

Faculty of Arts and Sciences

Department of Computer Science

CMPS 299 – Software Engineering

Spring 2016

Mentor: Dr. Ahmad Dhaini

Software Project Technical Documentation

|  |  |
| --- | --- |
| Class Section | 4 |
| Project Name | AUB Kiosk |
| Team Name | AUB Kiosk Team |
| Team Members | Sarah Fakhoury - Samer AL Masri - Ines Badji |

Table of Contents

1. Code Description
2. Used Modules
3. Scenes Description
4. Database Description
5. UI Description

1. Codes Description

We created various c# scripts (found in the assets folder) in order to implement the functionalities we have; below are the scripts that we created:

1. swipe.cs: used for the slide animation when navigating between the map and off campus menu.
2. Items.cs: a structure that keeps information about each restaurant or places to discover and links it to a click instance.
3. DynamicAdding.cs: used to add dynamically all the eating places or cafe as well as places to discover read from the database, this code uses an array of Items that is then shown on the screen.
4. cam\_move: used to move the camera using the coordinates found in the table BuildingsCoordinates, to be able to move the camera to the spot with the best view of the building (front door of the building for example) when the user searches for that specific building.
5. camera.cs: is the one responsible for the movement of the camera on touch (one finger to pan, pinch for zoom and three fingers to rotate) as well as the scaling of pins (when the camera moves or zooms). It also limits the camera in the vicinity of the AUB campus (for example, not allowing the camera to go below the terrain nor above the sky). Finally, it doesn’t allow the camera from getting inside objects (such as buildings) by 2 means:
   1. By adding a rigidbody and collider components to the camera, a function is triggered when a collision is detected forcing the camera to backtrack the last move.
   2. By constantly creating raycasts towards the ground and making sure the camera is never very close to the ground.
6. controlpanel.cs: controls the appearance of the panel and searches game objects.
7. course.cs: a structure created to hold information about a course.
8. eatOffCampus.cs: responsible of the button controls (open/close) in the map scene.
9. GetClasses.cs: searches through the database for the empty classes depending on the input given by the user. It stores the information in a list.
10. keyborad.cs: activates the keyboard when needed (for the searched on map scene and studyplaces scene)
11. LinkData.cs: is linked to the input search on the map scene and is responsible of searching through the database for the information requested by the user depending on the input as well as the chosen item in the drop down menu. This code is as well responsable of printing the result in a text box on the app scene and stops the camera from moving when the result panel is activated.
12. movpos.cs: is responsible for the movement to an object on click.
13. pins.cs: makes the selected pins appear/disappear depending on the click of the user. It gets the pins position from the PinsCoordinates table in the database.
14. trycams.cs: is responsible for showing the name of the building when clicked.
15. Reload.cs: is responsible for reloading after 10 seconds if no movement is detected.
16. Search.cs: is used in StudyPlaces scene, it is responsable of getting the input from the user and printing the information on the panel after retrieving the results from GetClasses.cs code.
17. switchFromMain.cs: is used to switch from a scene to another by clicking on the button attached to one of the methods.
18. TouchCamera.cs: detects a double tap on an object and moves the camera to that destination.
19. Path.cs: shows the path from bliss to any other building/area listed in the drop down menu.
20. Loading.cs: used in Loading scene in order to change the frames of a loading GIF file to an animation.
21. Info.cs: is responsible for the informations appearing on the screen when the “i” button on the bottom right is clicked.

2. Used Modules

Multiple modules were used throughout the project:

1. The NavigationMesh and ActorMesh included in unity to create the different paths and roads and to be able to render them.
2. The ocean and sky were imported from the Asset Store.
3. Tree packages found on the asset store as well as multiple other free online websites. We used the unity tree shaping module to be able to get the different needed shapes.
4. Terrain is a Unity module that facilitate the creation of hills and roads.
5. Event system: Used in order to trigger scripts based on button clicks and other input events.

3. Scene description

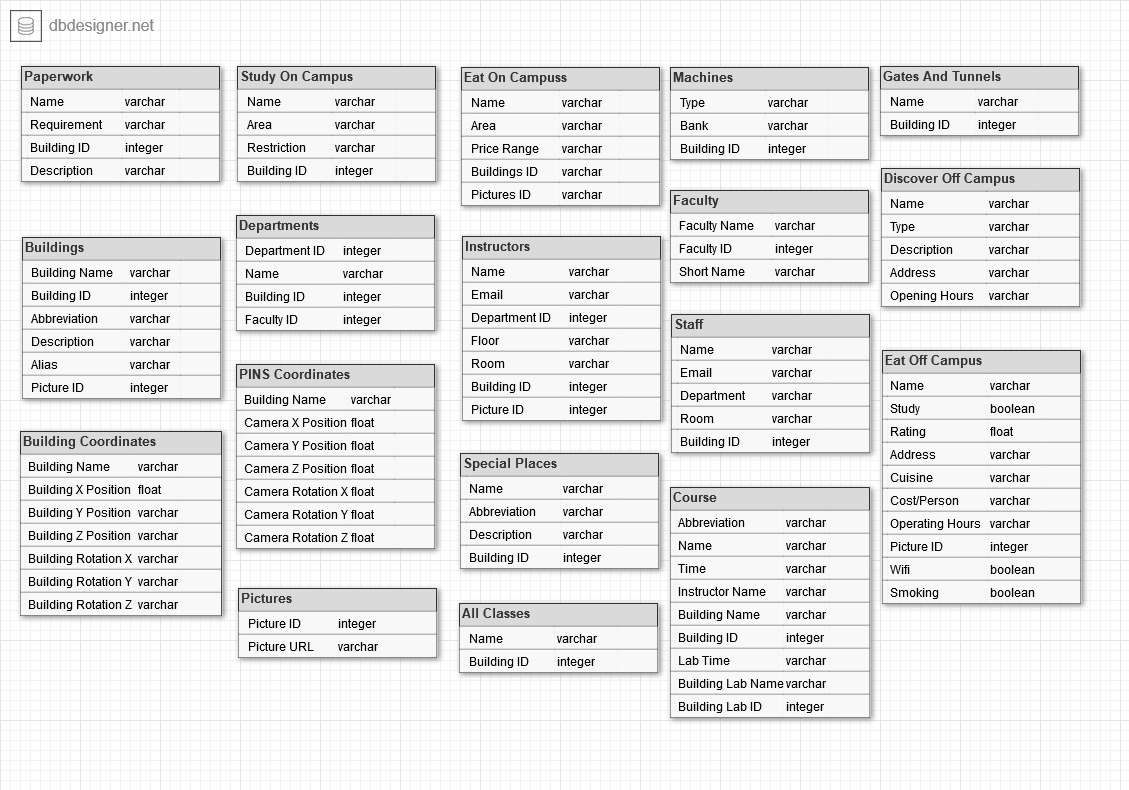
Our application contains 5 scenes (found in the assets folder):

1. Main Scene.unity: this is the main scene from which all the map and all map-related features are accessible.
2. Off Campus.unity: this scene is created for the user to choose between the 2 off campus features we currently have: viewing places to eat off campus, or discovering new places in the AUB neighborhood.
3. Eat.unity: allows the user to view and search through restaurants we have in our database. Also includes a search feature to filter the results accordingly.
4. Study Places.unity: allows the user to search for free classes to study in at any time between 8AM and 6PM.
5. Loading.unity: When coming back from the off campus mode to on campus, Main Scene takes a significant amount of time to load (mainly due to the map), hence we added a load scene in between which shows a loading animation while the user is waiting.

4. Database Description

The database was created and accessed using MySQL code, it was built to facilitate the retrieval of the information based on the planned UI.

