TECHNISCHE UNIVERSITÄT MÜNCHEN

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

Pandemia Munich: Prototype

Authors: Adrian Philipp, Clemens Kamm, Markus Siglreithmaier
Date: November 9, 2016

Contents

1. Physical Prototype

1.1. Prototype

The key goal of an prototype is to build something that distils out the core gameplay. This can be done in a software prototype. But often you will start with a physical one using paper, glue, and other materials like from board or card games. Even in a physical prototype, it is important to include the core game mechanics.
For our project we build such a physical prototype. In the following subsections, we will explain how we transformed our idea to a physical prototype and how it is played.

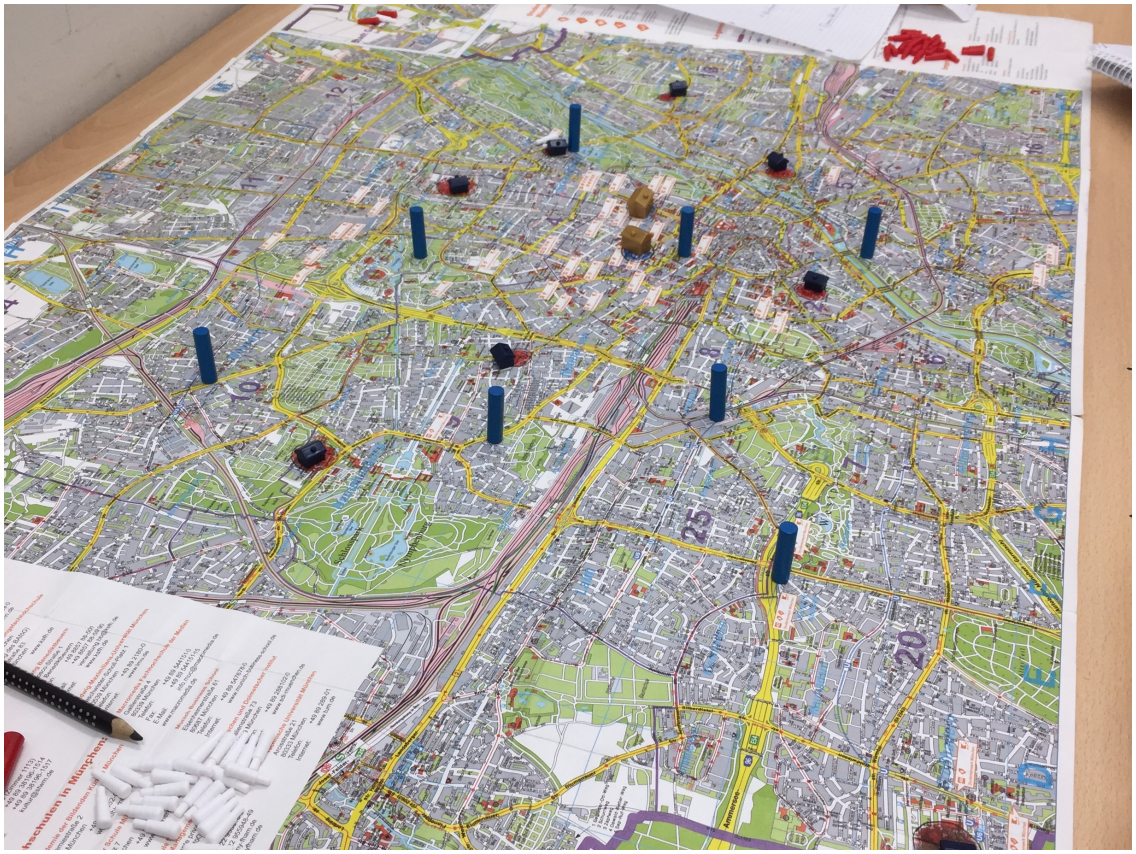
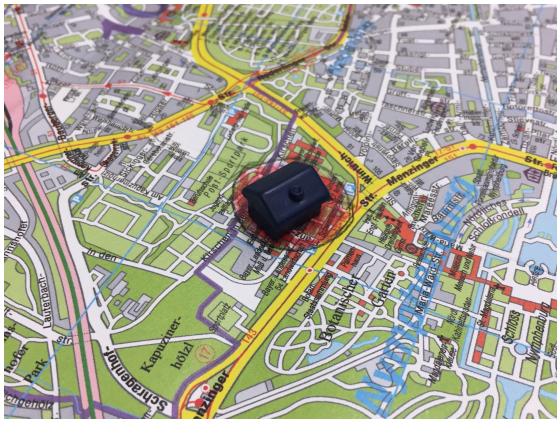


Figure 1 Shows the playfield after some turns.

