 BSHC(E)2, BSHBIS(E)2, HCC(E)2

Interdisciplinary Team Project

Sem.2 2015-2016

**Group 22**

**Technical Report**

Conor Grant - x13522167

Stacey Lynch - x14110890

Gavin Johnston - x13477082

Louise O’Connor - x13335131

Niamh Ní Bhriain - x14100274

**Table of Contents**

[Executive Summary 4](#_Toc386657037)

[Definitions, Acronyms, and Abbreviations 4](#_Toc386657038)

[1 Introduction 5](#_Toc386657039)

[1.1 Background 5](#_Toc386657040)

[1.2 Motivation 5](#_Toc386657041)

[1.3 Project Overview 5](#_Toc386657042)

[1.4 Target group 5](#_Toc386657043)

[1.5 Technologies 5](#_Toc386657044)

[1.6 Distribution of tasks 5](#_Toc386657045)

[1.7 Structure 5](#_Toc386657046)

[2 System 6](#_Toc386657047)

[2.1 Requirements 6](#_Toc386657048)

[2.2 User Requirements Definition 6](#_Toc386657049)

[2.3 Requirements Specification 6](#_Toc386657050)

[2.4 Functional requirements 6](#_Toc386657051)

[2.5 Use Case Diagram 6](#_Toc386657052)

[2.5.1 Requirement 1 <name of requirement in a few words> 6](#_Toc386657053)

[2.5.2 Requirement 2 <name of requirement in a few words> 8](#_Toc386657054)

[2.6 Non-Functional Requirements 9](#_Toc386657055)

[2.6.1 Performance/Response time requirement 10](#_Toc386657056)

[2.6.2 Availability requirement 10](#_Toc386657057)

[2.6.3 Recover requirement 10](#_Toc386657058)

[2.6.4 Robustness requirement 10](#_Toc386657059)

[2.6.5 Security requirement 10](#_Toc386657060)

[2.6.6 Reliability requirement 10](#_Toc386657061)

[2.6.7 Maintainability requirement 10](#_Toc386657062)

[2.6.8 Portability requirement 10](#_Toc386657063)

[2.6.9 Extendibility requirement 10](#_Toc386657064)

[2.6.10 Reusability requirement 10](#_Toc386657065)

[2.6.11 Resource utilization requirement 10](#_Toc386657066)

[2.7 Design and Architecture 10](#_Toc386657067)

[2.8 Implementation 10](#_Toc386657068)

[2.9 Graphical User Interface (GUI) Layout 10](#_Toc386657069)

[3 Discussion and reflection 11](#_Toc386657070)

[4 Conclusions 12](#_Toc386657071)

[5 Further Development 13](#_Toc386657072)

[6 References 14](#_Toc386657073)

# Executive Summary

Five individuals were given the task to generate an idea (I.T based) and to then implement that idea using technology they had never utilised before.

The idea: To create a family orientated, educational web application game.

This platform game would encourage children to complete typical daily tasks, bringing awareness to what steps are needed to complete a task from start to finish. It would target children aged between 4 and 10 years and would help them develop basic skills in a stimulating and playful environment.

The gaming market is increasing annually and educational games make up a huge percentage of that market. Not only can this unique web app be utilised by parents/guardians hoping to assist their children learn basic skills, but it has the potential to be used by playschools and assisted learning centres alike.

Unity was the chosen game engine decided upon for this project, why?

* Particular suited for 2D platform games
* Available for all group members to download free of charge
* Plenty of instructional videos & help forums available online
* Multiplatform- deploy across the web and all major mobile, console, desktop, VR& TV platforms



*Unity deployable platforms*

https://unity3whatd.com/unity/multiplatform

Because of its compatibility with Unity SQLite Manager was chosen as the management system for the database because of its speed, ease of use and advanced features.

All game graphic were created or edited in PS and saved as .png files so they could be imported into Unity.

Maximum 300 words. The abstract should mention the problem being addressed, describe the technical solution.

## Definitions, Acronyms, and Abbreviations

**Unity:** Unity (2D) a flexible & high end game solution particularly suited to creating platformgames**.**

**C#:**C# is a programming language used in unity

**HTML:** HTML(hypertext mark-up language) is a standardized system for creating world wide web pages that would allow us to showcase our game.

**CSS:** CSS(Cascading Style Sheets) is a style language that defines the layout of HTML documents.

**SQL Light Manager:** SQL (Structured Query Language) is a programing language with a special purpose for designing code found in a relational database management system.

**PS:** Photoshop is a digitally altering/ image-editing software.

**Web app:** Web application is a client-server software application which the user runs in a web browser.

# Introduction

This template for technical report is provided for your convenience. It should be seen as a guide rather than an obligatory form. Your individual report might require changes in terms of format or content (i.e., headings) or both.

Print on one side of the paper only (this will be the right hand side when the pages are bound).

## Background

Details about the chosen area

## Motivation

Why?

## Project Overview

What?

## Target group

## Technologies

How? - Brief description of the technologies used in the project. Do not copy & paste descriptions from websites here, but describe what it is and how it contributes to your project.

## Distribution of tasks

## Structure

Brief overview of each section

# System

## Requirements

## User Requirements Definition

This section describes the set of objectives and requirements for the system from the customer’s perspective. What are the clients saying they want?

## Requirements Specification

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

## Functional requirements

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system, in other words, what the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe how the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

Short, imperative sentence stating highest ranked functional requirement.

## Use Case Diagram

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

The Use Case Diagram provides an overview of all functional requirements.

### Requirement 1 <name of requirement in a few words>

The heading of this section should read, e.g., “Requirement 1: User registration” or “Requirements 1: Participant takes test”

#### Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

#### Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

**Scope**

The scope of this use case is to …….

**Description**

This use case describes the ………..

**Flow Description**

**Precondition**

The system is in initialisation mode……..

**Activation**

This use case starts when an <Actor>…………

**Main flow**

1. The system identifies the ………….
2. The <Actor> …………...(See A1)
3. The system …………..(See E1)
4. The <Actor> ………….

**Alternate flow**

A1 : <title of A1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 3 of the main flow

**Exceptional flow**

E1 : <title of E1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 4 of the main flow

**Termination**

The system presents the next ……….

**Post condition**

The system goes into a wait state

### Requirement 2 <name of requirement in a few words>

#### Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

#### Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

**Scope**

The scope of this use case is to …….

**Description**

This use case describes the ………..

**Flow Description**

**Precondition**

The system is in initialisation mode……..

**Activation**

This use case starts when an <Actor>…………

**Main flow**

1. The system identifies the ………….
2. The <Actor> …………...(See A1)
3. The system …………..(See E1)
4. The <Actor> ………….

**Alternate flow**

A1 : <title of A1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 3 of the main flow

**Exceptional flow**

E1 : <title of E1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 4 of the main flow

**Termination**

The system presents the next ……….

**Post condition**

The system goes into a wait state

List further functional requirements here, using the same structure as for Requirements 1 & 2. Most systems would have at least five main requirements.

## Non-Functional Requirements

### Performance/Response time requirement

After logging in and choosing a character, the game character responds to particular keyboard keys being pressed (this is all this is done in real time)

* Walk = left right arrow keys
* Jump = space bar key
* Attack = shift keys
* Climb = up and down arrows

### Availability requirement

Taskorama is available to everyone who has internet access. It is aimed at small children. The child /children’s parents may want to test the game out first but anyone who would like to play the game, can do so, provided they log in first. By loggin in they can save their scores and track their progress.

### Recover requirement

Taskorama backs up user details from the log in and will back up user’s progress in real time. This will help the user in case of any change in game playing that may result in loss of play. The game will back-up the stage of the game automatically letting the user start off from the last point of play.

### Security requirement

Taskorama has a log in requirement, therefore some personal detail is needed. An encryption on the server will prevent outside sources from gaining access to personal information belonging to the user. This will be implemented before the game goes live.

### Reliability requirement

The game code is free of errors. We will continue to runs tests to ensure it is accessible & preforms correctly on all browsers. This will be all done before the game goes live.

### Maintainability requirement

Taskorama will be maintained by members regularly checking the code, insuring the game is error free. Monitoring Taskorama will mean that the user will have an enjoyable experience. Online support & feedback options will be available to them via the website, if they do experience any issues or have any queries.

### Portability requirement

Taskorama will be compatible with most, if not all, web browsers. This means convenience for the user and unlimited access, provided they have internet access and a device.

### Extendibility requirement

To date we have three levels (tasks) that range in difficulty, available in Taskorama. Not only we will extend the amount of levels (tasks) but will offer templates, allowing parents/guardians, teachers, assistants the opportunity to customise specific tasks to suit a particular child/children. Taskorama will have basic levels. Reusability requirement

### Resource utilization requirement

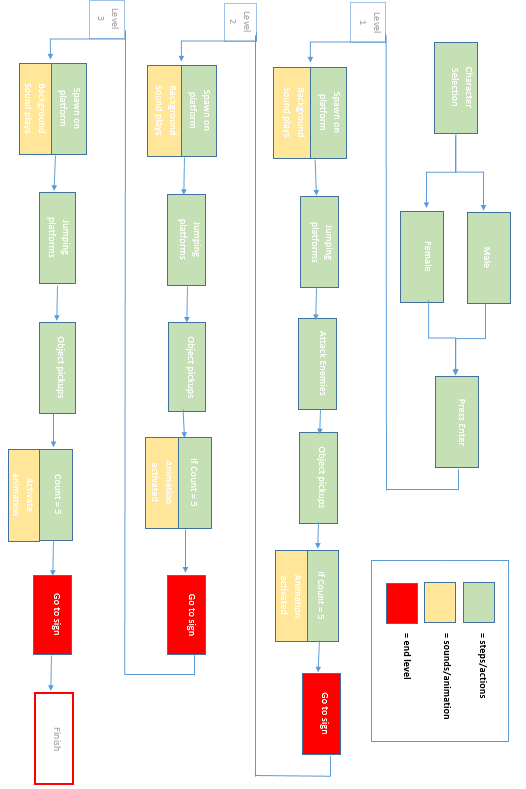
In order for the game to be played, the user will need to have a working device and internet connection.

## Design and Architecture

Describe the design, system architecture and components used. Describe the main algorithms used in the project. (Note use standard mathematical notations if applicable).

An architecture diagram may be useful. In case of a distributed system, it may be useful to describe functions and/or data structures in each component separately.

## Implementation



Describe the main classes/functions used in the code. Draw the Data flow diagram

Consider to show and explain interesting code snippets where appropriate.

## Graphical User Interface (GUI) Layout

Provide screenshots of key screens and explain.

# Discussions and reflections

Discussions and reflections regarding:

Learning – what you learned

Skills – what skills you developed

Process – the Project development process

Client Feedback Session

Project Module

**Discussions and reflections regarding:**

we were told that we had to do a web/mobile application but the thing was that the application had to be something that we haven’t done before. As a team, it was decided that a game application will be done as no one in the group had ever made a game application before.

**Learning – what you learned**

The big thing that was learned was Unity. This was a game engine that allows a person build a game in 3d of 2d. This was the main thing used in Taskorama. Unity proved a small bit tricky as we had to go off different YouTube videos. We got there in the end and came up with the Taskorama game. We also all learned how to work with new people, as a group. Working as a team, there needs to be participation by all team members. When working with people that is not known to you, it’s hard to know what they are like and how they work. Our group was fortunate as all members worked together well. The skills what was learned in doing this game was using Unity.

The game came together in the end. We came across some problems along the way. The most was just syntax errors, silly little errors that would stop the play and would be hard to find. With going off tutorials, there was some contradictions with the different tutorials that were watched but with a little bit of help from other group members, they were fixed. There was times when the game just wouldn’t work. Checking over the code, having other people to check over the code with no errors showing and still the game wouldn’t do what was needed to be done. This had one member stuck for a period of time. In the end it was a small little box that had to be ticked in Unity itself.

The game was done in stages. The first stage was getting the game player together. All sprites had to be put together and all movement attached to the sprites. While this was getting done, the garden scene graphics were getting assembled and the user log-in was created. These were shown in the prototype presentation.

The next stage was getting everything that was needed in the game done but all separately. The bathroom scene, the snow scene and the database were all done.

The final stage was putting them all together. This was proving a bit difficult as there was conflicting code. But all worked out in the end.

Taskorama was tested and feedback was great. The clients really enjoyed playing the game. They liked that the ninja could go underground, and kill the enemy. Also the way the platforms moved and the ninja would fall if it didn’t move with the platform.

# Conclusions

Without question, at one time or another, all group members experienced issues in completing the project. Time delays, technical problems & compiling errors played its part in the process.

Although creating a game application was a daunting prospect, particularly considering no member of the group had any prior gaming experience, there is no question that it has been very rewarding exercise.

# Further Development

With more resources, where could the results of this project lead to?

# References

It is recommended that students use the Harvard style:

References to web sites must include the access dates.

The library provides a study guide on Harvard style referencing.