Author: Michael Edgar

Supervisor: Robert Sheehy

Second Reader: Paul Collins

An Evaluation of the use of an adaptive controller with an eye tracker to facilitate accessibility in gaming

B.Sc. (Hons) in Computing with Games Development

Table of Contents

[Abstract 2](#_Toc35801792)

[1 Introduction 2](#_Toc35801793)

[2 Video Games and Accessibility 3](#_Toc35801794)

[2.1 Introduction 3](#_Toc35801795)

[2.2 Video Games 3](#_Toc35801796)

[2.3 Accessibility and Technology 5](#_Toc35801797)

[2.4 Case Study 1: Xbox Adaptive Controller 5](#_Toc35801798)

[2.5 Case Study 2: Tobii Eye Tracker 4C 6](#_Toc35801799)

[2.6 Case Study 3: Google Stadia 6](#_Toc35801800)

[3 Sprints 7](#_Toc35801801)

[3.1 Eye Tracking Unity Game 7](#_Toc35801802)

[3.2 Voice and Eye control in Windows 8](#_Toc35801803)

[3.3 Raspberry Pi Research 9](#_Toc35801804)

[3.4 Python/C# Eye Tracker Implementation 10](#_Toc35801805)

[4 Methodology 12](#_Toc35801806)

[4.1 Research Undertaken 12](#_Toc35801807)

[4.2 Research Question 12](#_Toc35801808)

[4.3 Proposed Project Implementation 12](#_Toc35801809)

[4.4 System Design 12](#_Toc35801810)

[4.5 Prototype 12](#_Toc35801811)

[5 References 13](#_Toc35801812)

# Abstract

Ignore for now

# Introduction

# Video Games and Accessibility

## Introduction

Video games are a form of interactive media that involve a player interacting with an artificial world on a computer. Accessibility is defined as how easy something is to reach, enter, use, see, etc. (Oxford University Press, 2019). Accessibility can be linked with people that have disabilities. To ensure accessibility for as many people as possible, many countries have laws in place to prevent the exclusion of those with disabilities.

In video games, accessibility can relate to the design of the controller, the use of flashing images, the inclusion or exclusion of subtitles, etc. Accessibility is something that is often overlooked in the design of video games. An example of where this is the case is referred to by Kevin Bierre et al. They present the example of a young boy with cerebral palsy who requires a specific adapted controller to play video games, but the boy in question struggles to find suitable video games as accessibility information is difficult to find (Bierre, et al., 2019).

## Video Games

A video game can be defined by the goal(s), the rules and the environment of the game. Video games are often categorised into various genres such as puzzle, first-person shooter, couch co-op, etc.

The first known video game was implemented by Josef Kates and was called ‘Bertie the Brain’ (Smith, 2014). Kates, who at the time was a consulting engineer in Toronto 1950, made the game as an exhibit for attendees of the 1950 Canadian National Exhibit (Simmons, 1975). The game was a simple Tic Tac Toe game in which the player would attempt to beat the computer by getting three consecutive O’s in a horizontal, vertical or diagonal line. This game was created to demonstrate power of computers (Smith, 2014).

A person standing in front of a store

Description automatically generated

Figure 2.2.1 Life magazine photo of Danny Kaye in front of Bertie the Brain at the Canadian National Exhibit in 1950 (Hoffman, 1950)

There were many games that followed Bertie the Brain, but the first video game created for entertainment purposes was ‘Tennis for Two’ (Smith, 2014). William A. Higinbotham created the game in 1958 for the Brookhaven National Laboratory (BNL) annual ‘Visitors Day’ to showcase the current research and development projects in the laboratory. Similar to Bertie the Brain, this game was purely an exhibition piece and was not released to the public. Tennis for Two ended up being dismantled after the 1959 Visitors Day (Nitray, 2015).

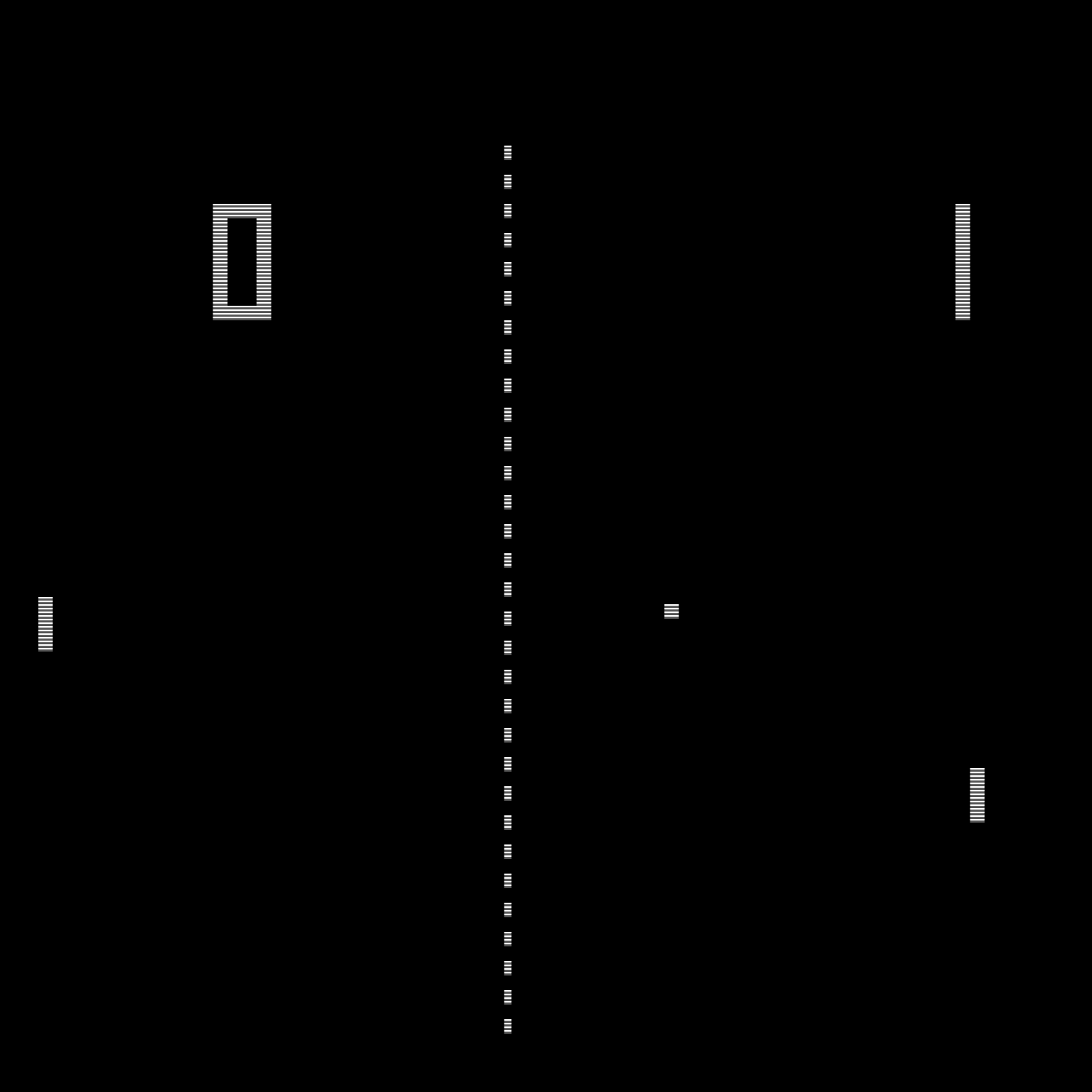
One of the first video games to be publicly released was Pong; based on the popular sport Ping Pong. The game consisted of two player-controlled panels and a ball that would travel back and forth between the two panels. The objective of the game was to make the ball object move passed the other players’ panel. It was released in 1972 for arcades (Wolf, 2007). An example of Pong is shown in figure 2.2 below.

Figure 2.2.2: Pong Game

## Accessibility and Technology

Technology has shown a significant growth in the last thirteen years with the introduction of many “smart” devices such as the smart phone, smart television and smart watch. With this growth, the desire to cater to accessibility needs has been highlighted. This includes options such as voice to text conversions and vice versa,

For as long as video games have existed, the issue of accessibility has been present for the people who play them (Wilds, 2020). Video games presented accessibility issues not only in the hardware configuration, but also in the software setup itself. One example of this would be Super Mario Bros from 1985, which lacked subtitles or audio control options; presenting issues for people who are deaf or hard of hearing.

The European Parliament and the Council of the European Union released a directive titled “Directive (EU) 2016/2102 of the European Parliament and of the Council of 26 October 2016 on the accessibility of the websites and mobile applications of public sector bodies”. This directive states that any website or app owned by a public body must be accessible to any person with a disability (National Disability Authority, 2017).

**MORE HERE**

## Case Study 1: Xbox Adaptive Controller

The Microsoft Xbox Adaptive Controller is a hardware device released in September 2018 that can be used to control video games on both the Xbox One and Windows PC. The controller was designed to cater for people who play video games with limited mobility. The controller meets accessibility needs with two main features: its configurability and the Co-pilot mode. The controller consists of 2 large programmable buttons, a D-pad, Xbox, View, Profile and Menu buttons, 2 USB ports, a 3.5mm headset jack, USB-C and DC power ports, a connect button and 19 3.5mm jacks for external inputs. The USB ports can be used to map the left and right analog stick inputs respectively from an external device, such as a joystick. The 19 3.5mm jacks correspond to each of the button inputs on a regular Xbox One controller; not including the movement of an analog stick. However, the controller does contain 2 extra inputs that can be mapped to any input of the controller including an analog stick direction input. The Xbox Accessories application can be used to modify inputs, while allowing for multiple configurations. The other main accessibility feature of the controller is the Co-pilot mode. This mode allows a player to connect a second Xbox One controller that would work alongside the Xbox Adaptive controller. This allows a person to use any of the inputs from a regular Xbox One controller while using the Adaptive controller; as well as allowing a second person to assist someone using the Xbox Adaptive controller.

Another example is the Logitech G Adaptive Gaming Kit is a set of buttons, switches and triggers; intended for use alongside the Xbox Adaptive controller.

## Case Study 2: Tobii Eye Tracker 4C

The Tobii Eye Tracker 4C is an eye tracking device that can be used in Windows 7,8,10, etc. released in 201?

## Case Study 3: Google Stadia

In 2019, Google released the Google Stadia, a new technology that promised a full gaming experience that one might expect from a traditional console or ‘gaming PC’, played directly from a smartphone or tv. The Stadia requires the Stadia controller and either a Google Chromecast or an Android smartphone that supported the Stadia application.

**MORE HERE**

# Sprints

## Voice and Eye control in Windows

|  |  |  |
| --- | --- | --- |
| **Sprint Number** | **Start Date** | **Finish Date** |
| 1 | 02/02/2020 | 07/02/2020 |

|  |  |  |
| --- | --- | --- |
| **Task Number** | **Details** | **Status** |
| 1 | Install Tobii Eye Tracking Application | Complete |
| 2 | Configure Eye Tracker | Complete |
| 3 | Enable Windows Eye Control | Complete |
| 4 | Experiment with Windows Eye Control to learn functions and possible uses within project | Complete |
| 5 | Enable Windows Speech Recognition | Complete |
| 6 | Experiment with Windows Speech Recognition to learn functions and possible uses within project | Complete |
| 7 | Configure Xbox Adaptive Controller | Complete |
| 8 | Connect Tobii Eye Tracker to Xbox Adaptive Controller and test functionality | See Notes |

In this sprint I was attempting to evaluate the current voice and eye tracking functionality within Windows 10, along with the functionality of the Xbox Adaptive Controller. This was to help shape where I would need to progress with the project in terms of what functionality I could incorporate and what functionality I would have to create myself. I learned that the Xbox Adaptive Controller could not directly process the inputs from the Tobii Eye Tracker. To solve this issue, I decided to process the inputs from the Tobii Eye Tracker and use an external device to feed these inputs directly to the Xbox Adaptive Controller. From discussions with my FYP supervisor, we decided to research the Raspberry Pi as a possible solution for this issue. The Raspberry Pi would process the inputs from the Tobii Eye Tracker and then feed those inputs into the relevant input within the Xbox Adaptive Controller, in a manner similar to a switchboard.

## Raspberry Pi Research

|  |  |  |
| --- | --- | --- |
| **Sprint Number** | **Start Date** | **Finish Date** |
| 2 | 11/02/2020 | 21/02/2020 |

|  |  |  |
| --- | --- | --- |
| **Task Number** | **Details** | **Status** |
| 1 | Research Raspberry Pi and potential sources within the college of obtaining one | Complete |
| 2 | Research local companies that work with Raspberry Pis | Complete |
| 3 | Contact IMaR to arrange meeting to discuss FYP and Raspberry Pi integration | Complete |
| 4 | Meet with ImaR Strand Leader of RFID & Internet of Things | Complete |

This sprint was focused on the possible implementation of a Raspberry Pi for processing the inputs from the Tobii Eye Tracker for use in a video game that wouldn’t support it natively. After many discussions, I came to the conclusion that the Raspberry Pi would only be necessary were I to be introducing this setup on an environment other than PC, e.g. Xbox One Console. For this reason, I began to research alternative implementations. From the meeting with IMaR’s Strand Leader of RFID & Internet of Things, we discovered a simple python script that would run from the command line and process the data from the Tobii Eye Tracker. This would form the basis of the next sprint.

## Python/C# Eye Tracker Implementation

|  |  |  |
| --- | --- | --- |
| **Sprint Number** | **Start Date** | **Finish Date** |
| 3 | 09/03/2020 | 21/03/2020 |

|  |  |  |
| --- | --- | --- |
| **Task Number** | **Details** | **Status** |
| 1 | Fork Tobii Eye Tracker project from GitHub (sajidbaloch, 2018) | Complete |
| 2 | Install GitHub Desktop | Complete |
| 3 | Clone Tobii Eye Tracker project | Complete |
| 4 | Run Python Script from Command Line | Completed with Errors |
| 5 | Troubleshoot Python Script Error | Completed |
| 6 | Research C# Tobii Implementation | Completed |
| 7 | Create C# Console Application in Rider | Completed |
| 8 | Import Tobii Interaction NuGet package | Completed with Errors |
| 9 | Troubleshoot NuGet import issue | Completed |
| 10 | Install Visual Studio 2019 | Completed |
| 11 | Create C# Console Application (.Net Core) in Visual Studio | Completed |
| 12 | Import Tobii Interaction NuGet package | Completed with Errors |
| 13 | Troubleshoot NuGet import issue | Completed |
| 14 | Create C# Console Application (.Net Framework) in Visual Studio | Completed |
| 15 | Import Tobii Interaction NuGet package | Completed |
| 16 | Complete following tutorial for a Console Application: <https://developer.tobii.com/consumer-eye-trackers/core-sdk/getting-started/> | Completed |
| 17 | Run application to ensure it works | Completed |

A picture containing computer

Description automatically generatedThis sprint focused on the task of parsing the data from the Tobii Eye Tracker in its simplest form. This means parsing the X and Y coordinates on the screen where the Eye is looking. At first, I attempted to do this using the Python script I obtained in the previous sprint. However, this proved to be an unsuitable approach as, after troubleshooting the issue with task number 8, I found out that in order to parse the data from the eye tracker I was using, I would require a special license that I would need to pay to use ([Tobii], 2018). From this, I learned that a similar approach that would not require the license could be implemented, requiring me to use C# instead of Python. This caused a further issue when attempting to import the required NuGet package. Neither Rider nor Visual Studio seemed to recognise the package after it was imported. I learned that this was caused by the project template I had used. Originally, I was using a .NET Core Console Application when I required a .NET Framework Console Application to use NuGet packages. Once I overcame this issue, I worked through a simple tutorial to retrieve the required data from the Tobii Eye Tracker.

Figure 3.1 Tobii Eye Tracker Gaze Data Coordinates

## VoiceBot Speech Recognition

|  |  |  |
| --- | --- | --- |
| **Sprint Number** | **Start Date** | **Finish Date** |
| 4 | 23/03/2020 | 27/03/2020 |

|  |  |  |
| --- | --- | --- |
| **Task Number** | **Details** | **Status** |
| 1 | Go to <https://www.voicebot.net/Download/> and download VoiceBot Installer. | Complete |
| 2 | Install VoiceBot by running installer and follow steps | Completed |
| 3 | Open VoiceBot and choose download profile | Completed |
| 4 | Download Rocket League Profile | Completed |
| 5 | Test Rocket League Profile in game | Completed |
| 6 | Click Edit Profile and Add button in dialog box | Completed |
| 7 | Name command “Boost” and ensure Use name as command checkbox is set to true. | Completed |
| 8 | Click Add Group button and name group “New” | Completed |
| 9 | Click Add, then Press Mouse Button and then open the dropdown menu and choose Left Mouse Click and then OK | Completed |
| 10 | Press OK again and test new command in game | Completed |
| 11 | Ensure profile runs on correct application only | Completed |

In this sprint, I investigated the VoiceBot application for it’s potential use within this project. For licensing reasons, I had to first contact the owners of the application and get permission for its use. After this, I tested a profile for the game Rocket League. I found that, after a small adjustment period, the application was perfectly suited for this project. I could set up an individual profile for any game I played, which would hijack the controls of the game based on the voice commands I set.

# Methodology

## Research Undertaken

The research undertaken for this project centred around accessibility issues and previous attempts to overcome these issues.

**MORE HERE (summarise key findings, approx. 2 Paragraphs)**

## Research Question

Can eye tracking and voice control be implemented alongside existing accessibility peripherals for use in a video gaming context. An evaluation of existing technologies, how they may work alongside each other and how beneficial they may be to players with less motor function ability

## Proposed Project Implementation

This project aims to use a number of hardware devices together in order to improve the experience of players with less motor function ability. This will also require software that communicates with the hardware devices and translates their inputs into inputs a video game may understand.

## System Design

## Prototype

|  |  |  |
| --- | --- | --- |
| **Prototype Number** | **Start Date** | **Finish Date** |
| 1 | 04/12/2019 | 09/12/2019 |

|  |  |  |
| --- | --- | --- |
| **Task Number** | **Details** | **Status** |
| 1 | Install Unity Hub | Complete |
| 2 | Install Unity 2019.2.6f1 | Complete |
| 3 | Install Rider 2019 IDE | Complete |
| 4 | Open Unity Hub | Complete |
| 5 | Create new 3D Unity Project | Complete |
| 6 | Download Tobii SDK from Unity Asset Store | Complete |
| 7 | Import Tobii SDK from Unity Asset Store | Complete |
| 8 | In Tools click “Add Tobii SDK Demo Scenes to Build” | Complete |
| 9 | Run Sample Scenes to learn about the Tobii SDK | Complete |
| 10 | Create “Scripts” and “Prefabs” folders | Complete |
| 11 | Create “AsteroidController”, “EyeTracking”, “GameManager” & “PlayerController” Scripts | Complete |
| 12 | Read documentation at <https://tobii.github.io/UnitySDK/scripting-api> | Complete |
| 13 | Populate Scripts with code shown in Figures 4.1 – 4.6 | Complete |
| 14 | Create 3D Sphere with the properties shown in Figure 4.7 and save as a prefab named “Asteroid” | Complete |
| 15 | Make the default Directional Light a child of the default camera object. Rename the camera to Player and save as a prefab with the properties shown in figure 4.8 | Complete |
| 16 | Create a UI Text object named “Exit Text” | Complete |
| 17 | Move the Exit Text Anchor and Position to Bottom, Left and change the Text to “Press ‘Esc’ to Quit” | Complete |
| 18 | Create an UI Image object name “Current Eye Position” | Complete |
| 19 | Populate the Current Eye Position with the components shown in Figure 4.8 | Complete |
| 20 | Run the Game and test it works | Complete |

EyeTracking Class

Used to track the movement of the player's eyes

Figure 4.1: EyeTracking Class

Using the Tobii Eye Tracker, along with Microsoft Services Voice, a game can be controlled without any physical input from the player, e.g. button presses. This project provided a clear use case for the eye tracker in video games; particularly how the eye tracker can be used without the need of a physical input device. The benefit of this is the ability of players without the ability to move their arms can be given the opportunity to play video games. This shaped the rationale for the rest of this project and provided the idea for the Eye Tracking Windows Control shown below.

# References

[Tobii], G., 2018. *[Solved] No Gaze data received?.* [Online]   
Available at: https://developer.tobii.com/community/forums/topic/no-gaze-data-received/  
[Accessed 12 March 2020].

Bierre, K. et al., 2019. *Game Not Over: Accessibility Issues in Video Games.* 1st ed. San Francisco: Games Accessibility Special Interest Group.

Bishop, M., 2018. *Brief History of Disability and Web Accessibility in the United States.* [Online]   
Available at: https://www.ebsco.com/blog/article/brief-history-of-disability-and-web-accessibility-in-the-united-states  
[Accessed 20 November 2019].

Hoffman, B., 1950. *Danny Kaye beats Bertie the Brain at Tic-Tac-Toe.* Toronto: Life Magazine.

National Disability Authority, 2017. *EU Web Accessibility Directive.* [Online]   
Available at: http://nda.ie/Publications/Communications/EU-Web-Accessibility-Directive/  
[Accessed 4 December 2019].

Nitray, K. J., 2015. *Tennis for Two.* [Online]   
Available at: https://sexedi.ga/09174746.pdf  
[Accessed 4 December 2019].

Nitsche, M., 2008. *Video Game Spaces : Image, Play, and Structure in 3D Worlds.* 1st ed. Cambridge, Mass: The MIT Press.

Oxford University Press, 2019. *Definition of accessibility noun from the Oxford Advanced Learner's Dictionary.* [Online]   
Available at: https://www.oxfordlearnersdictionaries.com/definition/english/accessibility?q=accessibility  
[Accessed 20 November 2019].

sajidbaloch, 2018. *Tobii-Eye-Tracker-4C.* [Online]   
Available at: https://github.com/sajidbaloch/Tobii-Eye-Tracker-4C  
[Accessed 09 March 2020].

Simmons, M., 1975. *Bertie the Brain programmer heads science council.* Ottawa: The Ottawa Citizen.

Smith, A., 2014. *Tennis Anyone?.* [Online]   
Available at: https://videogamehistorian.wordpress.com/2014/01/28/tennis-anyone/  
[Accessed 4 December 2019].

Smith, A., 2014. *The Priesthood at Play: Computer Games in the 1950s.* [Online]   
Available at: https://videogamehistorian.wordpress.com/2014/01/22/the-priesthood-at-play-computer-games-in-the-1950s/  
[Accessed 3 December 2019].

The European Parliament and the Council of the European Union, 2016. *Directive (EU) 2016/2102 of the European Parliament and of the Council,* Strasbourg: Official Journal of the European Union.

Wilds, S., 2020. *For All The Players: A History Of Accessibility In Video Games.* [Online]   
Available at: https://www.ladbible.com/technology/gaming-for-all-the-players-a-history-of-accessibility-in-video-games-20200124  
[Accessed 29 January 2020].

Wolf, M. J. P., 2007. *The Video Game Explosion: A History from PONG to Playstation and Beyond.* illustrated ed. Westport, Connecticut: ABC-CLIO, 2008.