Unity Workflow Toolkit

Final Major Project GDD

Contents

Document Information **3**

Document Information

Start Date: 02/10/2024

Current Version: 0.5

Last Modified: 10/10/24

Changelog

Date	Vers	Changes Made
03/10/2024	0.2	General Additions to the GDD
03/10/2024	0.3	Fixed spacing, added Risk Assessment / Mitigation
05/10/2024	0.4	Addition of Testing Plans

Project Overview

Title: Unity Workflow Toolkit

Platform: Unity

Genre: Plugin / Tool

Deadline: April 2025?

Artist(s): N/A

Programmer(s): Wattie

Description

Unity Workflow Toolkit is a set of Plugins / Tools for Unity. This project focuses on delivering QOL (Quality Of Life) Features, making it easier for developers to create and manage their projects.

This plugin is aimed at all developers, no matter their skill level - this tool simply streamlines boring and repetitive tasks.

Dependencies

This tool is designed for Unity **2022.3.18f1**, but I plan to allow this to work from Unity 2019 to the present. I will be explaining this in more detail in the Technical Design Document.

Deliverables

For this part of my Final Major Project, I intend to deliver this document - which can be classed as the **“Game Design Document”** - although it is not the game, a **Technical Design Document** that will be used to document the “Code” side of the plugins/tools, and everything else technical. I intend on delivering at least **4 Tools** - but I plan for **5-6**. These include:

Asset Organiser

This script will organise assets, files, scripts and more to ensure that they are in the correct folder; for instance, when developing games, I sometimes put the scripts in a random folder for testing and forget about it; this will ensure that the correct assets are in the proper spots.

Bulk Rename Tool

When Developing Games, you usually have Asset Names that might be annoying - for instance, I had many individual files to animate a button, and I wanted to change the names of them, but there were 50+ individual files I would have to rename; this script makes it very easy to change the name of all of them at once.

Post-Build Automation

The Post-Build Automation tool can automatically do a few different things; the most useful of it is that after your project has been built, it will compress the builds into a .zip file (while you also have the option to add more files, like a “Credits” file, or your Game’s Soundtrack to the zip file.

Deliverables

Script Counter

This tool is similar to a “word count” in document editing software; it will be able to count different types of scripts, text files, assets, and models your project has, you will be able to choose which directories it targets, and there will be settings, such as being able to copy/paste your script names, sort the list as alphabetically, see file size of scripts, and total file sizes and much more - this is very handy if you are writing documentation for your games.

Unused Assets Remover

This tool will search your files and remove unnecessary assets that might balloon your Game’s build size.

Notes

Adds a small window to make quick reference notes while programming; this tool will be powerful in late-night coding sessions.

Fancy Screenshots

Screenshot your game quickly; it will also be customisable (for example, add watermarks, drop shadows, and borders.

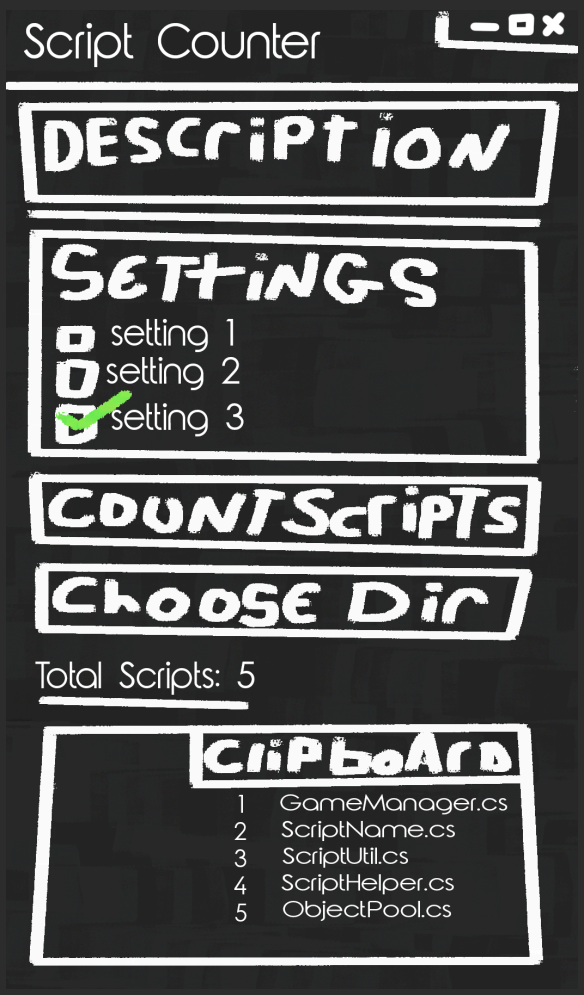
User Interface

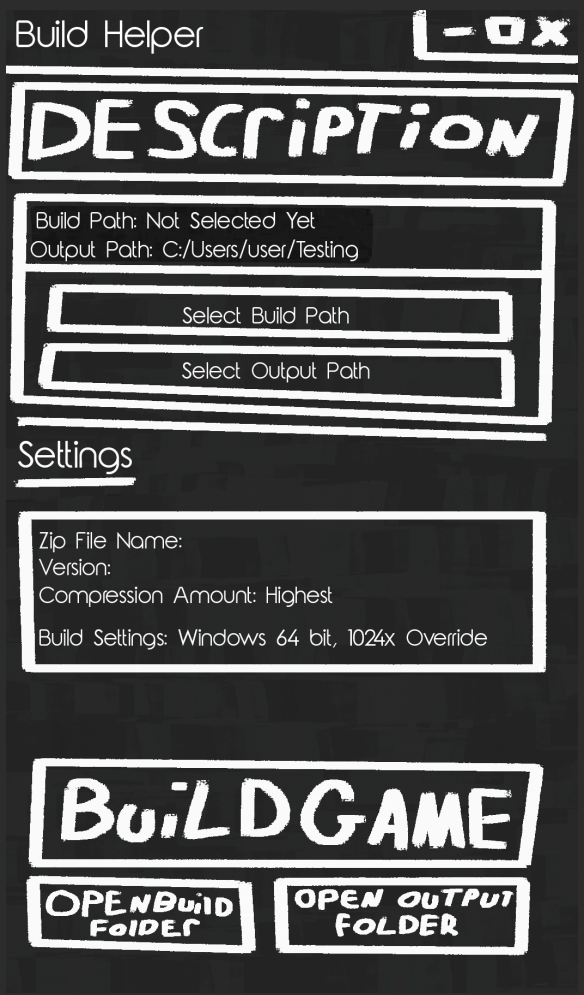
Style

The Tool's User Interface should be sleek and modern. Unity's in-built GUI controls for making tools are a little outdated, but I can draw boxes and make the script look a little nicer with this.

