TJ Time Jumper

System Specification

Project Name	TJ Time Jumper/TimeNexus	
Project Manager	Stefan Brandmair	
Document Owner	Stefan Brandmair	
Created on	23/11/2017	
Last modified	16/03/2005 19:54	
State	X	in progress
		submitted
		released
Document	System Specification	

Document History

Revision			Chapter	Modification	Author
Nr.	Date	Version			
1	1/1/2018	0.1	All	Initial Draft	Stefan, Arno
2	25/1/2018	0.2	2	Added a few functional requirements	Stefan, Arno
3	30/1/2018	0.3	2	More functional requirements	Stefan, Arno
4	3/2/2018	0.4	2	Even more functional requirements	Stefan
5	3/2/2018	0.5	1	Initial Situation and co.	Stefan
6	4/2/2018	0.6	3	Added NFR_001 and NFR_002	Arno
7	4/2/2018	0.7	1	Initial Situation improvements	Stefan
8	4/2/2018	0.8	6	Acceptance Criteria	Stefan
9	4/2/2018	0.9	1	Finished the Application Domain and Glossary	Stefan
10	4/2/2018	0.10	2	Finished all the functional requirements	Stefan
11	4/2/2018	0.11	4	Quantity Structure	Stefan
12	5/2/2018	1.0	3	Non-functional requirements and cleaning up everything	Arno

Content

1 Initial Situation and Goal

1.1 Initial Situation

While trying to find a puzzle game with time travel or time shifting as a main mechanic, we realized that there are not that many games in this genre. After narrowing our searching criteria to games which are also 3D and have a platformer aspect, because we wanted games that feel like a combination of Portal and Super Mario, we had a hard time finding any. We ended up finding 9 games which fit our criteria. Of those 9 games, only 2 truly fit the criteria. Namely: Quantum Conundrum [1] and Project Temporality [2]. However, the latter only allows you to create clones of yourself with a time machine.

So, we propose a game in which the player can not only control the main character, but, to some extent, also the world itself via an in-game device. This will allow the player to control the time of his environment. Compared to Quantum Conundrum, this game will actually allow time travelling instead of switching between different dimensions.

1.1.1 **Application Domain**

The game will be a 3D, first person platformer that uses time travelling as a main puzzle device. We chose time travelling as a puzzle device, because it is a unique and rarely seen concept in videogames. Also, time travelling comes with a large number of paradoxes and other challenges which make it even more interesting.

The main challenge in the game will be using the character's time travelling ability in order to solve various puzzles. Every level will consist of a series of puzzles that the player must complete to reach the end and go to the next level.

A platformer videogame presents the player with challenges that most people cannot do in real life because they are not fit enough. It is fairly similar to parkour, with a focus on not falling down from the platform that the player is currently on.

To make the game more immersive, it will employ a first-person camera. It will also be 3D. In other words, the camera will be inside the player's head and he will be able to see exactly what he would see in real life. This also has the advantage that the game can relatively easily be made into a virtual reality game.

1.1.2 Glossary

Platformer: A videogame type where the player has to jump from one place to another without falling down. An example of such a game is Super Mario.

Time scrolling: Time travelling, but only non-living things around you travel in time. This avoids a bunch of time travelling related paradoxes such as the grandfather paradox.

Portal (video game): A videogame where the player had to use portals that were capable of teleporting objects, including the player, to solve various puzzles. The game was a huge success and won multiple awards.

Time travelling paradox: A paradox caused by time travelling such as the grandfather paradox, where a person travels back in time and kills his own grandfather to prevent his own birth.

1.2 Goal Definition

The goal, which we are planning to achieve, is creating a third-person platformer game with puzzle elements and some creative new ideas and combining it in ways which have never been done before. It should fill a market niche that has been neglected a lot.

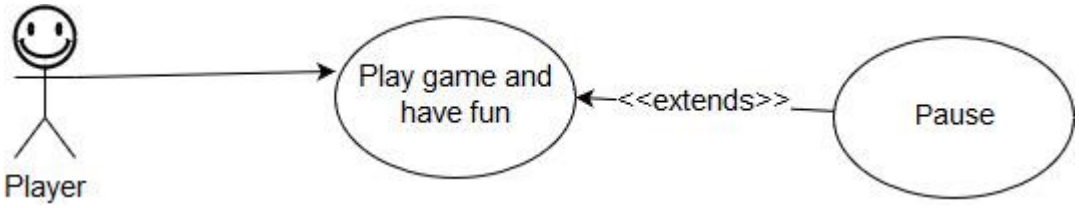
The target group of the game are all sorts of PC gamers, ranging from casual gamers who only play every once in a while, to hardcore gamers who play multiple hours every single day. We chose this target group, because almost every gamer knows how to control a character with the keyboard and a mouse. We are also assuming that our target group has access to a reasonably powerful PC.

Initially, the game will only be available on Windows, because it has the highest desktop market share. However, we plan to make it platform independent. The multi-language support will include German and English.

2 Functional Requirements

2.1 Use Case Diagrams

2.2 Pause - Use Case Details

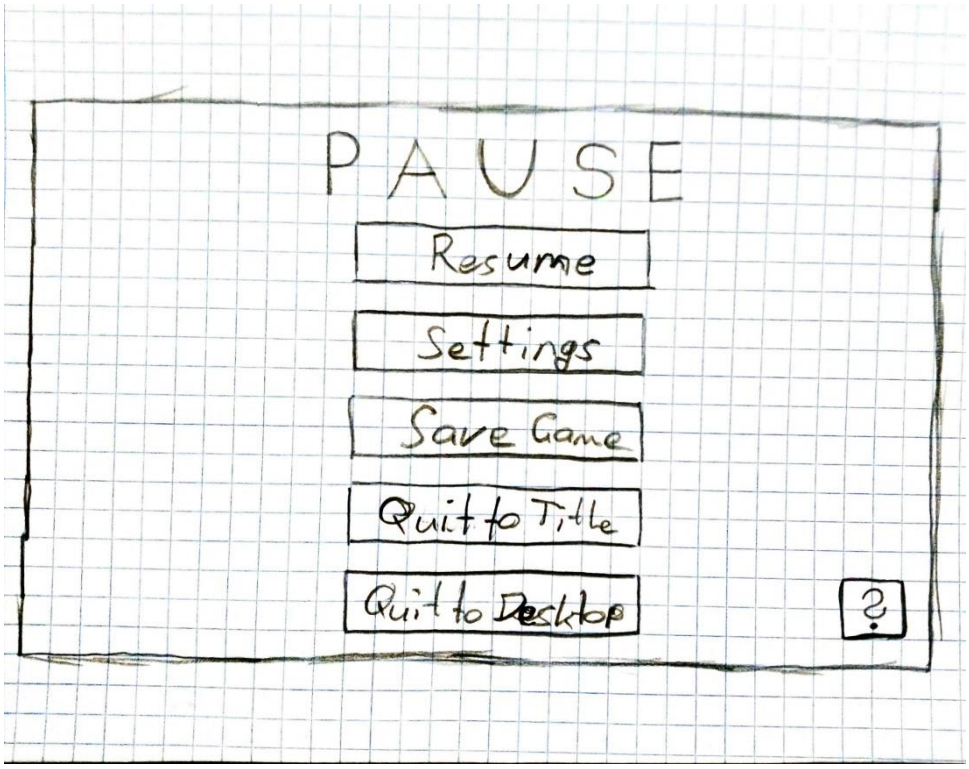


The function of this use case is to allow the player to change the settings, save the game or quit the game while it is paused.

2.2.1 Characteristic Information

Goal:	Pause all changes in the game scene and give the player access to various options
Precondition:	The player has to be in the game
Postcondition:	The game has to stand still, sounds muffled
Involved User:	Player
Triggering Event:	Pressing the key/button dedicated to this action

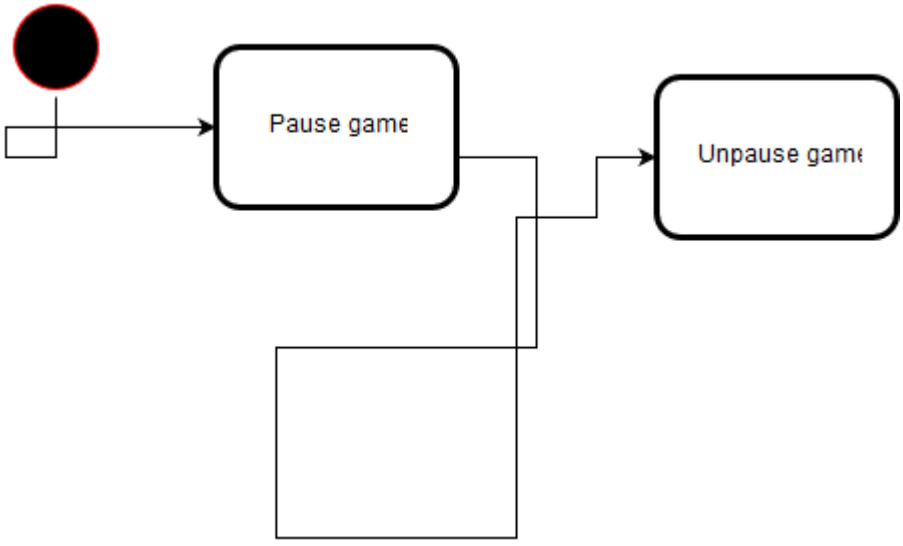
2.2.2 GUI to call the use case



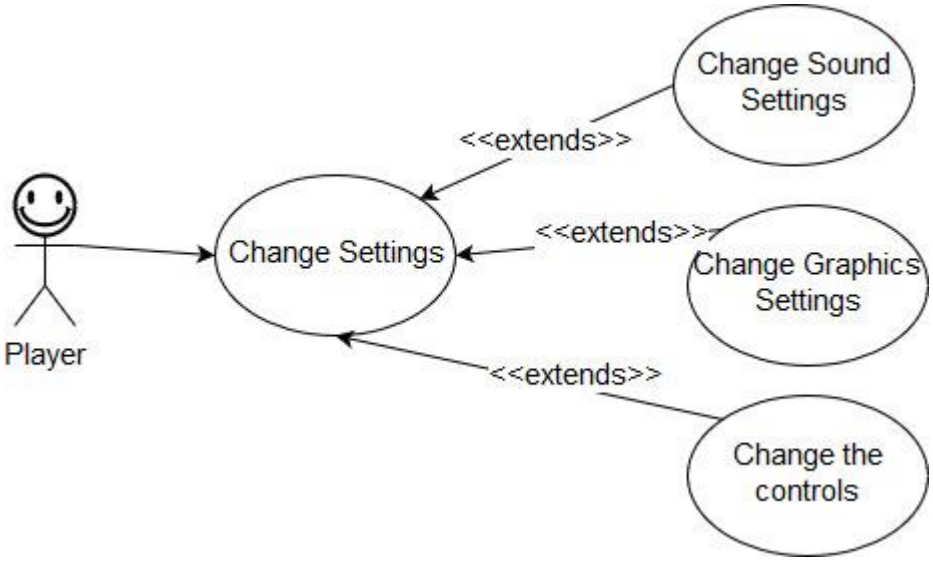
2.2.3 Scenario for the standard use (good case)

Step	Activity
Step 1	Press the pause button/key (to pause the game)
Step 2	Press the pause button/key (to un-pause the game)

2.2.4 Workflow



2.3 Change Settings - Use Case Details

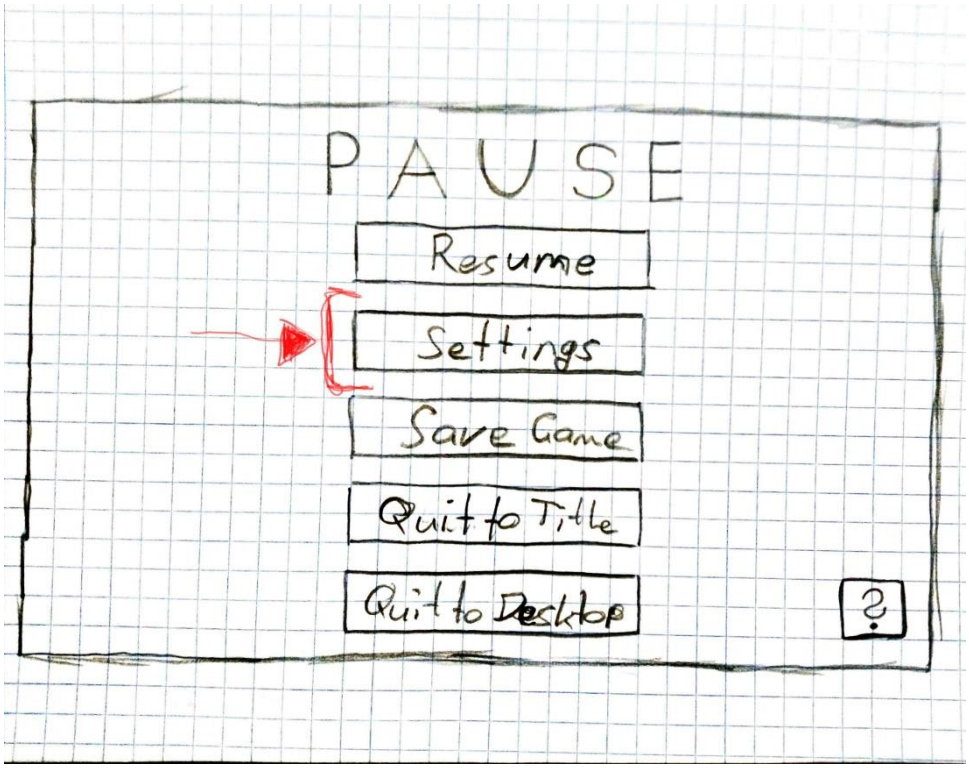


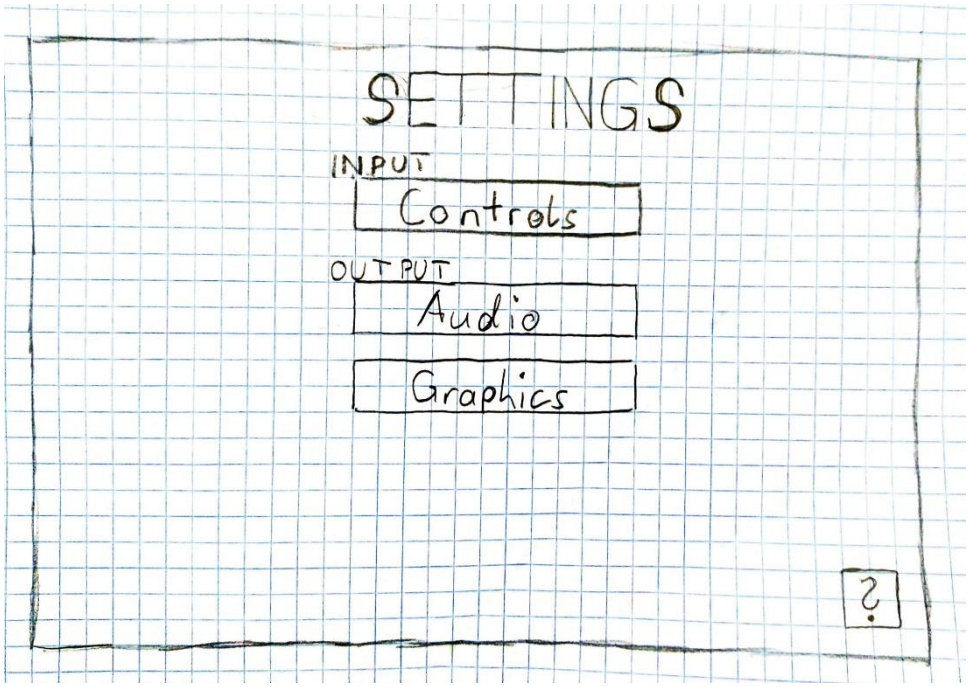
The user should be able to change various settings such as the graphic’s quality and computational intensity, the controls and the volume of the sound effects.

2.3.1 **Characteristic Information**

Goal:	Change settings like graphics or sound until they fit the user's needs.
Precondition:	The player is either in main menu or the game is paused.
Postcondition:	The changed settings are applied.
Involved User:	Player
Triggering Event:	The "Settings" Button in the pause menu or main menu is clicked.

2.3.2 GUI to call the use case



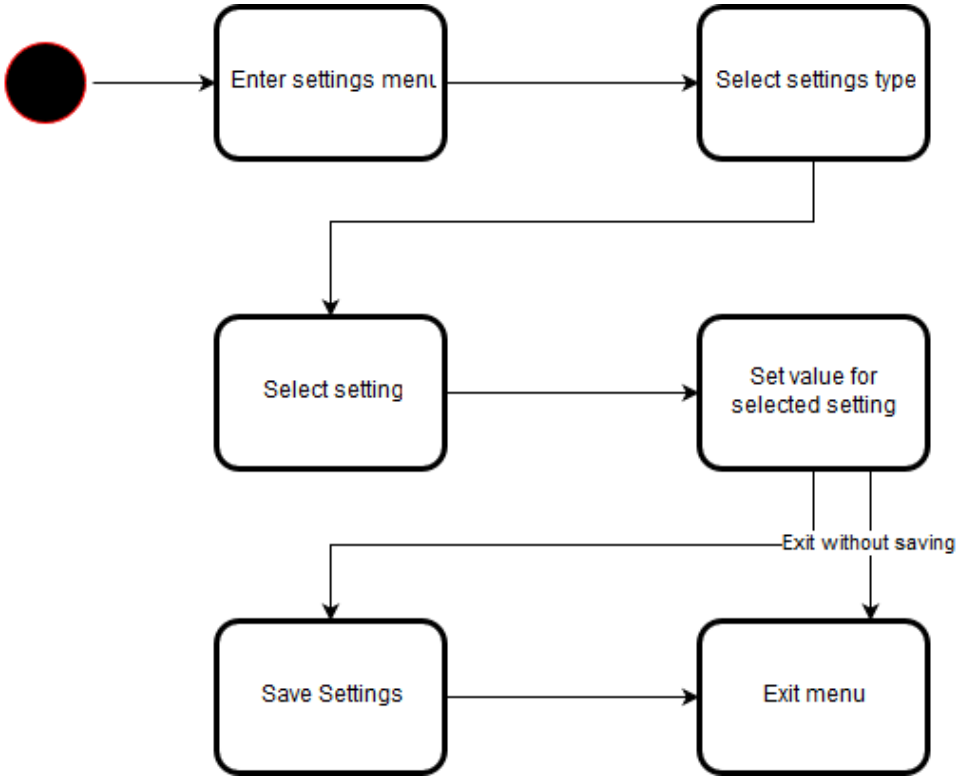


2.3.3 Scenario for the standard use (good case)

Step	Activity
Step 1	Enter the settings menu via the pause or main menu
Step 2	Select the type of settings you want to change (Controls, Audio, Graphics)

Step	Activity
Step 3	Select the specific Setting(s) you want to change
Step 4	Set it to the desired value
Step 5	Save settings
Step 6	Exit menu

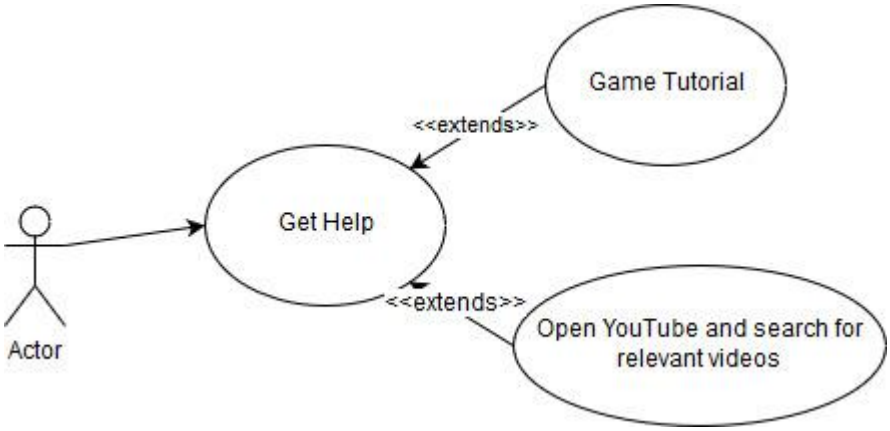
2.3.4 Workflow



2.3.5 Open Points

- Which settings should be changeable? This will be determined in the future, because it depends a lot on the software side of the project.

2.4 Get Help/Game Tutorial – Use Case Details

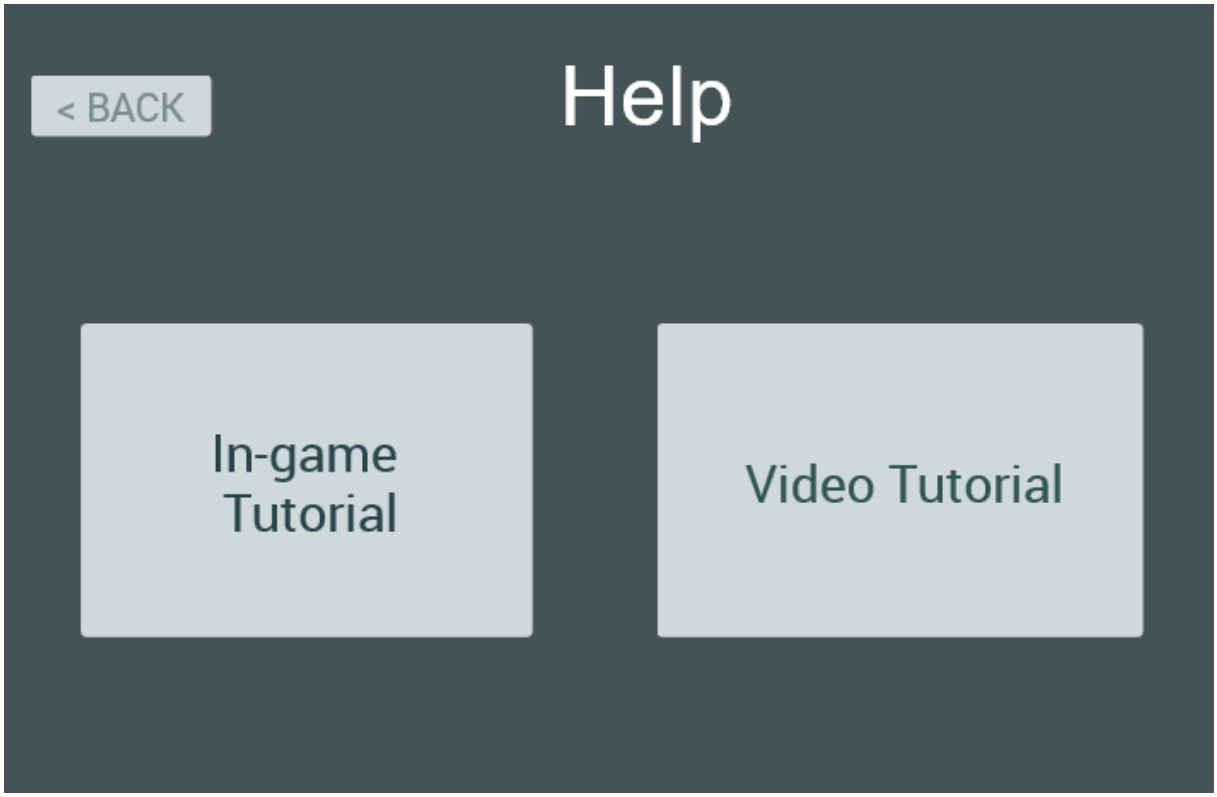


The user can get help on how to play the game. He will be able to choose between a game tutorial or opening a relevant YouTube video.

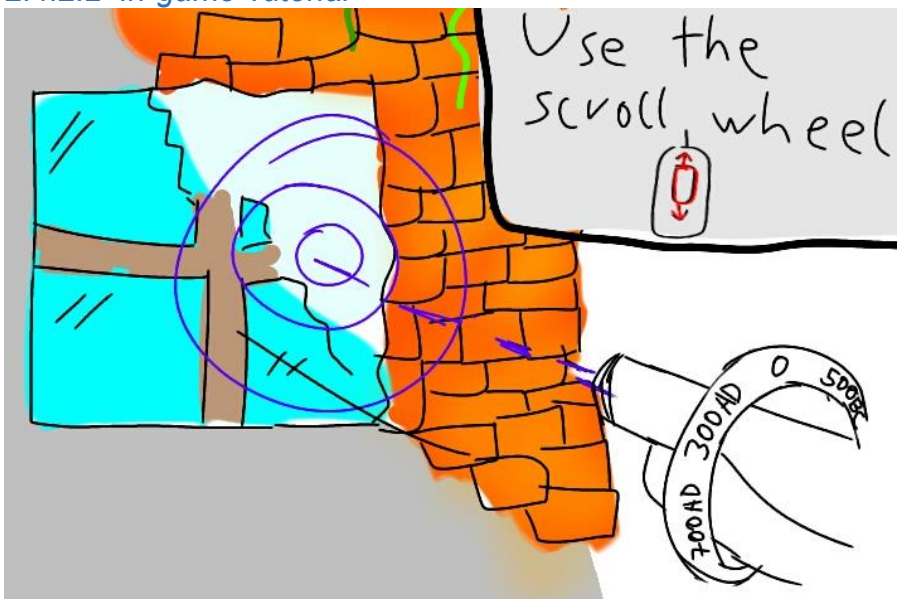
2.4.1 Characteristic Information

Goal:	To help the user with the basics of the game such as the controls.
Precondition:	The player does not know how to play the game.
Postcondition:	The player is capable of playing the game. This does not mean that he knows how to solve every single level.
Involved User:	The player
Triggering Event:	When the user presses on a button in the main menu or when the user starts the game for the first time, he will see the “Get Help” menu.

2.4.2 GUI to call the use case



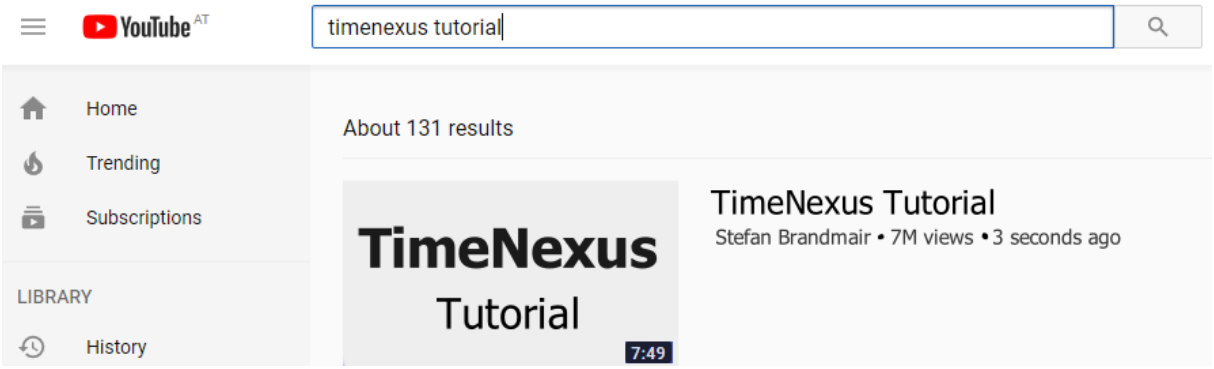
2.4.2.1 *In-game Tutorial*



2.4.2.2 *Scenario for the standard use (good case)*

Step	Activity
1	The „In-game Tutorial” button is pressed, which starts a tutorial game
2	The user finishes or skips the tutorial
3	The user presses the back button

2.4.2.3 *Video Tutorial*



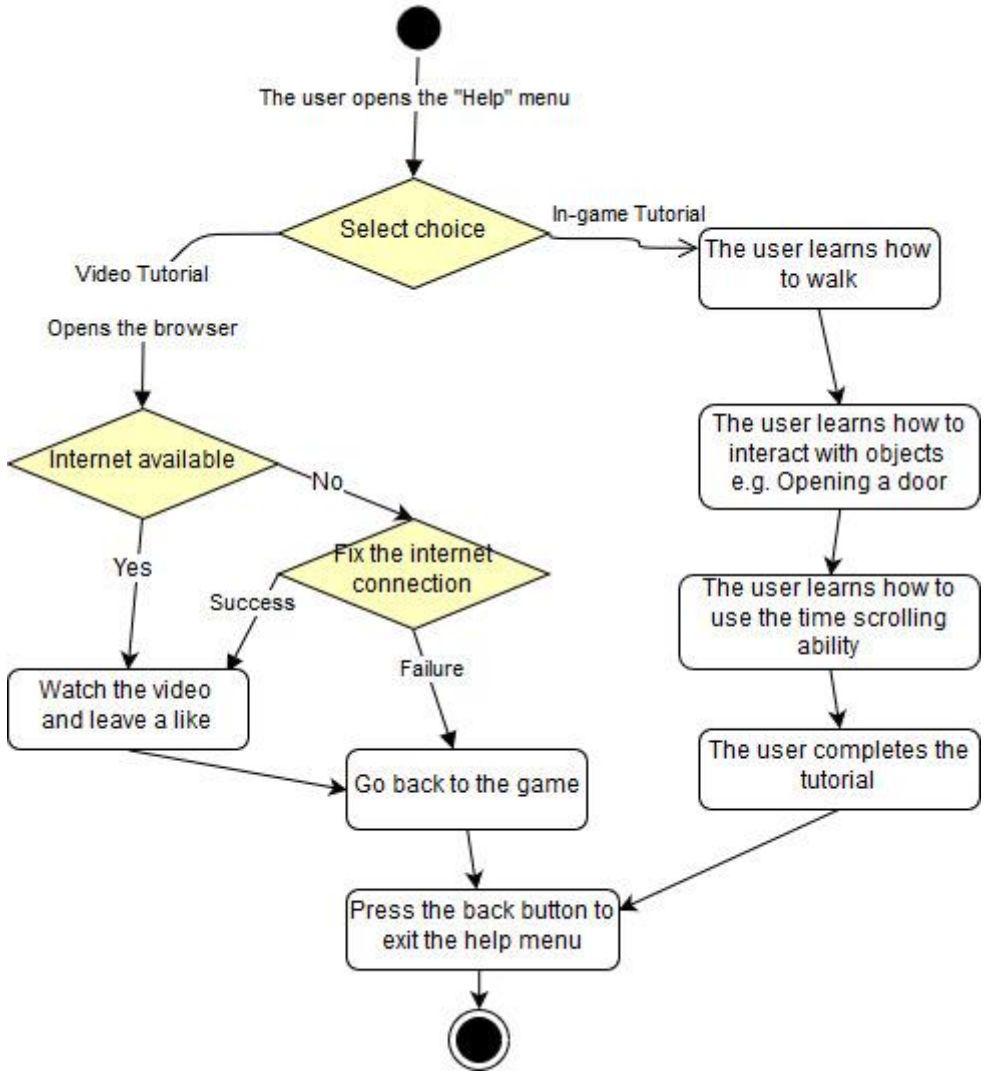
2.4.2.4 *Scenario for the standard use (good case)*

Step	Activity
1	The „Video Tutorial” button is pressed, which opens the browser
2	The user watches or skips the video tutorial
3	The user goes back to the game
4	The user presses the back button

2.4.2.5 *Scenarios for non-standard uses (bad cases or work around cases)*

Step	Activity
1	The browser is being opened but the user doesn’t have a working internet connection
2	Error message gets displayed

