

# Managing Affective-learning THrough Intelligent atoms and Smart InteractionS

Pending of approval from the European Commission

## D1.10 Learning Content Editor

<b>Workpackage</b>	WP1 - Exploitation, Business Plan and Market Analysis
<b>Editor(s):</b>	Thomas TECHENE, DXT; Jens Piesk, NG
<b>Responsible Partner:</b>	<i>DXT, NG</i>
<b>Status-Version:</b>	Final – v1.0
<b>Due Date:</b>	31/01/2019
<b>Submission Date</b>	31/01/2019
<b>EC Distribution:</b>	Other, Public
<b>Abstract:</b>	This document presents detailed and step-by-step instruction set on how to use the MaTHiSiS Standalone Learning Content Editor (desktop). It includes guidelines on the installation and configuration of the application.
<b>Keywords:</b>	Learning Scenarios, Learning Objectives, Learning Activities
<b>Related Deliverable(s)</b>	D3.2 <i>The MaTHiSiS Smart Learning Atoms (M24)</i> , D3.4 <i>The MaTHiSiS Learning Graphs (M24)</i> , D3.6 <i>Experience Engine (M24)</i> , D3.10 <i>MaTHiSiS Frontend Components (M33)</i>



## Document History

Version	Date	Change editors	Changes
0.1	17/12/2018	Thomas TECHENE, DXT	First version
0.2	07/01/2019	Thomas TECHENE, DXT	Section 2
0.3	09/01/2019	Thomas TECHENE, DXT	Update on the installation section
0.4	16/01/2019	Thomas TECHENE, DXT	Update section 3
0.5	21/01/2019	Thomas TECHENE, DXT	Update section 1.3
0.6	22/01/2019	Jens Piesk, Andrew Pomazanskyi, NG	Update Section 1, Section 4 and 5, Final Formatting
1.0	29/01/2019	Jens Piesk, Andrew Pomazanskyi, NG	Reviewers feedback, deliverable finalization.

The information and views set out in this document are those of the author(s) and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

# Table of Contents

---

Document History .....	1
List of Tables .....	3
List of Figures.....	4
List of Acronyms .....	5
Project Description .....	6
Executive Summary .....	7
1. Introduction .....	8
1.1 Document purpose and context.....	8
1.2 Document structure .....	8
1.3 Learning content Editor in a nutshell .....	8
2. Installation of the MaTHiSiS Standalone Learning Content Editor .....	10
3. Dependencies and preconditions .....	15
3.1 Dependencies.....	15
3.2 Preconditions.....	15
4. Commercialization of the component .....	16
5. Conclusions .....	19
6. References.....	20

## List of Tables

---

*Table 1: Definitions, Acronyms and Abbreviations..... 5*

## List of Figures

---

<i>Figure 1 MaTHiSiS Tools download section</i> .....	10
<i>Figure 2: Installation of MaTHiSiS Standalone Learning Content Editor (1)</i> .....	11
<i>Figure 3: Installation of MaTHiSiS Standalone Learning Content Editor (2)</i> .....	11
<i>Figure 4: Installation of MaTHiSiS Standalone Learning Content Editor (3)</i> .....	12
<i>Figure 5: Installation of MaTHiSiS Standalone Learning Content Editor (4)</i> .....	13
<i>Figure 6: Installation of MaTHiSiS Standalone Learning Content Editor (5)</i> .....	14
<i>Figure 7 - Installation of MaTHiSiS Standalone Learning Content Editor (6)</i> .....	14
<i>Figure 8 RAGE Marketplace</i> .....	17

## List of Acronyms

Abbreviation / acronym	Description
API	Application Programming Interface
CGDLC	Career Guidance Distance Learning Case
LAM	Learning Action Materialization
LCE	Learning Content Editor
LPR	Learner's Profile Repository
MF	MaTHiSiS Front-end
PC	Platform Configuration
SC	Sensorial Component
SLA	Smart Learning Atom
xAPI	Experience API

**Table 1: Definitions, Acronyms and Abbreviations**

## Project Description

---

The MaTHiSiS learning vision is to provide a novel advanced digital ecosystem for vocational training, and special needs and mainstream education for individuals with an intellectual disability (ID), autism and neuro-typical learners in school-based and adult education learning contexts. This ecosystem consists of an integrated platform, along with a set of re-usable learning components with capabilities for: i) adaptive learning, ii) automatic feedback, iii) automatic assessment of learners' progress and behavioural state, iv) affective learning, and v) game-based learning.

In addition to a learning ecosystem capable of responding to a learner's affective state, the MaTHiSiS project will introduce a novel approach to structuring the learning goals for each learner. Learning graphs act as a novel educational structural tool. The building materials of these graphs are drawn from a set of Smart Learning Atoms (SLAs) and a set of specific learning goals, which will constitute the vertices of these graphs, while relations between SLAs and learning goals constitute the edges of the graphs. SLAs are atomic and complete pieces of knowledge, which can be learned and assessed in a single, short-term iteration, targeting certain problems. More than one SLA, working together on the same graph, will enable individuals to reach their learning and training goals. Learning goals and SLAs will be scoped in collaboration with learners themselves, teachers and trainers in formal and non-formal education contexts (general education, vocational training, lifelong training and specific skills learning).

MaTHiSiS is a 39-month long project (originally 36-month long, extended after contract amendment), co-funded by the European Commission Horizon 2020 Programme (H2020-ICT-2015), under Grant Agreement No. 687772.

## Executive Summary

---

This document reports on the progress and final solution of the Learning Content Editor desktop application, developed as part of the MaTHiSiS project.

The document provides a detailed description of the application as well as a detailed guideline from the point of installation up to a full configuration of the Learning Content Editor.

Detailed guidelines about how to use the Learning Content Editor can be found in the **User Manual Guidelines**, and technical description of the tool are present in the **D3.2 The MaTHiSiS Smart Learning Atoms** [1], **D3.4 The MaTHiSiS Learning Graphs** [2], **D3.6 Experience Engine** [3] and **D3.10 MaTHiSiS Frontend Components** [4].

The Learning Content Editor is by default connected to the MaTHiSiS back-end in order to retrieve and save the Learning Content created with the editor. However, it is possible to still create content without an internet connection, since the editor has the abilities to open and save content from files.



# 1. Introduction

---

## 1.1 Document purpose and context

The purpose of this document is to provide the users of the MaTHiSiS Standalone **Learning Content Editor** tool with easy-to-follow instructions, in order to install, configure and use the application. The document starts by presenting the installation process with details that guide the user throughout the whole process step-by-step. An explanation of the applications User Interface (UI) follows. The document describes the functionalities of the UI buttons, as well as the flow of the user interactions, in order to use the application properly. In the last chapter, the applications dependencies and any additional programs or required files are reported.

## 1.2 Document structure

This document consists of the following chapters:

**Chapter** Error! Reference source not found.: Installation of the MaTHiSiS Standalone Learning Content Editor

**Chapter 3:** Dependencies and Pre-conditions

**Chapter 4:** Described further distribution of the component through third party marketplace

**Chapter 5:** Conclusion of the document.

## 1.3 Learning content Editor in a nutshell

The Learning Content Editor (LCE) is a visual-authoring graph-based tool, which purpose is to design non-linear Learning Experiences, based on pedagogical approved concepts, namely the Learning Scenarios, the Learning Objectives, and the Learning Activities. The Learning Experiences designed with this tool are generic and can be declined for several learner profile at the execution moment.

Teachers, tutors, trainer from both of educational and professional area are the targeted audience for the Learning Content Editor, who wish to build Learning Experiences, leaning on a novel pedagogical approach.

The Learning Content Editor contains a dedicated sub-editor for each main concept, to allow users the possibility to work separately at the different levels of abstraction for a given Learning Experience. Re-usability is one of the key notion present in the LCE, since the learning content produced is meant to be at the disposal of users for the construction of their graph learning experiences.

### **Learning Scenario Editor:**

A Learning Scenario consists of learning content components (learning goals and leaning objectives) and weighted relations between them. The learning scenario consists of all the components/concepts pertaining to **what-to-learn** during the educational process.

The learning objectives are atomic and complete pieces of learner knowledge, competencies and/or skills, which can be learned and assessed in a single, short-term learning process iteration. Learning objectives essentially comprise primordial learning goals, constituents of more advanced learning

goals, which cannot be further reduced to more primitive notions. They consist of the simplest of concepts pertaining to **what-to-learn** during the educational process.

Learning Goals describe learners' skills or knowledge over a comprehensive learning objective. In essence, learning goals consist of the particular competencies the learners need to acquire in order to achieve a specific learning objective.

In a few words, the Learning Scenario Editor is a tool for the creation/edition of high-level graphs called **Learning Scenario** composed of **Learning Objectives** and **Learning Goals**.

**Learning Objective Editor:**

A **Learning Objective** is made up of a list of **Learning Activities**.

**Learning Activities** stimulate and convey the learning process for one or more specific pieces of simple knowledge/skills (Learning Objective) to the learner. **Learning Activities** pertain to **what-to-do** in order to learn specific things. **Learning Activities** are device-agnostic and context independent.

In a few words, the Learning Objective Editor is a tool for the creation/edition of skills/competencies called **Learning Objectives** composed of **Learning Activities**.

**Learning Activity Editor:**

As described previously, **Learning Activities** are device-agnostic and setting independent, which needs a final abstraction level in order to precisely define how to interpret the Learning Activities, which is called the **Learning Action Materialization (LAMs)**. The LAMs are exactly the device-specific and/or Learning Material-specific interpretations of Learning Activities for different learning settings. **Learning Action Materializations** pertain to how-to-do what needs to be done in order to learn specific things.

## 2. Installation of the MaTHiSiS Standalone Learning Content Editor

The Learning Content Editor can be downloaded from the MaTHiSiS website <http://mathisis-project.eu/> under the Results section. More specifically, a dedicated page for distribution of the tools has been created where the Learning Content Editor can be found: <http://mathisis-project.eu/en/content/learning-content-editor-tool>

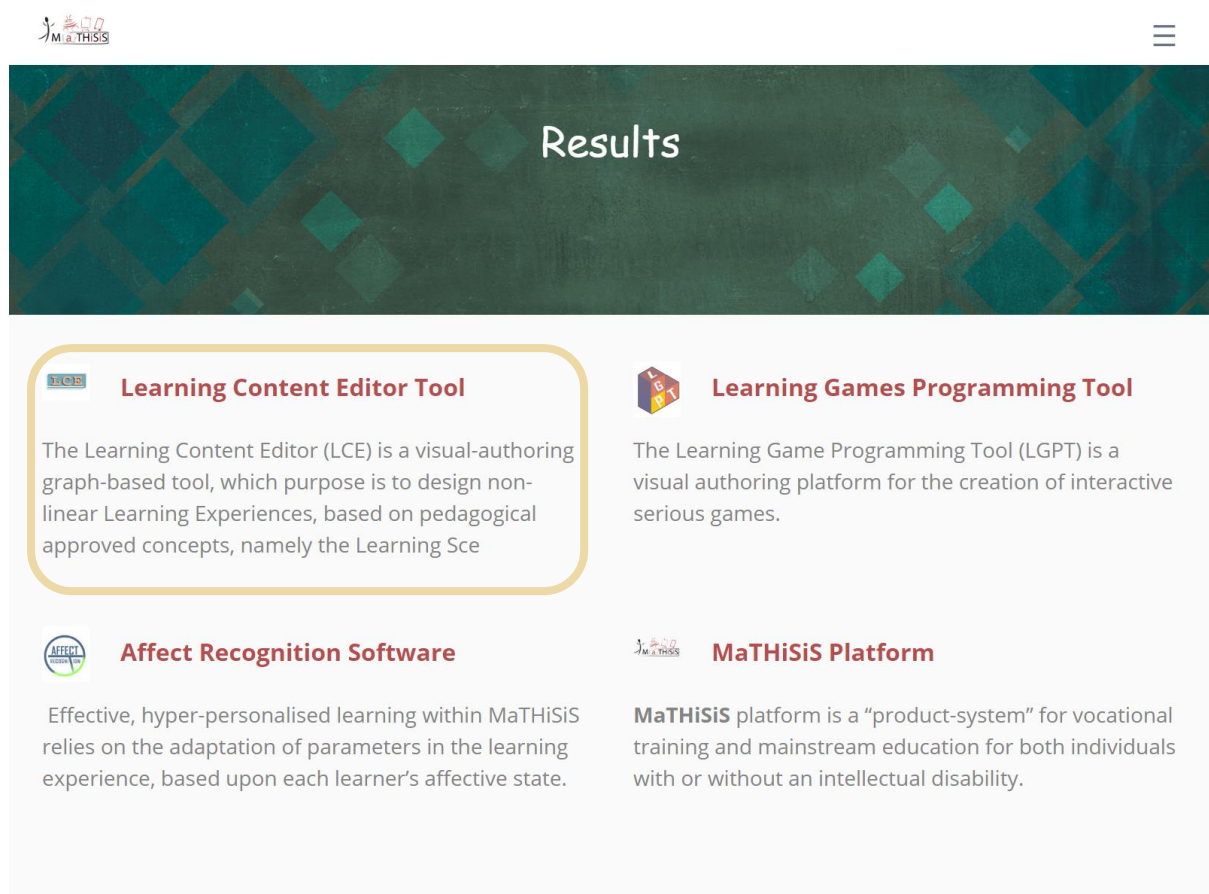


Figure 1 MaTHiSiS Tools download section

After following the tab of the **Learning Content Editor Tool** user will be exposed to some promotional material, component description and the executable, named “MaTHiSiS-LCE-x.x.x.exe” (x.x.x = version, current is 1.0.0).

After downloading the files, proceed with the following steps:

- Double click on the **MaTHiSiS-LCE-x.x.x.exe** installer and follow the wizard to install the software.

