Contents

[Document Information 1](#_Toc1178117307)

[Approval 1](#_Toc249082978)

[Development Software 2](#_Toc1778529786)

[Version Control 2](#_Toc1012735916)

[Code Testing 2](#_Toc551065399)

[Coding Conventions 2](#_Toc12482230)

# Document Information

|  |  |
| --- | --- |
| **Project name:** | VR Training Project |
| **Date:** | 24-01-2025 |
| **Author:** | Byron Griggs |
| **Contributor:** | Hayley Elahee, Owen Beeks |
| **Approver:** | Byron Griggs |
| **Document code:** | 2291/2305 |
| **Version:** | V4 |

# Approval

|  |  |
| --- | --- |
| **Date** | **Name and Signature** |
| 18/11/2024 | Owen Beeks |
| 25/11/2024 | Owen Beeks |
| 14/01/2025 | Owen Beeks |
| 27-01-2025 | Byron Griggs |
|  |  |

Note: The procedures in this document are suggestions, but team members are expected to follow them unless there is a good reason not to, at the risk of the project’s quality and maintainability.

Note 2: Whoever reviews this document, this is how I learned to do things. That doesn’t mean it’s the right way. Feel free to talk to me if you want to change something.

# Development Software

* Coding should be done in Unity and Visual Studio Code
  + Unity Version 022.3.36f1
  + Visual Studio 17.11
* Programming will be completed in C#

# Version Control

* All code (and Unity project files) are stored in GitHub first and Azure Devops as backup
* Team members must check out documents from GitHub and check back in when they are done. This must be done at the beginning and end of each working day, though team members may do it more often if they wish.
  + Documents should not be merged to the main branch without documentation of changes made, which is approved by a different team member.
    - It is recommended that codes are merged to main once per sprint
  + If team members do not know to check in and check out from GitHub, they may do so manually, by creating a branch and naming it after themselves and the date.

# Code Testing

* Team members are free to test and debug code as they wish, but any code that embodies a backlog task bust be tested before being merged into the main branch.
  + Tests must be created and performed by someone other than the primary author of the code. If authors are mixed, anyone can make the test.
* Testing should emphasize meeting the feature specifications in the SRS, as well as accounting for unexpected inputs/situations
  + Example: If a user tries to open the fridge while the kettle is boiling, the game should not crash.

# Coding Conventions

Table declaring essential /common conventions:

|  |  |  |
| --- | --- | --- |
| **Type** | **Convention** | **Example** |
| *Namespaces* | Pascal\_Casing | My\_Namespace |
| *Class Names //Methods* | PascalCasing | ClearData(); |
| *Local Variables / method arguments* | camelCasing | logEvent |
| *Abbreviations 3 char or more* | PascalCasing | HtmlHelper htmlHelper |
| *Type Names* | Type Names not System Names | string, not String  int, not Int32  bool, not Boolean |
| *Interface Names* | Prefix with I | IShape |
| *Curly Brackets* | Vertically Allign | FunctionName()  {  } |
| *Member Variables* | Declare at top of class | public class Example  {  public static string BankName;  } |
| *Enums* | Singular Names and DO NOT suffix | public enum Color  {  } |
| *Spacing / Indentation* | One tab key or the equivalent of 4 spaces | public enum Color  {  Tabbed code  } |

* AVOID Abbreviations
* DO NOT use SCREAMINGCAPS
* DO NOT use Underscores
* Organize Namespaces

Alongside these ensure you are using Standardised C# Coding and Naming Conventions. As well as the ones declared above, please use the following websites to ensure correct coding conventions:

* [C# Naming Conventions Cheat Sheet by GregFinzer - Download free from Cheatography - Cheatography.com: Cheat Sheets For Every Occasion](https://cheatography.com/gregfinzer/cheat-sheets/c-naming-conventions/#downloads)
* [C# Coding Standards and Best Practices](https://www.dofactory.com/csharp-coding-standards)
* [Identifier names - rules and conventions - C# | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/identifier-names)
* [.NET Coding Conventions - C# | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions)