|  |
| --- |
|  |
| Generic smartphone game controller  **Bachelor thesis** |
| Degree programme: Computer Science  Author: Quentin Flückiger  Thesis advisor: Prof. Marcus Hudritsch  Expert: Eric Dubuis  Date: 11 March 2019 |

Management Summary

Our first job was to learn about the architecture and possibility of Networking within the Unity3D API as it was the one chosen because of its “simplicity” and its cost.

Content

[Generic smartphone game controller 1](#_Toc11255995)

[Management Summary 2](#_Toc11255996)

[1 Introduction 4](#_Toc11255997)

[2 Project Management 5](#_Toc11255998)

[2.1 Vision 5](#_Toc11255999)

[2.2 Goals 5](#_Toc11256000)

[2.3 System Context 5](#_Toc11256001)

[2.4 Requirements 6](#_Toc11256002)

[2.4.1 Legend and additional information: 6](#_Toc11256003)

[2.4.2 Functional Requirements 6](#_Toc11256004)

[2.4.3 Technical Requirements 7](#_Toc11256005)

[2.4.4 Quality Requirements 7](#_Toc11256006)

[3 Phase 1 7](#_Toc11256007)

[4 Phase 2 7](#_Toc11256008)

[5 Phase 3 7](#_Toc11256009)

[6 Phase 4 7](#_Toc11256010)

[7 Conclusion and future work 8](#_Toc11256011)

[7.1 Results 8](#_Toc11256012)

[7.2 Improvements / Future work 8](#_Toc11256013)

[8 List of illustrations 9](#_Toc11256014)

[9 Contents of the table 9](#_Toc11256015)

[10 Bibliography 9](#_Toc11256016)

[11 Declaration of Authorship 10](#_Toc11256017)

# Introduction

Games, something that has travelled throughout history and is part of many cultures. They are even found in nature, as most animals learn by playing when they are newly born. It is a very old pedagogic way to teach, one that doesn’t even need a common language. Humanity pushed this process even further by developing board games, games that can sharpen the mind or which are there only to spend some time. The oldest board game, supposed to be the oldest, is the “Senet”. A game from Ancient Egypt created approximatively in 3’500 BCE. The Senet isn’t played anymore, as the rules where lost at some point in history, but another very old game is still played in our time. The “Go”, a game who saw birth in Ancient China as far back as 2’000 BCE and is still widely played and popular. The more commonly known game of “Chess” is only approximatively 1’400 years old.

With the technology produced by humankind, it allowed the human to develop other ways to play games, on computer, console and now smartphone as well, but we never stopped producing board games and inventing new one either. We even created Pen and Paper games, which are story told by one person where others, the players, interact by giving vocal input. And the popularity of games has been increasing incredibly, even more with smartphone as they allow people to play it very easily. Just a download and then hop it’s in the pocket, they can play it everywhere. “There are approximately 2.2 billion gamers in the world. Out of the estimated 7.6 billion people living on earth, as of July 2018, that means almost a third of people on this planet are gamers.” (Gaimin, 2018) Those numbers are crazy, and one can see the market opportunity in this field. But what hasn’t been made yet? What could be something new? Something interesting? Isn’t it possible to take benefits from both board games and video games?

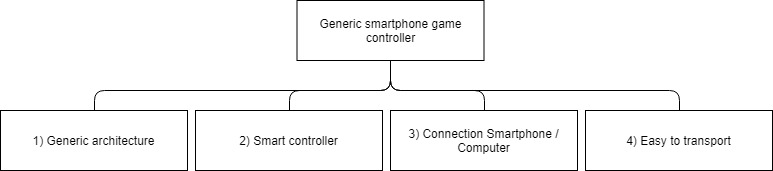
Those were the questions we asked our self and that led to the idea of this project.

