# Conceptual design and development of a serious game for learning topographical maps

## (Developing a digital learning game from an idea to a prototype)

## (Tõsimängu kontseptuaalne disain ja arendus topograafilise kaardi õppimiseks)

# Abstract

There is too little or no digital learning material about topographical maps which use techniques of repetition and instant 3D feedback. This paper introduces a digital learning game prototype that satisfies these needs and also provides description of the building processes.

# Introduction

# Theoretical background

## The need

“The Estonian Lifelong Learning Strategy 2020” and its program “A digital focus in lifelong learning” encourage creating digital learning materials and making them available in order to support the teachers’ and students’ digital competences (Government of Republic of Estonia, 2014).

* Teachers are interested in digital orienteering materials (the paper with Tartu University master student thesis.)
* Merging of subjects
* There is a need for digital learning materials (PROVIDE ARTICLES)

## The problem

* There is too little or no digital learning material which teaches topographical elements with instant 3D feedback.
  + Similar games exist, but without having both, drawing and exploration, options in them (Examples).

## The goal

* To create a topographical map learning game prototype.
  + Learners could instantaneously see how the contours (the lines that join the points of equal elevation) work.
  + Repetitive action benefits memorizing the symbols (FIND ARTICLES)
  + It is easier to start orienteering in a familiar environment (FIND ARTICLES).

# Methodology

* Design based research methodology was chosen in order to: get more information and practical knowledge on creating a digital learning game; find out if and how the needs of the target group shape the outcome; experiment which patterns help developing a digital game.
* To achieve the goals, the research was divided into the following steps:
  + research on other similar products
    - developing a unique prototype is more motivating
  + research on the needs of the target group
    - Lack of need for the prototype decreases: interest of possible sponsors for future development; buyers interest; and the motivation of development.
  + The research on what existing data can be used to develop the prototype
    - Regio, Maaamet, digital topographical information
  + Developing the prototype.
* To keep track of all the processes, hardships, successes and results, it was decided to record the progress to a diary.
* Development was decided to be conducted with the game development program “Unity”. An alternative is “Unreal Engine”. Unity seemed more appropriate, because… (PROVIDE STUFF: mobile support, mac platform, aesthetics, scripting possibilities, blueprint possibilities, personal goals (enhance C# instead of blueprinting, because companies hire more people with programming skills than blueprint skills (PROVE THIS!!!)))
* Since the development was planned to be conducted by a team of one person (the author), who has almost no experience with Unity or programming language C#, there was a high risk of not being able to deliver the prototype in time or the prototype would lack significantly in core functionality. Although some risk was reduced by the author’s intermediate programming skill in PHP and javascript language, there was a strong need to prioritize and to form a strategy in order to reduce more risk level. Thus the most suitable methodology to use was the Spiral Methodology (<http://www.itinfo.am/eng/software-development-methodologies/#chapter12>) and the MoSCoW method was used to set standards for the prototype.
* Considering the lack of experience, it was expected to confront numerous problems during development. For the most cases the PDCA (<http://www.problem-solving-community.org/experts-corner/posts/thibaut-castanet-1>) method was used to overcome difficulties. On one case, the problem was a higher mathematics equation which the author was incapable of solving. In this case there was a need to use the 8D problem solving model.
  + Creating team
    - Student studying mathematics, (Gabriel)
  + Defined the problem
  + No containment action needed
  + The root cause was already apparent
  + Defined theory how to solve the equation
  + Searched for solution and also for assistance to get to the solution (stackoverflow, programmer friend)
  + Analyzed the solution (studied the formula and methods of the code) so that in the future, similar problems would not occur or would be easier to overcome.
  + Congratulated the mathematics student and friend for helping.
* Spiral Methodology
* 
  + System requirements needed to be defined as much in detail as possible. Visit to “Regio” to gather intelligence of the use of the data and if there is a possibility to use any existing code.
  + The core mechanism – drawing mode, was developed by setting requirements for each following version:
    - First accomplish drawing a basic static line (not created by mouse movement)
      * Was a bad start, because the line should manipulate the terrain. If there is no terrain, the line alone does nothing. Thus:
    - manipulate one point of a terrain
    - … square shaped area
    - … circle shaped area
    - … custom shaped outline (the line manipulates terrain)
    - … custom shaped area
    - … line made by mouse movement
    - … create GUI button to activate the drawing tool
    - … create exploration mode and movement functions
    - … create GUI to switch between the modes
    - … create functions to cancel drawing when in exploration mode
    - … apply textures, colors to both modes
    - … plan the next version of the prototype (able to draw trees, water, drag-drop symbols, place self, fly-by mode when drawing)
* MoSCoW:
  + Must have:
    - Drawing mode
      * Canvas
      * Tool to draw contures
    - Exploration mode
      * Controls to look and move around
    - Basic GUI
      * Button to activate the line drawing tool
      * Small screen to see the 3D world change
      * Possibility to change between the drawing and exploration modes
  + Should have:
    - Map grid in drawing mode to have a basic idea about the scaling of the drawn objects
    - Authentic textures and colors in both modes
  + Could have:
    - Tools for drawing forest, water and some landmarks
  + Won’t have (this time):
    - Elevation smoothing
    - Sound effects
    - Visually more attractive textures and meshes.
    - Better looking GUI buttons
* To finish a prototype:
  + To have a digital learning material about teaching topographical map (find articles that say there is a need for this kind of product)

What is the most effective way to create the prototype?

# Results

* Importance of research (puts down the path what to achieve) – when finding out in the middle of doing the prototype, that the thing has been made, the motivation drops drastically – the production could come to a complete halt. So during research it’s possible to foresee what could have a negative impact to the production phase.
* Stress from having few time might have a positive effect of showing creative solutions on how to reach the destiny (like the usefulness of the map converter in this project)
* On building prototype:
  + Start with tutorials which you can already use in your code.
  + To achieve a difficult task, start by the simplest (terrain dot, squared area, circled area, area based on mouse movement, etc.).
  + Leave the aesthetics to the last. The functionality is the most important part! (https://medium.com/ux-power-tools/heres-everything-i-ve-learned-from-designing-10-000-ui-screens-as-a-lead-product-designer-7d2810bee810)
* The usability scenario should be the first thing to introduce to the target group!
* “Never tell teacher if they wanted to use a product, but instead tell them if the product was mandatory, what would make the product more pleasant to use”

# Discussion

# Summary

# Checkpoints

Checkpoints:

1. Idea – Ilya’s idea, that there should be a dlg which teaches topographical maps. My own interest and connection to this topic. This led me to make the project as my master thesis project.
2. Research
   1. Putting down how the game should work, what are its benefits, on which academy theories it relies on.
   2. Searching for similar programs (games and simulations) that have been made – in Estonian and English. Outcome: there were no game or simulation that uses drawing method to teach a map; drag-n-drop method exists (<http://www.teacherled.com/2008/06/01/map-maker/>), and also orienteering ([VirtualO](http://store.steampowered.com/app/529020/VirtualO/), [Suunnistussimulaattori](https://www.youtube.com/watch?v=uvBOFBvUZs4), [Skiddaw](https://vimeo.com/78057630)). Only map-maker practices creativity – the learner can make the map on his/her own
   3. Gathering feedback. From geology and military defense teachers – they would not use this kind of program in their lectures. (Probably because there was no usability scenario for them to get acquainted with; and “never tell teacher if they wanted to use a product, but instead tell them if the product was mandatory, what would make the product more pleasant to use”.) Although the teachers of “riigikaitse” would use a program that simulates orienteering in digital environment that is based on real-life locations, there would not be enough time to create the prototype satisfying these needs yet, but it would be possible in the future versions. This put the project to a halt – needed to find if the program could satisfy all the teachers. Final decision – to target the students directly, taking away the responsibility from the teachers -> added tutorial function to the idea.
3. Start of production: Rethinking the time scale – map converter idea. Map converter positive side – that would interest companies (potentially reliable target group); **reverse engineering method** to build the drawing game (saves time to use existing **authentic** data instead of creating entirely new data(-system)). Negative about it - A whole new product with no research information.
   1. Gathering authentic, widely-used data – to see if it would be too difficult to make a converter or a drawing game. Visit to Regio: interest in both (game and converter), suggestions to look up the data in maaamet.ee website.
   2. Examining the authentic topographical data – found out that the data can be read by 3D programs, Unity cannot read it before converting it to readable data. Planned to build the converter in Unity anyway (provide reasons: need to learn the game engine; the engine has prefabricated functions that save time to accomplish anything). Did not seem reasonable to convert data in order to convert the data. Decided to start building prototype
4. Building the prototype
   1. Goal to achieve the most difficult function (also the core functionality of the game) – the drawing mechanism. Because aesthetics should be the last thing (https://medium.com/ux-power-tools/heres-everything-i-ve-learned-from-designing-10-000-ui-screens-as-a-lead-product-designer-7d2810bee810). Start by searching drawing tutorials -> did not get too far with experimenting, because there was no need to actually alter the code -> therefore no progression. **Realized that there is** **no connection watching tutorials, if I cannot apply the knowledge to practice.** Decided to start with manipulating the terrain ASAP.
   2. Searching tutorials, manuals and instructions on how to manipulate Unity terrain data. When got the knowledge, it could be applied to the line drawing code in order to manipulate the terrain. The steps of terrain manipulation: manipulate a point; manipulate a squared area; manipulate an area in a circle; manipulate the area according to mouse movement; manipulating according to drawn line (bringing together line code and terrain code) – makes available amoeba-shaped outlines; manipulating terrain points inside the custom-shape outline.
   3. When finishing the most important function, there was already found theoretical solutions on: how to fill areas (forests, swamps, fields, water); how to drag-drop custom symbols on the map (houses, towers, churches); how to apply line-shaped objects on the map (roads, railways, ditches). Decided to concentrate on GUI.
   4. Creating exploration mode and making it switchable between the drawing mode (adding player controls, functions to stop drawing). Applying texture to the terrain and grass color to the map-drawing-canvas. Creating grid in order to get the idea of the scaling when drawing lines-of-the-same-height. **Thus the prototype is ready** for the writing part.

