Advanced Graphics

Assignment 2 – Using Physics, Ray casting and Texture to create a simple game.

Maximum points: 20

Due: At 2:59pm Thursday March 28th 2019. (Upload your single zipped javascript «**`firstname.js`**» file to dropbox)

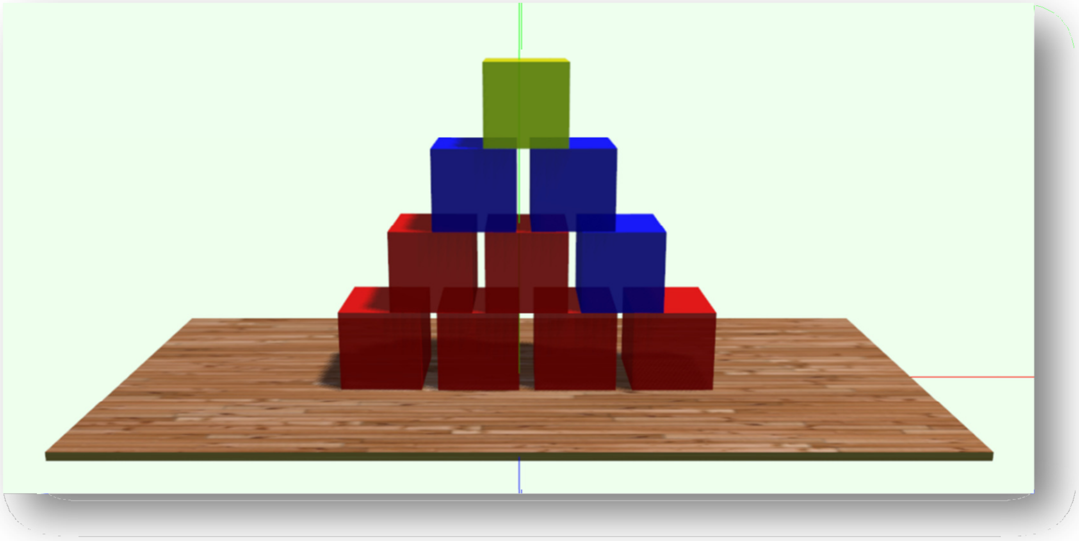
Objective for this assignment:

To prevent compatibility issues in marking and versioning, you will only use the javascript frameworks that is found in **`P:\NPERSH\COMP392\labs\lib`**

- To be able to setup threejs environment to use physijs.
- To use physiji objects
- Do all the assigned problems on your own.

The workflow for all of the labs in this will comprise of the following:

1. Create an appropriate folder structure for VS Code.
2. Add the necessary javascript libraries to the html page
3. Code the required javascript statements to complete the lab is a separate javascript file



`https://en.wikipedia.org/wiki/Glass_Tower_(video_game)`

Tasks:

Build a threejs game having:
3-D rectangular block
stacked on a table
when a block is clicked it is removed from the game
provide an interface to change aspects of the game.
A game (position, size and color of each block) is described by an external json file.
Although a orbit controller is not necessary for the game playing, you might find it useful during application development.

One mark for coding style

One mar for aesthetics

Three marks for difficulty level

½ Mark A scene with different sizes and colors of blocks on a surface. (Red, blue and yellow)

2 Marks A scene where a single block that disappears when clicked.

3 Marks A scene of stacked blocks on a surface, where the higher blocks falls when the lower ones are clicked. You will have to remove the blocks that fall off the table.

5 Marks Block configurations (which is actually a game) must be read from an external json file (minimum five files/games). These files should be read from the folder **`../assets/games`**. Use the convention «**`firstname[1..5].json`**» for your block configurations.

1½ Mark Record and display score as the game is played. You decide how points are awarded

1½ Mark Detect end of game (either a win or loss). You decide what condition will result in a win or a loss.

½ Mark Start a new game without restarting the application.

1 Mark Your own innovation such as changing the allotted time, gravity, friction or restitution while the game is being played.

Hints:

1. Put all the blocks in a collection such as an array, so it is easier to process them.
2. When starting a new game, you will have to remove all of the existing blocks.
3. When reading your game, make sure that your json file is in proper format.

Appendices:

```
function readFile(port, filename) {

    let url = 'http://localhost:' +
        port +                          //port number from data.gui
        '/assets/games/' +              //url path
        filename +                      //file name from dat.gui
        '.json';                        //extension
    //console.log(url);                 //debugging code
    let request = new XMLHttpRequest();
    request.open('GET', url);
    request.responseType = 'json';      //try text if this doesn’t work
    request.send();
    request.onload = () => {
        let data = request.responseText;
        //console.log(data);            //debugging code
        createGame(data);
        //createGame(JSON.parse(data)); //convert text to json
    }
}
```