



**DE MONTFORT
UNIVERSITY
LEICESTER**

CTEC3451 Development Project
**“Creating a Control Mapping
Program for Game
Accessibility”**
First Deliverable

Project Supervisor: Jethro Shell

Project Author: Ruya Kumru-Holroyd (P2512547)

Contents

Abstract.....	3
Literature Review.....	3
Functional Requirements.....	8
Use Case Diagrams (USD).....	8
Use Case Specification (USC)	9
Non-Functional Requirements.....	11
Performance	11
Usability	12
Compatibility.....	12
Localisation	12
Security	12
Dependability, Availability and Maintainability.....	12
Indicative Test Plan	13
Test Methodology.....	13
Test Objectives.....	13
Test Plan Strategy	13
Unit Tests	14
System Design	18
The Project.....	18
Prototype 1	18
Prototype 2	19
System Architecture.....	20
User Interface (UI).....	21
Implementation Report	22
References	23
Bibliography	23
Figures.....	25

Abstract

There are millions of video games available worldwide, but how many are actually accessible and well represented? The answer is less than you might expect, which is why this project was created to demonstrate that it is possible to do so and that it can be used and altered for future usage.

This report details the research into creating a control mapping program, as well as the two prototype games created to test the control mapping program. For these games, it was important to include the right representation and accessibility options, to really deliver the importance and the ability to be able to include these in video games. Therefore, time was taken to research different disabilities, types of game accessibility in current games as well as looking at controllers and mapping solutions to create this project.

Literature Review

Video games, “a game in which the player controls moving pictures on a screen by pressing buttons” (Cambridge Dictionary, (2020). VIDEO GAME), have become an everyday presence in today’s modern society. Being around since the 1950s, they have gone from military machines to arcade systems, to home consoles, to handheld consoles and mobile devices (History.com Editors, 2019), a brief timeline of video game history can be found in Mark J. P. Wolf’s book *The Video Game Explosion: A history from PONG to PlayStation and Beyond* (Wolf, M.J.P. (2008), pg. 17-21).

With the evolution of video games has also come the evolution of technology itself, with great improvement on new hardware, graphics, and performance. The number of people playing video games has also risen massively over the years, according to (Clement, J. 2021), the number of video game users in the UK is 44.32 million people, over 50% of the whole UK population. The game industry in the UK itself is the biggest in Europe and the sixth worldwide, with a market value of £5.3 billion, this makes it one the highest market in the entertainment industry, compared to music and film. And with more players come more different types of players and player styles. According to AbleGamers, a charity that aims to improve accessibility in video games, around 46 million video game players in the United States have a disability (Valentine, R. 2020). Which is around 1 in 5 of video game players in the United States, and according to (Clarysse, 2021), 1.75 billion people are dealing with disabilities in their everyday life, which is not including their friends and family, however, as Pascal says “the representation ratio in media is abysmally low, and worse, it’s almost condescending and tear-based content“. Which is why there is a need to have awareness and for action. Many of these players are unable to play popular games due to the

complexity and lack of accessibility options for them, and do not hold any disability representation or if they do, can be negative and inaccurate.

“Deciding to make a game accessible from day one instead of including it later is key” (Carrera, 2016. pg. 32). Accessibility is the level that a product is available according to the number of people. It is about treating everyone with respect and enabling them access to all possible cultural manifestations. It can be thought of as a system's or entity's "ability to access" and benefit from it. It is frequently linked to people with special needs and their rights to such organisations (Carrera, S. (2016), pg. 23). Therefore, adapting a game's hardware and software (such as game controllers, difficulty level, or feedback type) to individual needs, whether they have a disability, is what game accessibility is all about (Westin et al, 2011). Over the years, accessibility in gaming has grown in accordance with the new technological advances being made. The (AbleGamers, n.d.) shows the history of adaptive tech, and how far it has come, starting from 1986 where Nintendo created first accessible technology with the hands-free controller for the NES, September 2009 the PlayStation 3 adding button mapping to their OS system, becoming the first console to add accessibility features at OS level. 2011 saw AbleGamers themselves creating the Adroit controller with Evil Controllers, a controller with switch inputs for the first time, 2014 had Borderlands 2 add a colourblind mode in their game. Finally, in 2018 the Microsoft Adaptive Controller (XAC) is released.

Silvio Carrera with their book, *Accessibility in Games: Including people with disabilities*, explains that with the evolution of technology, “there was an increase on the amount of attention and control input necessary in order to play” (Carrera, S. (2016), pg. intro). They also describe that one of the issues that disabled gamers come across within video games “They might not be able to use the default controller the platform suggests, which means they won’t be able to do input in the game.” (Carrera, S. (2016), pg. 13), which is a sharp reason for the purpose of this project, to allow players to use their own controllers and map their buttons to the game prototype games and allow them to be saved. Carrera also mentions how the lack of flexibility in the control options, such as the ability to reconfigure buttons, makes it more difficult and unlikely for someone with special needs to tailor the game to their demands. As (NLS, 2015) explains, playing video games can help you be more creative, increase your problem-solving abilities, and foster teamwork. People with disabilities, such as those with movement impairments may not be able to utilise a normal game controller and therefore have fewer options for enjoying video games.

This is a project to create a control mapping program for game accessibility, there are currently programs such as this out there, for example, Rewired (guavaman.com, n.d.) is a comprehensive input system for Unity that contains a configurable and saveable controller map as well as a control mapper system that uses the Unity GUI to allow players to rebind controllers in real time. Another example is, reWASD (rewasd, 2017), which can remap

controller buttons on PC which can be used on different games just by loading and setting the mappings on the program, including saving profiles for different games. A similar example is a program called Input Mapper (Wobbles, 2020), it auto detects the device for the user and allows them to choose advanced macro and customisation settings, allowing them to adjust not only the device type, but also how inputs are translated and applied, Figure 3 shows the programs main page. Other examples are gaming console themselves, such as Xbox and PlayStation that include their own controller configurations and mappings, as can be navigated by (Hesse, B. 2021). This information can be considered when designing the control mapping program, as well as The Game Accessibility guidelines website, (Anon, n.d.) which includes a wide range of guidelines and accessibility design ideas for developers to include in their games, ranging from basic guidelines to advanced. This also includes guidelines on 5 different types of disabilities such as Motor, Cognitive, Vision, Hearing and Speech.

An important aspect of the two prototype games, that will be created for testing and use of the control mapping program, is the design and story of the game. To include the representation of disabilities in the characters and making sure they are as accessibly designed as possible for prototypes. Therefore, the first prototype game will be a 2D basketball shooter game, where the main character is in a wheelchair, to represent physical disability. This idea came about as I read through (Brody, 2020)'s article for the AbleGamers Charity called *The Need for More Disabilities in the Games We Play*. Where they discuss how there is little physical disability representation in games in our current society, and a good way to overcome this is to have a game such as Wheelchair Basketball, which could be like any other sport games out there. Another approach is to include disabled people in a wheelchair for basketball video games, which is the approach that was taken in this prototype game. Inspiration was taken from the android game Doodle Basketball, (Byril OOO, 2013) for gameplay and style. It is important to include physical disability representation even within a prototype game such as this to help convey how it is possible to include appropriate representation in the media, especially when in most mainstream video games, as (Ready Player, 2016)'s article explains, "a game protagonist is physically disabled in some manner, it's usually immediately fixed through the use of cybernetics, prosthetics, or even cybernetic prosthetics." This idea is shared with (TechTalk, n.d.) as disabled characters are the most likely to get "fixed."; "fixed" is being deliberately utilised to emphasise the fact that these are game constructions, and not easy to come by in real life, a catch-all fix is frequently thrown into the mix in video games. Ian Hamilton, an accessibility expert, agrees as he explains that there is still a notion in video games that people with disabilities are broken and need to be fixed, with tropes of these fixes being superpowers or

superhuman prosthetics. And “Moreover, games are often guilty of furthering the myth that a disability is rare, with all the impact that has on broader prejudice and discrimination.”

For the second game prototype that is being created for this project, the main premise is on mental health and representing that within a simple 3D shooter game. Where the main character will have a mental health issue such as depression or anxiety, and the aim of the game is to shoot away the negative thought bubbles. The reason why the second prototype game is about mental health instead of another physical disability is because it is not portrayed near enough in video games compared to physical disabilities. (TechTalk, n.d.) shows two diagrams that represent how mental health has been tackled in recent years, in Figure 1, in video games and the distribution of different types of disabilities portrayed in games most often, in Figure 2. Even then, the way that mental health is portrayed can be stereotyped and inaccurate. Or, as (Dunlap and Kowert, 2021, pg. 122) explored in their Mental Health in 3D article, mental illness is shown in video games in both traditional and novel ways, such as in-game mechanisms (e.g., sanity meters) and player-driven decision making. One example of this in a popular video game is (Grand Theft Auto, 2013) where the playable character has a mental state that can rise if the player causes too much violence. Few games truly look at mental health, which is the purpose of the representation in this second prototype.

Overall, positive representation of disabilities in video games is very important as it can bring awareness to more disabilities (Valentine, R. 2020), especially with accurate representation of disabilities such as Symmetra and Ana from Overwatch with Autism and with the loss of an eye, which are seen as disabilities in the context “that is, impairments that diminish their ability in the context of Western Society” (Cullen, Ringland, & Wolf, 2018). As well as Joker from the Mass Effect series, who was born with Vrolik syndrome which causes extreme bone brittleness, using crutches and leg braces to get around. (Blockfort, n.d.) this has been seen as great representation, as the best thing is that his disability does not define him, he is still fiercely independent despite the challenges he might face, (TechTalk, n.d.). Within the same article, Ian Hamilton says: “Representation of characters with disabilities is still rare. It is often simply not on people's radars. And when it is, fear of handling it badly can put people off.” And that game accessibility is a more prevalent topic in today's media. Which is why representation is being included within this literature review and the game prototypes.

In Conclusion to this literature review, Game technology has come a long way since the first electronic games in the 1950s. And with this evolution, has come rises in adaptive tech for disabled gamers, from the hands-free controller for the NES in 1986, to the Microsoft

Adaptive Controller (XAC) in 2018 including the ability to button map and use switch inputs. On the other hand, with representations of disabilities in games, whilst there are some good examples, the amount is lacklustre. With the information gathered in this literature review, this project can begin to take shape with ideas of control mapping programs, representation of disabilities and game accessibility design.

Functional Requirements

A functional requirement, as described by (Martin, M. 2019), is a statement that describes the service that the software must provide. It refers to a software system or a component of one. A function is nothing more than the inputs, behaviours, and outputs of a software system. A calculation, data manipulation, business procedure, user interaction, or any other unique functionality that defines what function a system is likely to execute can all be considered. Functional requirements are also known as functional specification in software engineering. For this project, it was necessary to identify the functional requirements by identifying who the users would be, construct a use case diagram, which depicts the user's (also known as the actor's) relationship with the application's numerous use cases, and then define each of the use cases with a use case specification, provide specific details regarding each distinct use case.

Use Case Diagrams (UCD)

In this project, there are multiple sections that require their own sets of functional requirements and use cases, as there is the main control mapping program and the two game prototypes. For each game prototype there is a main menu scene, which has its own use cases and functional requirements such as loading the game scene, changing game settings, and exiting the game. The menu's game settings are the highlight of this project, as this is where the control mapping program takes place and is used.