Database Schema:

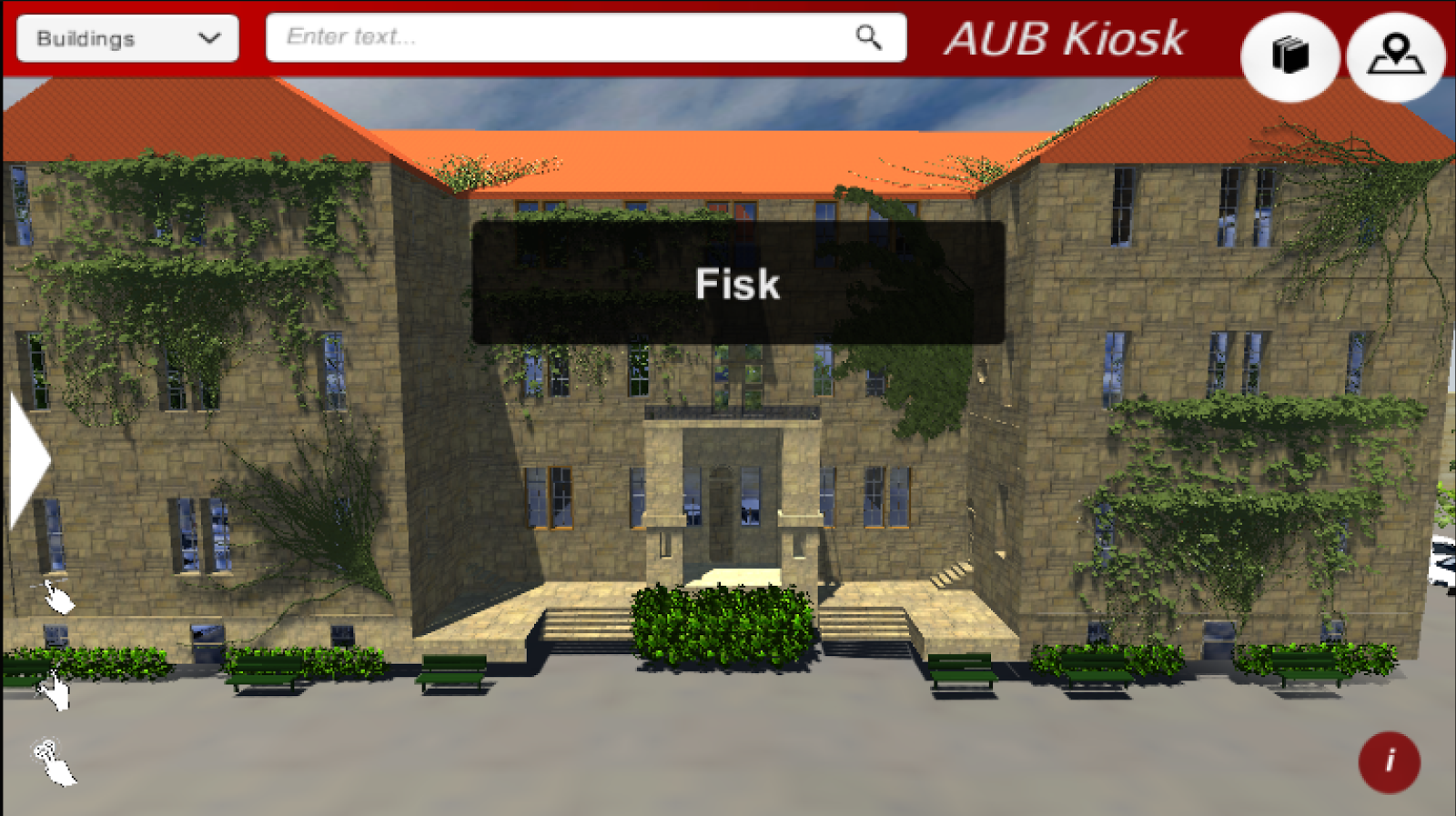


Due to our database being in developmental phase, and taking into consideration the simplicity of our database and the small amount of data that it holds relative to the computing powers today, we found out that the cons of adding foreign keys outweigh the pros (for instance the dependency it imposes when deleting records or updating tables). Therefore, we do not have any relationships in our database.