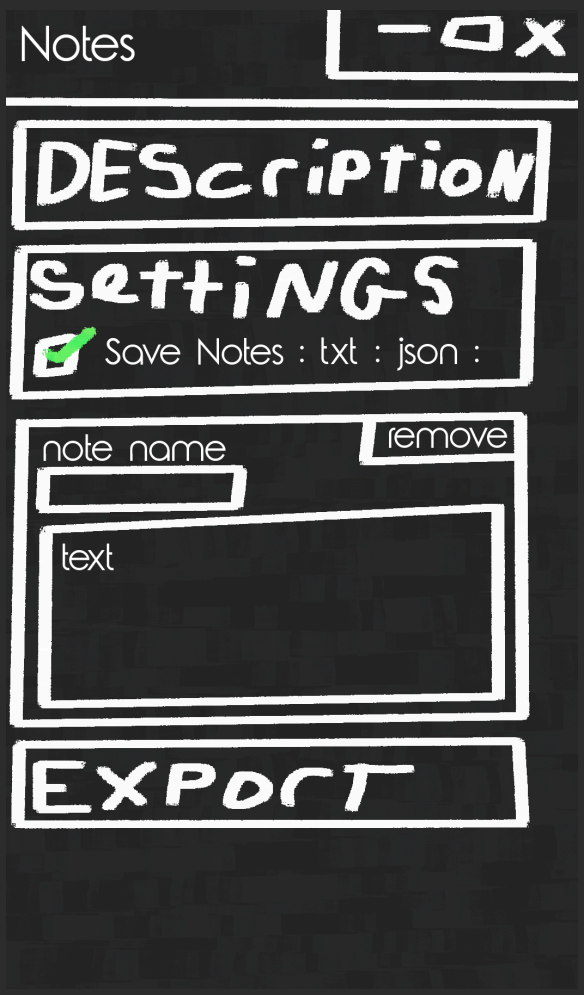
Mock-Ups

I have made some mockups of the tool's User Interface (that will be made with the in-built GUI workflow)





User Interface



Dropdowns

It has been suggested to me that Drop-Downs should be introduced in the tools, both by category (for example, settings) and individual features (for example, each setting in settings); this will make it so users of this tool have some sort of customizability.

Tool Usage and Workflow

What does UWT Solve?

Have you ever wanted a solution to tedious, repetitive tasks inside Unity Editor? I've noticed several areas where the Unity Editor could be optimised and isn't; for example, while developing games, I constantly had to build different game versions, manually adding files & folders, then zipping and naming and sharing them. This manual process was very frustrating and time-consuming; this is what inspired me to start this project.

Why UWT over others?

Some similar tools are already made, but they are made with the mindset of creating levels and customising the editor instead of managing boring and repetitive tasks; it is also more concise, easier to use and much more straightforward than other solutions.

How it works

Here's how you can use UWT and its tools effectively:

Tool Usage and Workflow

Post Build Actions

Step 1: Select the build folder (where your build files go) and output folder (where you want the zip file to appear)

Step 2: Customise file names, timestamps, versioning etc.

Step 3: Choose additional files if necessary (songs, licences. Readme files) and name the folder you want the files to appear in

Step 4: Press the “Execute Post Build Actions” button, wait for it to bundle everything together automatically, and save it for distribution.

Script Counter

Step 1: Select the folder containing assets/scripts you want to count

Step 2: Customise settings like file extensions (to only count the asset type you want)

Step 3: View the scripts or files, and press “Copy To Clipboard” if you’d like to paste it somewhere

Tool Usage and Workflow

Notes

Step 1: Select your desired settings (if you'd like to export the Note) and what file type the note will be (.txt, .json)

Step 2: Select Export Paths

Step 3: Click "New Note", name the note and write your note

Step 4: if you want to export them, click the "Export" Button.

Screenshot

Step 1: Set Export File Path

Step 2: Set naming conventions for screenshots (e.g. UUID, Animals, Random)

Step 3: Customise other screenshot settings (resolution size, border)

Step 4: Press the Export button to share your screenshot freely

Monetisation Strategy

Monetisation

It could be possible to monetise this toolkit; I do believe that once developers start experiencing the efficiency the tools give, I believe they would pay for it - I don't think I would price the bundle anything significant, but something to earn me a bit of money here & there - the more substantial challenge would be getting a Game Studio to use these scripts (and pay for it).

I prefer allowing these scripts to be used by anybody, as this would make a better contribution back to the Games Industry, but as a compromise, a minimal fee might work.

Distribution

Distribution could be handled through the Unity Asset Store. You can do all types of things here, like scripts, assets, tools, music/audio, and more, so this is an excellent place to start.

Another place I could put these projects is on Itch.io; they have a "Marketplace" similar to Unity Asset Store but lack some functionality in importing the tools.

User Feedback & Testing Plan

Objective

The main objective of the Testing Plan is to ensure the effectiveness, ease of use and integration of the Unity Workflow Toolkit (UWT) within Unity Developer Workflows; I will gather feedback if these tools improve Quality Of Life (QOL) and its impact on productivity. I will ask testers of different skill levels - 2 Novices, 2 Intermediate and 2 Advanced Developers to test this tool.

Stages

There will be three stages of testing; the first is “Initial”, - which will cover core functionality and make sure the tools work as intended; the second testing stage will be “Beta”, - which will focus on flexibility and integration with larger projects and the last stage “Release” will be for gathering final feedback on performance, documentation and satisfaction while using the tools.

Target Demographic

As this script is both for the novice Unity Developer and the more experienced, professional developers, the Testing Plan will specify which “type” of developer gave the feedback - this will help later when re-evaluating what tasks need priority (adding more functionality or bug fixing, etc.)

User Feedback & Testing Plan

Testing Methodology

Testing will primarily be moderated face-to-face for the first stage, as I would like to gather insight into the core functionality at the beginning. Beta and Release will turn into un-moderated, online testing sessions to allow the developers to use the tools in their own time, in which they will provide feedback via questionnaires/surveys or screen recordings.

Metrics

There will be quite a few metrics that I will be tracking (but are not listed here) to ensure that the next round of testing doesn't meet these same issues:

Time Efficiency: how much time it saves the developer if they were to do it manually instead, and is worth it)

Error Handling: (track any errors the testers have faced and the tool's ability to handle them gracefully

Usability: Users can rate different areas of usability on a scale of 1 -> 10 and write a small paragraph on what they would like me to improve in the lower-scored areas.

Customizability: Users can comment on how easy it is to adapt the tool to different projects or needs.

User Feedback & Testing Plan

Questionnaire

Here are the assets I have made to complete the questionnaire, including the questions in plain text so it is easy to understand. These assets can change, or I might even use Google's Questionnaire software if this is more convenient.

Questions

These questions offer a well-rounded feedback angle. I can see what I need to improve on in the future.

Q1. How much time did you save using the tool when doing the task manually?
Q2. Would you use these tools in your workflow regularly? why/why not?
Q3. Was there any part of the tool that took longer to set up/use than expexted?
Q4. Did you encounter any bugs or issues during testing? if so, what?
Q5. If Q4 was "Yes", did the tool resolve the issue? or did it break?
Q6. On a scale of 1-10, how easy was it to learn and start using the tools?
Q7. If you gave any areas a low score, what could be improved?
Q8. Was there a tool in the toolkit you found particularly useful?
Q9. Did you feel restricted by any part of the tool?
Q10. Were the options and/or settings sufficient? if no, what should be added?
Q11. What was your most and least favourite part of this tool, and why?
Q12. Would you recommend this tool to other developers?

Unity Workflow Toolkit Survey

Tool Information

This toolkit aims to provide Unity Developers with a better and more efficient experience when making a game,this survey is to see whether these tools are in fact, doing the job and to help me understand what I need to fix and add.

Thanks for testing :)

Who are you? (tick what applies)

Skill Level

- Novice/New Developer ☐
- Intermediate Developer ☐
- Experienced Developer ☐

Dev Stage

- Initial ☐
- Beta ☐
- Release ☐

Start:

End:

Questions (Answer on Question Sheet)

Q1. How much time did you save using the tool when doing the task manually?

Q2. Would you use these tools in your workflow regularly? why/why not?

Q3. Was there any part of the tool that took longer to set up/use than expexted?

Q4. Did you encounter any bugs or issues during testing? if so, what?

Q5. If Q4 was "Yes", did the tool resolve the issue? or did it break?

Q6. On a scale of 1-10, how easy was it to learn and start using the tools?

Q7. If you gave any areas a low score, what could be improved?

Q8. Was there a tool in the toolkit you found particularly useful?

Q9. Did you feel restricted by any part of the tool?

Q10. Were the options and/or settings sufficient? if no, what should be added?

Q11. What was your most and least favourite part of this tool, and why?

Q12. Would you recommend this tool to other developers?

