**PDTG**

1 **Development Environment**

* Html,javascript

2 **Framework**

* BABYLON.JS, Version 4.2 <https://www.babylonjs.com>

3 **Installation**

* **npm i** this commands install the dependency of the project.
* **npm run start**  This command will run the project on local host.

4 **Production**

* **npm run build ,** bundle.js file will be created in /dist/ directive

5 **Project Structure**

* index.html, canvas & progress circle is created in the file.
* **Directive src/scripts/**

List of classes

1. index.js
   1. This is the entry point of the application. key function is startScene() which is called in the index.html file to start the scene
2. SceneOptimiser.js
   1. This class can be used for the optimize the scene // refer the link for more details https://doc.babylonjs.com/divingDeeper/scene/sceneOptimizer
3. SceneManager.js
   1. This class is used for the multiple scene state, we have only one scene.
4. LoaderManager.js
   1. This is class is responsible for the load all the 3d glb models, it contains the loading progress & loading finish.
   2. how to load file, this.assetsManager.addMeshTask("room\_reference","","models/","roomscene.glb");
5. gui.js
   1. This is the class for the 2d UI, load 2D image, create rectangle, circles & text are implemented in that class.
      1. Key functions
         1. initUi()
         2. initMainMenu()
         3. drawMainMenu()
         4. initStageMenu()
         5. drawStageMenu()
         6. initRadialMenu()
         7. drawRadialMenu()
         8. initObjectiveMenu()
         9. drawObjectiveMenu()
         10. initResultShowMenu()
         11. drawResultShowMenu()
6. GameManager.js
   1. This class create the BABYLON Engine instance, it is render the our scene & handle the resize of scene
7. Enviroment.js
   1. This class create the instance of BABYLON scene, BABYLON camera, BABYLON lights.
8. audioManager.js
   1. This class load the audio effects & play that effects.

6 **GameComponent classes** directive :src/scripts/Components/

1. **acremote.js** 
   * Interaction with the AC-Remote & its functionality implemented in that class
2. **alcohalwipe.js**
   * This is responsible for alcohol-wipe cleaning functionality
3. **BpMonitor.js**
   * This class is giving the BP monitor value
4. **cabinet.js**
   * Interaction with cabinet & cabinet door animations implemented in that class
   * key functions
     + openCloseDoor()
     + doorAnimation()
5. **cabinetitem.js** 
   * This class is used for the interaction with Dialysis solution & Hand Sanitizer.
   * All the functionality of the dialysis solution & Hand Sanitizer like a drag, drop, validation is implemented in that class.
6. **doorobject.js**
   * Interaction with door and animations is implemented in that class.
7. **fanswitch.js**
   * Interaction with fan witch to turn on-off fan is implemented in that class.
   * key functions
     + startFan()
     + stopFan()
8. **GameTaskManager.js**
   * This class performs the game task like
     + Room setup.
     + Item setup.
     + Self setup
     + Check the result of levels, calculate the points for the practice mode & assessment mode
   * key functions
     + completeRoomSetUp()
     + completeItemSetUp()
     + completeSelfSetUp()
     + setPracticeMode()
9. **handwash.js**
   * In that class all the 7 bubble are created with drag & drop functionality.
   * key functions
     + createhandIcon()
     + updateStack()
     + drawhandWash()
10. **item.js**
    * This is main class for the interaction with all the table items like
      + Bp Machine
      + Connection Shield
      + Alcohol Wipe
      + Mask
      + Drain Bag
      + CCPD Record Book
      + APD Machine
    * All items functionality like interaction, drag, drop, validation, inspections, and use of items are implemented in that class.
    * key Functions
      + usebpMachine()
      + useccpdRecordBook()
      + useccpdRecordBook()
      + useMask()
      + showItem()
      + resetItem()
      + updateoutLine()
11. **lightswitch.js**
    * Interaction with light switch to turn light On/Off is implemented in that class.
12. **result.js**
    * This class is create the UI of the practice & assessment mode result screen.
13. **sinkitem.js**
    * Interaction with the sink area object Paper towel & Hand Soap, all functionality of that objects are implemented in that class.
14. **table.js**
    * Interaction with the table & table drawer animation is implemented in that class.
15. **trolly.js**
    * Interaction with the APD Machine & APD Trolley is implemented in that class.
16. **windowframe.js**
    * Interaction with the window & animation of window is implemented in that class.

**7 MainRoot Class** .directive :src/scripts/scene/

* **MainScene.js**
  + This is parent class for our all objects, all the objects variable & its instance are initialize in that class.
  + Using the instance of that MainScene class we can access all other objects property & functions.
  + Key Functions
    - initScene(),(all the objects instance are created in that functions).
    - objectiveListner(), (this function gets the record of the activity of the object like a fan on/off, objects placed on the tables, AC On/Off ).
    - resetScene(),(this function resets all the objects & instances to prepare the different modes).
    - createapdmachineText(), (it creates the apd machine text over the APD machine).
    - createconnectionItemValidation(), (it creates the validation date text & correct or wrong symbol).
    - createdrainBagValidation(), (it creates the validation date text & correct or wrong symbol).
    - createApdPackageValidatiion(), (it creates the validation date text & correct or wrong symbol).
    - createBpText(), (it creates 3 BP Machine values text ).
    - setCameraTarget(), (this function set the default camera view ).
    - setCameraAnim(), (this is the function to animate camera angle to focus the object).
    - setFocusOnObject(), (this is the function to animate the camera position to focus the object).
    - startFan(), (it start the fan animation).
    - initacParticle() (this function initialize the AC particle animation).
    - setGame(), (this function is called to start the game in different mode).
    - handleUI(), (pointer events of backbutton,backmenu,submit button is handle in that function).
    - enterScene(), (this function is called after loading page to enter the room ).
    - startGame(), (this function set the training mode objectives & practice/ assessment mode).
    - checkObjectiveTraining(), (this function check the training mode objective completion).
    - createccpdCanvas(), (this function create the input fields & title for the CCPD record book).
    - updateResult(), (this function get the record for practice mode & assessment mode result based on user activity).

**8 Events Of Objects interaction**

* BABYLON.ActionManager.OnPointerOverTrigger (this event occurs when out pointer comes over the mesh).
* BABYLON.ActionManager.OnPointerOutTrigger (this event occurs when out pointer goes out from the mesh).
* BABYLON.ActionManager.OnPickTrigger (this event occurs when pointer over the mesh & user click on the mesh).

**9 Table & Cabinet Objects functions property**

* **constructor(name,root,meshobject,pos,placedpos,rotation)**
  + It is the constructor prototype with parameters
  + name, pass the name of the object in string, according to type of the object.
  + root, this is the mainscene class instance;
  + meshobject, this is the 3d mesh object which we load from loadmanager class
  + pos,position of the meshobject
  + placedpos, this is the table top place position of the object
  + rotation, this is rotation value of object when place on the top of table
* initDrag()
  + it initialize the drag event of the objects
  + DragObservableEvents
    - onDragStartObservable when drag start
    - onDragObservable when object is dragging
    - onDragEndObservable when drag end
  + initAction()
    - initialize the object interaction events like
      * OnPointerOverTrigger
      * OnPickTrigger
      * OnPointerOutTrigger