Project Management

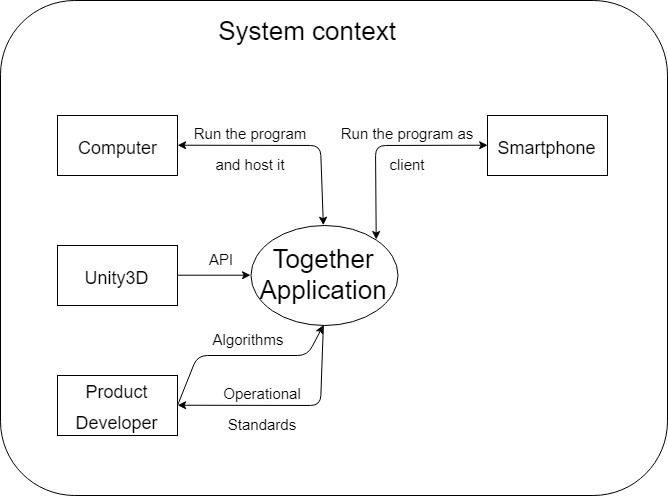
## Vision

Creating a generic gaming software/platform/library which will use a computer/tv screen as the view and smartphone as a controller to allow people to play together at video games without having to carry, or buy, a heavy controller. Most of the people have a smartphone which can be used as a substitute and can even be made “smart”. This library will allow games to be easily built with its component (network, controller, …).

## Goals



## System Context



## Requirements

### Legend and additional information:

Table columns definition

• Nr := identification number

• P := Priority (3 levels expressed in with numbers: 1;2;3)

• V := Variability (3 levels expressed in with numbers: 1;2;3)

• C := Complexity (3 levels expressed in with numbers: 1;2;3)

• R := Risk calculated from {P;V;C} (numerical value)

• Status := planned (will be done) / stand by (nice to have) / done

• Goals := goals reference(s)

### Functional Requirements

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Number** | | **Description** | **Status** | **P** | **V** | **C** | **R** | **Goals** |
| 1 | | Connection |  |  |  |  |  |  |
|  | 1.1 | Connection from smartphone to computer. | Approved | 3 | 1 | 2 | 1.83 | 3 |
| 2 | | Build generic |  |  |  |  |  |  |
|  | 2.1 | The system must be designed to allow for optimal extensibility and reusability. | Approved | 3 | 3 | 3 | 3 | 1 |
| 3 | | Extendable |  |  |  |  |  |  |
|  | 3.1 | Dynamic HUB which accept new games | Approved | 2 | 1 | 3 | 2.17 | 1 |
| 4 | | Smartphone possibility |  |  |  |  |  |  |
|  | 4.1 | Exploit the components of the smartphone to deepen the singularity of the concept. | Approved | 3 | 1 | 1 | 1.33 | 2 |
| 5 | | User Interface (UI) |  |  |  |  |  |  |
|  | 5.1 | Design and implement a UI that allows the user to interact with the system on smartphone using touch technology. | Approved | 3 | 1 | 1 | 1.33 | 2 |
|  | 5.2 | Design and implement a UI that allows the user to interact with the system on computer. | Approved | 3 | 1 | 1 | 1.33 | - |

### Technical Requirements

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Number** | | **Description** | **Status** | **P** | **V** | **C** | **R** | **Goals** |
| 1 | | Software |  |  |  |  |  |  |
|  | 1.1 | C# | Approved | 3 | - | - | - | - |
|  | 1.2 | Unity3D 2018.3.7f1 or lower 2018 version. | Approved | 3 | - | - | - | - |

### Quality Requirements

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Number** | | **Description** | **Status** | **P** | **V** | **C** | **R** | **Goals** |
| 1 | | Platform compatibility |  |  |  |  |  |  |
|  | 1.1 | Compatibility with Android / Apple phones | Approved | 2 | 1 | 2 | 1.83 | - |

# Understanding Unity3D Network

Gathering ressources, knowledge

* Old projects, screenshot, beginning

The first step we took in our project was to decide which network technology we should use to achieve our goals. There were mainly two options, Photon and UNeT. As we never had any experience with Photon we decided to use UNeT. UNeT is the built-in technology from Unity, one can find the documentation and tutorials on the official Unity website. This technology is separated in two categories: High Level API (HLAPI) and Low Level API (LLAPI).

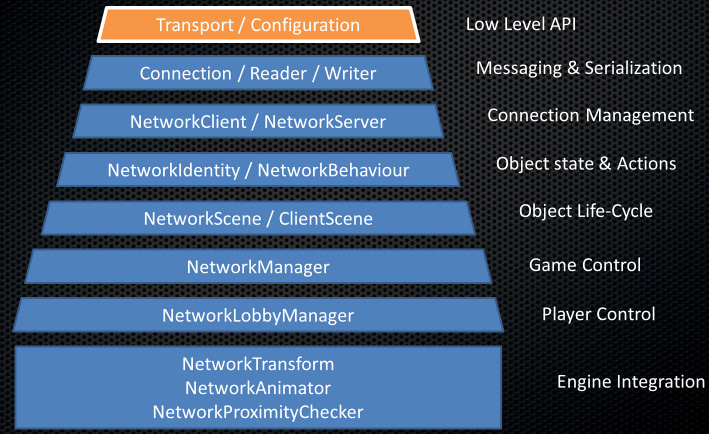


Figure 1: Unity3D Network Layers

The LLAPI, or Transport Layer API, let developer built their own networking system. It is best used when we are already experienced in network programming and as well when we need a very specific network. As it requires to be built from the very base.