• UWT Survery • Copyright © 2024 • Chomp Studios •

UWT Question Sheet 1/2

Q1

Q2

Q3

Q4

Q5

Q6

• UWT Survery • Copyright © 2024 • Champ Studios •

UWT Question Sheet 2/2

Q7

Q8

Q9

Q10

Q11

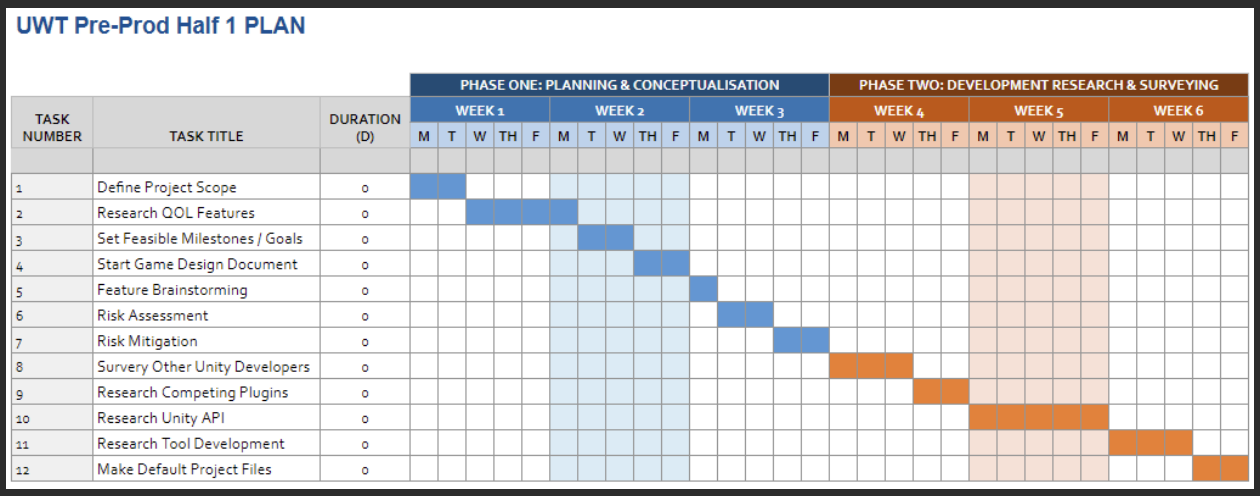
Q12

• UWT Survery • Copyright © 2024 • Champ Studios •

Production Plans

Contains Gantt charts, plans and timelines

To aid me during the production process



UWT Pre-Prod Half 1 PLAN

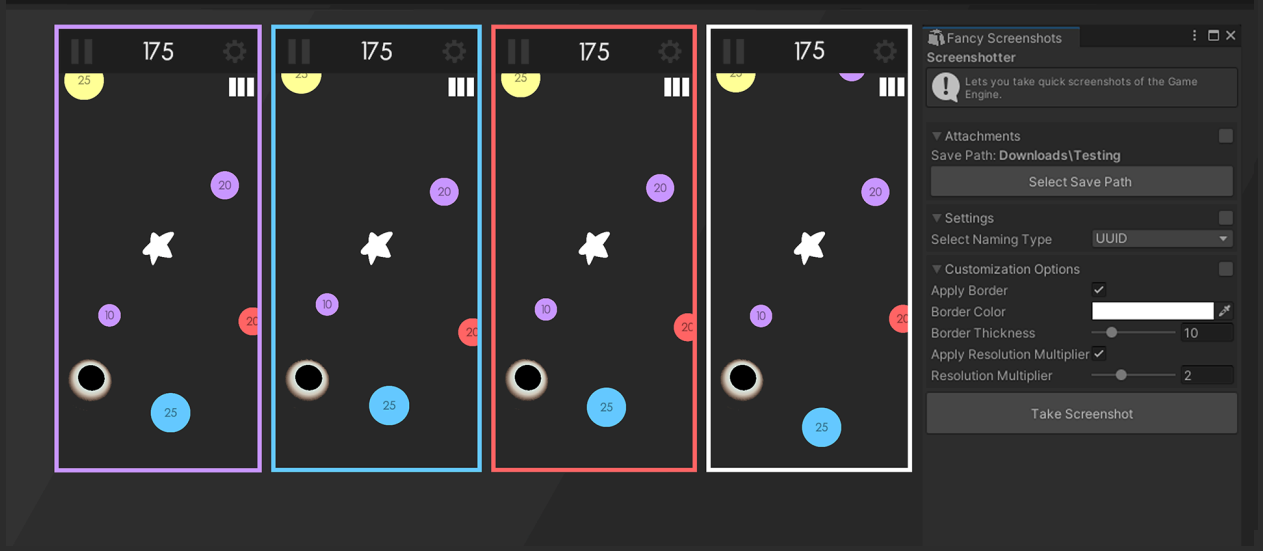
TASK NUMBER	TASK TITLE	DURATION (D)	PHASE ONE: PLANNING & CONCEPTUALISATION															PHASE TWO: DEVELOPMENT RESEARCH & SURVEYING														
			WEEK 1					WEEK 2					WEEK 3					WEEK 4					WEEK 5					WEEK 6				
			M	T	W	TH	F	M	T	W	TH	F	M	T	W	TH	F	M	T	W	TH	F	M	T	W	TH	F	M	T	W	TH	F
1	Define Project Scope	0																														
2	Research QOL Features	0																														
3	Set Feasible Milestones / Goals	0																														
4	Start Game Design Document	0																														
5	Feature Brainstorming	0																														
6	Risk Assessment	0																														
7	Risk Mitigation	0																														
8	Survery Other Unity Developers	0																														
9	Research Competing Plugins	0																														
10	Research Unity API	0																														
11	Research Tool Development	0																														
12	Make Default Project Files	0																														

Marketing Assets

Unity Asset Store

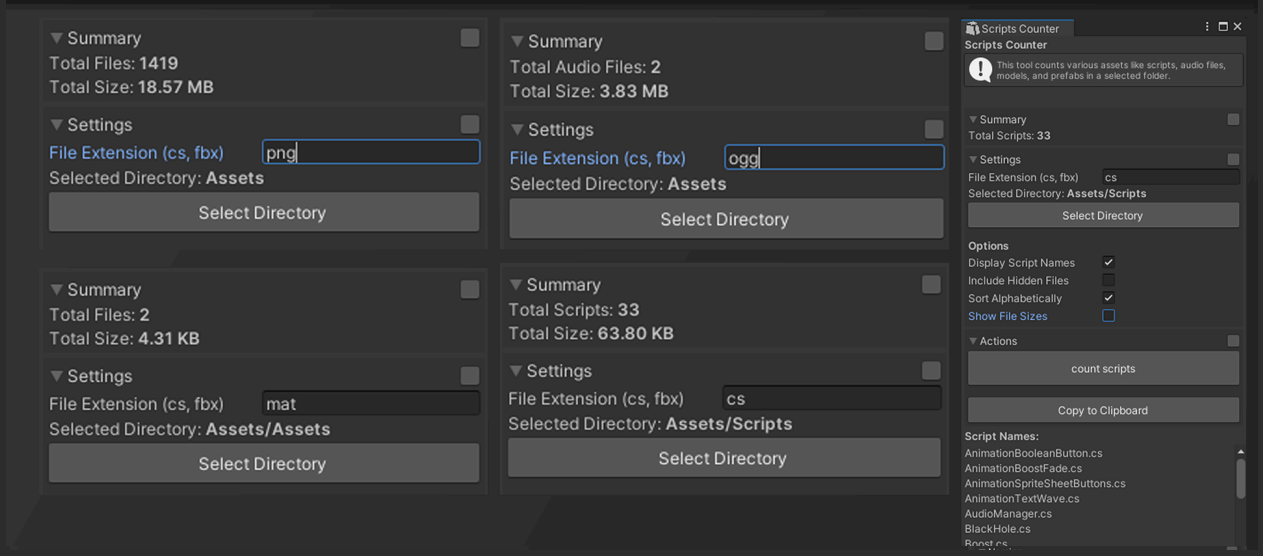
Take Screenshots

Easily take customisable screenshots inside of the editor.



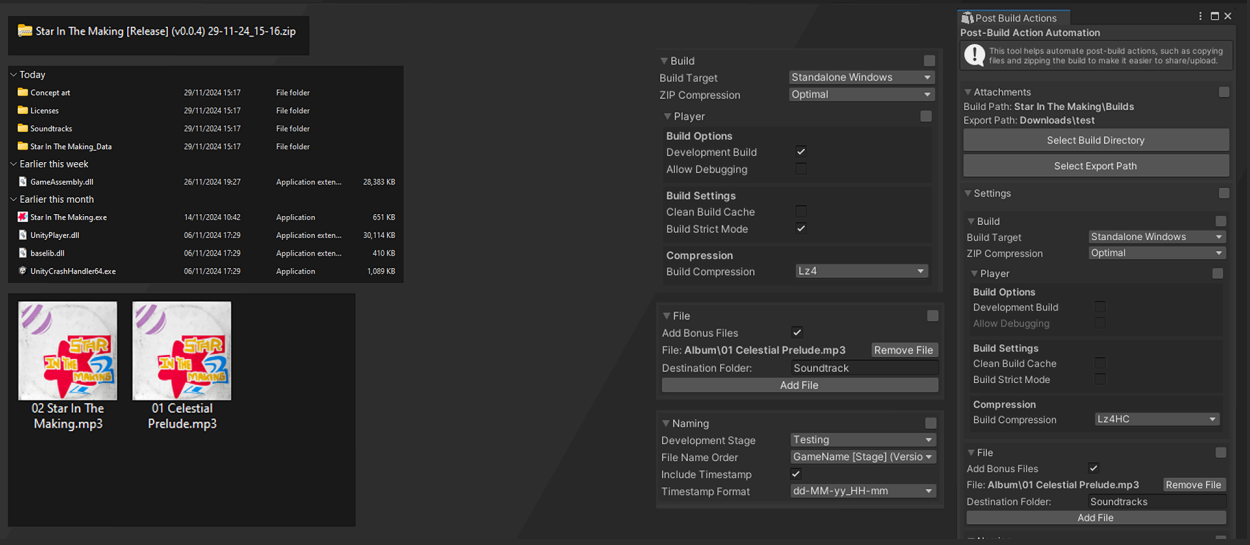
Count Assets

easily count your scripts, build files and more.



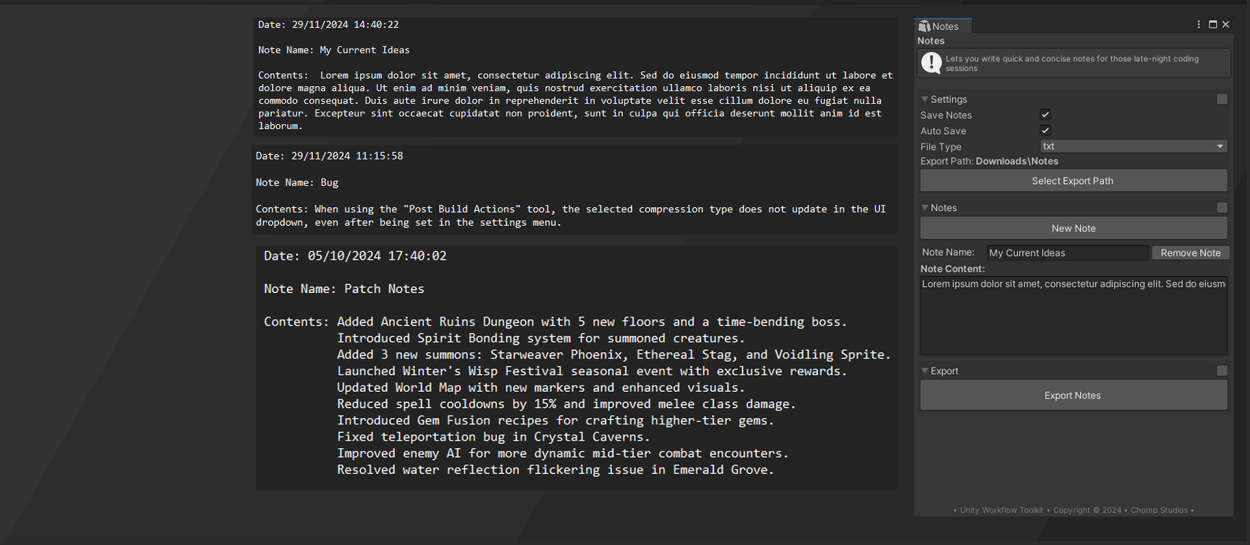
Build Games

Zip your games with your License files, Soundtracks and more.



Take Notes

Write down your ideas, bugs & more inside the editor



Risk Assessment & Mitigation

Explanation

When Planning projects, it is always good to assess risk, no matter how likely or unlikely they are. I have assessed certain risks and the probability of them affecting me if they cause me, what the impact is, and how to bounce back from them.

This is done as it is better to be ready for risk and know how to deal with them individually. I will be using a **Risk Assessment Matrix** to help out.

Risk Matrix	Trivial 1	Minor 2	Significant 3	Major 4	Severe 5
Almost Certain 5	Medium (5)	High (10)	Very High (15)	Extreme (20)	Extreme (25)
Likely 4	Medium (4)	Medium (8)	High (12)	Very High (16)	Extreme (20)
Moderate 3	Low (4)	Medium (6)	Medium (9)	High (12)	Very High (15)
Unlikely 2	Very Low (2)	Low (4)	Medium (6)	Medium (8)	High (10)
Rare 1	Very Low (1)	Very Low (2)	Low (4)	Medium (4)	Medium (5)

Risk	Impact	Likelihood	Mitigation
Scope Creep	**Significant** It could go overboard from the original plan and become hard to complete properly	Unlikely (6)	Check every few days whether I am still in the scope; if not, recall features; if too far away, start working on new features
Compatibility	**Minor** I may encounter compatibility issues with Unity versions or OSs when testing.	Rare (4)	Look up fixes, try plugins on different builds, etc.
Performance Issues	**Significant** Long load times or frame drops	Moderate (9)	Read code and look for memory leaks or inefficient code, update it, delete unnecessary method calls, etc
Technical Debt	**Significant** Accumulating technical debt due to botch fixes	Likely (12)	Consistently patch out botch fixes, document them and work on making a proper fix ASAP.
Poor Feedback	**Minor** Testers leave bad/unconstructive feedback, leaving giant bugs in	Rare (2)	Have a couple of more “Testers” in reserve, or contact friends/family for feedback

Risk	Impact	Likelihood	Mitigation
Scope Creep	Significant It could go overboard from the original plan and become hard to complete properly	Unlikely (6)	Check every few days whether I am still in the scope; if not, recall features; if too far away, start working on new features
Corrupted Hard Drives	Severe Could remove all progress due to corrupted source code	Rare (5)	Regularly back-up project source code, both on Offline (**USB**) and Online (The Cloud, **Github**)
Mental Health Problems	Severe It will be challenging to progress, work on bugs, the project, documentation, etc	Moderate (15)	Take regular mental health breaks, avoid hyper-focusing, and drink water.
Burnout	Significant Lack of the ability to work on the project	Almost Certain (15)	Take a few days off to do more research-based tasks until I can code again
Medication Shortage	Major It is not a problem on its own, but it makes it harder to progress and increases burnout & mental health problems.	Likely (16)	This is an ongoing issue, but it should be resolved soon. However, it is not something I can fix myself.

Budget

Expected: 40.00

Maximum: 100.00

Used: 22.20

I have always been accepting of buying assets for my projects; as a Programmer, I am not as creatively gifted as others, so I have set a budget for this Workflow Toolkit if I have to spend money, for example, on logos.

Asset	Reason	Date	Cost
Logo	Needed a redesign	08/10/24	£5.40
Two-year Domain Two-year SSL	Connecting UWT to the documentation website I made	11/10/24	£9.16 £7.64

Bibliography

Credits