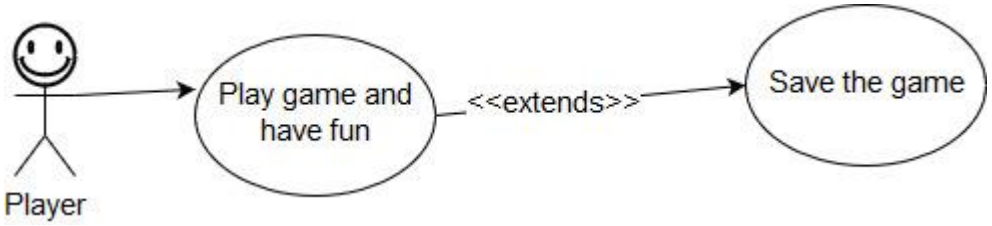
2.4.3 Workflow



2.4.4 Open Points

- Internet connection must be available
- The user must have a browser installed

2.5 Save - Use Case Details

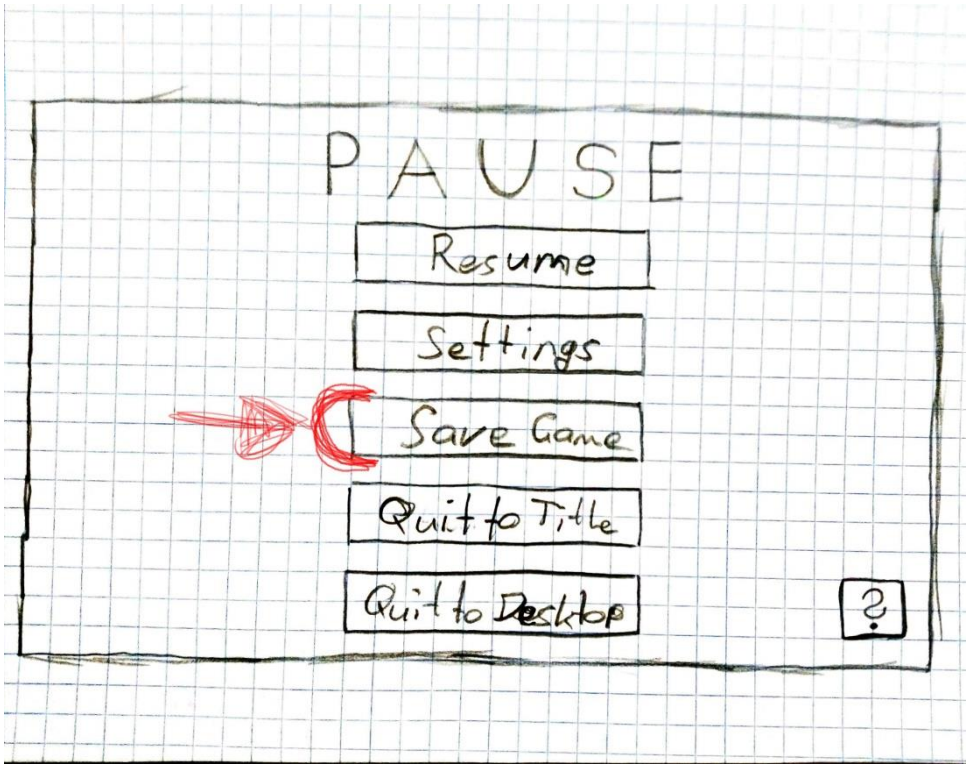


This use case serves for saving the game in order to keep the progress and reload it later.

2.5.1 Characteristic Information

Goal:	Save all relevant data of the current game
Precondition	The game is running.
Postcondition:	The game is saved.
Involved User:	Player
Triggering Event:	A click on the Save-button in the pause menu or reaching a point where the game saves automatically

2.5.2 GUI to call the use case



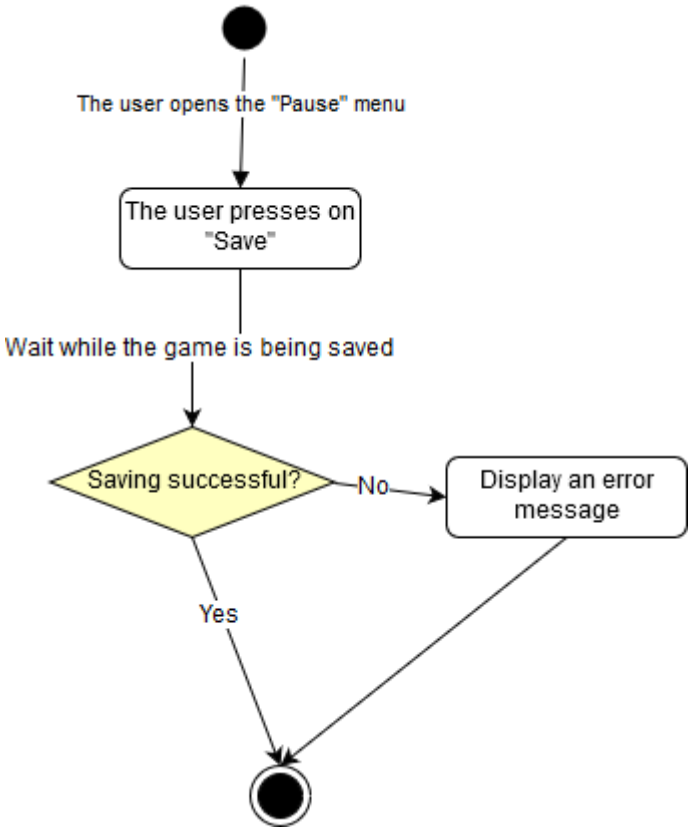
2.5.3 Scenario for the standard use (good case)

Step	Activity
Step 1	Press the save button

2.5.4 Scenarios for non-standard uses (bad cases or work around cases)

Step	Activity
1	The game cannot be saved because the file system doesn't let the user write stuff to a file. (Insufficient permissions, not enough space, etc.)
2	Error message gets displayed

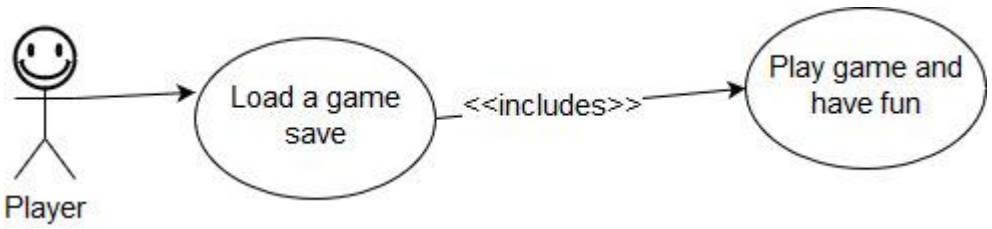
2.5.5 Workflow



2.5.6 Open Points

- Should the user be able to save the game himself or should the game have an auto-saving functionality?

2.6 Load/Start the Game - Use Case Details

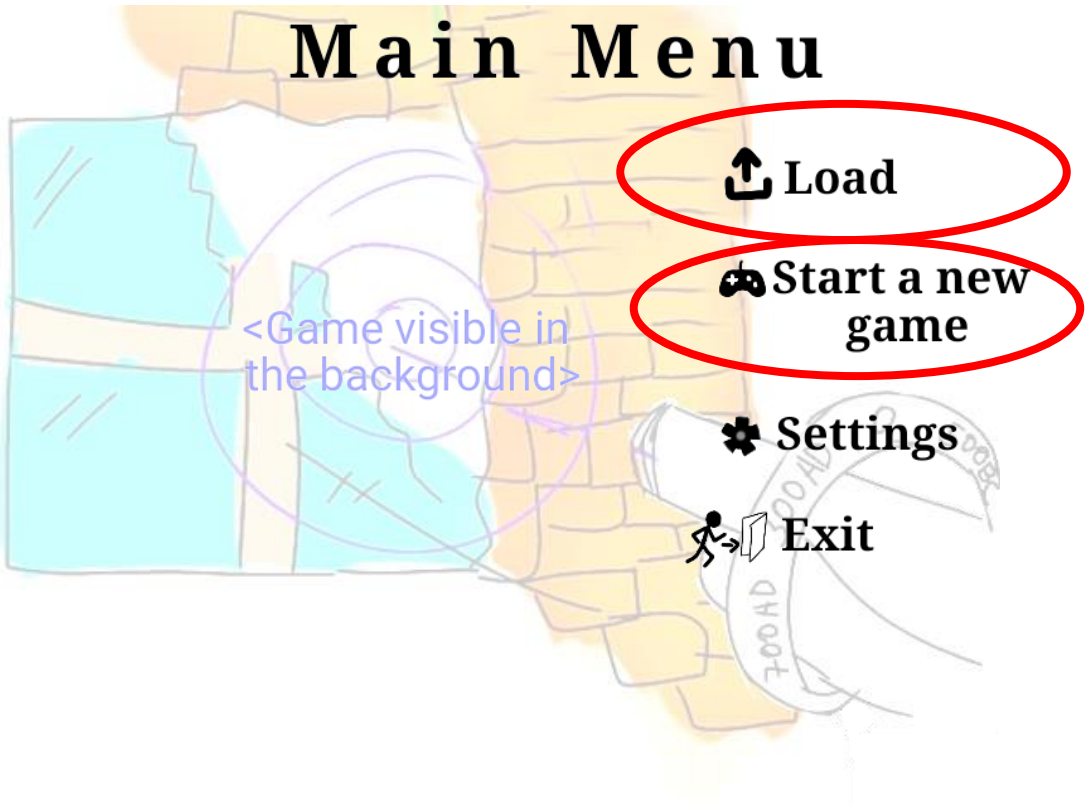


The user can start a new game. Alternatively, he can load a previously saved game save. If he chooses to do so, he will resume the game at the previously played level.

2.6.1 Characteristic Information

Goal:	The user should be able to load a game save.
Precondition:	The user has saved the game at some point.
Postcondition:	The user is where he saved the game.
Triggering Event:	The user pressed on the load game button in the main menu.

2.6.2 GUI to call the use case



2.6.3 Scenario for the standard use (good case)

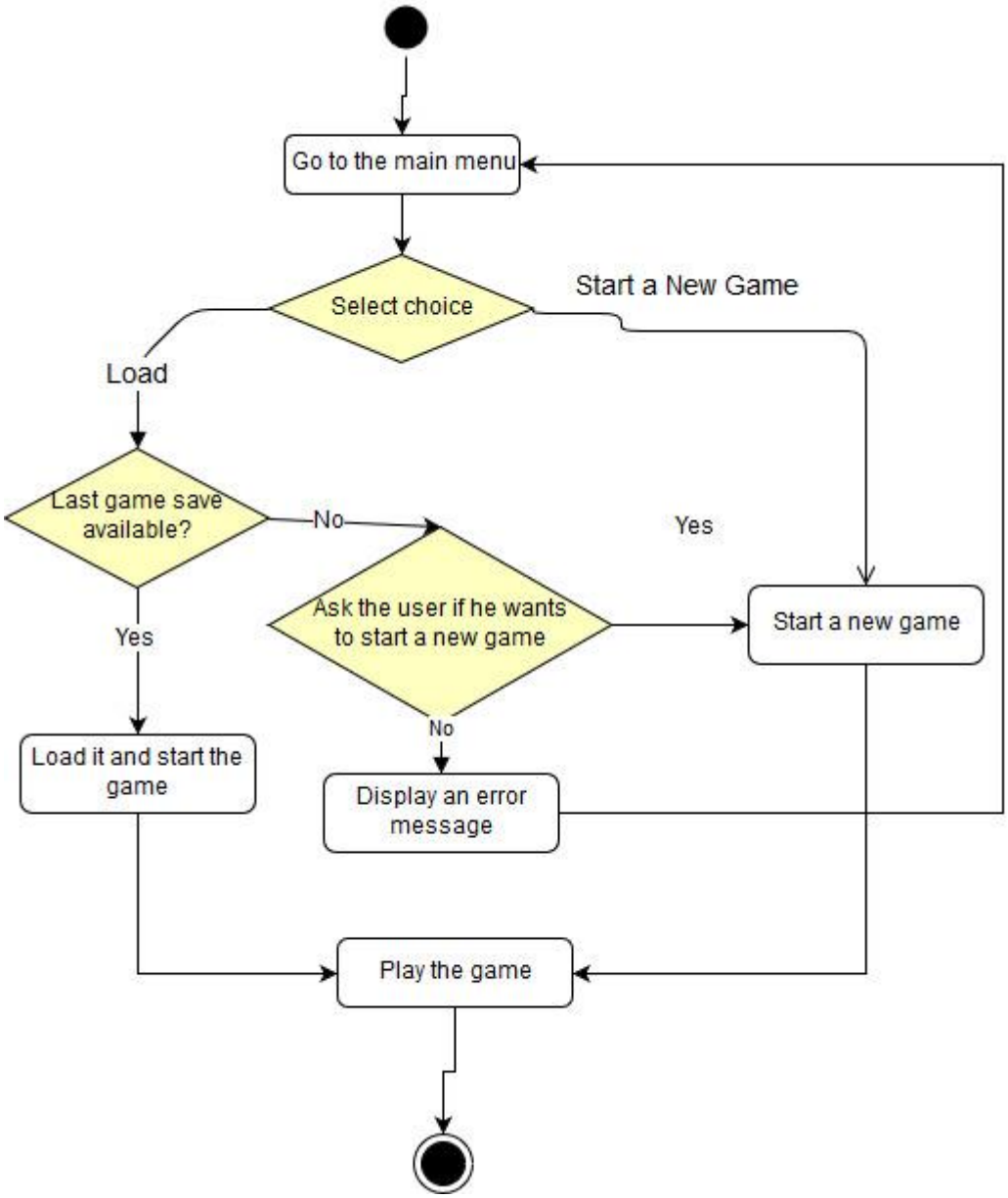
Step	Activity
1	Press on one of the buttons
2	Play the game

2.6.4 Scenarios for non-standard uses (bad cases or work around cases)

Step	Activity
1	The last game save cannot be loaded
2	Error message gets displayed

2.6.5 GUIs for the non-standard uses

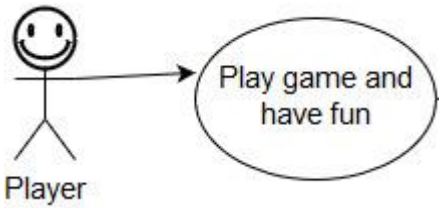
2.6.6 Workflow



2.6.7 Open Points

- Should it be possible to load older game-saves?

2.7 Play the game and have fun - Use Case Details



The player should be able to actually play the game. This is the main use case.

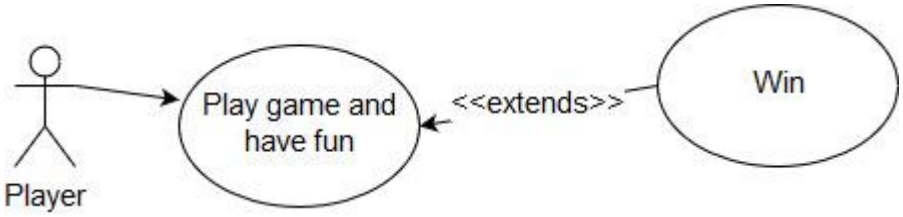
2.7.1 Characteristic Information

Goal:	Making the player happy and glad that he bought this game.
Precondition:	The user wishes to play the game.
Postcondition:	The user is done with playing the game.
Triggering Event:	The user started a new game or loaded an existing game save.

2.7.2 Open Points

- The game still has to be developed.

2.8 Winning - Use Case Details

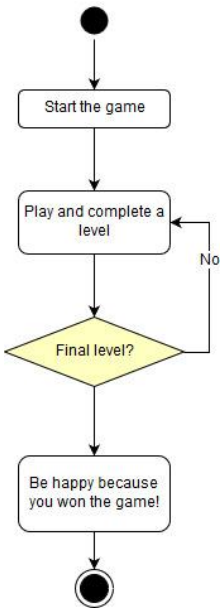


It should be possible to complete the entire game and win.

2.8.1 Characteristic Information

Goal:	Winning is the goal of the game itself.
Precondition:	The user is playing the final level of the game.
Postcondition:	The user completed the game.
Triggering Event:	Completing the final level

2.8.2 Workflow



3 Non-functional Requirements

ID:	NFR_001
Name:	Frame Time
Type:	EFFIC
Description:	The game should run with at least 30 FPS on most systems. This equates to about 33 milliseconds per frame.
Assigned use cases:	Play the game and have fun

ID:	NFR_002
Name:	Lag
Type:	USE
Description:	The FPS have to be consistent while playing.
Assigned use cases:	Play the game and have fun

ID:	NFR_003
Name:	Languages
Type:	USE
Description:	The game should have support for English and German.
Assigned use cases:	All

ID:	NFR_004
Name:	Languages
Type:	MAINT
Description:	It should be possible to implement another language simply by adding a language file with all needed texts.
Assigned use cases:	All

ID:	NFR_005
Name:	Copyright
Type:	LEGAL
Description:	The game may not violate any licence terms or other restrictions of used assets.

Assigned use cases:	All

ID:	NFR_006
Name:	Content
Type:	LEGAL
Description:	The game may not contain any unconstitutional ideology (i.e. symbols which look something like this Chinese Character: 卍)
Assigned use cases:	All

ID:	NFR_007
Name:	Transparency
Type:	MAINT
Description:	Sections of the code that are difficult to understand or messy shall be clearly commented to help with debugging.
Assigned use cases:	All

ID:	NFR_008
Name:	Levels
Type:	MAINT
Description:	Levels should have an uniform interface to add them into the game.
Assigned use cases:	Load/Start the Game

4 Quantity Structure

The game will have hardware requirements that are similar to those of other 3D videogames that have been released in 2017, because we aim to have graphics that is up to modern standards.

Processor:

64 bit due to game engine requirements. However, this is not an issue, because these days, 32 bit computers are becoming rare.

Intel Core i5-2400s @ 2.5 GHz because it will be a fairly resource intensive, 3D game. However, most gaming PCs have better CPUs.

Graphics:

OpenGL 3 capable GPU due to game engine requirements.

Memory:

4 GB of RAM (minimum) due to game engine requirements.

Hard Drive Space:

The majority will be taken up by the assets such as images and 3D models. An average compressed 3D model is about 10MB. The code for the game should take up at most 500 MB. For starters, we are planning to make about 10 levels with various assets.

50 shared assets: 50*10 = 500 MB

20 assets per level: 20*10*10 = 2000 MB

Code: 500 MB

Total: **3000 MB** = 3 GB (gigabyte)

Taking the average gamer PC specifications [3] into account, the game should run with at least 30 FPS on the average gamer PC.

5 Acceptance Criteria

5.1 AC_001 – Pausing the game

Test Step	Expected Behaviour
Press the pause button	The pause screen gets displayed. The game, including the physics and music, is frozen.

5.2 AC_002 – Changing settings

Test Step	Expected Behaviour
Change the sound settings	The user should be able to change the volume of the background music, the ambient or environment sounds, the player's voice and various other sounds. It should be possible to change all of them individually.
Change the graphics settings	The user can choose between a few graphics settings such as fancy, good and fast.
Change the controls	The user can remap every single control.

5.3 AC_003 – Get Help/Game Tutorial

Test Step	Expected Behaviour
Open the tutorial video	A tutorial video that explains how to play the game should be opened in the default browser. At the end, the user should be able to play the game.
Play the in-game tutorial	A temporary tutorial game should get started. The user should see various hints and explanations that will help him complete the tutorial. At the end, the user should be able to play the game.

5.4 AC_004 – Saving and loading

Test Step	Expected Behaviour
Saving	A saving icon appears and the current progress gets saved.
Loading	The player should be where he previously saved the game.

5.5 AC_005 – Playing the game and having fun

Test Step	Expected Behaviour
Playing the game	The user should play the game for at least 30 minutes before stopping and doing something else. He should enjoy playing the game.

5.6 AC_006 – Winning the game

Test Step	Expected Behaviour
Completing the final level	The user won the game. He is presented with a “You Won” screen. After that, he will see the game’s credits.

6 References

[1] „Quantum Condrum,“ February 2018. [Online]. Available: http://store.steampowered.com/app/200010/Quantum_Conundrum/.

[2] „Project Temporality,“ February 2018. [Online]. Available: http://store.steampowered.com/app/290320/Project_Temporality/.

[3] „Steam Hardware Survey,“ February 2018. [Online]. Available: http://store.steampowered.com/hwsurvey/.