

FINAL PROJECT

Team: quackscroft

Members: Ismagil Uzenov 1873718

This project depicts a game in which the player controls a student character that is trying to finish his project for one of his classes. As with all projects, there is a deadline that needs to be met, and a certain amount of time and effort that needs to be spent to complete the project. The player is bound by his stamina and motivation parameters which determine how fast he can progress in his work. There are several interactable objects in the scene which allow the player to perform actions, as well as issues occurring over time during the completion of the project. The objective of this game is to complete the project as much as possible before the time is over.

Environment and Tools

This project was created with the use of THREE.js framework. In this project the following tools were used for loading objects and their materials into the scene:

- 1) GLTFLoader.js
- 2) OBJLoader.js
- 3) FBXLoader.js
- 4) MTLLoader.js

For the postprocessing effects and object selection the following tools were used:

- 1) CopyShader.js
- 2) EffectComposer.js
- 3) RenderPass.js
- 4) ShaderPass.js
- 5) OutlinePass.js
- 6) FXAAShader.js

Object Models

The object models used in this project were either created by the team or imported from external sources. All the models are described below.

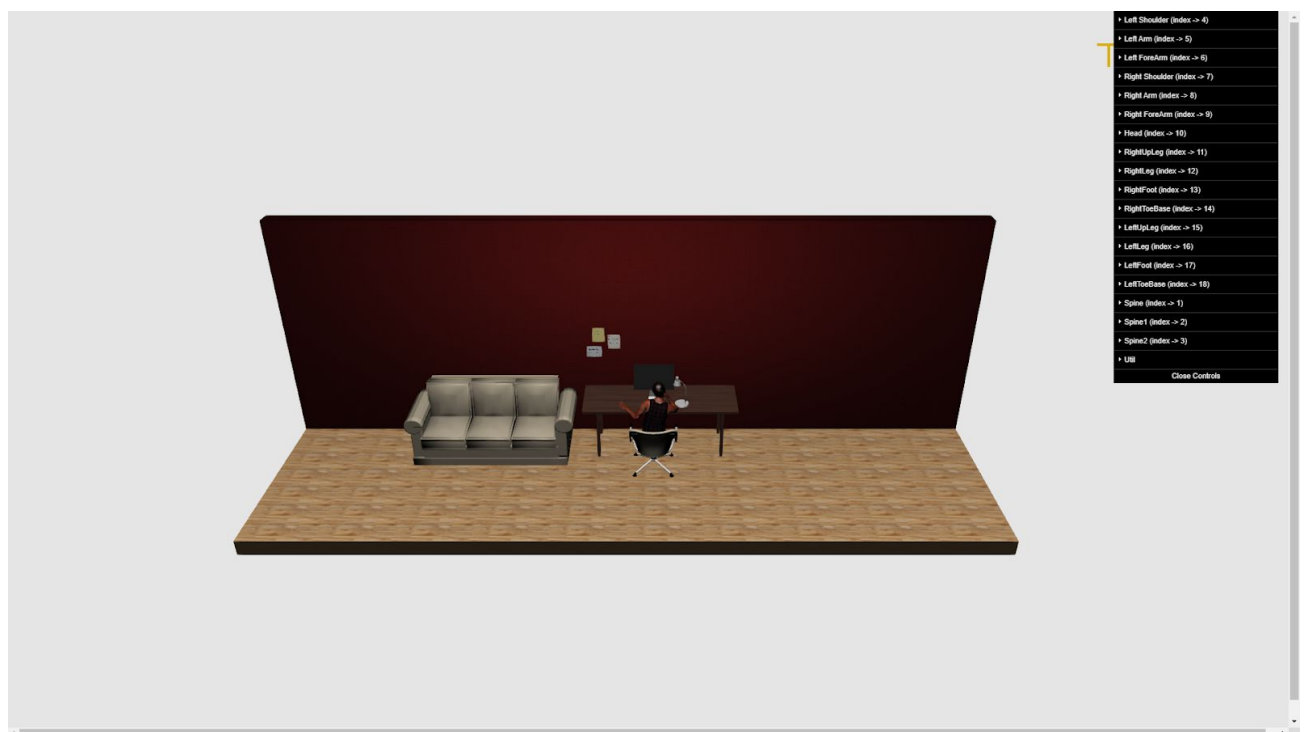
The man model, which was created using the Adobe Fuse software. This software allows us to create human characters with a high degree of freedom. From clothes to facial features, all abstract details of the model were selected by me. The created model is an .obj model which does not have any defined bone structures and cannot be used for animations. This is why this

obj model was sent to the Mixamo framework, which is a software supported by Adobe, that allows the generation of hierarchical bone structures. In the end, the hierarchical model that will be able to execute created animations was built. This model was loaded to the scene by using the FBXLoader library. The bone frames of interest were pushed to an array and their coordinates were pushed to a GUI object to be manipulated, in order to create pose frames that would be used to create animations.

Several models were imported from Clara.io, which is an online platform that allows users to create and share license free 3D models. The chair, lamp, mouse, keyboard and couch objects used in the game were downloaded from this platform. These models were uploaded to the scene by using MTLLoader and OBJLoader libraries to load the model materials and 3D objects, respectively.

The remaining objects in the scene were created manually. The wall, floor and notebook papers object were created with a simple box geometry and rescaled to suit the desired shape. After that the respective materials were assigned to each object. The table object was also created with the same method, but since it is an object with multiple parts, it was created in hierarchical structure.

The overview of the scene with the GUI object open can be seen below.



