

GRAPHIX Machine Project- No Man's Submarine (By 2-3 People)

[100pts] General Instructions:

The Machine Project involves making an ocean exploration simulator game prototype similar to the game Submarine Commander where the players pilot a submarine and duke it out with the other subs, but our prototype currently does not have any combat involved and includes a third person view of the area similar to the space section in No Man's Sky.

Submarine Commander



No Man's Sky



General Specifications

- The game should contain at least 7 different 3D Models that aren't used in class.
 - 1 Fully Textured and Normal Mapped Model for the Player's ship
 - 6 other models scattered on the map representing the enemy subs.
- Skybox representing the ocean depths.
- Ship can be controlled by the Player to explore your area.
- You can only use the libraries and APIs discussed in class.

Feature Checklist

The following is a more detailed Breakdown of the Specifications of the Project:

Feature	Score	Description
3D Models	10	<p>The game should atleast have the following different Models:</p> <ul style="list-style-type: none">• (Player Model) 1 Fully Textured Model with Normal Maps• (Enemy Subs Model) 6 Textured Models <p>*The Enemy Sub models should never overlap / collide with each other</p>
Skybox	10	<p>The game should feature a Skybox representing the ocean depths / floor</p> <ul style="list-style-type: none">• The Skybox shouldn't have been used before in class
Lighting	15	<p>The game should feature the following lighting:</p> <ul style="list-style-type: none">• Point Light that illuminates the front of the player sub• Direction Light coming from the top of the ocean down <p>*You can cycle through the intensity of the Point light using the F key (Low, Medium, High)</p>
Cameras	25	<p>The game should feature the following Camera's</p> <ul style="list-style-type: none">• 3rd Person Perspective Camera on the Player's ship<ul style="list-style-type: none">◦ The view can be controlled by using the mouse◦ You cannot see as far in this view• First Person Perspective Camera<ul style="list-style-type: none">◦ You can see much further in this view◦ Cannot be controlled by the mouse

			<ul style="list-style-type: none"> ○ You can only see objects in a single shade of color in this view (Similar to Sonar) • Orthographic Top / Birds-eye View Camera overlooking the whole area by default <ul style="list-style-type: none"> ○ You cannot move the ship in this view ○ You can pan the camera around using WASD <p>*You can swap 1st / 3rd Person Views using the number 1 key. *You can enter Top / Birds-eye view using the number 2 key.</p>
	Player Controls	10	<p>The Player ship can be controlled using WASDQE</p> <ul style="list-style-type: none"> • W/S – Forward / Back • A/D – Turn Left / Right • Q/E – Ascend / Descend <p>*Sub can only be controlled when in 1st / 3rd Person view *Sub cannot go above 0 in the Y axis *Print out the current depth the sub is in the console window using cout</p>
	OOP	10	<p>You should split the coding using atleast the following classes:</p> <ul style="list-style-type: none"> • Model Class • Light Class • Camera Class • Shader Class • Player Class <p>*The classes should properly contain the Properties / Functions belonging to it.</p>
	Documentation and Other requirements	10	<p>Your code should be fully documented and submitted along with other things to be discussed in the Submissions and Appendix.</p>
	Overall Aesthetic	10	<p>Game Prototype shouldn't look just randomly thrown together.</p>
	Total	100	
Optional Features			

	Spot Light	+3	Implement a Spot Light from the front of the ship pointing forwards instead of a point light
	Drag in Orthographic	+3	Implement Drag Controls in Orthographic view that lets you change the viewing angle
	Multiple Textures	+3	Implement a Model that uses multiple textures (Does not include the ones using Normal Maps)
	Depth Indicator	+2	Implement a current Depth indicator on the screen (Not just printed in the console)

***You can only get bonus points when your base score is at least 75 and your overall score won't increase beyond 100**

Important Dates:

DEC 02, 2022 (F)	Start of Submissions (Online)
DEC 09, 2022 (F)	End of Submissions w/o Deductions (Online)
DEC 12, 2022 (M)	Deadline of Submissions (Online)

Submissions

- Submissions should be a zip file uploaded in Google Drive
- The zip file should contain the following Files and Folders
 - o GAME – This folder should contain the exe and all the other files required for it to run
 - o FeatureChecklist.docx – Filled out Feature Checklist Document (See Appendix)
 - o README.txt – Members, Github link for the source code
 - Don't push the Debug and Release folders in Github
 - Don't push your Release Builds on Github
- The Google Drive link should be accessible to those with the link
- Submit the Google Drive link via the submissions page

Appendix – FeatureChecklist.docx

Members:

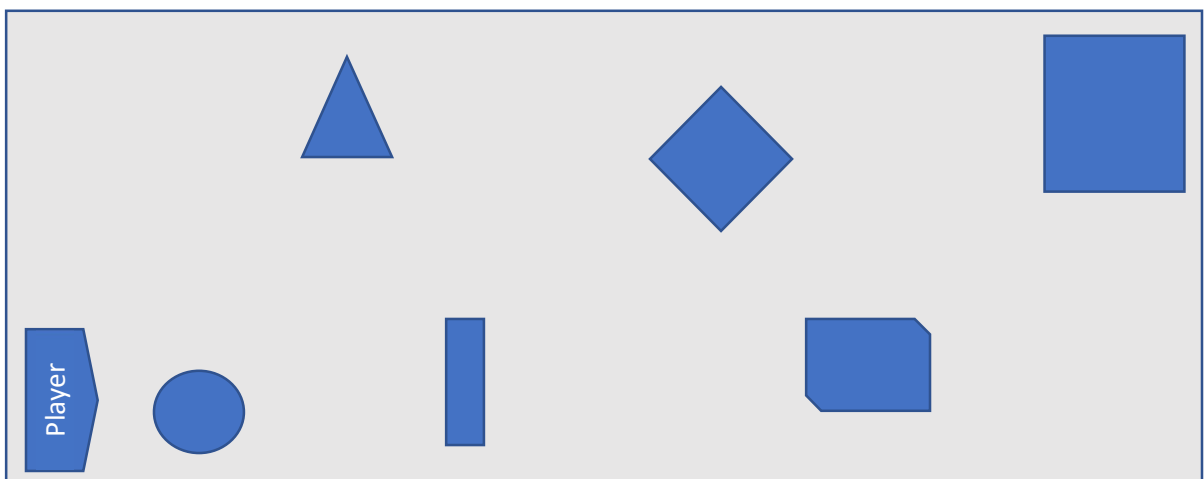
Github Link:

Implementation Summary

Feature	Summary
3D Models	Indicate in this column a short summary on how the respective features were implemented
Skybox	Remove the features that were not implemented on time or were skipped-
Lighting	You can also indicate which member implemented what feature. -
Cameras	If only partially implemented- indicate what only made it into the Game
Player Controls	Polled event every end of the while loop ... - By JR Cala (This is a sample)
...	
OOP	You don't need to include the documentation part in this table

Game Map

- Include a simple map / illustration on where you placed the models by default to help me navigate your game.
- Also indicate at what depth the subs can be found



Models

- Include Screenshots of the models as how they should appear originally
 - o 3D Model download sites often have these
- Credit the source of the models here as well

Planet Model



Barbara (Genshin Impact) : <https://genshin.hoyoverse.com/ja/news/detail/5885>

Debris 1



3D Pirate Chain Coin ARRR by Strob - <https://www.turbosquid.com/3d-models/3d-pirate-chain-coin-arr-model-1876405>