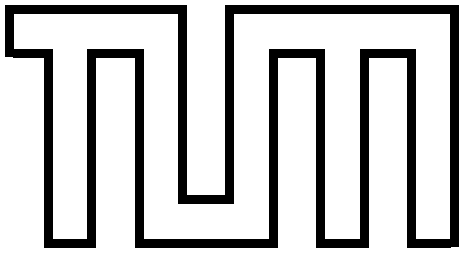


Technische Universität München

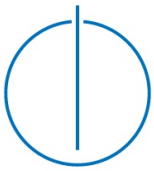
Department of Informatics

Master Practical Course Games Engineering
Augmented Reality

Rune TD

Serious Celtic Augmented Reality Tower Defense Game

Summer term 2018



Authors

Stefan Stark, Florian Hösch

Dennis-Florian Herr and Michael Felleisen

Supervisors

Prof. Gudrun Klinker, Ph.D.

Dipl.-Inf. David A. Plecher, M.A.

Contents

1 Introduction

"The year is 50 B.C. Gaul is entirely occupied by the Romans. Well, not entirely... One small village of indomitable Gauls still holds out against the invadors. And life is not easy for the Roman legionaries who garrison the fortified camps of Totorum, Aquarium, Laudanum and Compendium...", Robert Steven Caron. [1]

Similar to the famous opening quote from the Asterix comics, our game is set in the middle of the last century before Christ. The Celtic culture is at its peak and shaped Central Europe and the British Isles during the last centuries. [2] But conflict arises between the Celts and the Romans when Gaius Julius Caesar is leading his military campaigns northwards to conquer Gaul, known as the *'bello Gallico'*.[1]

During his campaigns, Caesar encounters several *'oppidum'*, fortified Celtic villages or cities. [4] The player is taking control over the defenses of a small Celtic *'oppidum'* and has to defend the village using powerful Celtic runes. After the village is fortified enough to fend of wildlife, the player has to fend of wave after wave of the Roman campaign to conquer his village. Who will come out on top? The organized roman empire or the naturalistic Celts with exceeding knowledge in ancient runes?

The game is supposed to teach players the meaning of ancient Celtic runes in a playful and engaging way. While playing the campaign players gradually unlock runes, that they must use with physical, printed markers[2] to control the augmented reality world. When a rune is unlocked the player gets educational information about the rune and an explanation of its effect in-game. The original meaning of the rune is related to their in-game effect to improve the learning experience by creating coherence between the educational information and the game experience.

We chose knowledge about Celtic runes as the educational goal, because the Celtic culture is an important part of Europe's history, but little is taught about Celts and especially about runes in the general education system. So even though most people might have heard about 'runes' through games, there is little to no knowledge about their original usage and meaning.

[1] Celtic and Gallic are for our purposes synonyms. Celtic comes from the Greek term *'Keltoi'* whereas Gallic comes from the Latin term *'Galli'* and refer to the same folk. [3] But as Gaul was also used by the Romans to describe an area that is mostly equivalent to what is now France, we stick to the terms Celts and Celtic instead of Gauls and Gallic to describe the people and culture that spread throughout Central Europe.

[2] Our system of rune markers is explained in Chapter 2.

The following chapters are meant as a documentation for our prototype implementation. We start with describing the game design choices, followed by an explanation of the mechanics used to play the game. Afterwards we discuss problems during development, especially in regard to augmented reality and present a short user study of our prototype as presented at the TUM Demo Day of the summer semester 2018. Finally we discuss future work that was out of scope of the practical course.

2 Game Design

This chapter covers our game design in regard to game play and the educational aspect.

2.1 General Concept

Our game design evolves from the idea of a serious game utilizing augmented reality techniques available on larger mobile devices, i.e. tablets or phones with a display size of at least 7 inch. The educational goal is to convey knowledge about Celtic runes and culture through an engaging and entertaining game experience. Learning is supported by a connection between the matter and the game mechanic, so the player improves his skill in the game by mastering knowledge about runes and vice versa.

The game is designed as a classical Tower Defense (TD) game. Players build towers to defend their city from increasingly difficult hordes of enemies. This is explained in more detail in Section 2.3. We use augmented reality to give the player control over the towers and buildings with printed physical markers, our 'rune markers' (cf. Section 2.2). This is meant to increase the learning success by bringing the runes into focus while providing a refreshing game experience through novel control mechanics. Knowledge about runes and Celtic history is presented gradually using a game campaign, where the player unlocks new runes and additionally information after increasingly more difficult levels. This is thoroughly explained in Section 2.4.

