Tools Scripting 2018-2019

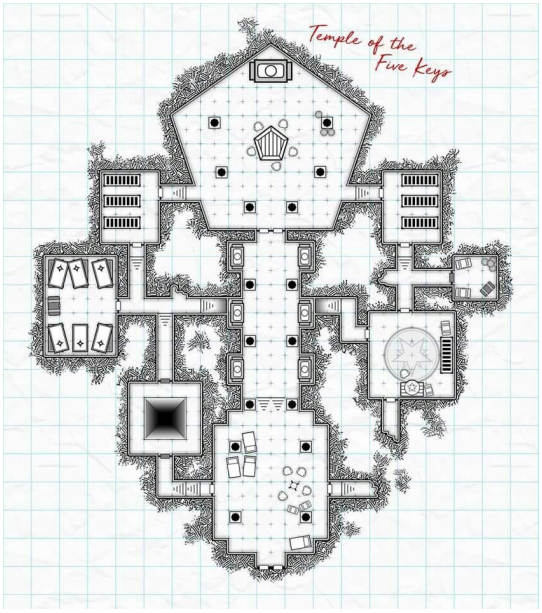
Final Work

Content: Maxscript & C++ Tools

Exercises:

1. Design your own level (Whitebox)

Design and create a single whitebox map by providing an sketch of the map and the .max scene with the content. Whitebox must be block based.



Note 1: Student can use map designs from google or create its own. Max scene must be of his own.

Note 2: Follow the principles from the given links

https://www.gamasutra.com/view/feature/131736/beginning_level_design_part_1.php

2. Export scene information into the engine using the tools we created.

The scene must be ready to be exported from max and load into the engine without any error. The scene you created must be the scene to be loaded on engine launch.

3. Create a new component (movable platform e.g).

- The student should be able to attach a component into an object from a 3dsmax scene, export this object with its component attached to it (must be linked in the json).

- The student should be able to read this component and load it into the engine. The component must appear in the UI to be edited.
- Implement add new component button in the inspector window. Atleast two type of components could be added.

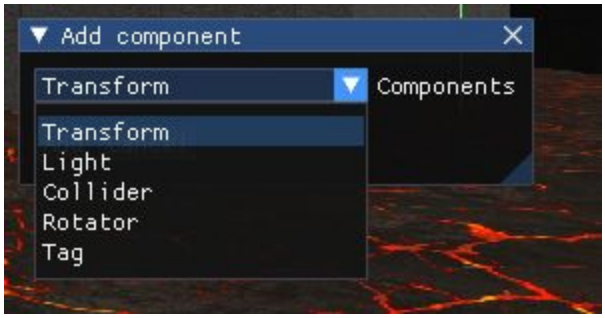


Fig 1.0: Example on add component window

Note: The process to add a component may be different from the one given on the previous image.

4. Complete the renderer debug menu (material parameters support):

Extend the material debug window on the inspector by adding more parameters on the editor. Render all the debug textures, shader type... The number of parameters and edition settings is up to the student. Use unity or unreal as inspiration.

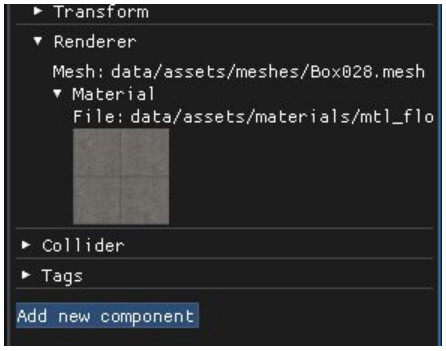


Fig 1.1: Uncomplete render debug menu

Note: Implementation on debugRender method at mesh component.

5. Add script logic on console editor:

The user should be able to execute scripts from the console by introducing commands (move player to position, spawn item..). Create atleast one new command and describe its functionality in the readme.

Fig 1.2: moveplayer command executed on the console

Note 1: Implementation on BuildCommand method at ConsoleModule class.

Note 2: Following links might give an idea of the different commands that games use in their scripting system to allow the developer and the player to interact with the engine.

https://nwnlexicon.com/index.php?title=Category:Beginning_Scripting

https://scripts.zeroy.com/

6. Finish loading json in max (scene changes between engine/max):

The student must be able to save the current engine scene to a json, as seen on class and must provide implementation on the the methods updateTransform and updateCollider at mvd_scene_load.ms file in maxscript folder in order to update the objects with new information that the json provides. (The object position was changed at the engine therefore it must be updated on max)

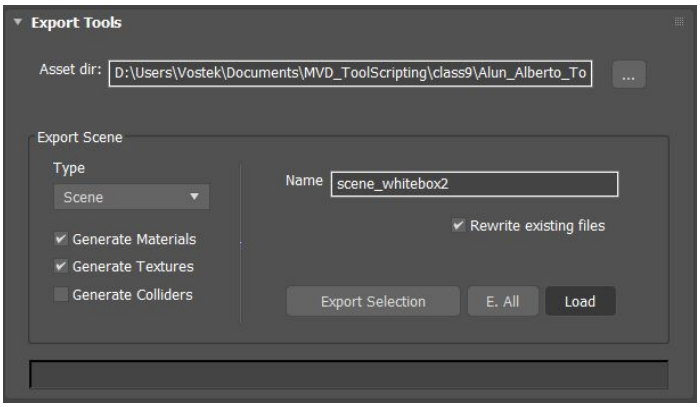


Fig 1.3: Updating scene_whitebox2 on 3dsmax

Note 1: To update an scene on max, write the scene name you want to load from scenes folder and press load button as seen on fig 1.3

Note 2: You will need to place theNewtonsoft.Json.dll provided in your 3dsmax root folder in order to be able to load jsons in maxscript, as seen on class.

7. Add a button to delete entity:

The user should be able to delete the current selected object from the scene by pressing a button on the inspector or pressing the “del” button.

Submission:

- Limit date [6/2/2019 – 23:59:59]

- The student must provide a .zip file (or github repo) with the engine and tools files required by the app. Readme and map design image must be included.

- Content must be sent to alberto@mobilemediacontent.com before limit date.

- Mail header must be [MVDTools-MyFullName1-MyFullName2]

- Work must be done by pairs of two students or individually.

- Provide a readme (.txt file) with the description of the work done.

- The engine must execute without errors.