The HLAPI is built on top of this layer and others which adds functionality. It is a set of networking commands which can be found in the namespace: UnityEngine.Networking. It allows the developer the same ease of use as the rest of Unitys component system as they can be added the same way as another component would be.

Between the two possibilities we choose to use the HLAPI because it would fasten the development of our software. If we were doing this project as our project two as well, we would have chosen the LLAPI as it would have suited us more to have an own made networking system with our own messages. Another reason why working with LLAPI would have been better is because of UNeT being deprecated in the near future and the only information Unity has been given is that the HLAPI would not be compatible but the new system would be close to the LLAPI allowing to go from the latest to the newest. But as we tried in the beginning of the project to develop a little example with the LLAPI we acknowledged that it would take us too much time.

For this example, we created two different projects, a server and a client, established a connection between a given port and sent some messages from one to the other. Bellow you will see the architecture of the initialisation of the server.

public void Init()

{

NetworkTransport.Init();

ConnectionConfig cc = new ConnectionConfig();

reliableChannel = cc.AddChannel(QosType.Reliable);

HostTopology topo = new HostTopology(cc, MAX\_USER);

hostId = NetworkTransport.AddHost(topo, PORT, null);

webHostId = NetworkTransport.AddWebsocketHost(topo, WEB\_PORT, null);

Debug.Log(string.Format("Opening connections on port {0} and web port {1}.", PORT, WEB\_PORT));

isStarted = true;

}

Afterwards on each update a method “UpdateMessagePump” would be called and its duty was to listen to any network event and act accordingly.

switch (type)

{

case NetworkEventType.Nothing:

break;

case NetworkEventType.ConnectEvent:

Debug.Log(string.Format("User {0} has connected through host {1}.", connectionId, recHostId));

break;

case NetworkEventType.DisconnectEvent:

Debug.Log(string.Format("User {0} has disconnected.", connectionId));

break;

case NetworkEventType.DataEvent:

BinaryFormatter formatter = new BinaryFormatter();

MemoryStream ms = new MemoryStream(recBuffer);

NetMsg msg = (NetMsg)formatter.Deserialize(ms);

OnData(connectionId, channelId, recHostId, msg);

break;

default:

case NetworkEventType.BroadcastEvent:

Debug.Log("Unexpected network event type.");

break;

}

As said before this would have been to time consuming for this project, so we decided to go with the easier to use HLAPI.

After choosing the technology we would be using for our project we started to browse forums, youtube videos, udemy courses and documentation on the Unity website to acquire knowledge and example material which would be used later on in the project.

Following tutorials, we created a very simple 2D prototype using the network manager and its HUD. It was a very simple one where the player was a cube which could move around 2 axis and fire at other player or boxes. The health and transform position would be synchronized. At this stage we could connect with the smartphone, but we would all see the same thing, be it on the computer screen or the smartphone one.