Follow the installation wizard:

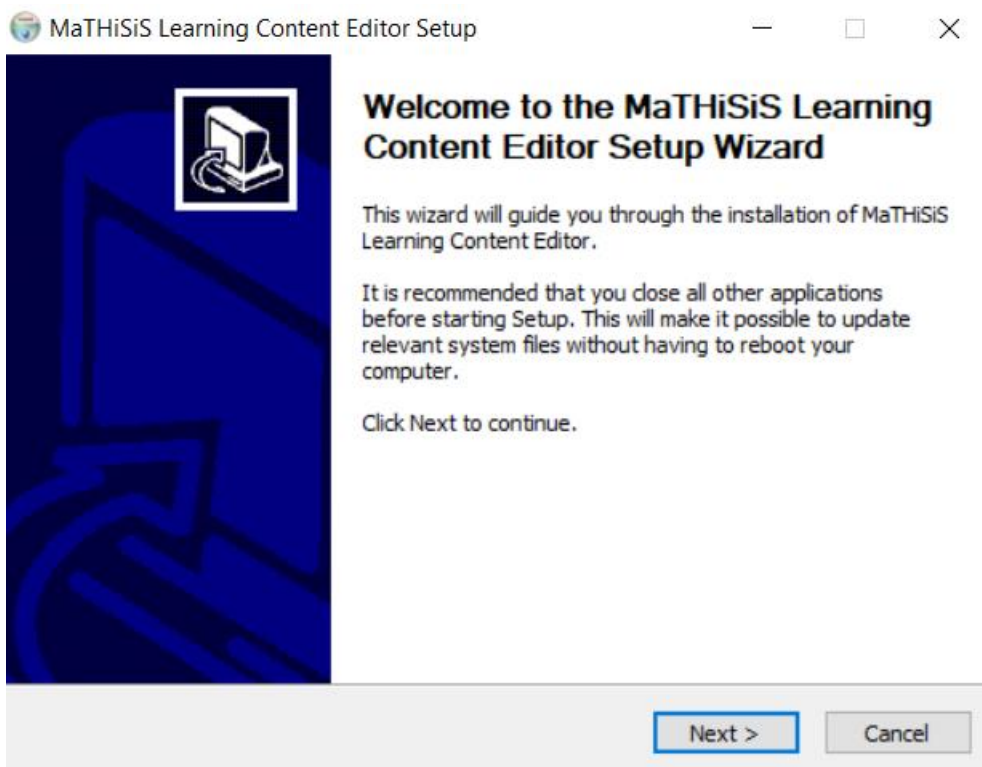


Figure 2: Installation of MaTHiSiS Standalone Learning Content Editor (1)

- Click on the “Next” button

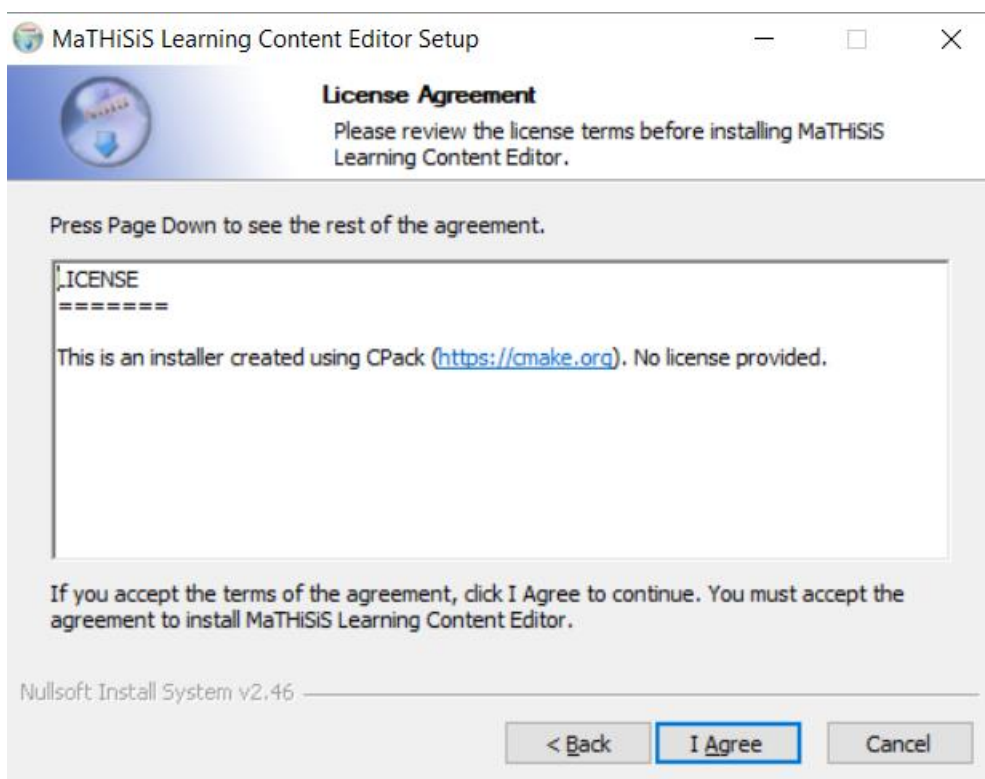
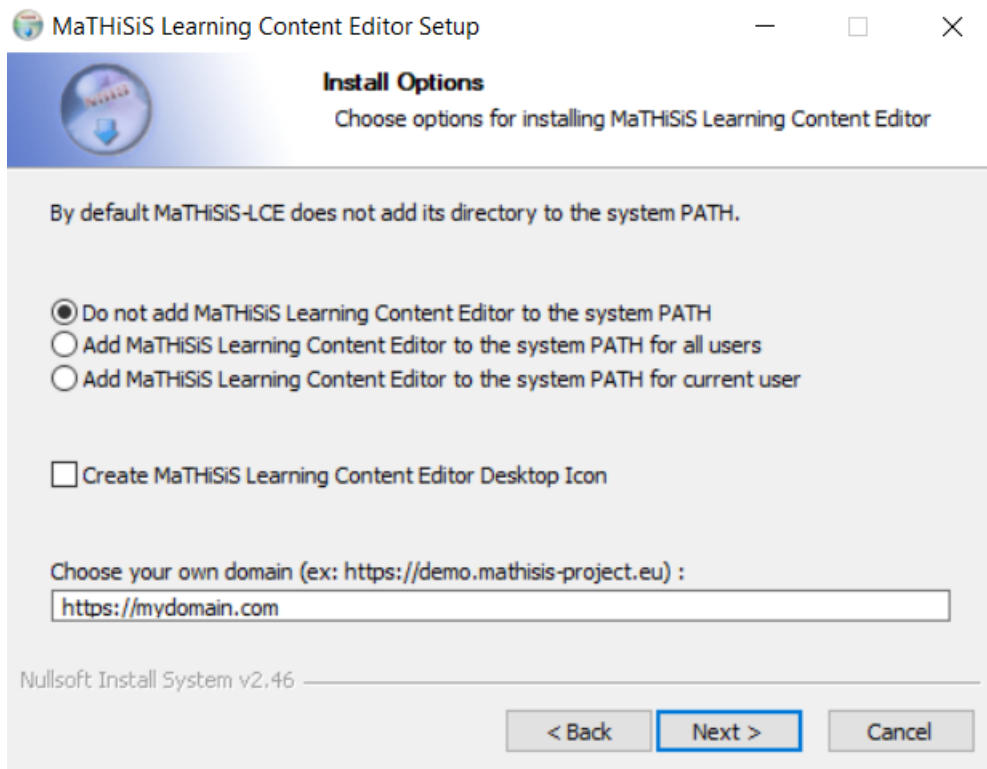


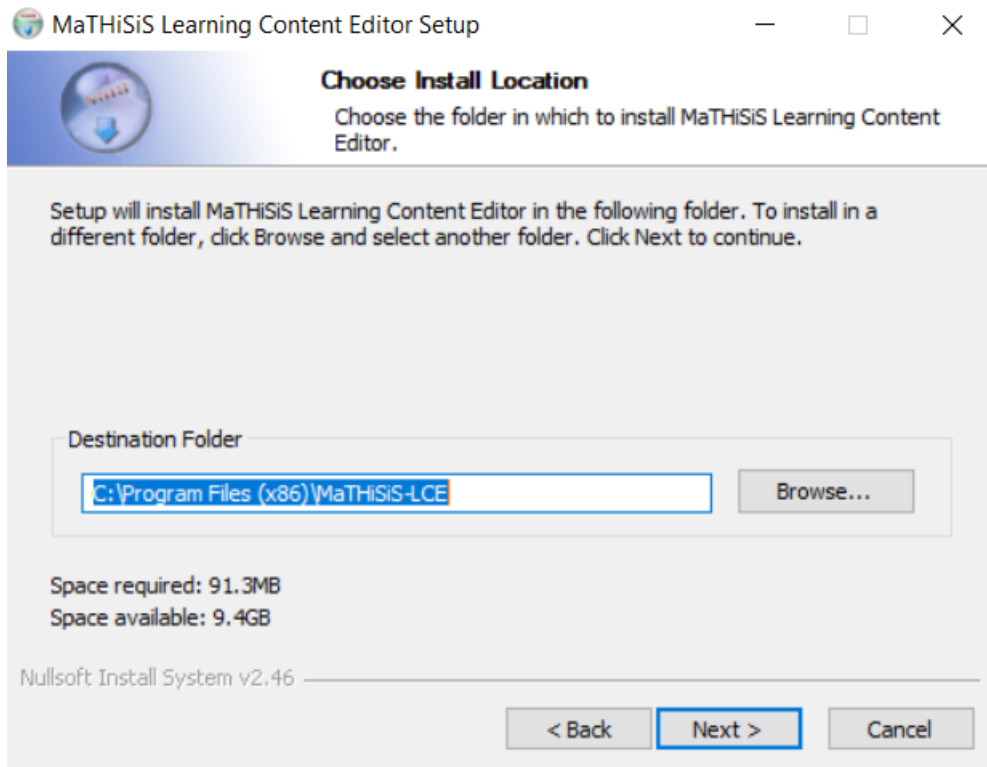
Figure 3: Installation of MaTHiSiS Standalone Learning Content Editor (2)

- Accept the license agreement, clicking on “I agree”



**Figure 4: Installation of MaTHiSiS Standalone Learning Content Editor (3)**

- Choose whether a desktop icon will be created for the application
- Choose if the software will be inserted in the PATH environment variable for one user, all users, or not added. Note that this option is not mandatory, by default the selected option will not add the LCE in the environment variable.
- Insert the domain on which the platform will be, so the LCE knows where to find the different libraries.



**Figure 5: Installation of MaTHiSiS Standalone Learning Content Editor (4)**

-Choose the installation location folder

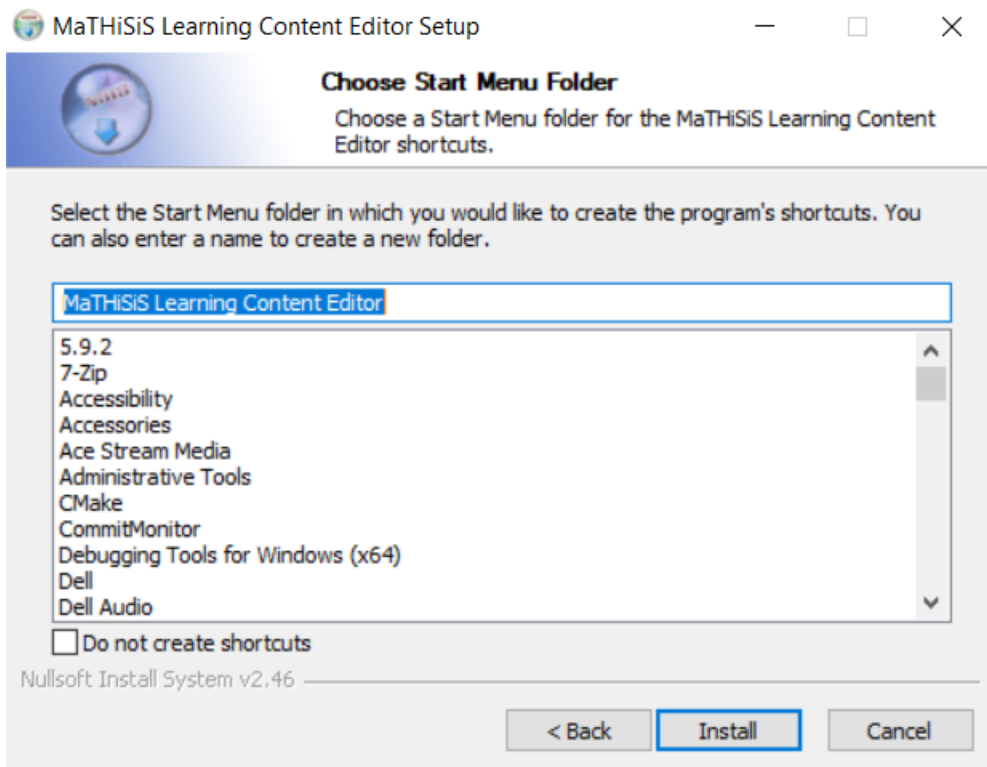


Figure 6: Installation of MaTHiSiS Standalone Learning Content Editor (5)

- Choose of the Start Menu Folder

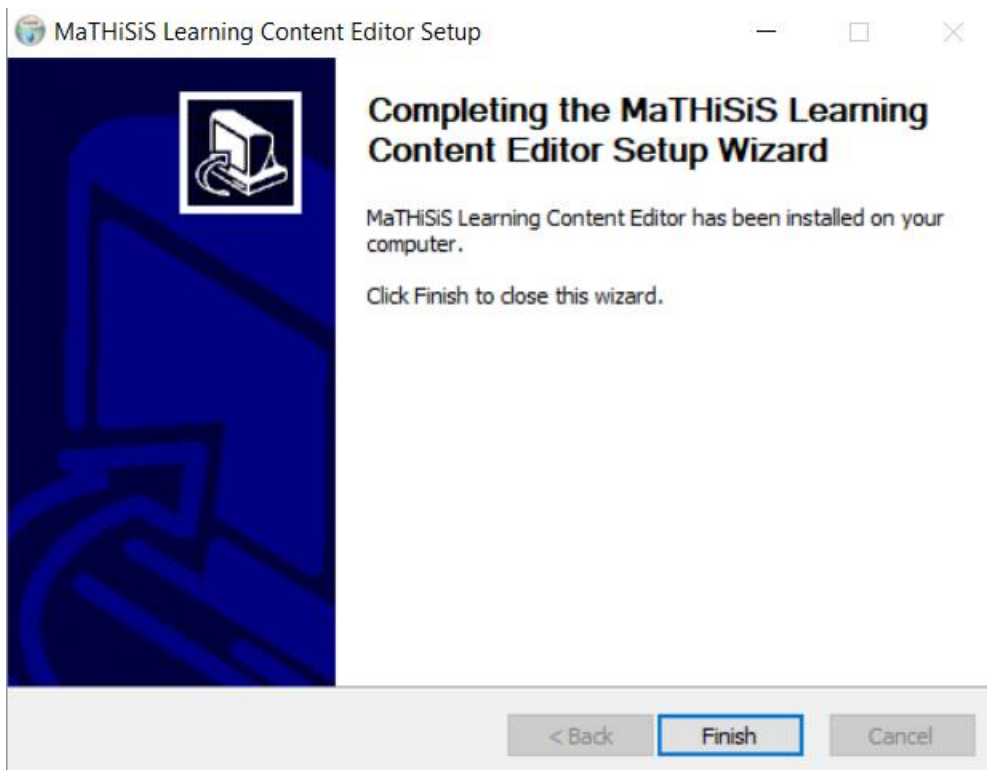


Figure 7 - Installation of MaTHiSiS Standalone Learning Content Editor (6)

## 3. Dependencies and preconditions

---

### 3.1 Dependencies

The application has the following dependencies:

- Microsoft Windows 8 or newer
- Python 2 (downloaded with the installation files)
- KinectSDK-v2.0\_1409: Provided with the installer
- Visual C++ Build Tools 2015: Provided with the installer

### 3.2 Preconditions

Since the Affect Recognition tool needs to communicate with the MaTHiSiS cloud services, the network of the workstation that runs AR-d must provide access to **ports 80,443 and 7000** and also the following addresses:

1. <https://prod.mathisis-project.eu/> and
2. **wss://prod.mathisis-project.eu:7000** (Please note that this is *NOT* http).

Firewall restrictions to the web and the aforementioned locations must be disabled.

For institutions and corporations that **have very strict firewall restrictions**:

- Some ports have to be manually allowed from the system administrator of the venue.
- The main port (7000) is where the AR-d tool communicates with the cloud. This needs to be open, otherwise the platform won't communicate with the devices at all.

In case you notice that the tool is not connecting to the network in a particular institution/workplace that might have network and firewall restrictions, please contact whomever manages the internet connection configuration and ask them to allow the connection to the ports above.



## 4. Commercialization of the component

---

To ensure successful exploitation of the tool, certain market penetration assumptions have to be made and visibility of the solution provided. Project partners have performed an analysis and evaluation of various ecosystems that could potentially allow future discovery, monetization, marketing and facilitation of MaTHiSiS tools adoption in various domains. The European Commission realizes the potential and importance of applied and serious gaming solutions being part of the Horizon 2020 Programme funding scheme and supports the technology penetration in various domains that are deemed to address various societal challenges not in education only, but social cohesion, security and health. Such commitment of the policy makers becomes apparent with the provision of additional subsistence to a number of platforms and marketplaces that serve as a bridge between research institutions and industry players making software tools and open APIs available for further research and market adoption.

Thus, MaTHiSiS has established certain liaison activities with RAGE<sup>1</sup> marketplace to benefit from the value proposition the platform offers and at the same time take advantage of distributing such components as LCE. The value propositions are described as the following<sup>2</sup>:

- 1) an interoperable set of advanced technology assets
- 2) proven practices of using asset-based applied games in various real-world contexts,
- 3) centralised access to a wide range of applied software modules, services and resources,
- 4) an online social space (the RAGE Ecosystem) that arranges and facilitates collaboration that underlie progress and innovation,
- 5) workshops and online training opportunities for both developers and other stakeholders,
- 6) assets-based business cases that support the industries at seizing new business opportunities, and
- 7) a business model and launch plan for exploiting the RAGE Ecosystem beyond the project's duration.

---

<sup>1</sup> <https://www.gamecomponents.eu/>

<sup>2</sup> [https://cordis.europa.eu/project/rcn/194166\\_en.html](https://cordis.europa.eu/project/rcn/194166_en.html)

The screenshot displays the RAGE Marketplace website. At the top left is the RAGE logo with the tagline 'Realizing an Applied Gaming Ecosystem'. Navigation links include COMPONENTS, GAMES, EXPLORE, STAKEHOLDERS, LOGIN, and a shopping cart icon. The main content area features a search bar with the text 'What are you looking for?' and a 'Search' button. Below the search bar, it indicates '37 results found (15 ms)' and 'Page 1 of 5' with a 'Next' link. On the left side, there are filters for PRICE, TAXONOMY (with a 'SHOP' dropdown), and CATEGORIZATION. The CATEGORIZATION section shows a list of categories with checkboxes and counts: Component (37), Analytics (7), Avatar Systems (7), Affective State Detection (2), Gamification (1), Language Processing (8), and Personalisation (9). The main search results area displays three items:

- Virtual Labs Authoring tool**: Includes a 5-star rating, 'FREE' status, and a description: 'Authoring your own Wind Energy and Chemistry Virtual Labs'.
- goedle.io Unity Tracking SDK**: Includes a 5-star rating, 'FREE' status, and a description: 'Analytics SDK for Unity to track telemetric/ event data in order to make use of predictions and dynamic difficulty adjustment with goedle.io'.
- Real-time Emotion Detection on Asset**: Includes a 'FREE' status and a description: 'The asset detects real-time emotion based on facial expression of humans. The asset functions in three manners: 1) using a single image file, 2) using a recorded video file, and 3) using a real-time webcam stream.'

**Figure 8 RAGE Marketplace**

Figure 7 depicts the RAGE marketplace that offers a number of categories where the components can be submitted, discovered and eventually acquired. To ensure that the components developed within MaTHiSiS project meet the requirements of the RAGE marketplace software tools submission process, a dedicated metadata document has been created as part of the commercialization package which in turn will enable partners to further submit their technologies to third party platforms, marketplaces and repositories thus providing yet another mean to get Return On Investment (ROI) on the tools developed. The metadata document followed the following structure:

- The name of the component
- Component description that is also used during the search queries
- Short non-technical description
- Technical description of the component including (1) what the component does; (2) what it could be used for, (3) what inputs are needed; (4) what outputs it produces.
- Picture and logo as a visual cue/tile in the search results
- Optional release date
- Optional language settings
- Optional access URL in case a home page or a GitHub page exists
- Development environment
- Target platform
- Programming language used
- Applied computing concepts according to the ACM Computing Classification System
- Optional possible goals or purpose followed when developing a component/module
- Optional indication of the component category that will be listed in the public RAGE Ecosystem Portal
- Optional keywords with module attributes

- Version of the exploitable result, its notes, development status and a commit URL in case the software is in a versioning system
- License type with an optional indication to a license URL or a description of any specific conditions and restrictions applying to the use of the component
- Contact details of the legal owner and creators of the component
- Finally, a detailed technical description explaining what the component does, what inputs it needs, how it functions, how it can enhance applied games, technical requirements, operational constraints, source code, possible supporting documentation such as the user guides and tutorials, help files, demonstrations, design models, setup files, installation scripts, configuration data, setup guides, demonstration
- Other resources such as videos, snapshots, etc.

## 5. Conclusions

---

D1.10 provides an evidence that the Learning Content Editor tool is delivered as a packaged software and available through the project website with a clear plan for further distribution of the component.

The deliverable provides installation guidance for the end user whereas the interaction and usage instructions are provided in a separate user manual.

D1.10 is a supporting document to a software release whereas individual approaches to commercial use are going to be described in a dedicated business plan related to the exploitation of the MaTHiSiS system as a whole and the individual tools developed during the project runtime.

## 6. References

---

- [1] DIGINEXT (ed.): D3.10 *MaTHiSiS Frontend Components (M33)*. Deliverable of the MaTHiSiS project, 2018.
- [2] DIGINEXT (ed.): D3.2 *The MaTHiSiS Smart Learning Atoms (M24)*. Deliverable of the MaTHiSiS project, 2017.
- [3] DIGINEXT (ed.): D3.6 *Experience Engine (M24)*. Deliverable of the MaTHiSiS project, 2017.
- [4] CERTH (ed.): D3.4 *The MaTHiSiS Learning Graphs (M24)*. Deliverable of the MaTHiSiS project, 2017.