Cross Platform Development – Project Research Workbook

This workbook will help you focus your research for your project.  
Once you have answered these questions, use this information in your Technical Design Document.

|  |
| --- |
| Briefly describe the cross-platform application, game or simulation you are researching.  (This is your initial idea to focus your research. The application described in your design documents or your final build may end up being different from this description) |
| Description: I will be developing the game *Potion Shopkeeper*.  The player is the potion maker of a small village and the game takes place in their house and garden. The player gathers ingredients from their garden, combines those ingredients in their laboratory to make potions, and gives those potions to villagers who present with varying problems. The player must experiment with different ingredients to make different potions which address the needs of the villagers.  Potion Shopkeeper is a relaxed, 1-player, 3D side-scrolling point & click game. The game is a simplified version of Potion Craft: Alchemist Simulator.  Potion Shopkeeper will be developed in Unity (2021.3.13f1) with Visual Studio 2022 with release builds for 3 platforms:  • standalone PC application (Windows 10).  • web browser (Chrome).  • mobile phone (Android App Bundle). Mandatory requirements: Using this document as the basis for my cross-platform development prototype, I must implement:   * Creating a 3D environment * Creating start, help, main and settings menus * Animations * Visual effects * Unity 3D player character with controller * Deploy to PC * Deploy to web * Deploy to Android * Create appropriate installers * One or more of:   + Adding music   + Adding on-click sound effects  Optional extras:  * A side-scrolling camera locked to the player * Adding an on-mouseover glow to objects * Pausing the music * Adjusting the music volume * Adjusting the effects volume * Changing the resolution * Changing the language * Creating drop-down menu fields * Creating menus which expand on button press |

|  |
| --- |
| List the software you will use to create your project.  Include any third-party plug-ins, APIs or libraries, if known. |
| |  |  | | --- | --- | | Software | Purpose | | Microsoft Visual Studio | IDE | | Microsoft Word | Project management | | Microsoft Excel | Project management | | Draw.io | Project management | | Microsoft PowerPoint | Prototype presentation | | Unity | Game development (all) | | Android studio | Game development (mobile) | | itch.io | Game development (web) | |

|  |
| --- |
| With reference to the above list, what legislative frameworks or organisational standards govern the use of this software (including any third-party plug-ins, APIs or libraries).  For example, include any End User Licence Agreements (EULAs), terms of service, copyright notices, licencing information, developer guidelines, coding standards, or similar.  (Information in the AIE Student Handbook may also be relevant in relation to the use of software on campus machines.)  Include URL links where relevant. |
| **General information:**  [Media classification ratings in Australia](https://www.classification.gov.au/classification-ratings/what-do-ratings-mean)  **PC & General development:**  [Visual Studio Community 2022 Software Licence Terms](https://visualstudio.microsoft.com/license-terms/vs2022-ga-community/)  [Unity Terms of Service](https://unity3d.com/legal/terms-of-service)  **Mobile:**  [Android Software Development Kit Licence Agreement](https://developer.android.com/studio/terms)  [Developing for Android](https://developer.android.com/develop/ui)  [Google Play Store developer policy centre](https://play.google.com/about/developer-content-policy/)  **Web:**  [itch.io terms of service](https://itch.io/docs/legal/terms)  [itch.io content creator quality guidelines](https://itch.io/docs/creators/quality-guidelines)  [itch.io community rules](https://itch.io/docs/general/community-rules)  **UI development guides:**  [Intro to UI versus UX](https://www.interaction-design.org/literature/topics/ui-design)  **Apple:**  I’m not developing for Apple devices, but Apple is an industry leader in aesthetically pleasing, effective, efficient and economical UI design.  [UI Dos and Don’ts](https://developer.apple.com/design/tips/)  [Human interface guidelines](https://developer.apple.com/design/human-interface-guidelines/)  **Android:**  [Designing for Android](https://developer.android.com/design/ui) |

|  |
| --- |
| List the cross-platform installers and installation methods you will use, or the specific binary formats that are required to deploy the game.  This list should include all platforms you plan to deploy your game or application to.  (Your game or application must be deployed to at least two different web browsers, and at least two different digital devices – one of which may be PC) |
| PC:  Mobile:  Web 1 (Chrome):  Web 2 (Edge): |

|  |
| --- |
| What IDE will you use?  Identify your reasons behind this choice (ignoring the pre-configured environment on the campus computers). |
| **Microsoft Visual Studio (VS):**  VS is a professional benchmark for game development and is the IDE I have experience with. VS appropriately and reliably interfaces with Unity, and its IntelliSense is advantageous both for development efficiency and review. |