Lights

There are two light sources used in this project, both of which are `PointLight` objects of `THREE.js` library. The first one is an ambient light source that is always on, which was created to simply illuminate the scene. The second light source is the lamp bulb that can be switched on and off by mouse selection.

Animations

All the animations in the project were created manually. The `utils.js` document contains the codes of the functions used to create animations. The `animations.js` file contains the codes of the animations themselves.

The animations were created by using the 3D interpolation technique. This technique allows an object to smoothly transition from one pose to another. To this end, the GUI allowing us to manipulate the bone structure of the human model was used to obtain and save pose frames that are saved in the `frames.json` file. These frames are then used with 3D interpolation function, `lerp3D`, in the `animations.js` file. The transition from one animation to another is handled by the use of flags. Each animation has its own call flag, as well as, each transition in the animation. Once completed, each transition cancels its own flag and activates the next ones. Also, once an animation is completed, it cancels its own call too. The list of animations created and used in this project can be seen below:

- 1) sleep: Sleeping animation.
- 2) walking: Walking animation towards couch or chair.
- 3) watching: Transition from idly sitting to watching pose.
- 4) tired: The animation to indicate that stamina is low.
- 5) typing: The animation used to depict progression in the project.
- 6) toChair: Transition to sitting on the chair.
- 7) fromChair: Pushing oneself from the table and getting up.
- 8) toCouch: First sitting and then laying down on the couch.
- 9) fromCouch: First sit up and then get up from the couch.

Interactions

The interactions in the scene are varied. First of all, the menu and action selection interactions are done by HTML buttons. They are created in the `project.html` file and their style components are characterized in the `main.css` file. Their functions are defined in the `main.js` file, whether they are used to toggle animations, start the game or see the instructions.

The image below depicts an example of such objects.



The other type of interactions in the scene is the mouse selection interaction. The couch, computer screen and the lamp objects are interactable in this way. These interactions were created with the use of THREE.Raycaster library. The mouse movement and mouse click event listeners were created that are able to track the movement of the mouse on the canvas. Said objects were pushed to an array that was then used with the Raycaster to detect if there are any intersections of the mouse with any of the objects. The mouse click event detects and determines what will happen as a result of a mouse click on a selectable object. The OutlinePass library was then used in order to highlight the selected object. These interactions are:

- 1) Couch selections: The couch object consists of thirteen smaller objects. By only pushing these objects to the selectables array, we would not achieve a singular couch selection event, rather every part would be selected individually. In order to overcome that a traversal event was created to be executed every time an object belonging to the couch was selected. With this event, every object belonging to the parent would be added to the array that is used to highlight the selected object. Therefore, all the objects in the couch are highlighted simultaneously once one of them is selected. Upon selection of the couch, two button objects appear for a small amount of time. These are “SLEEP” and “LAY DOWN” buttons. Their functions will be explained later.
- 2) Computer screen selection: The screen consists of two parts and the object selection procedure is the same as explained in the couch selection. Upon selection, the screen produces two button options for a small amount of time. These options are “WORK” and “SUBMIT”.
- 3) Lamp selection: Upon selection of the lamp, the PointLight object associated with it either switches on or switches off.

Gameplay

The initial screen of the project is the main menu with three options: “START”, “INSTRUCTIONS” and “DIFFICULTY”. The “INSTRUCTIONS” button will lead to the papers hanging on the wall which contain brief information about the game and some tips. The game has two difficulty levels which can be changed by selecting the “DIFFICULTY” option. The easy level is called “HARD WORKER”, which gives the player 80 stamina and 80 motivation in the beginning of the game, as well as 360 seconds of total deadline time. The hard level is called “PROCRASTINATOR”, which gives the player 65 stamina and 65 motivation, and a total of 240 seconds of gameplay.

The game starts by clicking the start button. The player needs to explicitly select most of its actions. These actions are “WORK”, “SUBMIT”, “SLEEP” and “LAY DOWN”.

The “WORK” action is selected after clicking on the screen and prompts the working event which increases the progress of the project and the motivation of the student, but decreases the stamina. The level at which the progress is increased is determined by a formula which depends on stamina and motivation.

The “SUBMIT” action needs to be clicked explicitly even after the project is completed. This action ends the game and allows the player to collect the deserved grade. The grade, ranging from 0 to 30, is determined by the progress in the project.



The “SLEEP” action can only be activated if the stamina is below 30. Upon activation the player moves to the couch and goes to sleep for 10 seconds. This action regenerates some stamina and motivation.

The “LAY DOWN” action can be selected at any moment and it prompts the student to go to the couch and lay down. This action slowly regenerates stamina and motivation up to some point and can remain active as long as the player desires. The player can exit this state by clicking the “STOP” button which appears after the student lays down.

Apart from that, the game contains problems that occur upon progress in the project. These problems are: the release of a new season of the student’s favourite TV show, accidental deletion of a big file and a call from friends to go outside. For the TV show the player has options to binge watch the whole series which regenerates stamina and motivation, but costs a lot of time. It can watch a couple of episodes which regenerates a smaller amount of stamina, but decreases a small amount of motivation and costs some of its time. Or, the player can reject to watch which does not cost any time, but results in loss of a big amount of motivation.



The screenshot of the first problem can be seen above.

The deletion of a file, results in the loss of some progress and a big amount of motivation.

The player can either continue as it is or give up on the project which would result in getting a 0 grade for the project.

The friends' call can either be accepted or rejected. Upon acceptance, the student will move out and come back after a certain amount of time. This will result in a loss of time and stamina, but also in a surge in motivation. Upon rejection, the player loses a big amount of motivation.