TECHNISCHE UNIVERSITÄT MÜNCHEN

Computer Graphics and Visualization Group
Computer Games Laboratory WS 16/17

Pandemia Munich - Interim

Authors: Adrian Philipp, Clemens Kamm, Markus Siglreithmaier

Date: November 30, 2016

Contents

1. Interim Report

1.1. Progress

As stated in the project schedule we started with the implementation of our game right after finishing the physical prototype. By now, we have finished the functional minimum layer and are mostly done with the low target. In general, we are happy with our progress, especially with the user interface design and the graphical representation of Munich. Furthermore, we think based on the work we already done, we can quickly make progress in the development of the game. Nevertheless, they are still lots of things to implement and test. Especially, the part of balancing the game can become quiet a lot of work as we think.

1.1.1. Progress in Layers

In this subsection we will describe the progress of our game in more detail. This will be based on our goals for each layer as we stated in the first chapter. We will mostly focus on the first two layers, but will also mention some aspects of the desired target layer and what needs to be done.

Representation of Munich Our approach to build a realistic representation of Munich is based on the "‘Open Street Map"’ data, which is available for free use. The OSM data format is a form of XML, divided into three main areas: nodes (the most general structure), ways (consist out of nodes) and relations (consist out of nodes or ways). With the help of this data, we were able to rebuild Munich with real data in a procedural way. This topic includes a lot of computational effort and parsing of data, but in the end we think this will give us the best results to realistically display Munich. As you can see in Figure 1, we can now represent Munich in a simple 3d way. There are still not all tags used that OSM provides and the art-style is not yet finished. Furthermore, complex structures are not yet fully renderer (wholes in the polygon). The functional minimum and the low target can be seen as reached in this area. For the desired target there is still missing some visual highlights and especially the visualization of the spread of the virus.

Simulation: Crowd and Virus Based on the approach of using OSM and therefore the possibility of generating a graph of all roads in Munich, we first implemented a version of the A* algorithm. Quickly, we realized that this approach would be to computational heavy to do this for all citizens. Therefore, we decided to follow two approaches. Firstly, a really simple one which just calculates the next step based on a really simple heuristic, and secondly implementing a routing table for each node in the network. As by now we are finished with the simple one and are well into the routing table approach. Nevertheless, there is still a lot of work to do. For example there is the need to enhance the daily routines of the citizens and how the virus is spreading. At the moment this is all done in a really simple manner, but we think that we can make a realistic simulation with the help of the