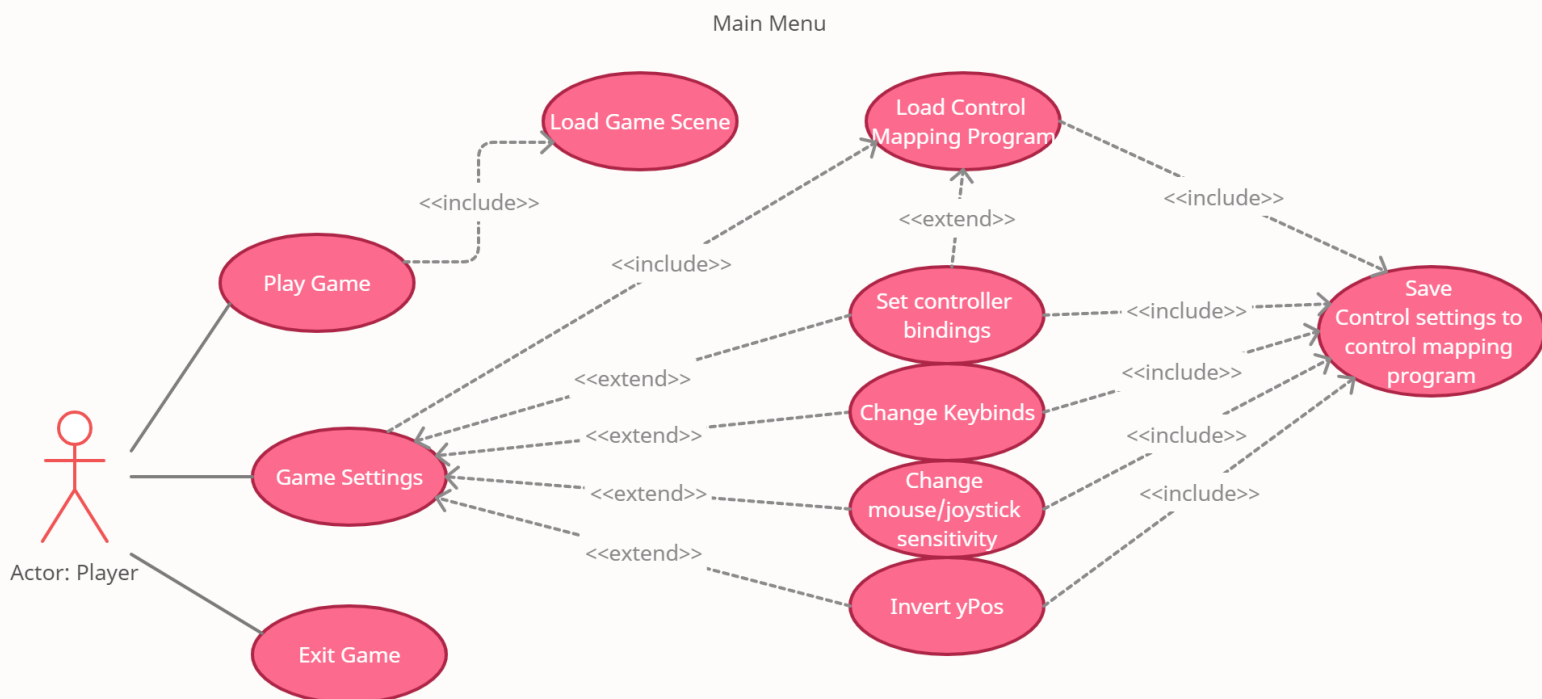


Figure 4: Use Case Diagram for the Main Menu scene

Although there are two game prototypes in this project, they typically share the same functional requirements from the actor as they navigate through the games. These functional requirements being, game scene loaded, pausing game, player control, generating assets, loading game settings and more.

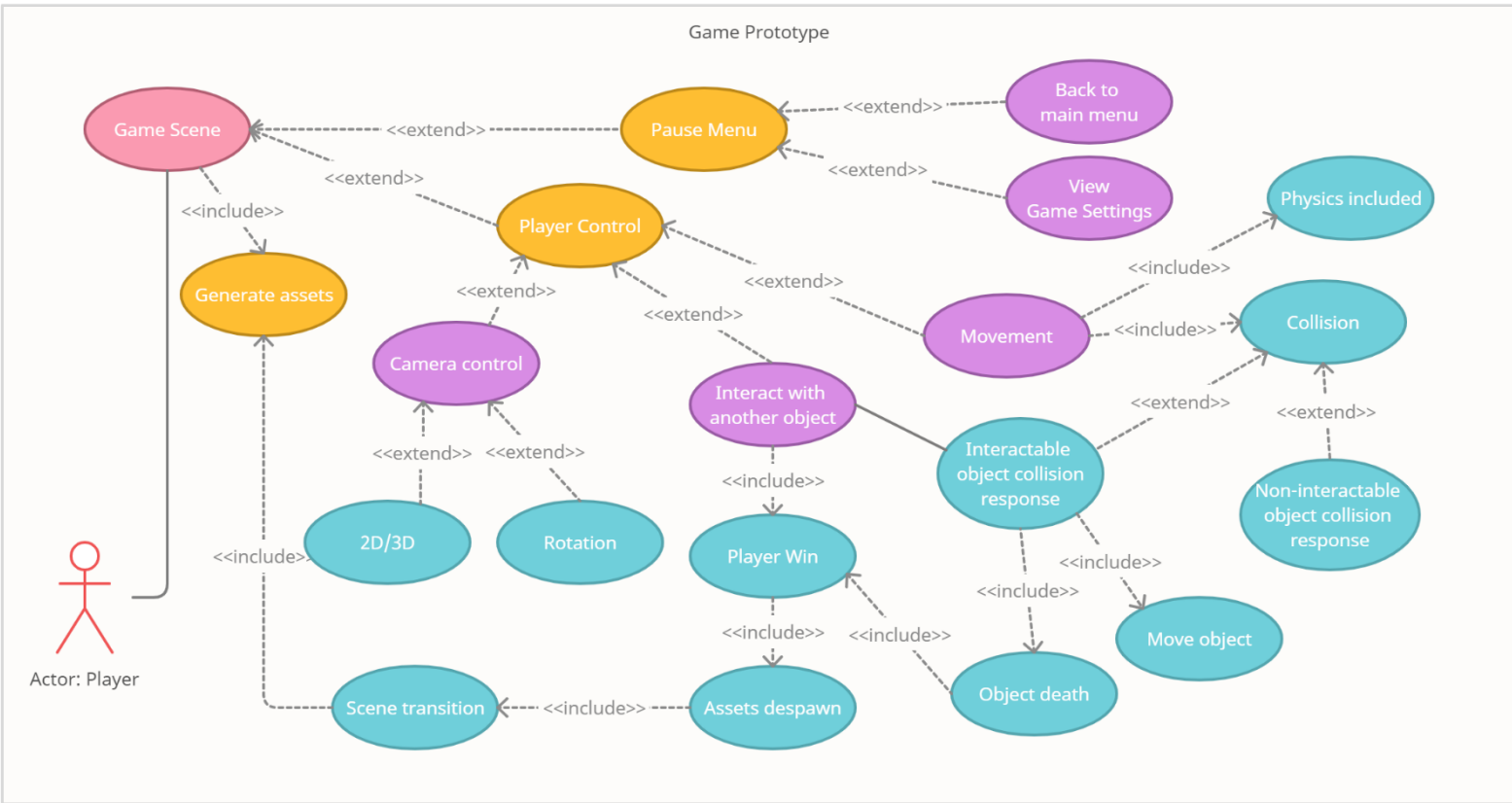


Figure 5: Use Case Diagram for the Game Prototypes

Use Case Specification (UCS)

These following use cases are descriptions of the use cases in the UCDs above, giving more detail and understanding to the project and its functional requirements.

Main Menu:

Use Case	Description
Play Game	Button from that takes actor from main menu to the game scene.
Load Game Scene	Load the prototype game scene and all its levels.
Exit Game	Exit the game prototype application.
Game Settings	Change the game settings within the game, includes the load control mapping program which is the heart of the project.
Load Control Mapping Program	Loads the pre-set control settings that have been saved from one of the game prototypes. Any that have not been set will remain at default unless changed. This includes the saved mapped controls already on the program.
Set Controller Bindings	If the actor is using a controller, set the controllers button bindings.

Change Key-binds	Change the key binding for PC keyboard if actor is using it or change the bindings to another controller button or other equipment for controlling.
Change Mouse/Joystick Sensitivity	Change the sensitivity for the mouse or joystick for the game prototypes.
Invert yPos	Invert the vertical position Y for inputs such as mouse or joystick to best suit actors' preferences
Save Control Settings to Control Mapping Program	Save changed game settings to the control mapping program so that it can be loaded onto other games and the data can be transferred.

Game Prototypes:

Use Case	Description
Game Scene	The whole game scene, with included levels and information.
Pause Menu	Button that makes the whole game pause, includes options such as returning to the main menu or looking at the game settings.
Back to Main Menu	Button in pause menu that takes actor back to main menu of game prototype.
View Game Settings	Button in pause menu that takes actor to the game settings to change or load settings.
Player Control	All cases that contribute to the controlling of the player.
Movement	The movement of the player object in the game prototype, including physics that occur with movement and collisions that may occur.
Physics Included	The physics that happen as the controlled player object is moving through the game space.
Collision	Collision that happens with the player object that is being controlled by the actor. This can be with an object that is interactable or an object that is not.
Non-Interactable Object Collision Response	The response that happens when the player object controlled collides with an object that is non-interactable.
Interactable Object Collision Response	The response that happens when the player object controlled collides with an object that is interactable. This includes an object being moved or being deleted.
Move Object	An object that has been collided with moves to a different position.

Object death	An object that is dies after a collision due to game rules, this includes the rule that the player can win due to this object dying.
Interact with another object	Like <i>interactable object collision response</i> , it is associated with it, this includes move object and object death as well as player win. This also interacts with another object due to player control with other inputs instead of collision such as game rules or controllers.
Player Win	Player win can happen when an object dies and the game is over, when this happens it includes score counting and assets de-spawning and this transitions to a new scene which could be a new level scene or a menu scene.
Assets despawn	When the player wins, despawn all the assets to load the new scene
Scene Transition	Load the next scene, which can be a new level, game over scene or the main menu, this includes Generate assets for the assets needed for the next scene.
Camera Control	The camera that is controlled by the player and follows it, this is primarily for 3D camera compared to 2D camera which will be stationary.
2D/3D	Setting the camera for either 2D or 3D depending on the game prototype.
Rotation	The camera rotation that happens when the camera follows the player in a 3D environment.

Non-Functional Requirements

As well as functional requirements of this project, there are some non-functional requirements, they specify how the system will operate, that can be laid out in terms of the game prototypes and the control mapping program. The most common types of criteria to evaluate include performance, usability, compatibility, localization, security, dependability, availability, and maintainability.

A list of some of the project's non-functional requirements follows:

Performance

- The prototype games will run at 30 frames per second
- The prototype games will load and unload resources such as player assets

Usability

- The game settings allow player to change controls and save them
- The user interface is simple and easy to understand

Compatibility

- The software uses Unity3D
- Will need some sort of input device
- Keyboard and Mouse needed for default settings

Localisation

- Project created in British English

Security

- There will be no personal data from the player stored
- All game data is kept within the game files

Dependability, Availability and Maintainability

- The prototype games will be accessible as possible
- There will be no problems to the player, any system errors will be dealt with
- The prototype games will be offline
- All code, scripts and assets will be version control and well maintained

Indicative Test Plan

Test Methodology

Testing is an important part of a project's development and progress, to ensure a program/application is functional and without problems or errors. As a project has functional requirements, tests can be used to check that these requirements are being met. The paragraphs that follow will go into my testing strategies and include unit test cases that explain what each test is and the results of those tests.

Test Objectives

The objectives to the tests are to test the functionality of the project and the various parts to it, for example, the two prototype games to test that all scripts work, and the games are playable and accessible for the user, no errors occur and each level runs and works as supposed to, and mechanics such as physics and gameplay are functional, and performance is not deterred in any way. The control mapping program is also another test objective, to ensure that the player can set their controller buttons or specific key-bindings as they want and that they are saved and loaded to each prototype game.

Test Plan Strategy

Agile methodology is the way that this project is being managed, it is broken down into several phases and is being consistently developed throughout the project time. With these several phases, come unit testing as each phase needs to be tested and ensured it is working before the project can move onto the next one. Such phases can be things like creating a C# script for a prototype game and testing that to see it is working, or making assets and testing if they load in. Testing in this way is preferred because it enables any errors or problems to be dealt with quickly as and when they occur, rather than leaving it to a later stage and having to rewind the project's progress to fix an issue.

Unit Tests

Unit tests focus on specific areas of the project and test the functionality of the existing code and development. In this project's development thus far, majority of the failed unit test cases are due to not being implemented yet.

Unit Test Cases

Case ID	Test Case Objective	Test Case Description	Expected Result	Pass or Fail
1.0		Main Menu		
1.1	Test images and buttons are loaded to the main menu scene	Open Game application/Main menu	Asset images and buttons load to scene	Pass
1.2	Test that the play game button loads player into game scene	Click on Play Game button	Redirected to Game scene	Pass
1.3	Test that the game settings button loads player into game settings scene	Click on Game Settings button	Redirected to game settings scene	Pass
1.4	Test that the exit game button exits the game application	Click on Exit Game button	Game application is exited	Pass
2.0		Game Settings		
2.1	Test that game settings scene is loaded with assets, buttons, and text	Load into game settings scene	Text, buttons, and assets loaded	Pass
2.2	Test that key-bindings can be changed when press key on button and text changes to new key	Click button on key-binding to change and press new key	Button text shows new key set and that is new binding	Fail
2.3	Test that checkbox for inverting y position can be checked	Click on checkbox	Tick appears on checkbox	Pass
2.4	Test that set controls and key-bindings changed can be saved to database	Click on save button	New controls and key-bindings can be saved	Fail
2.5	Test those controls mapped that have been saved in database can be loaded to game	Click on load control maps button	Saved controller/key-binding settings in database loaded to game settings and buttons changed	Fail
2.6	Test back button leads player back to the Main Menu scene	Click on back button	Redirected to Main Menu scene	Pass
3.0		Game Prototype 1		

3.1	Test game runs and plays on load, with all assets and UI loaded and scripts running with no error	Load into game scene	Images and assets loaded, player loaded, text and button UI loaded, enemy loaded	Pass
3.2	Test 2D colliders loaded and collide with relevant objects	Move player or ball towards a 2D collider	Player and ball collide, stop moving on collision	Pass
3.3	Test player and ball objects are moveable with set or default controls	Use controls set or default A and D keys to move player and ball	Player and ball objects move	Pass
3.4	Test that player animation occurs when moving or shooting	Move player or shoot ball	Player animation occurs	Fail
3.5	Test that physics in game scene work correctly, gravity working	Load into game scene	Ball and Player use gravity when moving and shooting	Pass
3.6	Test ball spins and bounces when collided with	Ball falls to ground once shot	Ball spins and bounces	Pass
3.7	Test ball is kinematic when not shot	Move player and ball	Ball does not move from player hand when not shooting	Pass
3.8	Test ball is dynamic when shot	Shoot ball	Ball moves freely in the game scene, with physics reacting	Pass
3.9	Test when ball and basketball net collide, ball and player reset	Aim ball and shoot at basketball net	Player and ball reset to original start points (checkpoints) when ball has collided, and ball is reset to kinematic state	Pass
3.10	Test when ball and basketball net collide, score count increases by 3	Aim ball and shoot at basketball net	Score count UI on game scene is increased by 3 points	Pass
3.11	Test player and ball do not move using move left and right controls whilst ball is dynamic	Move left and right when ball is dynamic	Player and ball should not move using the move left and right controls	Pass
3.12	Test that ball only allows one mouse drag once shot	Drag on screen as shooting	Ball should not move	Fail
3.13	Test there is a line renderer and is correctly drawn to screen when player aiming to shoot	Drag on screen as shooting	Line renderer appears on screen that aims the ball towards the basketball net	Fail

3.14	Test there are multiple levels, each one loaded after the next	Complete a level	Game reset and new level loaded with new scripts/game mechanics	Fail
3.15	Test the key bindings can be changed using the key binding script in Unity Inspector	Change key-binding script using inspector	Controls changed to different/new keys and can be used	Pass
3.16	Test the on screen drag shoots the ball with increased power the further the distance dragged	Drag on screen as shooting, try different distances of start and end points of drag	The longer the drag, the more power the ball has when shooting and vice versa	Pass
3.17	Test that game mechanic scripts are working as intended and no errors appear on console	Run and play game and look at console	No errors appear on console and game runs smoothly	Pass
4.0	Game Prototype 2			
4.1	Test game runs and plays on load, with all assets and UI loaded and scripts running with no error	Load into game scene	Images and assets loaded, player loaded, text and button UI loaded, enemy loaded	Pass
4.2	Test that the camera follows the player as they move around the game scene	Run game and move player	Player moves around game scene using controls set or default controls	Pass
4.3	Test that pistol is positioned on the camera correctly and does not move or wander	Run game and move player and look around with camera	Pistol positioned in place and does not move out of place when moving/looking around	Pass
4.4	Test the player movement of moving left, right, forward, back, rotate left and right and jumping	Use set keys/controls to move player	Player moves around game scene smoothly	Pass
4.5	Test Pistol animation when shooting	Press key/control button set to shoot	Pistol moves up and down in an animation motion to reflect shooting	Pass
4.6	Test when shoot button pressed, bullet projectile fired, and bullet instantiated	Press key/control button set to shoot	Bullet fired from pistol in a straight-line using force	Fail
4.7	Test terrain and world objects colliders are set and collide with player	Move player towards terrain/world objects to collide	Player stops moving when colliding with objects	Pass

4.8	Test when bullet collides with enemy object, bullet and enemy disappears	Shoot bullet at enemy and colliding with it	Enemy and bullet objects destroyed	Fail
4.9	Test when bullet collides with enemy object, score increases by 1	Shoot bullet at enemy and colliding with it	Score count and UI increased by 1	Fail
4.10	Test when bullet collides with terrain or other world objects, physics occurs	Shoot bullet at objects or terrain and colliding with it	Bullets ricochet off objects and terrain/ground, bouncing off or lay on ground	Fail
4.11	Test there are multiple levels, each one loaded after the next	Complete a level	Game reset and new level loaded with new scripts/game mechanics	Fail
4.12	Test game lighting works correctly	Run game scene	Lighting is on and shows as expected, not different in anyway	Pass
4.13	Test enemy defeated animation occurs	Defeat an enemy	Explosion or fading animation	Fail
4.14	Test the game audio works, walking steps with player moving and gunshot sounds	Run game scene	Hear game audio as player moves around hear footsteps, and gunshots when shooting occurs	Pass

System Design

The purpose of the system design document is further the detail of the functional and non-functional requirements, as mentioned above, and implement them into a full system design of requirements, architecture, user interface, inputs, and outputs.

The Control Mapping program is designed to help with game accessibility, capturing the mapping of controllers/keyboards, how these mappings can move from two different playable prototype games and help benefits disabled players gaming experiences, making games more accessible and easier to play. The player is supplied with two playable prototype games that include their own menu and game settings section where they can load the control mapping program and set their own controller buttons/key bindings, which can then be saved to the program and that data is kept outside the games in the program which can be loaded to a different game. The program and prototype games will be made using Unity3D and will run on PC.

Controls, per Game Prototype, will be relatively simple as it is important to keep them easy to understand and learn, "if they are too difficult to learn or if the game experience becomes uninteresting then you're creating barriers that only skilled and patient players will be able to overcome," (Carrera, 2016).

The Project

To keep the control mapping program simple, easy to understand and accessible for all, the code of it is within the back end of the project, and the front-end which includes the two prototype games, and their menu/game settings UI is what controls and sets the program.

Prototype 1

Firstly, prototype 1 which is titled "Wheelchair Basketball Shoot" is as simple as the name suggests, it is a 2D basketball shooter game where the player aims and releases a ball to a basketball net and gains points for each shot, there will be multiple levels that get more difficult as the game advances, with obstacles and moving nets. The player can be controlled default with A and D keys on a keyboard to move left and right on the screen, the ball following along with it, and then the ball can be aimed and released by using the mouse and holding down the left mouse button and dragging down on the screen to aim the ball into the basketball net with a line renderer that shows the path the ball will follow when the left mouse button is released and the ball is fired. The strength at which the ball is fired depends on the speed and duration of the mouse button being held down.

The action needed to aim and shoot the ball, is by dragging on the screen and aiming at the basketball net with a line renderer, this method is accessible because the player can start and end the drag position at any point on the screen, that the player would like, instead of a specific control area. And simple two switch key controls make the game controls easy to learn and use as well as simple to bind and change to other switch keys or controller buttons that are mapped using the control mapper program.

As this is a prototype there is only going to be 3 levels in this prototype, the first one being very simple, just aim and shoot to the basketball net, the second will have the basketball net move up and down the screen slowly, the third will have an obstacle in the way and the moving net.

The representation of disability in games, for this prototype, as explored within the Literature review, has come from (Brody, 2020) AbleGamers article about the need for more disabilities in the games we play, where they suggested that sports games can be adapted to include disabilities such as basketball games can have a wheelchair basketball player.