# References

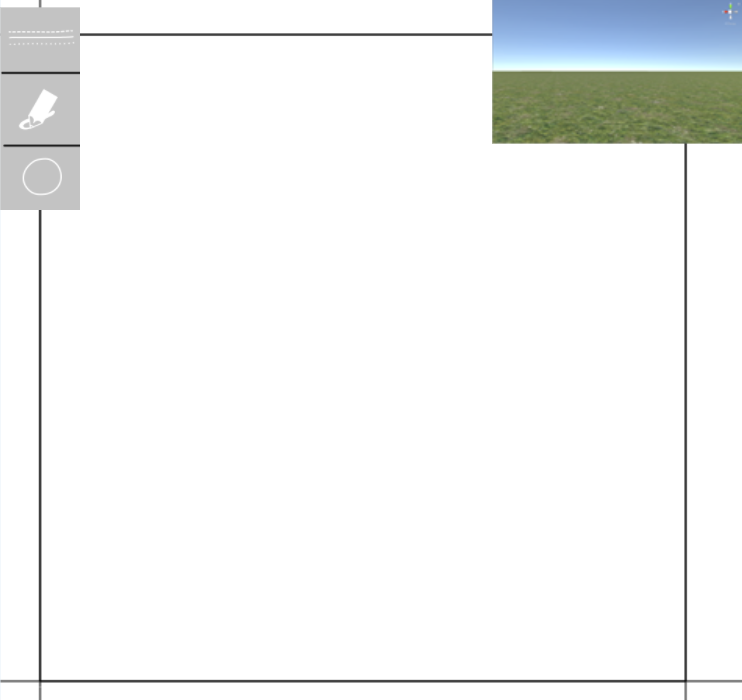
Government of Republic of Estonia. Estonian Cooperation Assembly, the Estonian Education Forum, the Ministry of Education and Research. (2014) *The Estonian Lifelong Learning Strategy 2020*. Tallinn.

