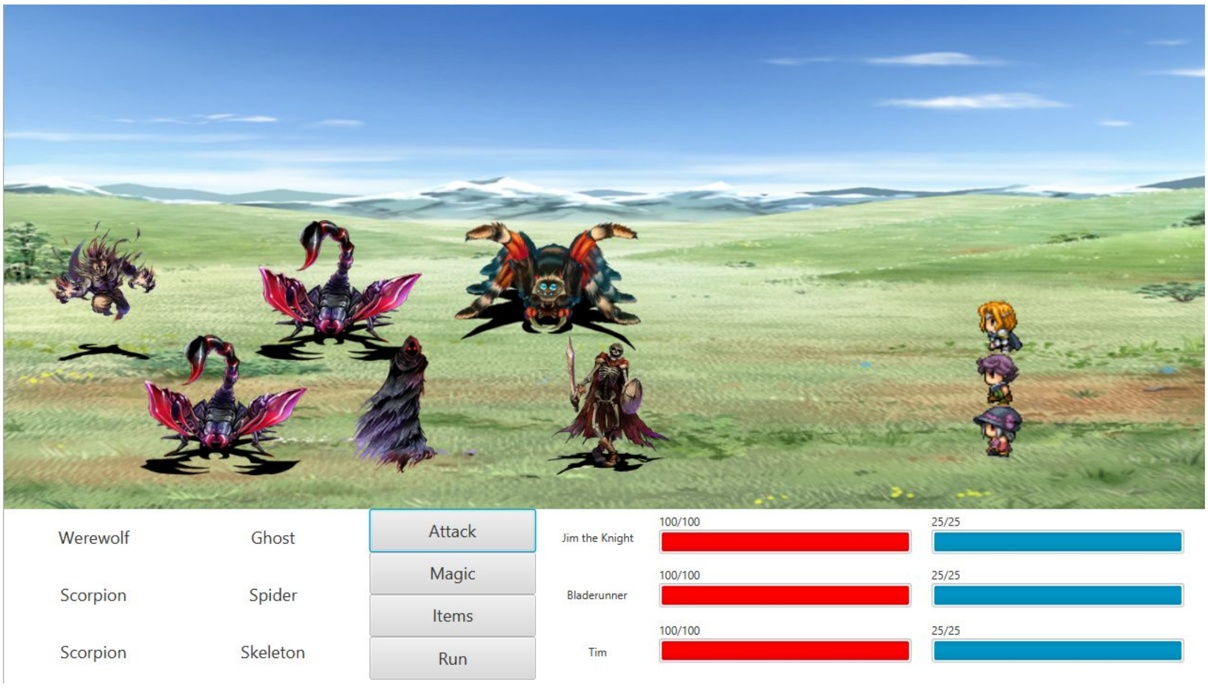
Progress reflection:
Our goals for the third Sprint are shown in this task sheet which we created at the beginning of the sprint.



A	B	C	D
Task	Subtask	People	Size of Task
Battle System			
	Battle System Logic	Maren, Brett	XL
	Battle System GUI	Brett	L
Shop			
	Shop Functions	Brett	M
	Shop Field	Brett	S
	Shop Menus	Brett	M
Overworld Generator			
	Logic	Maike	XL
	making maps with editor	Maren	M
Items			
	Implement Item Uses	Nicholas	M
Random Monster Encounters			
	Percent Chance on Tiles	Nicholas	S
	Logic Connecting to Battle	Nicholas	M
	Monsters	Nicholas	S
Editor			
	creating new features	Maike, Nicholas	L
	implementing tileChunks to editor	Nicholas	
	clean transition between different	Maike	
	grounds(like gras and sand)		

The dungeon generator can now link one field to another (see Editor tutorial part 2). The dungeon generator can now add fields into the dungeon, which are represented as small rooms depending on the dungeon type. These rooms are linked in the “DungeonHelper“ class to a specific dungeon type.
The world map consists of 3x2 smaller map chunks (those were made in the editor first) which are loaded from the hard drive and are pieced together randomly into one big field.
The editor was overhauled and now contains new functionality which will be described in part two of the Editor tutorial.

While moving the player, there is a percentage chance to start a battle, which changes the screen to a battle screen and loads a random party of monsters, containing a minimum of one monster and a maximum of six. We have different types of enemies ranging from low to high levels, depending on the level of the characters. Currently, the battle system does not completely work as the battle logic and the GUI are not completely linked together yet.



As of now you can see the names of the spawned enemies, the options you have as a player, and the three player actors. The player has four options: Attack, Magic, Items and Run. When pressing “Attack” you can choose an enemy and then it should perform a basic attack, the damage depending on the weapon the player has equipped. This is not implemented yet but should get done in the last week. When selecting “Magic” you can also choose the enemy and then choose one of four spells of the player. Every type of player has their own set of spells. In the beginning they have one spell, then as the level increases they will either get a new spell or the available spells receives higher damage rates.

The current spells available are described below:

Knight

Shield Bash: Shield attack, 60% chance to stun the enemy
Whirlwind: Damage on two targets
Berserk: More damage when the player’s health points are under 25%
Massive Sword Slash: High damage

Mage

Fireball: Damage on one target
Chain Lightning: Damage on multiple targets (depending on the level of the spell)
Heal: Healing a character and with higher level, the whole party
Frostbite: Disable enemy for one round

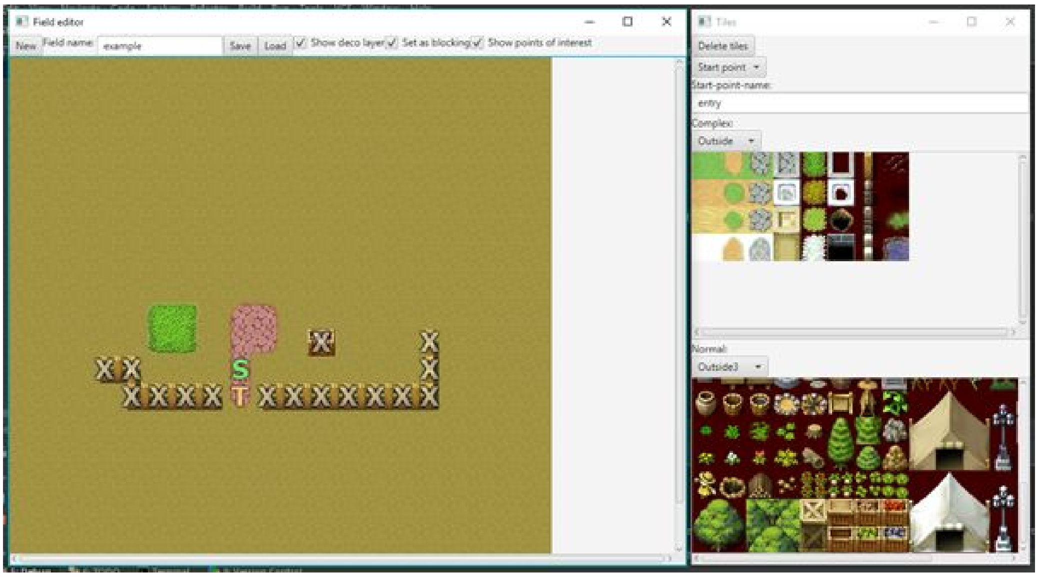
Thief

Ambush: 50% critical chance to attack with double damage

Mutilate: Two attacks
Execute: More damage when the target has health points under 25%
Shuriken Toss: Low damage on all targets

In the next sprint we will implement the functionality of the “Items” and the “Run” button.

Editor tutorial (part2):



Field editor

New | Field name: example | Save | Load | ☑ Show deco layer ☑ Set as blocking ☑ Show points of interest

Tiles

Delete tiles

Start point

Start-point-name: entry

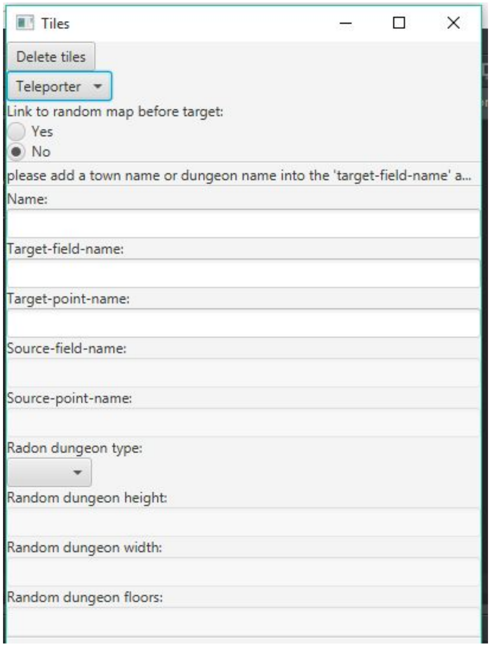
Complex: Outside

Normal: Outside3

First thing changed is the overview of the editor. We have reduced the number of windows from two to three to make it more convenient to use.

The left window shows the map. The functions of this window are mostly the same as in the first version, but the tools of the removed third window have been moved to the upper part of the window. Furthermore, it is now able to display points of interest, such as teleport points with a yellow “T” and start points with a green “S”. Finally, it shows which tiles are unable to be walked upon by the player with a gray “X”. These features can be seen above in the map window.

When clicking on the map to place tiles, you can now press the control key to copy a tile or the shift key to edit all tiles at once.



Tiles

Delete tiles

Teleporter

Link to random map before target:
○ Yes
● No

please add a town name or dungeon name into the 'target-field-name' a...

Name:

Target-field-name:

Target-point-name:

Source-field-name:

Source-point-name:

Radon dungeon type:

Random dungeon height:

Random dungeon width:

Random dungeon floors:

The right window has been completely overhauled from the first version.

The teleport points are more complex than in the first version of the editor as now you can choose if whether or not the teleport point should be set on a random map. If you choose to have a teleport point set to a point on a random map, you will need to specify how the game should generate the random dungeon. The first three fields will remain the same, but may have slightly different meanings:

- “Name”: This simply defines the points name

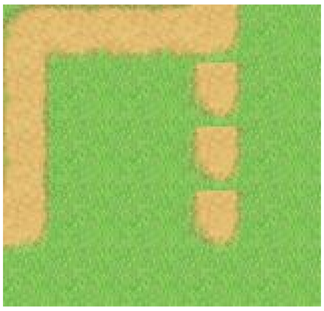
- “Target-field-name”: This defines on which field the teleporter should teleport; or if random is set to yes, defines the target field after the random dungeon
- “Target-point-name”: This defines the target start point within the target field

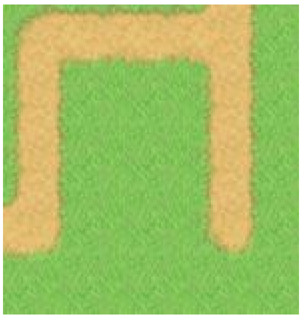
Along with these three fields, you will have to fill the following fields if the teleport point should link to a random map before the target:

- “Source-field-name”: The name of the field, where the player goes if he decides to return from the dungeon without completing it (Should be the name of the current field, where the teleporter lies)
- “Source-point-name”: The name of the start point on the source field
- “Random dungeon height”: The height of a random dungeon floor in number of tiles
- “Random dungeon width”: The width of a random dungeon floor in number of tiles
- “Random dungeon floors”: The number of floors for the random dungeon
- “Random dungeon type”: The type of the random dungeon

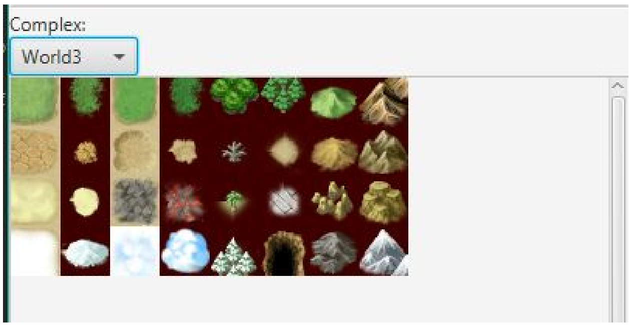
There is currently only one test dungeon type, which will have the dungeon look like a basement. Other types to be implemented may be things like wooden, mountainous, cave, castle, prison, etc.

Unlike normal tiles, which are still present further down on the left window, complex tiles will be rendered differently depending on their neighbours. That means, if you create a path, mountains or fences with these tiles, they will connect to each other naturally.