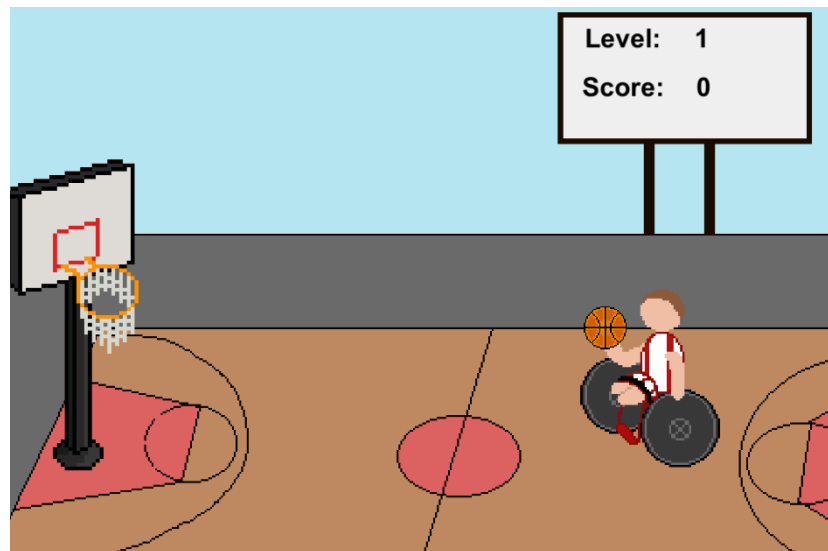


Figure 6: Screenshot of Prototype 1 game

Prototype 2

The second prototype game is titled “Bye-Bye Thoughts”, it is a simple 3D first person shooter which represents mental health. The player works their way through a 3D environment which is darkened, and they must go around and shoot the “Bad Thoughts” away before it consumes them and the entire screen goes dark, the more “Bad Thoughts” they shoot, the brighter the scene becomes until it is normal daylight and all the thoughts have been expelled. The player gains points as they play, with multiple levels they will continue to rack up as many points as possible before player loses, and score will be kept in a high-score table.

The player can be controlled by default with Keyboard and Mouse, using WASD respectively for Forward, Left, Back and Right, Space bar for Jumping, And Left Ctrl or Left Mouse Button to shoot. As per the project’s control mapping program these can all be changed within the game settings, accessed from the Main Menu.

As it is a 3D game, there is a 3D camera that is essential to the game, a follow camera is used to create the perception of a first-person shooter, with a pistol in the front view, attached to the player and camera so that it moves with the two as they navigate through the 3D world.

The representation within this prototype is Mental Health, as mentioned briefly, with these “Bad Thoughts” highlighting people with depression who feel clouded by their own negative

and intrusive thoughts and this is a game that could feel therapeutic in a way, to expel the negative thoughts and escape from them, even for a little while.

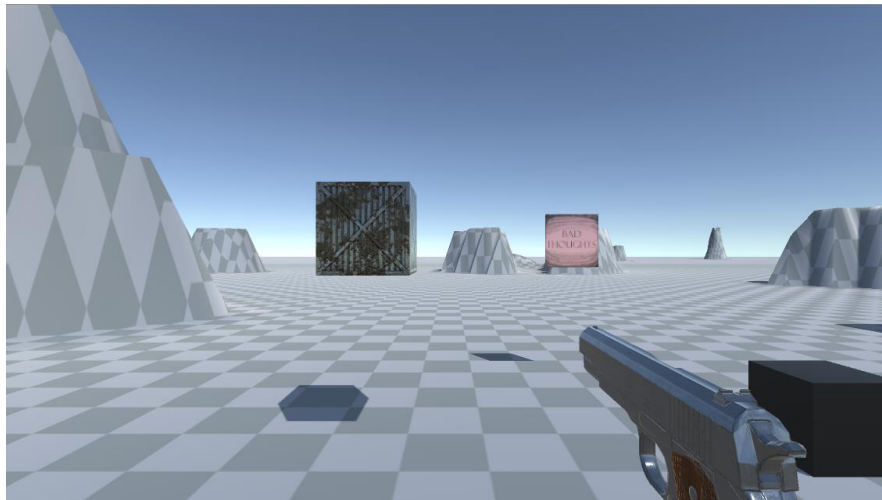


Figure 7: Screenshot of basic Prototype 2 game

System Architecture

For the System Design, the method of Object-Orientated Programming (OOP) was used for classes when creating C# Scripts in Unity. The Unified Model Language (UML) Diagram below shows the full systems design, with both prototype games and the control mapping program classes shown in an OOP environment. This helps highlights the functional requirements of the project within the codes and classes that are needed to create it.