|  |
| --- |
| Identify the cross-platform libraries, plug-ins, or APIs you will use.  Mention any restrictions or limitations that exist with these libraries on each target platform.  For example, some parts of the .NET class libraries implicitly depend on threads, but some platforms (WebGL) do not support threads. |
| The principal cross-platform functionality I require is Unity’s *Input System* which will enable a single collection of character navigation commands to be applicable to multiple platform builds (PC, web and mobile).  I haven’t identified any other cross-platform libraries, plugins or APIs that will be necessary for my prototype. |

|  |
| --- |
| What issues exist, or do you expect might exist when developing for the target platforms you have identified? |
| I believe that the most pressing issue for me will be publishing to mobile and web, with learning and utilising the Unity ‘new’ Input System second, and UI scaling and proportionality third.  All knowledge required for mobile and webGL design, development and publishing will be new to me.  I have very few hours of experience with Unity, so building code as a deliverable in-engine is new to me, and I will have to learn how to create installers, and use itch.io. Game engine sound, animation and physics will all be new to me, so will require more of my attention than UI, raw C# coding, documentation and testing, or giving an oral presentation. |

|  |
| --- |
| List any areas in your game where pre-written scripting packages could aid in development.  For at least one of these items, identify a package from the Unity Asset Store (or another source) that may be suitable. |
| My prototype has been designed from the beginning to capitalise on my User Interface body of work, together with the *Visual Effects* cross-platform development tutorial, and so the only scripting packages I should need to write will be for implementing:  **Player**   * The Player object itself.   + I will need a 3D model, a skeleton for it, relevant animations, as well as the script assets to make it work. These assets are provided for us through the *Visual Effects* tutorial.   **Laboratory**   * The Laboratory object itself.   + I have a working prefab inventory from my UI tutorials, but I need to turn the standalone screen-space overlay UI inventory into a UI which is presented when a clickable 3D object in the world is clicked. * The ‘brew’ function of the laboratory to turn 3 Ingredient objects into 1 Potion object is new, but I have assets already built for the relevant inventory, buttons, text boxes, etc.   + I believe I should be able to amend the drag and drop ‘swapping’ effect here to conditionally turn 2/3 Ingredients into blank items, and 1 into the Potion.   **Villager**   * The Villager object itself is new.   + Inventory is a pre-existing asset I can re-use.     - I have a working prefab inventory from my UI tutorials, but I need to turn the standalone screen-space overlay UI inventory into a UI which is presented when a clickable 3D object in the world is clicked.   + Menu assets can be re-used from UI. * The Villager’s requested Potion.   + In the prototype this will simply be a single prompt asking for a single potion of non-changing type (eg always a healing potion). This will be as simple as re-using menu prefab assets.   **Items**   * Ingredients and Potions themselves require a trivial amount of work to implement, as I have a system of inventory items pre-existing from the UI tutorials, written as ScriptableObjects.   **End day function**   * Functionally a ‘reset’ button for the scene to renew the ingredients and the request from the Villager. Not mandatory for implementation, but ideal.   **User interface**   * I have already downloaded and used three free user interface kits from the Unity Store for cosmetic improvements to the prototype. The first of these, *Quick Outline*, includes a pre-written scripting package. In addition, my pre-existing UI tutorial assessments form the basis of my gameplay loop and UI design.   + [*Quick Outline* by Chris Nolet](https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488)     - Adds a ‘glow’ outline to 3D game objects in Unity on mouseover.   + [*Simple Buttons* by Nayrissa](https://assetstore.unity.com/packages/2d/gui/icons/371-simple-buttons-pack-97516)     - Button icons which I have made use of in my UI.   + [*Basic RPG Icons* by Poneti](https://assetstore.unity.com/packages/2d/gui/icons/basic-rpg-icons-181301)     - Sprites which I have used for Potion and Ingredient item inventory icons |

|  |
| --- |
| List the pre-written scripting packages or plug-ins you will use during development.  (Include a URL for each package or plugin) |
| [*Quick Outline* by Chris Nolet](https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488)  Adds a ‘glow’ outline to 3D game objects in Unity on mouseover.  The 3D object, animations, particle effects and sounds from the cross-platform development *Visual Effects* tutorial.  My own pre-existing UI tutorials and assessments. |

|  |
| --- |
| List the game engine and any additional development tools you will use. |
| **Unity:**  While I am very new to Unity, it is the game engine I have more experience with (Unreal was the only realistic alternative).  **Additional tools:**  Code implementation will be completed in Visual Studio. Android studio will be my developer tool for mobile, which I will be testing on my personal phone. itch.io will be the publication platform for webGL.  I haven’t identified any other code development tools that I will be using. |

**NOTE TO SELF: PRESENTATION WAS APPROX 2:30 ON 21/09/2023**