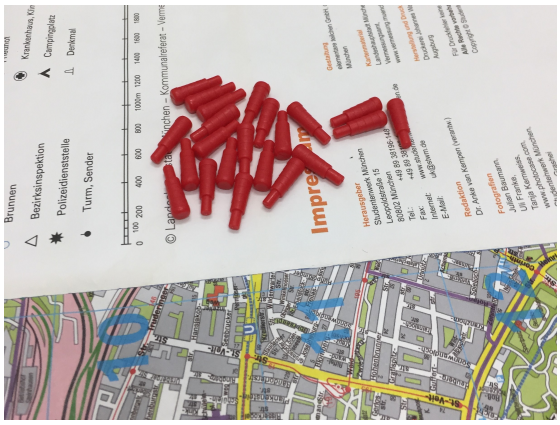
1.1.1. General

As stated in the game proposal, we want to build a game in which the player fights a virus that is spreading in Munich in real time. In the game prototype we changed the gameplay to a round-based one to focus more on the actions of the player and the AI. The technical achievement for our game is to simulate the citizens and the spread of the virus realistically. The prototype cannot achieve a realistically simulation therefore we changes this to a random approach, similar to the description of our functional minimum in the game proposal chapter. Based on this changes, we created the game prototype.

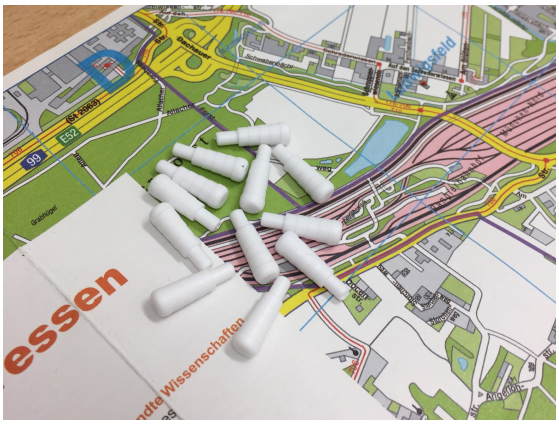
(a) Detector that can be placed by the player.



(b) One of the hospitals that are on the map.



(c) Small red rods that are used for the infected.



(d) Small white rods for the research points.

Figure 2 Several items that are used in the prototype.

1.1.2. Materials

The most important material is a city-map of Munich, which is used as a play field. On the map we highlighted mayor hospitals in the area of Munich. Furthermore, we used the grid of the map for tracking the position of infected entities. Entities are represented by small red rods. Detection devices can be placed on the intersections of the map grid and are represented by blue rods. A dice is needed for different actions in the game for example which action the AI will perform. Lastly, small white rods are needed to display the amount of research points.

1.1.3. Game Rules

This subsection describes the game rules of the game. How the player can lose and how they can win the game. Furthermore, all player and AI actions are described. We will begin with the start of the game.

Game Start

The game starts with the initial positioning of 10 infected entities on the city map. Randomly each entity is allocated with a position. The position is based on the grid of the map, for

example D5 or E4. The player does not know where the infected are. They are only visible for the AI on a sheet with the positions. Now the player has in each round 3 "‘action points"’ which they can use for different actions. The goal for them is to find the infected entities, bring them to the hospital and heal them. The player can identify infected through the placing of detectors on the map and the manual searching of fields. If the player has used all there "‘action points"’, it is the turn of the AI. The AI can move, create new and kill entities.

How to lose? How to win?

The player loses when more than 10 entities are dead. They can die due to the actions of the AI or if the player tries to heal them. Another possibility to lose is when more than 20 entities are infected at the same time.
The player wins when they heal all infected entities. They can heal entities if they detect them and then heal them in the hospital. Furthermore, the player can win if they find a cure against the virus. A cure is found when the player gained 15 research points. Research points are gained if a entity is healed or dies in the hospital. Per entity one research point is gained.

Player Actions

The player has in total five different actions they can choose from. Each action costs "‘action points"’ (AP). The player has 3 AP per round.

- Place a detector on the intersections of the map grid: A detector can identify infected entities that stand in one of the adjoined grid fields. Entities in this area are no longer invisible and will be marked on the map via a small red rod. (2 AP)
- Search grid manually: The player can search three grids. If an infected entity is found, it will be marked on the map via a small red rod. (1AP)
- Hospital transportation: Bring a detected entity to the nearest hospital. If the hospital is full you cannot transport it to the hospital. (1AP)
- Try to heal an infected entity: If a entity is in a hospital, the player can try to heal the entity. Based on the throw of the dice different things happen: (1AP)
  - 6: Entity is healed. Player gains one research point and one additional AP. (H)
  - 3,4,5: Entity stays alive. (A)
  - 1,2: Entity dies. Player gains one research point. (D)
- Update a hospital: The player can update a hospital of there choice. (2AP)

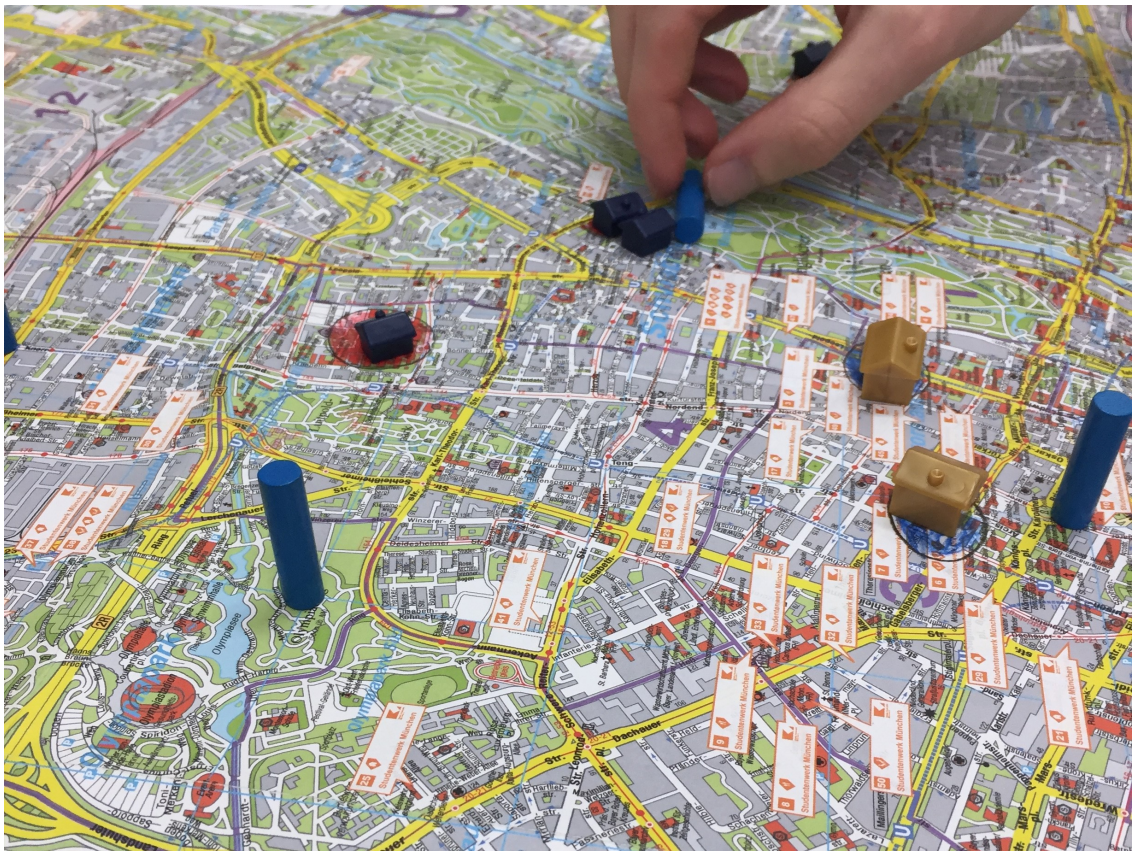


Figure 3 Player places a detector on the map. The hospital by his hand is already at stage 2.

AI Actions

The AI has three actions it can do. Which action is performed, is decided with the throw of a dice. The following listing shows the different actions with the corresponding dice number.

- 1,2: One or two infected are placed into a field in which already an infected entity is placed.
- 3,4: The AI can move three or four entities on the map. One entity can move 2 fields.
- 5,6: The AI kills randomly one infected entity.

Hospital

In the hospital infected entities can be moved and only there they can be healed by the player. Based on the stage of expansion they differ in capacity and how good they can heal infected. The list shows the attributes for the different stages.

- Stage 1: Capacity is 2. Normal statistics on healing probability.
- Stage 2: Capacity is 3. H:6; L:2,3,4,5; D:1
- Stage 3: Capacity is 4. H:5,6; L:2,3,4; D:1

1.2. Experience

Our overall experience throughout playing the prototype was quite positive, but we came across several balancing issues during playtesting, which we tried to at least partially fix in later iterations.

In our current setup we are starting with ten infected citizens to make it easier for the computer player to keep track of all of them. Unfortunately, this number turned out to be much too small for the high resolution grid of our map. Because of that, it took a long time until first infected persons were detected. Also, it was very difficult to congest hospitals, making it useless to update those.

Initially, we limited the movement of infected persons to one cell. However, this decision made the game to static and we needed to increase their mobility. In this first iteration of our prototype, it was also possible to use action points for research. Since this system could be easily be abused - the player was guaranteed to win by spending all action points on research - we changed the research system to what was mentioned in the game rules section.

We learned from these experiences, that finding the right balance between gameplay elements is very difficult. This applies in a prototype, which omits several parts of the game, so this will be a big challenge in the development of the final game.

1.3. Design Revisions

In this section we will discuss the changes of our game concept from the 1st milestone with regards to the presented physical prototype and the feedback we received for our initial concept.

1.3.1. Prototype

We tried to transfer the insights gained during playtesting and creating our physical prototype to our game idea concept. As we only modelled the core part of our game concept we can't review all gameplay features.

The virus detection turned out to play a larger factor in our gameplay and we want lay stronger focus on it (detectors). Alongside with this change comes a scale adjustment for our map as want to reduce the virtual representation of Munich to the city center. In similar fashion we transfer to more area based spreading and detection of the virus than handling it separately for each house.

1.3.2. Critiques

For our initial proposal we received a lot of feedback from our fellow students (big thanks!). The overall feedback was pretty good and we tried to extract the common aspects mentioned for improvement:

Satisfaction mechanic Our satisfaction level system for the citizens was frequently mentioned. We have to further elaborate this topic to achieve a solid part of the game concept, especially take it more into consideration for balancing to enforce a dynamic gameplay.

Learning based AI The virus AI represents a major aspect of our gameplay in terms of a balancing and has a large impact on the success of our game. The feedbacks emphasized this aspect multiple times and suggested different behavior strategies which we consider for our upcoming implementation.

Research tree We got many interesting suggestions for the improving our research tree, which could vastly improve this part of the gameplay. Having different virus types and making the research depend on it is something we already consider. An different idea, which came up, was to add minigames for advancing the research.

Crowd behavior Another mentioned point is the dynamic behavior of the crowd on the current game state (e.g grouping for riot at low satisfaction). This would be a desirable target for us as it would make the overall game more dynamic and fun.

Beside points listed above there are several good other suggestions, which probably lay out of the scope for this lab like multiplayer or a story mode due to the limited time. Even tough we would really like to implement something similar.

List of Figures