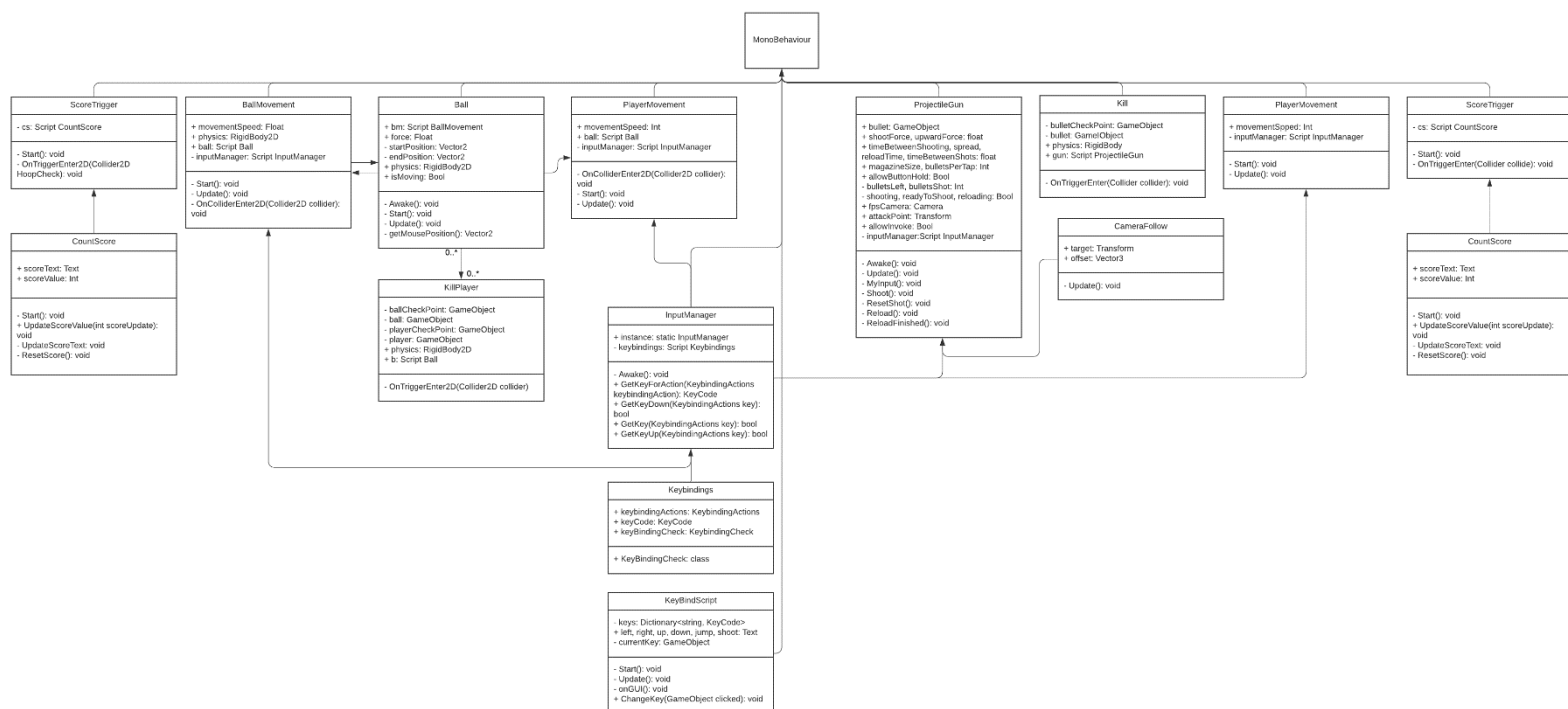


Figure 8: UML Diagram

User Interface (UI)

The user interface is incredibly important as it needs to be accessible and easy to use, as well as clear to see the control mapping program working as the mapped control data is saved and loaded through the game settings in each game prototype. It must be fully functional and interact with things within the game prototypes and for the control mapping program. Specific UI that will be used in this project is the use of text and buttons to set key bindings and button mapping for the game settings and controls, as well as buttons that save and load control mapping data to and from a database. Other UI within the game prototypes is texts such as score count, level number and ammo amount, as well as buttons within main menu's to navigate through the project and game prototypes.

(Below are some UI examples, they are not final)

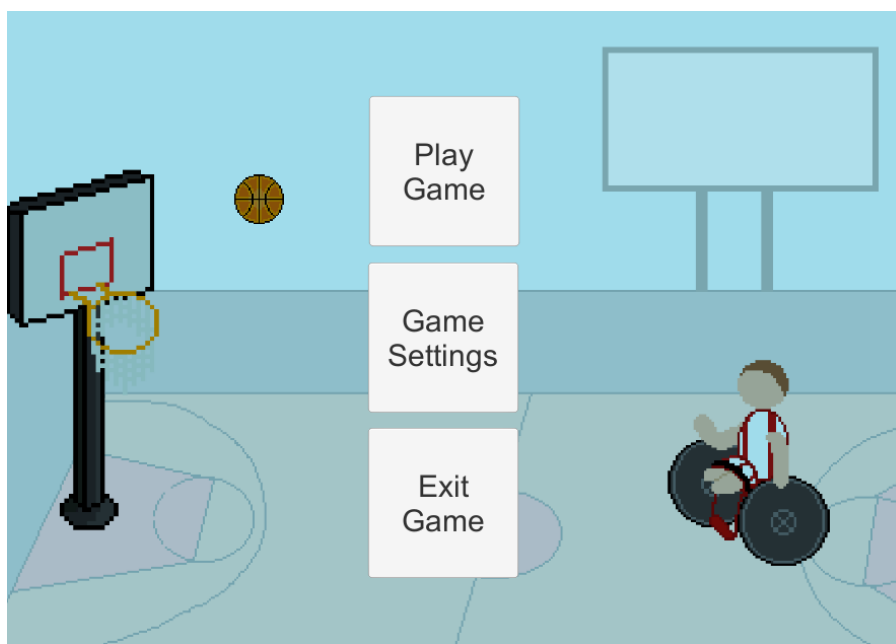


Figure 9: Prototype 1 Main menu



Figure 10: Prototype 1 Game settings

Implementation Report

The first meeting of the second term, had the project's current development shown to the supervisor, Dr Jethro Shell, which included the start of two game prototypes shown in Unity3D, as well as some scripting and back-end code that were relevant to the control mapping.

The first game prototype, the front-end showed a playable game with a player and ball object that can be controlled as well as the ability to shoot the ball into the basket, with collisions on grounds and walls, and a scoring system in place. When the ball collided with the net, the score goes up by 3 and the player and the ball is reset. The back-end showed the various scripts, using C#, that were used to create the game and controls, specifically showing the code and physics that create the ball movement when shot, as that was something that had taken time to create and perfect.

For the second game prototype, the front-end showed a simple 3D game scene, with a moveable player and follow camera, a first-person shooter game was created with a pistol and animated shooting motion with a terrain and multiple objects in the scene. This prototype was less developed compared to the first with errors such as bullet projectiles not working. This was shown to the supervisor and advice on how to counter this problem was given.

Along with these prototypes, within the back-end of the project were beginnings of key bindings and an input manager that will be used towards the mapping program and the accessibility of the game prototypes. The first prototype had the ability to change the key bindings to whatever key on the keyboard, the next step after would be to allow the user to use different inputs such as a controller and bind and map those buttons to the prototypes.

References

Bibliography

AbleGamers. (n.d.). History of Adaptive Tech. [online] Available at:

<https://ablegamers.org/history-of-accessibility-in-gaming/> [Accessed 10 Jan. 2022].

Anon, (n.d.). *Game accessibility guidelines | A straightforward reference for inclusive game design*. [online] Available at: <https://gameaccessibilityguidelines.com/> [Accessed 10 Jan. 2022].

Blockfort. (n.d.). Top 10 Video Game Characters with Physical Disabilities! [online] Available at: <https://www.blockfort.com/character-lists/physicallyimpaired/> [Accessed 10 Jan. 2022].

Brody, C. (2020). *The Need For More Disabilities In The Games We Play*. [online] ablegamers.org. Available at: <https://ablegamers.org/the-need-for-more-disabilities-in-the-games-we-play/> [Accessed 6 Jan. 2022].

BYRIL OOO (2013) *Doodle Basketball*. [GAME] Android. Russia: Byril OOO.

Carrera, S. (2016). *Accessibility in Games: Including People with Disabilities*. Germany: LAP LAMBERT Academic Publishing.

Cambridge Dictionary (2020). VIDEO GAME | meaning in the Cambridge English Dictionary. [online] Cambridge.org. Available at: <https://dictionary.cambridge.org/dictionary/english/video-game> [Accessed 8 Dec. 2021].