Left picture: this is how the path looks like when you add it to a map.
Right picture: this is how the path looks like after the next render (for example you can disable and enable the deco layer which will trigger the editor to re-render the whole field.)

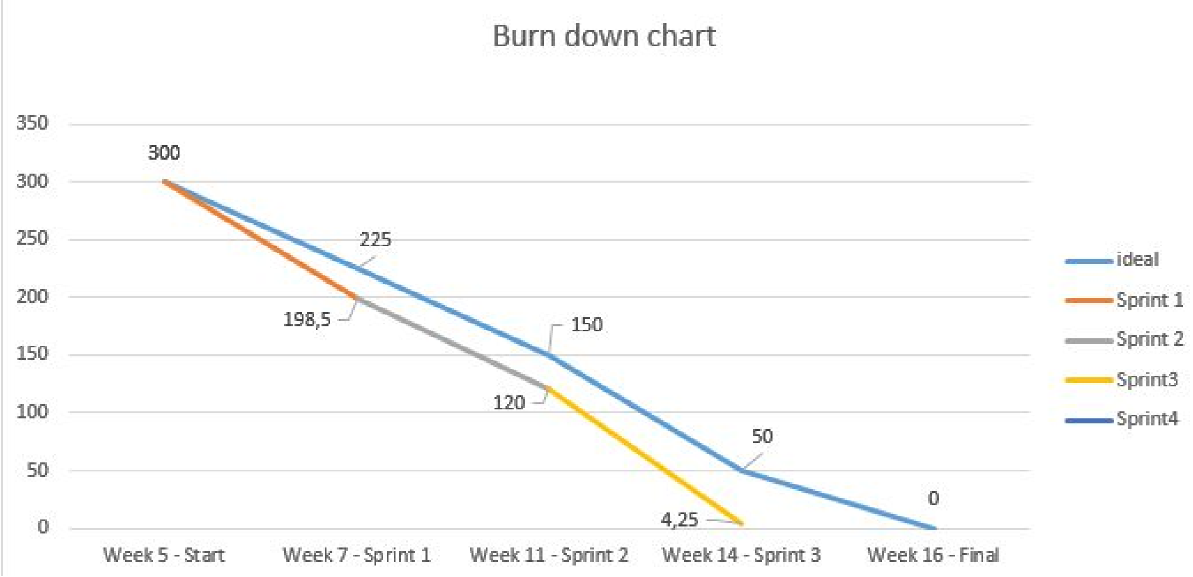


Complex:

World3

Problems encountered:
Developing of the clean transition between tiles and generating of a random world was more time needed than firstly anticipated. There were difficulties in trying to use an FXML file with its controller class with the battle logic, but this issue was eventually solved.

Burn down chart:



As seen in the burn down chart we worked more hours than expected again, a total of 295,75 hours in 9 weeks. Also seen in the journal is that the hours aren't evenly distributed but which probably results from the tasks that everyone was given. Some of them were easier to implement as we thought so we needed less hours to complete them. All in all we do think we are on a good way to finish our project in the next 2 weeks.

Projected Progress:
Most of the editor features are done. Tile chunks are implemented for three different chunks: a four by four tree section, a two by two hole and finally a two by two waterhole. Furthermore, this sprint we were able to implement a clean transition feature that allows for smooth transitions between different ground types such as grass and sand. The clean transition between different grounds is created through a feature called complex tiles. Those are rendered differently depending on their neighbours (see the editor tutorial for pictures). The "generateDungeon" method is now complete with rooms. From here on it only needs to be filled with content, for example more types of dungeons. The worldmap is now generated randomly by piecing together large pre-defined map chunks. For now we have test map chunks, but in the next days we will create the map chunks that will be used in the final presentation. We will be creating these by using the editor.

One of the things that needs to get done is to finish up the battle system with connecting the implemented logic with the GUI. So the base is set, we only need to use the created methods.

Conclusion:

It was a good idea to write a task sheet for the third sprint. So everyone exactly knew what he/she had to do. While not everything on the chart got done, everyone knew what they had to do this sprint. Our main focus for the final sprint will be finalizing the battle system. We will also be trying to implement the shop and finally after those two features are implemented we will deal with any remaining bugs and problems. We are very confident we will be able to implement all the major features before the final presentation.