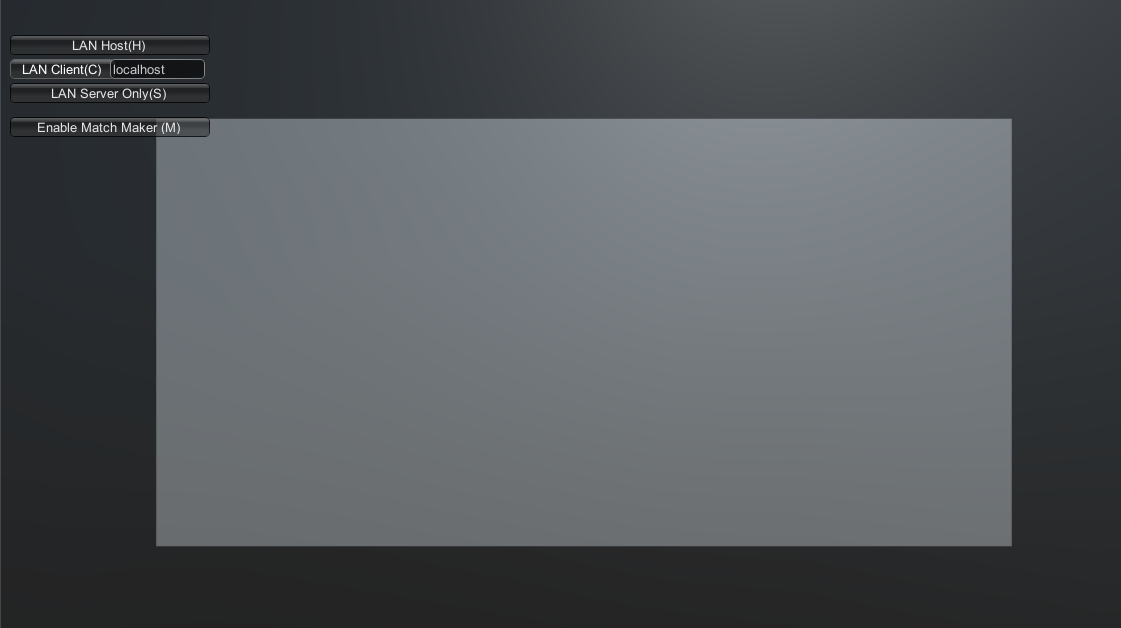


Figure 2: First try multiplayer lobby

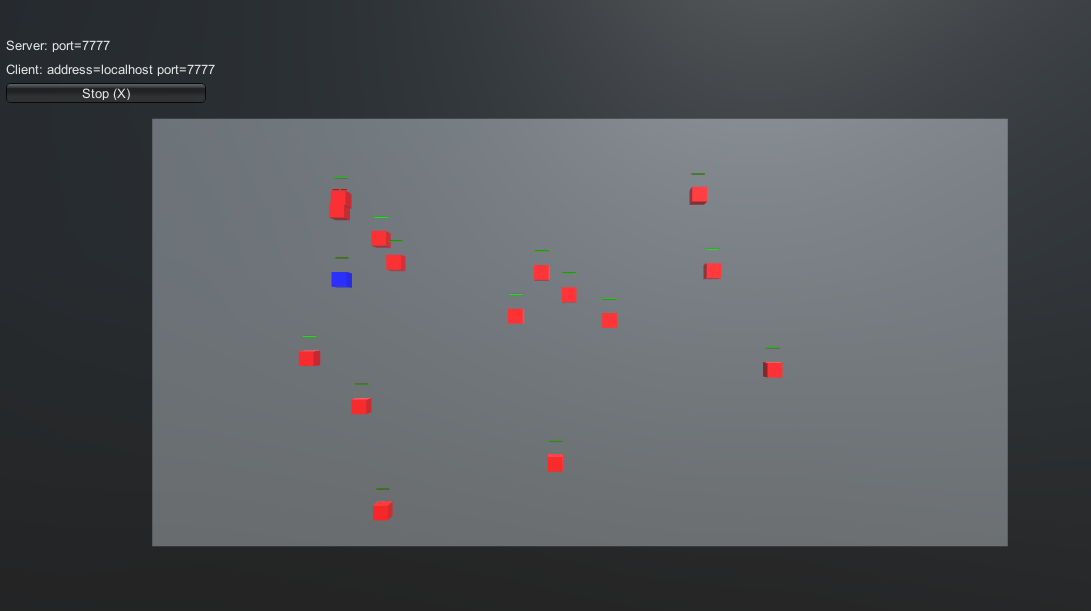


Figure 3: First try in game

# Phase 2

Snake

# Phase 3

Lobby

# Phase 4

Domino

# Conclusion and future work

## Results

## Improvements / Future work

# List of illustrations

[Figure 1: Unity3D Network Layers 7](#_Toc11271999)

# Contents of the table

[Table 1: Et ut aut isti repuditis qui ium 3](#_Toc371573636)

# Bibliography

**Literary Entry**

*Author’s name, Author’s first name, book title, publisher, place, edition, year* 7

**Literary Entry**

*Author’s name, Author’s first name, book title, publisher, place, edition, year* 9

**Literary Entry**

*Author’s name, Author’s first name, book title, publisher, place, edition, year* 11

# 

# Declaration of Authorship

I hereby certify that I composed this work completely unaided, and without the use of any other sources or resources other than those specified in the bibliography. All text sections not of my authorship are cited as quotations and accompanied by an exact reference to their origin.

Place, date:

Signature: