



School of Design and Informatics

Session 2022/23

Module Code: **CMP405**

Module Title: **Tools Programming**

Module Deliverer: **Matthew Bett**

Unit of Assessment: **Unit 1 of the module assessment – 100%**

Learning outcomes assessed:

1. Assess, select and critically evaluate platforms and methodologies for tool development
2. Describe and utilize databases, 3D models and bespoke assets as part of tool development.
3. Critically evaluate and synthesize workable user interactions as part of tool development.
4. Design, develop and critically evaluate a tool application designed to operate on game pipeline data.

Submission date: **11:59 p.m. on Tuesday 16th May 2023**

Feedback Return Date: **Ten working days post submission.**

Feedback Type: **Feedback will be provided via MyLearningSpace. .**

Assessment overview

Task 1: Using the tool framework: Develop extensions to the functionality and usability of the tool.

the repository for the tool source code to build on is:

<https://github.com/savantgarde/WFFC-Edit>

As given, the sample tool is very basic: It will read information from an SQLite database and display the objects in the world as they are. While the actual data in the database reflects much more complex data than simply the rendering and basic transforms, there is no way to edit this information.

There is also a basic interactive camera with which you can manipulate your view of the world. There are a number of ways in which you can enhance this system but it must include at least **one** enhancement of usability and **one** enhancement of world editing capability as a **MINIMUM**.

Item included in tutorials that are expected as standard in the submitted coursework

- Improving the camera and navigation system through a dedicated Camera Class. ✓
- Object selection VIA mouse picking. ✓

Usability enhancement examples:

- Click and drag for manipulation of objects
- Arcball Camera motion (Orbit around object with a button held) as a separate camera mode
- Undo – redo
- Focus on selected objects (Move camera to centre on selected) ✓
- Object highlighting and Multiple object selection and de-selection VIA picking ✓
- Axis widget on object highlighting. Showing local transform.

World Editing enhancement examples:

- Terrain editing / generation / customisation
- Object manipulation ✓
- Object attribute editing (any and all object information, models, textures, data etc) Such as an object inspector
- Creation of new objects
- Copy / pasting objects ✓

- Automatic distribution of objects
- Audio previewing / assignment
- Camera paths / multiple cameras/ camera view previews
- Further Database work
 - Coming up with a solution for large amount of objects – how do we select them in a menu? Select category then submit another query for all objects in that category
 - Any other work that makes meaningful use of further Database queries and searches in appropriate fashion.

The chief limitation on this is that you CANNOT change the structure of the database. You can make new entries and manipulate them but you cannot create new tables or columns. This is a static restriction. Consider that the database is being used by a game application elsewhere and to change its structure would break compatibility.

Multiple additions or complex additions of functionality will be able to achieve a higher grade.

Replacements or augmentation of the MFC components and Rendering system will also be considered viable improvements to the system which will gain you credit. But your work must still improve the usability and the world editing capabilities.

If you have an idea and are unsure if it is appropriate or not, please discuss with me.

DEVELOPMENT IDEAS:

Included here is a list of possible development paths for functionality. This was produced in conjunction with blizzard based on features present in its WOWedit system.

Picking one of these complex systems will provide inspiration, direction and more than enough improvements for a good coursework grade. Note that some of these systems are complex and even attaining part-functionality will be a good amount of work.

Water /River Generation: A system for creating / carving out water features in the landscape. With the possibility of creating and rendering in special geometry to show the water surface. Note: That as this is not currently supported in the database system in terms of surface geometry. “saving” the water plane is not required if you were able to compete this, having an editor only preview would be sufficient.

Wall Placement / Alignment: A custom toolset for placing, aligning and joining sections of wall on the terrain. Castles and other fortifications are commonplace and having a tool to quickly fabricate this component would be very useful.

AI Pathing: AI rely on paths governed by control points which generate a curve to animate along or walk along etc. This would require the creation, editing and rendering of these curved paths in the editor. Note that Catmull-Rom splines are probably the direction you want to go with this.

Live terrain painting: Use of the mouse and an indicated brush shape on the terrain, the live painting of blends between multiple terrain textures to achieve visual interest. Terrains are commonly composed of multiple layers of textures and this would be a great way to edit and create these blends.

Live terrain Sculpting: Similar to Texture editing but instead, the live editing of terrain height using specially created tools. Allowing the artist to sculpt and shape the terrain into the desired appearance.

Object inspector / editor: A dedicated dialogue for the creation, previewing and editing of object parameters for creation or customisation. Includes a 3D preview window

Terrain information Panel: This is a minor, but interesting feature, A pop-up indicator on mouse-click on the terrain that give the precise Coordinates used at that point. This is useful for designers as it allows them to check coordinates at a point before allocating them to an object. Etc. In wow edit it also provides information about what texture layers are used at this point.

Task 2: Documentation

You must provide documentation for your application. This will be approx. 6 pages as a minimum guide. This documentation should include:

1. Summary of what features you have added to the tool. (1 paragraph)
2. Definition of controls / how the features are used (key mappings) point and click etc. This is so that I understand how to use your feature and can investigate it fully!
3. For each feature/ system you have added to the tool
 - Full explanation of what it is, does and is designed to do. With reference to why this would be useful in the context of content creation for the WOFC game. Refer to any User experience goals or design intended to be UX friendly or reference systems in other tools (unreal, Photoshop etc) that you used as a reference point.
 - Technical discussion how you implemented the feature. HOW you did it. Algorithms, coding structure and technical discussion into what you created and the techniques that you used to achieve them.
4. Conclusion: Provide reflection and critical analysis of your work. What went right what went wrong and why? What would you do differently next time?
5. References for any code or techniques incorporated in your application. Cases of plagiarism will be taken very seriously.

Things I do not need in report: Huge chunks of pasted code, there is no need. You can refer to functions classes etc but don't paste the code into your report.

Task3: Video Demonstration

To facilitate the marker process to ensure all features you have created for your tool are best demonstrated for effect, you are required to create a video using screen capture software provided in the labs or your own preferred software package. The video must be uploaded to a video sharing site of your choice (YouTube etc). Make sure the video is not set to private, but either public or unlisted. Provide a direct link to the video in the Project documentation. Also, if you can ensure that the video remains up for a year, it can serve as a useful example to the next year's students.

Submission

- The submission should include a ZIP file with the following items
 - Visual studio project (delete .sdf / .db files to reduce size)
 - A word document or PDF containing your written report and any relevant design documentation.
 - The word or PDF report must contain a link to your Video Demo.

All submissions must be uploaded to the appropriate location within the MLS system. The deadline for submissions is 11.59pm and the system is likely to be busy at that time. Please ensure that you TEST your submission before you submit it.

Grading scheme

Note that the grade is an indication of overall performance and not computed.

Literal grade	Evaluative descriptor
A+	<p>Excellent overall. Demonstrates an excellent grasp of the subject matter. Excellent standard of coding and integration into the existing framework. Excellent Improvements / additions to the tools functionality. In both usability and world editing capability. Excellent technical discussion of each new feature with detailed justification and explanation of its purpose and design. Excellent conclusion and reflection on the work. In addition, exceptional in at least one of the above.</p>
A	<p>Excellent overall. Demonstrates an excellent grasp of the subject matter. Excellent standard of coding and integration into the existing framework. Excellent Improvements / additions to the tools functionality. In both usability and world editing capability. Excellent technical discussion of each new feature with detailed justification and explanation of its purpose and design. Excellent conclusion and reflection on the work.</p>
B+	<p>Very good overall. Demonstrates a very good grasp of the subject matter. Very good standard of coding and integration into the existing framework. Very good Improvements / additions to the tools functionality. In both usability and world editing capability. Very good technical discussion of each new feature with detailed justification and explanation of its purpose and design. Very good conclusion and reflection on the work. In addition, excellent in at least one of the above but overall performance deemed to be very good.</p>
B	<p>Very good overall. Demonstrates a very good grasp of the subject matter. Very good standard of coding and integration into the existing framework. Very good Improvements / additions to the tools functionality. In both usability and world editing capability. Very good technical discussion of each new feature with detailed justification and explanation of its purpose and design. Very good conclusion and reflection on the work.</p>
C+	<p>Good overall. Demonstrates a good grasp of the subject matter. Good standard of coding and integration into the existing framework. Good Improvements / additions to the tools functionality. In both usability and world editing capability. Good technical discussion of each new feature with justification and explanation of its purpose and design. Good conclusion and reflection on the work. In addition, very good in at least one of the above but overall performance deemed to be good.</p>
C	<p>Good overall. Demonstrates a good grasp of the subject matter. Good standard of coding and integration into the existing framework. Good Improvements / additions to the tools functionality. In both usability and world editing capability. Good technical discussion of each new feature with justification and explanation of its purpose and design. Good conclusion and reflection on the work.</p>

D+	<p>Satisfactory overall.</p> <p>Demonstrates a satisfactory grasp of the subject matter but limited</p> <p>Satisfactory standard of coding and integration into the existing framework.</p> <p>Satisfactory Improvements / additions to the tools functionality. In both usability and world editing capability.</p> <p>Satisfactory technical discussion of each new feature with some justification and explanation of its purpose and design.</p> <p>Satisfactory conclusion and reflection on the work.</p>
D	<p>Adequate.</p> <p>Achievement of all threshold standards but grasp of some subject may limited.</p>
MF	<p>Marginal fail.</p> <p>Performance just below the threshold standard. A reasonable expectation that a pass is achievable by reassessment without the need to repeat the module.</p>
F	<p>Fail. Performance well below the threshold level. Some limited evidence of achievement of the outcomes.</p>
NS	<p>No assessments submitted.</p>