Project Specification

1. Content

Project Specification 1
1. Content 1
2. Description 1
3. Objective 2
4. Game requirements 2
4.1. Game Play 2
4.2. Window dimensions 2
4.3. Grading Rubric 3
4.4. Display controls info on request 3
5. Technical requirements 3
5.1. Development environment 3
5.2. Code and design 3
5.3. MiniGame 3
5.4. Project name 4
6. Reporting 4
7. Milestone and deadline 4
7.1. Milestone: 20 % of the total score 4
7.2. Deadline: 50 % of the total score 4
7.3. Milestone and deadline submission 5
7.4. Presentation 5
7.5. Retake 5
8. Weekly submissions 6

2. Description

The project consists of programming an offline 2D game in C++ using the programming2 framework.

Each student does the project on a strictly **individual basis** and doesn't duplicate other student's code. Also read the article 75:

"Irregularities In Connection With Exams, Examination-Related Disciplinary Decisions and Appeal" (English)

"Onregelmatigheden in verband met examens, examentuchtbeslissingen en beroep" (Dutch)

in the OER-document:

"Education and Examination Code Academic year 21-22" (English) available on https://www.howest.be/en/study/study-at-howest/educational-organization

"Onderwijs- en examenreglement academiejaar 21-22" (Dutch) available on https://www.howest.be/nl/studeren/inschrijven-en-info/studiecontract

3. Objective

The student demonstrates with this assignment that the topics of the module are well understood and can be applied in a specific application.

4. Game requirements

4.1. Game Play

You choose an interesting level of an existing 2D (arcade) game.

Choose wisely!

- It is preferred to have a nice finished limited project over a huge game that is only partially implemented.
- The time you can spend on your game will be limited due to other courses such as 3D.
- There is a milestone in April right after spring holiday.
- Do not procrastinate this assignment, the load due to other courses will increase as the end of the semester approaches.
- You can choose to make a smaller, simple game and know that the max grade for such a game will not be 20, even it is finished.
- Use the rubric to self-evaluate and to match your game with the requirements. If your game does not have what is in the rubric, no grades for that part will be given.
- Look for a game with existing sprite sheets. Making them yourself will take days or even weeks!

4.2. Window dimensions

The size of the game window should be no larger than 1280 x 800.

4.3. Grading Rubric

The game requirements such as camera, animations, interactions, game play, HUD, sound … are described in the **GradingRubric** document. This document helps you to build an excellent project. It is needless to say that this document shouldn't be ignored. Use it as a guideline, priority, and game selector.

The same document will serve as a guide for the lectors to define your grade.

4.4. Display controls info on request

On request – when the I key is pressed – info about the game control keys has to be shown.

5. Technical requirements

5.1. Development environment

The game is developed using **Visual Studio 2022**.

The game is built using the framework published on Leho. You are not allowed to use game engines or other frameworks. However, you are allowed to extend or modify the framework with your own code. You can use xml or json parsing libraries as long as you mention it in the code and tell your teachers during presentation.

Detection of memory leaks must be activated.

5.2. Code and design

The programming language is C++.

It is important that your **program is readable and maintainable** by applying the principles and techniques you have been taught. Use the coding standards document as a guide. These Code and programming guidelines are lined up in the standards document under 00_General.

5.3. MiniGame helper

During the first labs we will make a small game in which you'll see:

- How to transform (translate, rotate, scale) objects.
- How to manage multiple similar objects using a container.
- How to convert a level image into an svg image, which can be read in by the code available in the framework and converted into a series of line segments.
- How you can use the Raycast functions in the framework to determine whether an actor hits the level (consisting of line segments) and how deeply the actor has penetrated the level.
- How to implement a camera by translating the OpenGL Modelview matrix.
- How to add music and sound effects to your game using SDL_Mixer

The MiniGame folder contains a working example.

5.4. Solution / Project name

Name_firstname_GameName

6. Reporting

This folder contains a file named **Name_FirstName_Game.xlsx** in which you report your progress. You can write in English or Dutch.

Rename the file using your own name and chosen game.

Copy this report file in your Visual Studio game project folder and add it to the **Resource Files** filter in your Visual Studio game project.

Continually update the content of this document. Do not wait until the Leho upload. It should always be up to date.

It contains 2 worksheets in which you indicate how much of the game is realized and works.

- **Basic Game rubric**: in which you indicate how much you've realized of the game rubric. Only indicate a part as being realized when that part **works correctly** and is **testable**.
- **Game extras**: in which you indicate the working and testable extras.

7. Milestone and deadline

7.1. Milestone: 20 % of the total score

For the milestone you should get to the "**Meets Standard**" column for the following parts:

- Camera
- Animations
- Interactions
- Game implementation

The "**HUD and UI"** and "**Sound"** rows are not considered, they do not yield any points.

The quality of code will also be considered during the grading of the milestone.

Before **Monday April 18th 2021 at 22:00** you upload your game for milestone grading. (After that date, there's only three weeks left !)

7.2. Deadline: 50 % of the total score

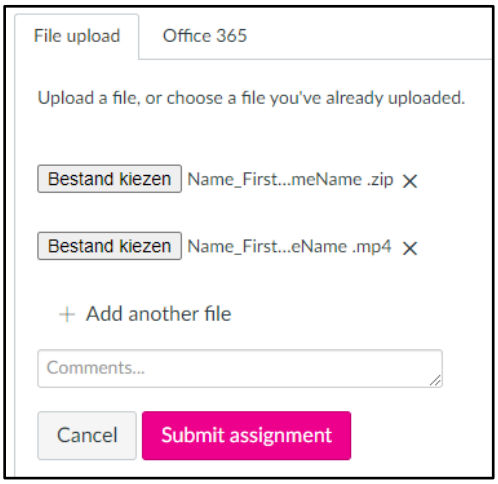
The presentation is as indicated in the exam timetable (May-June).

You will need to **upload** the **evening before** you have your **presentation** to ensure you have a good night sleep.

You will present/defend the project to programming teachers without the presence of other students.

7.3. Milestone and deadline submission

You upload two files with name **1DAEXX_Name_FirstName_GameName** into the dedicated assignment.



1. In the Leho assignment tab "File Upload", a **Name_Firstname_GameName** compressed file, containing your C++ project including all resources so that we can run it. Notice that this project should also contain the report file **Name_firstname_Game.xlsx**, with up to date content.

2. In the Leho assignment tab "File Upload", a **Name_Firstname_Presentation.mp4 file**: A **video file** in which you **present** the working of your game.
   This **video** includes the **captured game window** (so no need for a camera). The **audio** is a **mix** of the **game audio** and **your voice** explaining what you are demonstrating. The video does not take longer than 5 minutes. The timeline of the video follows the rows of the rubric from top to bottom. So first you show the working of the camera, then the animations …. You don't have to explain the code nor tell what you're planning to work on. You just show the working of the game.

7.4. Final Presentation

The presentation is done on your own laptop. During both presentations, you:

- Play your game or show a video with you playing your game.
- Answer questions about the code. With these questions we want to assess your insight and profoundness of your knowledge regarding the programming of the game.

7.5. Retake

There is retake in August for the project part of Programming2:

- You can continue the same project or choose another game.
- The milestone result is no longer considered.
- The retake score counts for 70% of the total retake score. The remaining 30% is for the theory quiz.

8. Weekly submissions

Each week before the start of the first lab session, you upload your game project as part of the folder 1DAEXX_YY_Name_Firstname folder, this is the folder that also contains your Visual Studio solution WYY with your solutions of the lab exercises. Notice that we expect you to upload only **one** compressed folder a week.