Requirements Specification for:

<VR ‘Ramparts’ Fitness Game ‘oVRthrow’>

**Prepared By:**

**<Benjamin Belay, Isaac Ray, Christopher Stuttard and Johnathon Yeap>**

# 1 – Introduction (20% - Max 2 Pages)

* 1. - Overview and justification (purpose):

IBM are looking to reimagine the game 80’s ‘Ramparts’ in VR a game that combines puzzles, combat and strategy together. Reimagining this game in VR opens up endless new opportunities to develop this classic to a new level. COVID-19 has affected everyone across the world over the past year and a half and in particular it has had a major effect on peoples fitness and physical health as they have not been able to leave their house to exercise in the usual way. Combining an exciting game with fitness challenges will hopefully be one of the great motivations people need to get back in to the exercise routine that they once had.  
  
This game will benefit lots of different age ranges. The minimum age range for VR Gaming is 7+. So, everyone over the age of 7 will be able to play this game. It will not contain any exercise that is too complicated for the older generation to also take part in.  
The VR experience will also include Watson Text-to-speech service allowing the user to shout commands and fully interact with the game. This is an added bonus for people who have had little interaction in recent times as they can immerses themselves in an entirely new experience.

In the rest of Section 1 the outline of the project and the scope of the project are given with a description of everything we are aiming to achieve. Then in Section 2 we will identify all of the functional and non-functional requirements for this project as well as any potential risks and issues we could also encounter in the development of the project as well as risk of the project itself. Finally in Section 3 we will outline the development approach we will take to complete this project and the schedule in which we will complete each phase.

1.2 - Project scope:The scope of this project is to create a VR game. In this VR Game there will be an available play space where each of the players are required to construct their own castle by building walls within a given time limit. If they don’t manage to build their castle in time, then they will lose. We must create this play space and the control panel for users to build walls with a form of exercise. Once this phase of the game from the original ramparts is complete the combat phase can begin. The main boundaries we encounter in this phase are the ability to create an appropriate play space for the users to move and build in. Its possible that future version are when this play space really comes to life and isn’t just a simple field.  
  
The second phases of the game is the combat which is where the real exercise and fitness will take place. The user will pick up a cannonball and throw it using the motion of a kettlebell swing. A harder swing will make the cannonball go further and do more damage. A lighter swing will cause less damage. This should be entertaining but at the same time quite strenuos and purposeful as you will feel out of breath afterwards. The game will get harder as the game goes on as you are more tired and the opponents walls get stronger. Watson-text-to speech will also be used to shout commands such as ‘raise shields’ to invoke power-ups.  
We will also need to create several other screens within the game. A login screen to choose how many players and rounds you would like to play. Screens between phases to outline what you need to do on your first few times playing the game only. And a screen for when you lose as your castle was destroyed to much hence it couldn’t be rebuilt in time.  
In terms of back end we will need to store all the different power ups and lots of imagery and artwork for this VR project. The game is played from a fresh every time you play and doesn’t need to save any details, nor do we need to score any sensitive data. We may store high scores and other data, or this could be something incorporated in future versions of the game.  
  
We will be dealing with IBM who are the main stakeholder and have the most input into how we create this game. Other stake holders include players of the original ramparts game who we hope will want to play our new version so we must make sure they agree we keep enough features true to the original. We also need to make sure the game is challenging enough to make it a worthwhile fitness game for all ages, so we have stakeholders from each age group and gender to consider.

1.3 - System description: The Existing Game – The current game Ramparts was released in 1990 and featured on some old consoles including the SNES. It was distributed in Japan by Namco. This 2D game is now very outdated compared to modern technologies. It contains no fitness whatsoever as you sit down and play it with a dpad. It does however have this combat and repair phase where you shoot cannonballs and then quickly scurry to rebuild walls which we intend to keep in our version of this game.  
  
Comparable Games – There a few VR fitness game on the market although unfortunately none of our team has the equipment to be able to play these to see their benefits and flaws. Instead, we have read the feedback of many users to see what they did and didn’t enjoy within such a game genre. Pistol whip is 1 such rhythmic shooter. It received fantastic reviews as people enjoyed the level-based system with difficulties for all. People seem to thrive of a in intense game rather than something that is relatively simple. We intend to apply this pressure when rebuilding your castle. In other games common complaints people had been the game going on too long that people were too tired out or just fed up of repetitive exercises with no real end goal to work towards. Other games didn’t clearly register an exercise when it was completed which was frustrating for users.   
  
Our Game (oVRthrow) – Our system brings the existing game which is now significantly outdated back to life in an exciting new format. An open world experience that provokes intense gameplay and fitness as well as puzzles and strategy. The software to be built is the play space an open world mountainous area the players can move around in. Choosing a spot to build walls in a grid system similar to that of other game like ‘Minecraft’ and ‘Clash of Clans’. Players will stretch and jump to reach and build the walls they desire while the clock tick downs on them. If they don’t complete their castle the unfortunately, they will be eliminated.  
  
Watson-text-to speech software must be used to allow the users to shout command in order to use a relevant power up during the repair or combat phase which causes a reaction in game immediately. We are planning to use UNITY which uses C# as language to develop the game itself as well as some 3D Design software to develop the stunning visual we intend to achieve. The system will not be required to store any data so no complex databases are needed.

# 2 – Solution Requirements (50% - Max 10 Pages)

2.1 - Functional requirements:Describe the system functions requirements and include requirement models e.g. use case, activity charts where useful (do not do diagrams for every function). You can also use data flow diagrams, flowcharts and decision tables if required. List each system function as a subsection with a brief description. An example of this could be something like:

|  |  |
| --- | --- |
| ID, type and title | e.g. FR1.2 Mobile application – filtering results |
| Description | A use can filter results in a list or a map. When viewing the results in a list or a map a user should be able to filter the results. Filtering options provided are:  • Filter by choosing a restaurant type  • Filter by a specific food dish |
| Priority | e.g. High |
| MuShCo | e.g. Must have |
| Dependencies | e.g N/A or FR7 and FR20 |
| Expected results | e.g. when filtering the results, only the existing results area is affected and a new search query should not be sent. |
| Exception handling | e.g. The user may abandon the search at any time  e.g. error message generated e.g. data for filtering does not exist |

2.2 - Non-functional requirements:Identify a number of non-functional (quality) requirements for the system, such as security, scalability, usability, etc. These requirements vary from project to project. Example:

|  |  |
| --- | --- |
| Type | e.g. Space requirement, Security requirement, Ethical requirement etc … |
| Metrics | How to measure the NFR, e.g. sub-component A needs to have a response time <1ms, 95% of the time. |
| Security | Only users holding the role “Customer advisor” or “Supervisor” can update the customer record |
| Constraints | Are there any constraints on the NFR? E.g. hardware to be used, OS, DBMS, I/O devices, Standards, data structures, language, documents … |

2.3 – Risks and Issues:Briefly identify and discuss any potential risks (or issues with the potential to become risks) that could possibly impact the project. Discuss any potential limitations of the group/hardware/software/current systems etc., and the problems that these could cause and the mitigation proposed. Finally classify the risk or uncertainty vs loss.

Recommended 1 page.

# 3 – Project Development (20% - Max 2 Pages)

3.1 – Development approach: Discuss the SDLC approach the group would like to use for this project. You should justify your reasons for selecting any particular approaches as opposed to others.

Recommended 1 page.

3.2 – Project Schedule:Provide a project schedule, identifying deadlines for key aspects of the project. This can be provided in whatever format you deem most suitable