## Cross Platform Development: Assessment Brief

### Suggested Project Brief – Duck Hunt

#### Overview:

Although virtual reality is not new, and in fact has a history dating back to the 1060’s, the modern iteration of virtual reality platforms are still very much in their infancy. Yet this hardware captures the imagination of both consumers and developers, and provides new opportunities for established studios and independent developers alike.

Yet developing a VR or AR experience is not without its challenges, and ensuring your game functions well and is fun to play on all release platforms can sometimes be challenging. Differences in hardware processor speed, input methods, screen aspect ratio and resolution are just some of the hurdles that must be considered and planned for.

In this assignment you are tasked with developing a fun, casual virtual reality game for PC and at least one VR platform (Oculus Go or Google Cardboard recommended).

#### Implementation:

*Duck Hunt*, by Nintendo (1984) was developed for the Nintendo Entertainment System and utilized the *NES Zapper* light gun– a physical gun that allowed players to shoot objects within the game.

The game itself was very simple, with the player required to shoot the wildlife appearing onscreen without exhausting their allocated ammunition. The goal was to maximum your points through shooting accuracy.



You can familiarise yourself with the original game here: <https://www.youtube.com/watch?v=x-daxzVxrQI>

Your task is to create a modern iteration of the classic game *Duck Hunt*, using virtual reality.



You are tasked with creating your own virtual reality version of *Duck Hunt*. Ensure your game has the following features:

* Your game must be made using a 3D game engine, like *Unity 3D* or *Unreal Engine 4*.
* It must target at least one virtual reality platform
* Your game must use 3D primitives, positioned in a 3D world.
* Your game must include a GUI showing the player score, or other game information. Consider adding a main menu, high-scores screen, or other screens.  
  Ensure your UI aligns with current industry best practice for VR. For example, you should consider adding diegetic UI elements within your game.
* Your game must run on at least 2 different platforms (for example, PC and Oculus Go or mobile using Google Cardboard) and must take advantage of the different features of those platforms (for example, gyroscopic input on mobile devices).   
  Note, your PC version does not necessarily need to be ‘VR’ if you are also targeting the Oculus Go, for example. But you must ensure input is optimised for each platform.
* It must be possible to build all versions of the game from the same source code/project.
* Create an installer for your PC version, and ensure your mobile build is packaged appropriately (i.e., create an installable .apk for an Android build). The PC installer must also contain an uninstaller.

In addition to the game, you also need to create some planning documentation. Create a document that:

* Outlines the design of your game;
* Identifies any issues in cross-platform and VR development. This may include:
  + User interface mock-ups specifying input or display adjustments required across platforms,
  + An identification and discussion of player control issues across platforms,
  + Screen size and aspect ratio differences,
  + API or software version requirements,
  + Deployment methods,
  + Platform-specific features or constraints, and
  + Industry standard tools, APIs, or methods for handling cross-platform development issues.
* Lists any third-party tools or libraries used, or the sources for any artwork or animations not created yourself;

Include in your planning document an analysis of how extended realities are used in games and/or alternative industries. For your selected industry (i.e. games, simulation, serious games, etc.) include:

* A brief discussion on the history of AR/VR for the selected industry,
* Research on target markets and demographics for the selected industry, and
* A list of successful or competing AR/VR products for the selected industry.

This discussion will help guide and justify your own design decisions regarding industry best practice in extended reality application design and development.

You are also to report on the technology you will use. Ensure you discuss the physiological constraints of the selected platform, including how these constraints are addressed; whether haptic feedback is supported, and a discussion of its potential in applications or games; optimization and performance considerations; market size; technical specifications; and any other pertinent information.

#### Seek and Evaluate Feedback

Conduct play testing to gather feedback on your game across the platforms you have targeted. Review and evaluate this feedback to identify areas where your game could be improved. Document this process, ensuring you include any prepared testing plan or playtest questionnaires along with the player feedback.

* Record play test results and a feedback evaluation in a written document.
  + Your report must include an evaluation of player feedback identifying areas of improvement.
  + Include any prepared testing plans or playtest questionnaires along with the documented player feedback in a Word or PDF document, to be submitted along with your other deliverables.

#### Code Documentation

Finally, you are to use an integrated or third-party documentation tool (such as Doxygen) to create and maintain code documentation.

#### Submission

You will need to submit the following:

* A Release build of your gamer for each target platform that can execute as a stand-along program
* An installer / uninstaller for the PC build of your game
* A planning document identifying and discussing cross-platform development issues and solutions
* Play testing feedback and evaluation document
* Source code documentation, created with a third-party documentation tool (such as Doxygen)
* Your complete game project, including source code

Be sure to remove any temporary build folders (i.e., the Debug and Release folders, and Library folder) and unused assets included in your project. Only project files, source code files, and any resource files used should be included in your submission.

Package all files in a single compressed archive file (.zip, .7z, or .rar)

***Submission Checklist:***

This submission checklist is used to assist your assessor in marking your assessment.

A copy of this checklist can be downloaded from <https://aie.instructure.com/> and must be submitted with your project.

**General**

|  |  |
| --- | --- |
| **Description** | **Y/N** |
| All submitted projects compile without errors *Programs that don’t compile cannot be assessed* |  |
| The program includes a “readme” or document explaining how to compile, execute and operate the program |  |
| The program performs as described in the general description |  |
| The program contains no logical errors |  |
| The code is sufficiently commented and clean |  |
| An attempt has been made to increase the program’s efficiency |  |
| Code compiles without no warnings |  |
| Program executes without crashing |  |
| Program has no memory leaks, and closes all files after use |  |
| A release executable has been made and included in the submission |  |
| Project files and source code are included in the submission |  |
| All files are packaged in a single compressed archive |  |

**Required Features**Complete the following table by providing the class name or file name, along with the line number, to show where you have implemented each feature.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Class/File** | **Line Number** |
| Player input is handled in an industry standard way across different devices |  |  |
| Game world objects are manipulated using code |  |  |

|  |  |
| --- | --- |
| **Feature** | **Y/N** |
| You have made a VR game that executes on at least one VR platform |  |
| You have written a planning document that identifies and discusses cross-platform and VR development issues |  |
| The game compiles and runs error free |  |
| The game executes without crashing |  |
| One game project is used to build release builds for all target hardware |  |
| You have written game code in a programming language such as C# or C++ |  |
| The game includes a GUI |  |
| Code is executed in response to GUI events |  |
| The GUI is handled correctly across different devices |  |
| Play testing has been conducted and a play test evaluation document has been written |  |
| A release executable has been made for each target device and included in the submission |  |
| An installer/uninstaller has been included for the PC build |  |
| Source code documentation has been created (using a program like Doxygen) and included in the submission |  |