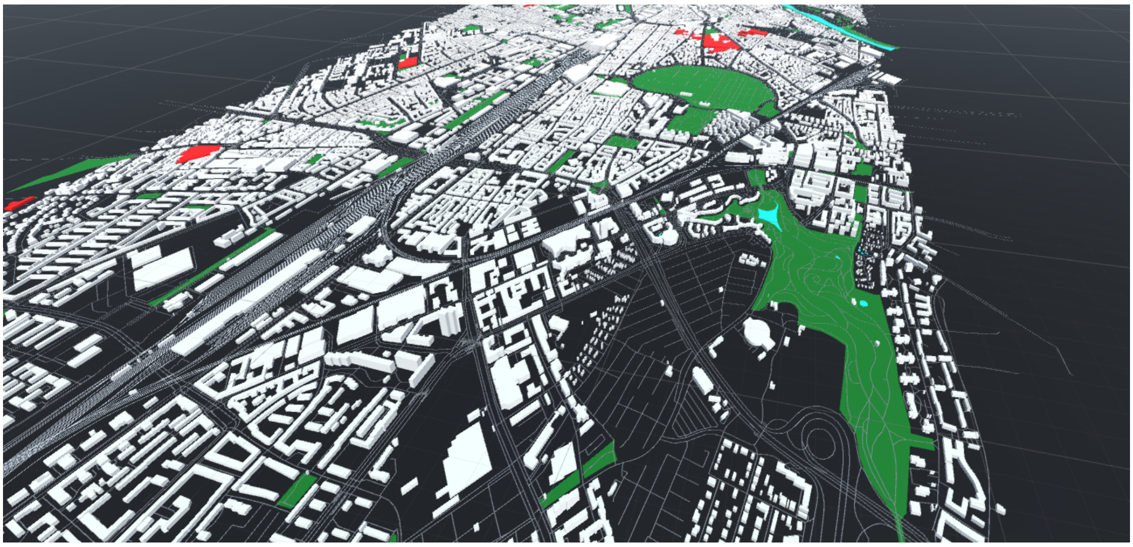

Figure 1 Current version of the graphical representation of Munich.

routing tables. In summary, the functional minimum is definitely reached and nearly the low target. To desired target there is definitely missing the implementing of realistic daily routines and a more sophisticated spread of the virus.

Player Interactions and User interface The user interface design is a very important part of the game because nearly all actions are controlled over the interface. We are very happy with the overall design of it, but there is still room for improvements, especially to give the player more feedback about the current state of the game. To this point we implemented the basic player interactions like sending medical troops to an area, placing detectors and investing research points. More details are explained in the following section. All in all, the functional minimum and the low target is reached. For the desired target, it is now necessary to improve feedback and to include more player interactions with the game.

Resource Management, Research Ladder and Upgrades This is the area where most of the work is still need to be done. In the resource management we made a design revision. Now all player actions just cost research points and there is no money any more. The player can spend it to build detectors, upgrade buildings, do research and find the cure. This is already implemented, but still needs a lot of polishing. The research ladder itself is not yet fully implemented, but the user interface is finished. In the area of upgrading buildings there is still many work to be done. In total, we finished the functional minimum. The low target is mostly done, but still needs polishing and some more work in the upgrading part. The good thing is that this can easily be enhanced once the structure is finished and should be definitely be in a good state at the end of the implementation phase.

1.1.2. Forecast

Overall, we think we should manage to finish our project plan in time. In some areas, we are a little bit behind the schedule, in other areas like the graphical representation and the user interface we are a little bit ahead. At the moment, they are two mayor areas, which could endanger our schedule. Firstly, the simulating of the citizens could become to heavy loaded,

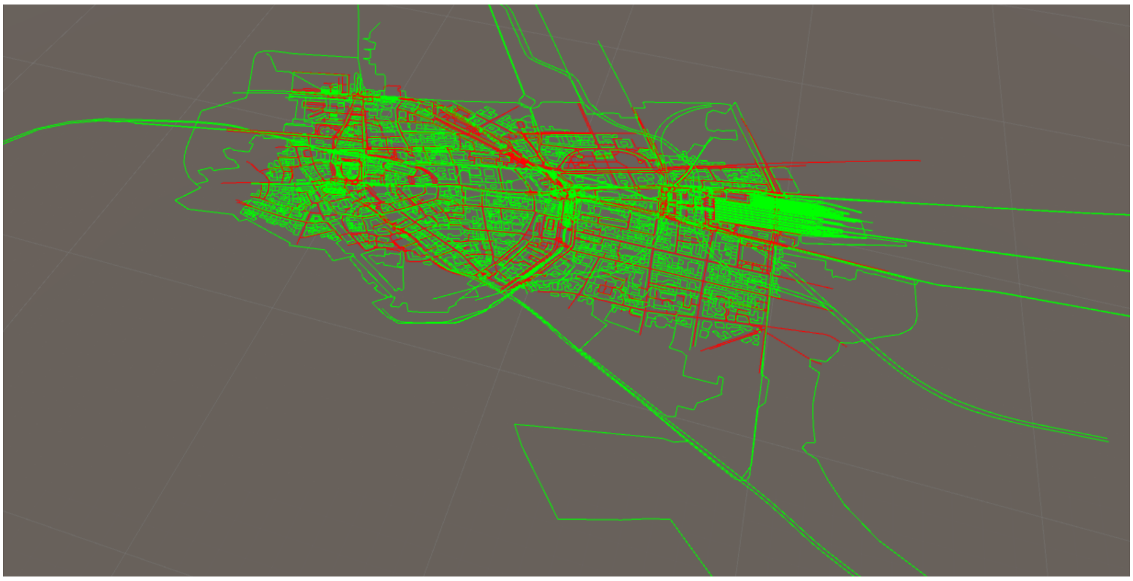

Figure 2 First trail to representing Munich.

especially with daily routines and the sophisticated spread of the virus. This could lead to an unplayable game. To stop this from happening, we are always profiling our game and the frame rates. Furthermore, we need to do a trade-off between performance and simulation in the end. Secondly, the balancing of the game could consume to much time and therefore the game might in the end not as much fun to play as it could be. As a precaution, we try to start the balancing as early as possible and give the player as many feedback as possible.

1.2. Interaction

1.2.1. Control

The player sees the city in a top-down view from a large-scale perspective. The whole game can be played by using a mouse only, but the view can also be panned with WASD or the arrow keys optionally. We implemented common interaction schemes from 3D editors, which fit quite well for our gameplay. The viewport can be adjusted by zooming in with the mouse wheel and panning via right-clicking and dragging the map. All interactable buildings and objects are clickable by hovering the mouse over the specific object in screen space.

1.2.2. User Interface

A large part of the player interaction is done via the GUI. We try to provide a clean and easily understandable interface for the user to be able to focus on the actual game and provide him good overview of the current game state. Our current progress can be seen in Fig. 3 at an early stage in the game. On top we provide show the basic condition of our city in order to allow the player to see how far the virus has already spread. The progress bar visualizes the advancement in the research towards finding the cure for winning the game. In the right panel the player can build new detectors and place it on the map or make progress in the research area using the gained Research Points (RP). Fig. 4 shows the popup menus for hospitals and

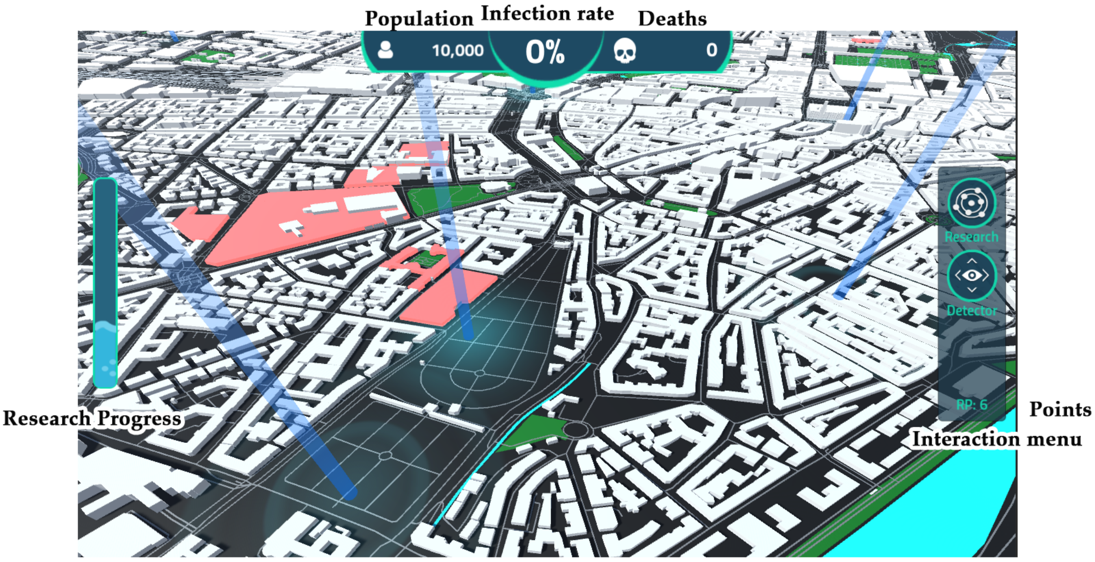

Figure 3 GUI overview: Main HUD of the game including different UI elements: Research bar (left) visualizing the current progress until finding the cure, city status on top showing the current condition of the citizens giving a rough estimation of the game state. The menu on the contains the main interaction control with detector building and upgrading the research tree

the research tree. In the hospital menu you can control your medic troops to send them to waypoints on the map, where they can treat citizens in the specific area. For the research tree we plan to provide the player upgrade options for better virus detection, treatment, etc, but this is currently work-in-progress.

1.2.3. Gameplay

We have made several gameplay changes with respect to our initial proposal after building the physical prototype. The current game lays more focus on finding infected citizens and trying to heal them in specific places using medic troops. Over the time the player gains research points, which are our current form of currency the player can use to build new detectors, make progress in research and upgrade hospitals. Detectors will show all people in a certain radius with their current infection status. The red colored buildings in Fig. 3 are hospitals, which have associated medic troops the player can control. Each troop can be sent to a certain location on the map, where it will treat the infected citizens. For a successful treatment the player will gain additional research points which can be further invested.

1.3. Challenges

One big challenge is to make the game run smoothly. Especially the simulation of the population is a very demanding task for the CPU. Our first idea was to use the A*-algorithm once a citizen chose a goal to calculate a path towards it. As mentioned before, it turned out to be very time consuming to find a path for each citizen that way. Precomputed paths for each possible start and goal combination on the other hand takes too much memory for a network

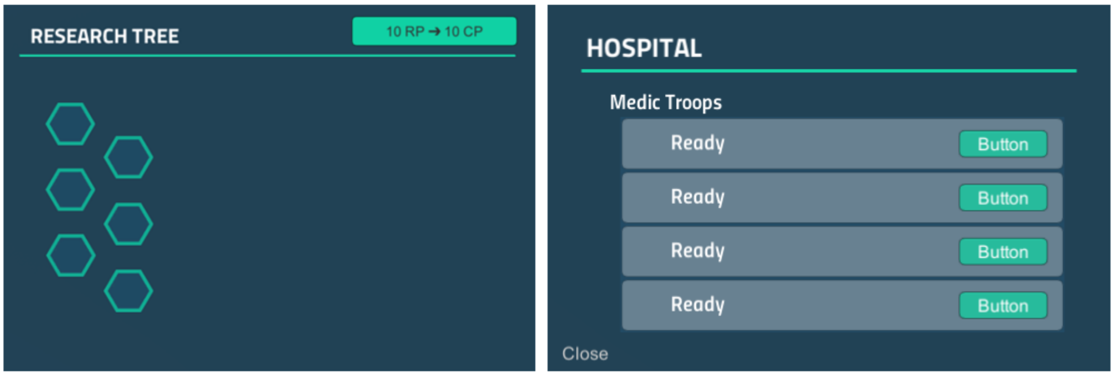

Figure 4 Popup menus: Research tree (left) for unlocking new features and advance in the development of a cure. Hospital menu controls the medic troops, which you can emit for treating citizens.

with more than 50,000 nodes. Because of this, we plan to compute a routing table at each node that stores the neighbor with the shortest remaining way for every goal node. To send medic troops to a position in the scene, it was necessary to find the nearest node to the position of the player's mouse click. Since it costs a lot of time to iterate through all nodes and calculate the distance to each of them, we do the search using some kind of k-d tree.

Another challenge was to understand how the OSM data are organized and how to process them. It turned out, that there are several inconsistencies in the OSM data set. E.g. some faces are stored in clockwise order while others are stored counterclockwise. Also, it took a long time to create the objects at the beginning of the game, so we now load models, which we precomputed beforehand, instead. Since there are many buildings in Munich, which don't change over time, we combined them to big game objects.

List of Figures