Clarysse, P. (2021). *The Genesis of Big Karma — Positive Disability Representation in Gaming*. [online] Medium. Available at: <https://pascalclarysse.medium.com/the-genesis-of-big-karma-positive-disability-representation-in-gaming-681692587af2> [Accessed 10 Jan. 2022].

Clement, J. (2021). *Video game market in the United Kingdom - Statistics & Facts*. [online] statista.com. Available at: <https://www.statista.com/topics/1763/gaming-in-the-united-kingdom/#dossierKeyfigures> [Accessed 8 Dec. 2021].

Cullen, A., Ringland, K. and Wolf, C. (2018). A Better World: Examples of Disability in Overwatch. *Department of Informatics, University of California Irvine*. [online] Available at: https://www.researchgate.net/publication/324687611_A_Better_World_Examples_of_Disability_in_Overwatch [Accessed 27 Jan. 2022].

Dunlap, K. and Kowert, R. (2021). Mental Health in 3D: A Dimensional Model of Mental Illness Representation in Digital Games. Loading... The Journal of the Canadian Game Studies Association [online] 14(24), pp.122–133. Available at: <https://loading.journals.publicknowledgeproject.org/loading/index.php/loading/article/view/339/319> [Accessed 10 Jan. 2022].

guavaman.com. (n.d.). *Rewired Documentation | Control Mapper*. [online] Available at: <https://guavaman.com/projects/rewired/docs/ControlMapper.html> [Accessed 10 Jan. 2022].

guavaman.com. (n.d.). *Rewired Documentation | Overview*. [online] Available at: <https://guavaman.com/projects/rewired/docs/Overview.html> [Accessed 10 Jan. 2022].

Hesse, B. (2021). *How to Remap Your Gaming Controller's Buttons on Any Platform*. [online] lifehacker. Available at: <https://lifehacker.com/how-to-remap-your-gaming-controllers-buttons-on-any-pla-1846783156> [Accessed 10 Jan. 2022].

History.com. Editors (2019). *Video Game History*. [online] Available at: <https://www.history.com/topics/inventions/history-of-video-games> [Accessed 8 Dec. 2021].

History of video games timeline: Wolf, M.J.P. (2008). *The video game explosion : a history from Pong to PlayStation and beyond*. Westport, Conn.: Greenwood Press.

loc.gov. (n.d.). *Video Gaming Accessibility*. [online] Available at: <https://www.loc.gov/nls/resources/general-resources-on-disabilities/video-gaming-accessibility/> [Accessed 10 Jan. 2022].

Martin, M. (2019). *What is a Functional Requirement? Specification, Types, EXAMPLES*. [online] Guru99.com. Available at: <https://www.guru99.com/functional-requirement-specification-example.html> [Accessed 6 Jan. 2022].

Ready Players. (2016). *Little Metal Limbs – Portrayals of the Disabled in Games*. [online] Available at: <https://readyplayers.wordpress.com/2016/06/23/little-metal-limbs-portrayals-of-the-disabled-in-games/> [Accessed 10 Jan. 2022].

rewasd. (2017). *Use reWASD to remap controller buttons on PC completely*. [online] Available at: <https://www.rewasd.com/blog/post/map-controller-to-keyboard-hardware-mapping> [Accessed 10 Jan. 2022].

Rockstar Games (2013). Grand Theft Auto V. [online] Xbox, New York, Rockstar Games

TechTalk. (n.d.). *MENTAL HEALTH & DISABILITIES*. [online] Available at: <https://techtalk.currys.co.uk/tv-gaming/gaming/diversity-in-gaming/games-and-disabilities.html#:~:text=Physical%20disabilities%20are%20portrayed%20in,re%20most%20likely%20to%20tackle>. [Accessed 10 Jan. 2022].

Valentine, R. (2020). *Three-quarters of all US households include someone who plays video games*. [online] gamesindustry.biz. Available at: <https://www.gamesindustry.biz/articles/2020-07-15-three-quarters-of-all-us-households-include-someone-who-plays-video-games> [Accessed 9 Dec. 2021].

Westin T., Bierre K., Gramenos D., Hinn M. (2011) Advances in Game Accessibility from 2005 to 2010. In: Stephanidis C. (eds) Universal Access in Human-Computer Interaction. Users Diversity. UAHCI 2011. Lecture Notes in Computer Science, vol 6766. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21663-3_43 [Accessed 10 Jan. 2022]

Wobbles (2020). *Input Mapper 1.7*. [online] InputMapper. Available at: <https://beta.inputmapper.com/> [Accessed 24 Jan. 2022].

Figures

Figure No.	Image	Description
1	<p>Mental health has only been tackled in the last few years</p> <p>No. of games</p> <p>1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018</p> <p>■ All disabilities ■ Mental health</p>	Graph that shows how mental health has been tackled in video games over the last few years.

2	<div><table><thead><tr><th>Disability Type</th><th>Percentage</th></tr></thead><tbody><tr><td>Physical</td><td>54%</td></tr><tr><td>Physical/Mental</td><td>20%</td></tr><tr><td>Sensory</td><td>20%</td></tr><tr><td>Mental</td><td>3%</td></tr><tr><td>Sensory/Physical</td><td>3%</td></tr></tbody></table><p>Physical disabilities are portrayed in games most often</p><p>Physical disabilities might include amputees, burn victims, victims of ballistic trauma, wheelchair-bound characters and more. For developers, these are the disabilities they're most likely to tackle. More nuanced issues, like anxiety and depression, are far less prevalent in games.</p></div>	Disability Type	Percentage	Physical	54%	Physical/Mental	20%	Sensory	20%	Mental	3%	Sensory/Physical	3%	<p>Pie chart that shows how much different types of disabilities are portrayed when disability is included in games.</p>
Disability Type	Percentage													
Physical	54%													
Physical/Mental	20%													
Sensory	20%													
Mental	3%													
Sensory/Physical	3%													
3	<div><p>The screenshot shows the InputMapper application window. On the left is a sidebar with navigation options: Devices (selected), Announcements, Profiles, Settings, and Donations. The main area displays settings for a 'DUALSHOCK®4 USB Wireless Adaptor'. It includes a 'New Profile' button, a 'Device Tuning' dropdown, and a 'Monitor' button. Below this, there are two sections: 'Data Rate' with a green progress bar, and 'Player 1' with a '[Default]' dropdown, another 'Device Tuning' dropdown, and a 'Monitor' button. A 'Current App' section at the bottom left shows 'InputMapper' as the active application.</p></div>	<p>Main page of Input Mapper program, showing two different controllers (input devices)</p>												