# Extras

## Scenario of usage

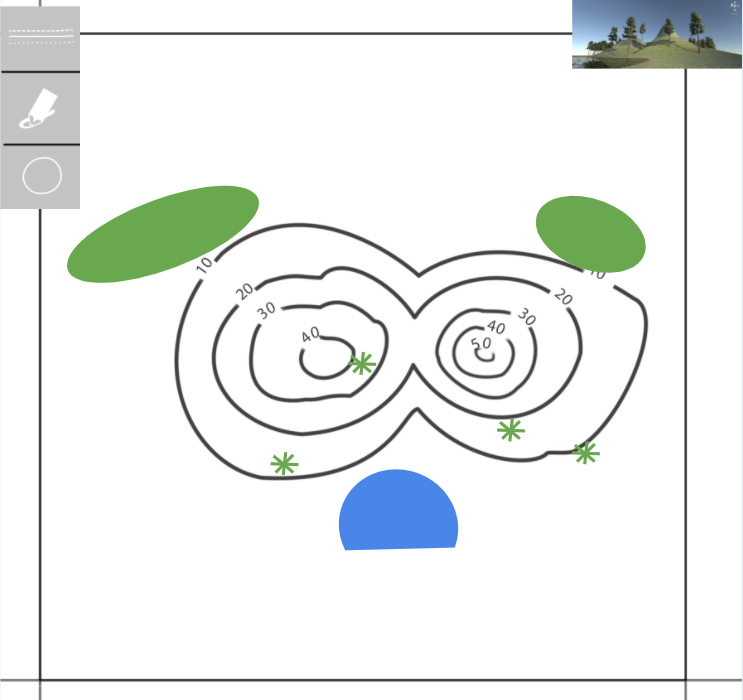
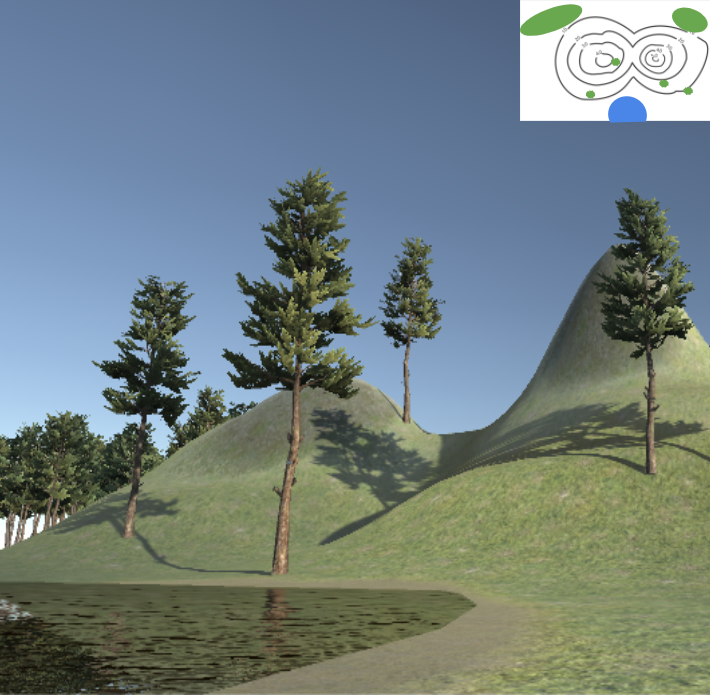
Nate has just started studying geography in 6th grade and the teacher has assigned the first homework for the pupils. The assignment is to download the digital learning game “Topographics” to their smartphones or tablets via Google Play; then start and complete the tutorial in the program, and post results to e-school, or send them via e-mail to the teacher. Unfortunately Nate does not own a smartphone, so the teacher suggests Nate to use the program in the computer class instead.

In the computer class, Nate logs in on a PC, and opens the learning game “Topographics” by double-clicking the program’s icon on the desktop. The program opens and shows options to “start drawing a map”, or to “start the tutorial”. Nate chooses the option for the tutorial, and the screen changes to “the drawing view”. This view contains a drawing canvas, a toolbar on the left of the screen and to the top-right corner of the screen there is a small window showing a flat landscape.



A text “Click here” appears next to the toolbar, with an arrow pointing to a “line” icon in the toolbar. Nate clicks on the icon, and the text changes to “Click & drag to draw a shape”, and an arrow pointing to the canvas. Nate clicks and drags on the canvas and sees a line appearing where the mouse cursor is moving. When he releases the mouse button, he notices the landscape change in the landscape window on the top-right corner: the landscape has raised in the exact shape as Nate drew the line. Next to the landscape window appears a new text “Congratulations! Now you know how to raise the terrain! Click on the terrain window to have a closer look!”

After clicking on the terrain window, the canvas disappears and the “landscape view” is shown in full screen. Nate is introduced how to look and move around in the landscape and also how to switch back to the drawing view. Then the tutorial continues to introduce other tools, such as colors and symbols. After another minute, Nate has created two hills, a lake and some trees which he is eager to start exploring in the landscape view.

Now the tutorial introduces Nate the final task: the canvas zooms out, revealing the grid of the map on the canvas, and ordering Nate to fill all the 9 squares of the grid with any of the tools found in the toolbar.



After finishing the final task, the tutorial shows Nate how to save the work and share it online or by e-mail. Nate sends the work to his teacher, but also chooses to share images of the newly created world on social media, so others can see his creation too.