Finally we based the style of the game on available historical data about Celtic cities, defenses and enemies, which is covered in Section 2.5.

2.2 Runes and Rune Markers

As the primary goal of playing the game is to learn about Celtic runes, the runes are the core of our game experience. Therefore we split the game into two distinct phases: A preparation phase where the player interacts with physical representations of the runes, our rune markers, and a game phase where the player interacts with the augmented reality device.

Our rune markers are printed out representations of runes that each have a unique effect in the game and need to be properly positioned by the player. Figure 2.1 shows a rune

marker for the *'Algiz'* rune, which is used to build and position towers in-game. As a player can use some runes multiple times, the rune markers have an additional unique boarder that enables the augmented reality tracking to distinguish similar runes.

As the basis for our runes and rune markers we used the 24 runes of the *'Elder Futhark'* [5], the oldest form of the runic alphabets. Almost[3]all of the 24 runes have a special meaning in the game that closely relates to their original meaning. For example the Algiz rune has the meanings *'Elk', 'Protection', 'Defense'* [6] and is used to build towers. Though, it must be noted here though that runes can have multiple meanings and the meaning is not always absolutely clear. Rieckhoff and Biel point out that there is not Celtic historiography, literature or religious writings. [7] The absence of a large basis of usage, makes interpreting runes especially difficult. A complete table of the runes we used, the original meaning we used as a basis and their usage in-game can be found in Appendix TODO.

Figure 2.1: An Algiz rune marker for building towers

To properly play the game a player must recognize a rune and it's function in-game. The player has as much time as he needs to study, recognize and properly place the runes in the preparation phase.[4]In combination with the coverage of most runes in-game and a close relation of their traditional meaning to the in-game usage, this measures support the learning success of the player.

2.3 Tower Defense

We use the basic and popular tower defense concepts for our core gameplay mechanics. Towers are manually placed on the map (through the position of the rune markers relative to the augmented reality map) and shoot incoming waves of enemies. The enemies have to be killed before they can reach the village, which is positioned with the *'Mannaz'* rune marker and the anchor point of the augmented reality map. If enemies reach the village, players loose health. When health reaches zero, the game is lost and the player has to start over.

Players can upgrade their towers by placing buff runes close to their tower rune. For

[3]Unfortunately, we had to skip some runes as there was no reasonable way of mapping the rune to an in-game functionality.

[4]Though this can help him learn the runes, this was also a necessity due to the nature of augmented reality on a tablet computer.

example the *'Kenaz'* rune adds fire damage to the tower it's linked to. Additionally players can earn gold by building farms and harvesting them during the game phase, which adds additional interactivity to the game phase. The gold can be used to buy additional lives during the preparation phase.

2.4 Campaign

Because of the sheer number of runes the player has to learn before he is able to actually play the game, we slowly introduce the runes step by step in a campaign/ tutorial mode. That way, no one is overwhelmed by too many different rune symbols, their meanings and their meanings in-game, but is able to learn in small manageable steps.
After playing the campaign, a free-play mode is unlocked to offer a challenge for more experienced players without playing the campaign again.

Way points change per level. player has to adapt.

2.5 History

Manching nachempfunden

3 Implementation

Unity. Vuforia extended tracking. Voxel Engine. Marker generation. Map placement. Upgrade handling. marklight (+rich text).

3.1 Marker Generation

Since we chose Vuforia for marker tracking, all markers had to be unique. The game design however requires multiple markers for each rune, as most buildings and upgrades can be in play multiple times. Furthermore the markers have to be readable and unambiguously assignable to a specific rune by the player.

Our solution was the marker design seen in game. The rune represented by the marker is placed in the center and fills most of the space. In order to make them unique, the marker borders are seamed with small versions of all 24 runes in random order. Because many markers had to be created like this (4 per rune) we created a Java Script program to do the random placement of the rune images. This design has the additional advantage of always giving the player an overview of all runes, as well as making the correct orientation of the center rune more obvious.

3.2 UI

To create the games' user interface we used MarkLight.[8] MarkLight is a Unity framework offering a declarative design language similar in syntax to HTML. It can be used to create user interfaces with code only, bypassing the need to drag and drop UI-elements in the Unity scene view.
In order to further separate the UI from the core code, most text is handled using separate JSON files. This includes primarily the Wiki, as well as the campaign flavour text. Fortunately MarkLight supports the Unity Rich Text markup format, so we used it to make the partly huge walls of text easyer to read.[9]
At this point it should be noted, that all the content presented in the ingame wiki is taken from the wikipedia page corresponding to the rune, as well as the overview page of the elder futhark.[5]
Another important job the UI had to fulfill was showing the player if one or several of

the markers has lost tracking. For this purpose every marker detected by Vuforia gets a colored sphere attached.

- green means the marker is actively tracked
- yellow means the marker is tracked using extended tracking only
- red means tracking was lost or position is invalid

It is important, that all runes the player wishes to start the level with have at least a yellow tracking level.

3.3 Rune Combining

There are 3 types of upgradeable runes: towers, farms and the village rune. All other runes are either elemental (which are used to change the damage type of towers) or upgrade runes that change the behavior.

Every upgrade rune has a list of valid upgradeable runes which it can connect to. Out of that list it will select the closest one if it is not out of range. This is visualized by a linerenderer.

Every upgradeable rune will collect a list of all connected upgrades when the START button is pressed so they are not changed while the game is running. This list will be consulted to identify the behavior and stats of the upgradeable runes.

3.4 Voxel Framework

We decided to use a Voxel Framework as a base for our project for multiple reasons:

- We had already started to build a Voxel Framework so we were able to reuse most of the code.
- It is easier to generate nice looking Assets for a Voxel style game.
- Objects from the Voxel Framework can be changed at runtime (e.g. creating a farm on the terrain changes the grass to dirt.)

Most objects that are used in the game, such as the towers or the terrain, are displayed by the Voxel Framework. The data for every object is stored in a 3D Grid. Every data point stores the information about the material at its position. From this grid we generate one optimized mesh that uses a Texture Atlas and UV coordinates to display the color information of the objects.

The terrain is split into multiple chunks to reduce the complexity of the mesh optimization and therefore improve the performance. Every chunk has its own mesh but the all share the same materials and textures to reduce the amount of draw calls.

Besides the rendering the Voxel Framework also includes basic terrain generation which we adapted to our needs for this project. Furthermore it supports growing plants on farms.

4 Encountered Problems

4.1 Markertracking

Since we were creating our project in Unity we decided to use Vuforia as our Augmented Reality tracking library, but the performance was not sufficient in the beginning.
The tracking was not stable enough for the amount of markers we intended to use, so we had to change a few aspects of our game.

The first thing we scraped were the enemy path markers. Instead the path spawns relative to the base rune. This solved stability issues as well as problems with very efficient path patterns made by the players. By removing this player variable we had an easier time creating a balanced campaign mode.

The second thing we improved were the markers themselves. The first hand-drawn markers did not bring the image recognition performance we needed.
Therefore all the markers we used in our project were generated by a script. In the center of each marker is the central rune the marker is representing and the outlines of the marker consists of the whole rune alphabet in a random order. Because of that, we could have multiple different instances of the same rune markers without using e.g. plain numbers.

4.2 Marker printing

The first time we printed all of the, close to 100 markers, we used a very simple printer and very basic printing paper. This resulted in white vertical lines across most of the markers, which was very bad for the image tracking process. Moreover, after using the markers a few times, they started to bend which made the tracking even worse. After that we used an industrial printer with very thick paper and got much better results in tracking and marker durability in general.

4.3 ARCore/ARKit

ARCore and ARKit was anounced in August 2017 and promised heavy performance boosts in the markerless tracking technology. Those advances in tracking stability could have been a great benefit for the game. Unfortunately, the support release for phones that are not flagship level phones (like the Galaxy S9 /iPhone 8) is still in the making. Dynamic environmental lighting and AR plane detection were the features we especially looked forward to.

5 Userstudy

Our userstudy took place on Demo Day what allowed us to get a more or less diverse group of 12 test users, that were willing to take some of their time to fill in our questionnaire.
The majority was very familiar with AR/VR which is most likely due to us handing out the questionnaire during Demo Day, the most relevant event for games engineering students.
On a scale from 1 (bad) to 5 (good), previous knowledge about Celts was self-rated at 1.92. After playing the game the self-rating on how much the test users learned about the Celtic culture was at 2.5. The test group was able to score 3.3 out of 6 possible points in the drawing quiz part, where they had to recognize 3 runes and write their meaning/name and 3 different rune names, where they had to draw the corresponding rune.
Overall, most of the testers liked the general tower defense concept with a score of 3.9 and liked our game with a score of 4.1. So the Celtic aspect increased the fun the testers had while playing the game.

5.1 Suggestions for improvement

The most frequent suggestion was, to reduce the amount of unlocked runes for every level in the campaign mode. For Demo Day we did not create a fully fleshed out campaign. We tried to keep the campaign relatively short to make it playable in a shorter amount of time, to make the game not as time consuming as it would be in the final version.
Furthermore, the marker-tracking performance was an issue for some testers despite our efforts to make it as stable as possible. Possible ways of improving AR further would be using a better device (computing power and better camera) and/or using advanced AR-libraries (ARCore/ARKit).

6 Future Work

6.1 Free Play Mode

The Free Play Mode was supposed to be played after completing the campaign. It would be designed to test and strengthen the knowledge gained about runes, their meaning and how to use them most efficiently.
To achieve this, the player would fend off randomly generated waves of enemies with all runes unlocked. Before each wave starts, information about the type of attackers is displayed in form of rune symbols and/or names. If the player was shown the runes "Ehwaz" (speed) and "Kenaz" (torch) for example, he would know to expect a wave of fast enemies with fire element and could adjust his strategies accordingly.
This mode could easily be implemented as a two player mode as well, with one player creating the waves of attackers using markers and the other fending them off.

6.2 Wearables

For a game like this, wearable Head Mounted Displays (HMD) could enhance the player experience quite a bit. HMDs like the Oculus Rift, the HTC Vive or Google Cardboard in combination with the Unity engine are quite easy to implement. Player input on the other hand would be more difficult.
Using a simple mouse would break the immersion a lot. Using a controller would be acceptable, but the game requires the player to use their hands to place runes so the player would have to put down and pick up the controller constantly. Gesture recognition would be the optimal solution but is very difficult and time consuming.

References

[1] Robert Steven Caron. *Beginning of most Asterix comics*. URL: http://comedix.de/lexikon/db/vorwort.php (visited on Mar. 13, 2018).

[2] Bernd Liermann. *Herkunft der Kelten und Lage des keltischen Kulturkreis*. URL: http://www.antikefan.de/kulturen/kelten.html (visited on Mar. 13, 2018).

[3] Sabine Rieckhoff and Jörg Biel. *Die Kelten in Deutschland*. German. ausgeliehen TUM TB Stammgelände GES 410f 06.2004 A 159. Stuttgart: Theiss Verlag, Sept. 2001, p. 18. ISBN: 9783806213676.

[4] John Collis. "Die Oppidazivilisation". German. In: *Das keltische Jahrtausend*. Ed. by Hermann Dannheimer and Rupert Gebhard. 2., erweiterte Auflage. Vol. 23. ausgeliehen TU München Universitätsbibliothek 94 A 484. Prähistorische Staatssammlung München und Verlag Philipp von Zabern, Mainz, 1993, p. 102. ISBN: 3-8053-1514-7.

[5] Wikipedia. *Elder Futhark*. URL: https://en.wikipedia.org/wiki/Elder_Futhark (visited on Mar. 15, 2018).

[6] Wikipedia. *Algiz*. URL: https://en.wikipedia.org/wiki/Algiz (visited on Apr. 9, 2018).

[7] Sabine Rieckhoff and Jörg Biel. *Die Kelten in Deutschland*. German. ausgeliehen TUM TB Stammgelände GES 410f 06.2004 A 159. Stuttgart: Theiss Verlag, Sept. 2001. Chap. Geld und Schrift, p. 217. ISBN: 9783806213676.

[8] Ex Makina AB. *MarkLight Homepage*. 2017. URL: http://www.marklightforunity.com/ (visited on Apr. 9, 2018).

[9] Unity Technologies. *Unity manual entry of rich text*. URL: https://docs.unity3d.com/Manual/StyledText.html (visited on Apr. 9, 2018).

Appendix

A Appendix A

a