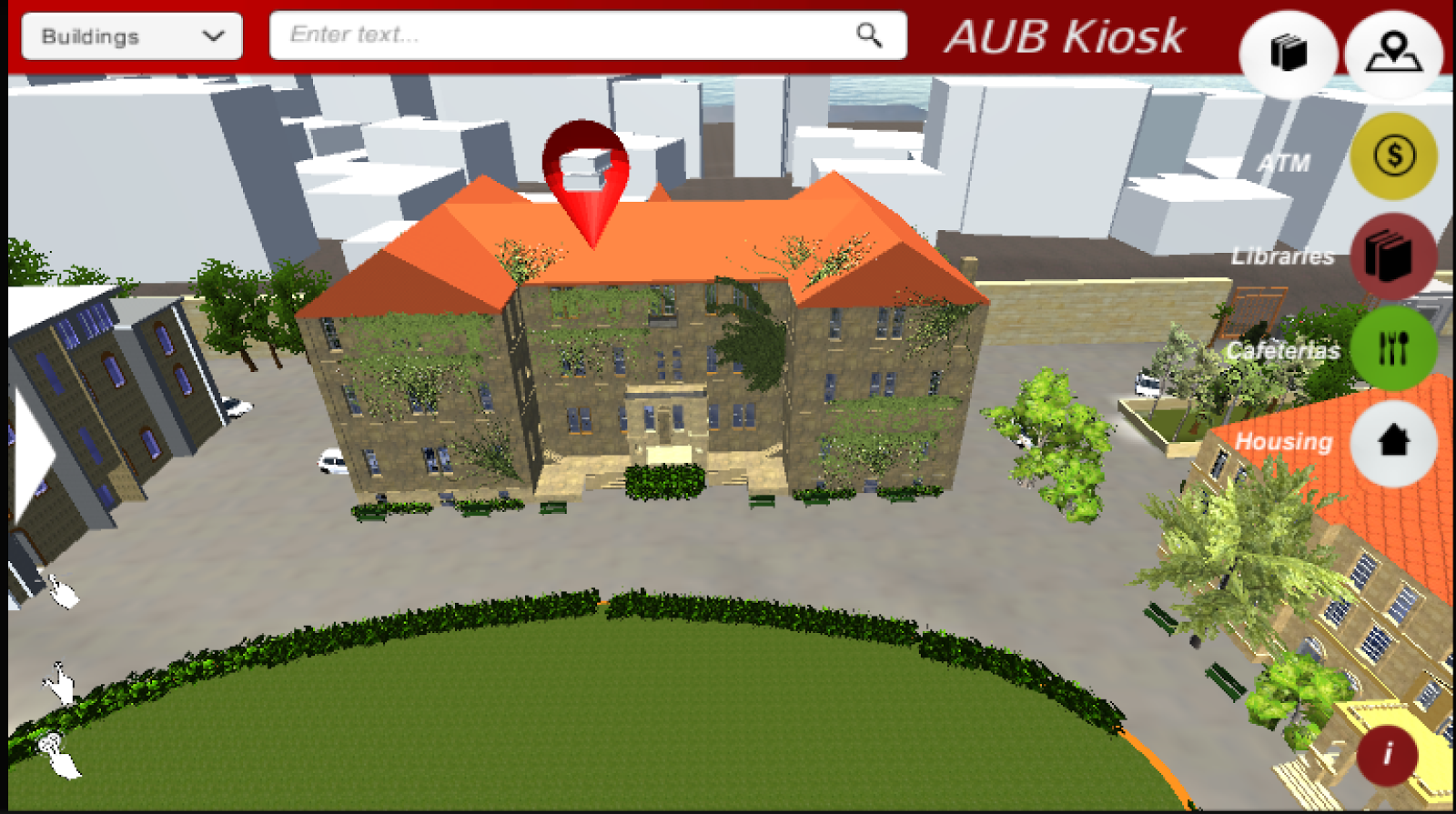
5. UI Description:

The UI was designed and built by the AUB kiosk team. The buttons and input fields are attached to specific scripts in order to operate correctly; below is a mapping of each button to what script(s) it is attached to:

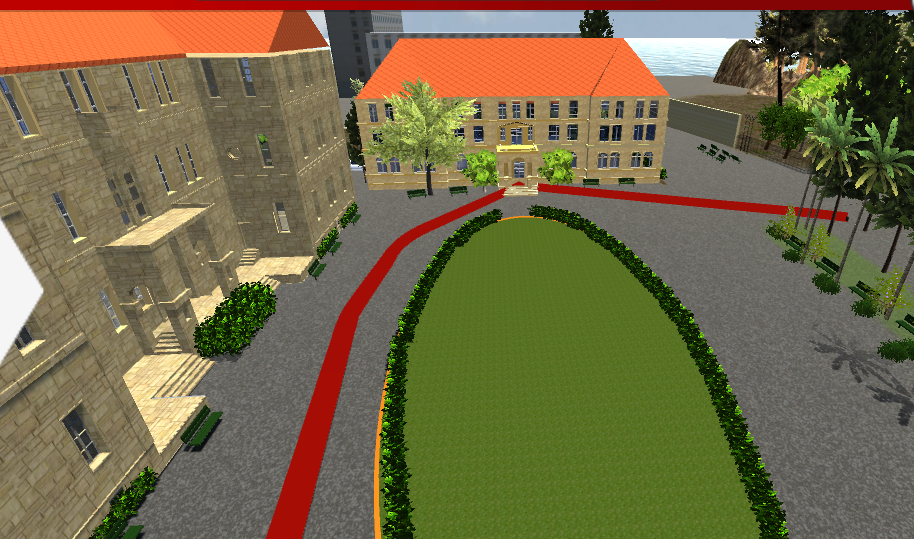
1. The main scene:



1. The middle search bar uses linkData.cs in order to search dynamically through the database and return the results in a new panel.
2. The drop-down menu for customized search (with Buildings as the default value) use searchoptionshide.cs to get the show the hidden menu.
3. The book icon directs the user to the study places scene using the switchFromMain.cs
4. The info bottom left button is linked to the eatOffCampus.cs;
5. The white arrow on the side uses the switchFromMain.cs to slide the canvas to the off campus mode.
6. The canvas is linked to multiple scripts allowing them to run anywhere in the program:
   1. cam\_move.cs
   2. reload.cs

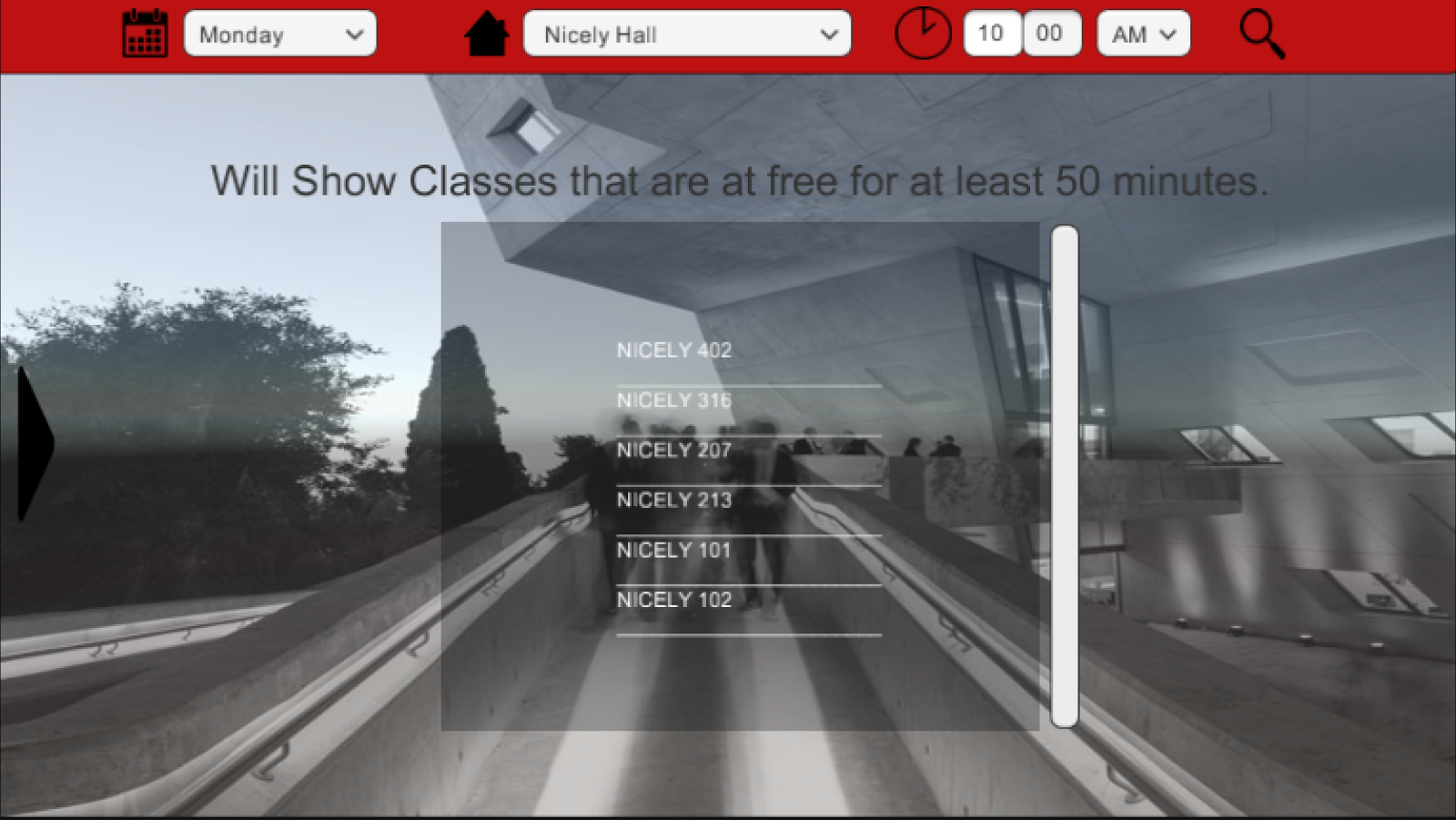


1. The pins button on the upper right corner shows the drop down menu using pins.cs. When the pins button is clicked, it uses eatOffCampus.open() to open a hidden dropdown menu; Each of the categories buttons (ATM, Libraries, Cafeterias, and Housing) uses pins.<Name of Button>() to instantiate corresponding pins.



1. The path button uses the path.cs code to show the options, the search button as well as creating the path when the choice is made

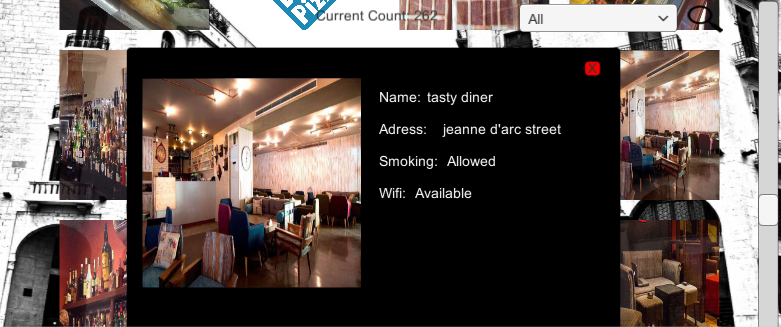
2. The study places scene:



Using search.cs, course.cs and getcourse.cs, the search button in this scene shows the free classes based on the values specified by the user on the left drop down menus (day of the week, time, and building).

3. Off campus scenes:





In both pictures the Dynamicadding and Item.cs is used to search and show the results