OLM VR

NITZAN WINKLER

BAR NEWMAN

JONATHAN WEIZMAN

TABLE OF CONTENTS

INTRODUCTION

We developed a VR Platform for a research following a demand made by a researcher in psychology from Emek Izrael Institution.

The experiment focused on the subject's ability to memorize the objects in the environment and identify changes in the positions. The environment simulates a Savana with bushes meant to be memorized and foxes as a distraction.

To make the subject more comfortable inside the VR environment we created an experimental environment of a modern office. The movement in this environment is the same as in the main experiment so the subject won't get distracted by the feeling inside the VR.

Experiment Description: the subject will have one minute to memorize an environment of 24 objects (bushes), afterwards he will be shown the same environment with some changes in the positions of the bushes. The subject need to identify all the changes. In one version of the experiment there will be distractions (foxes), the subject will have two tasks, the first to memorize the bushes and the second to identify the distractions by clicking when they appear. The experiment's results will show the timestamps he clicked and will show if he missed the fox or hit it.

The Experiment will be divided into 2 different kinds of movements (1 minute each):

1. Guided tour experiment – the subject will stay in his place while the environment moves around him in a path (feels like a car movement).
2. Free movement – the subject will be able to move around (slightly, cable length and room limitation) and will be able to explore the environment only from his spot.

SYSTEM

We created a platform that runs on computer and played with VR gear, targeted specifically for the HTC Vive.

The application was developed using Unity 2018.2.1 (64-bit) game engine.

Equipment required

1. HTC-Vive Headset
2. HTC-Vive controller

For a more comfortable usage of the platform we created an application in windows forms (C#).

Some relevant links:

https://unity.com/

SCENES OVERVIEW

The demand for the experiment included 2 different environments and 2 different movement options.

1. Modern Environment (office) –
   #scene1: Free movement in the middle of the office.
   #scene2: Guided movement around the office.
2. Experiment Scene (Savanna) –
   #scene1: Free movement in the savanna valley **with no** distractions.
   #scene2: Free movement in the savanna valley **with** distractions.
   #scene3: Guided movement in the savanna valley **with no** distractions.
   #scene4: Guided movement in the savanna valley **with** distractions.
3. Summery scene (Savanna) –
   #scene1: Free movement.
   #scene2: Guided movement.

MODERN ENVIRONMENT

The modern environment scene allows the subject to explore the VR environment. Most of the subjects didn't use VR before and can be enthusiastic and it can affect their results in the experiment.

Each subject will be assigned a movement method (Free or Guided) and will experience this movement in this stage.

THE EXPERIMENT SCENE

For this scene we chose a savanna environment with pre-defined 24 similar bushes.

Each subject will be assigned a movement method (Free or Guided) and will experience this movement in this stage.

Part of the subjects will have distractions – a fox will run across in front of the subject and will distract them from doing their memorizing mission.

When the subject sees a fox he use the vive controller in order to mark that he saw the fox (by clicking the vive trigger), if you didn't click while the fox is visible it will be counted as a "miss".

SUMMERY SCENE

The scene's environment is the same as in the Experiment scene with the same bushes but in different locations. The subject needs to mark the bushes he think changed their location. He will have a constant ray out of the controller. The ray will be red when the object is not clickable, blue when you point to a bush that is not selected yet and green when you already selected it.
The movement in these scenes is the same as chosen in the previous scene (free or guided).

APPLICATION OVERVIEW

In order to simplify the usage of the platform we created a wrapping application.

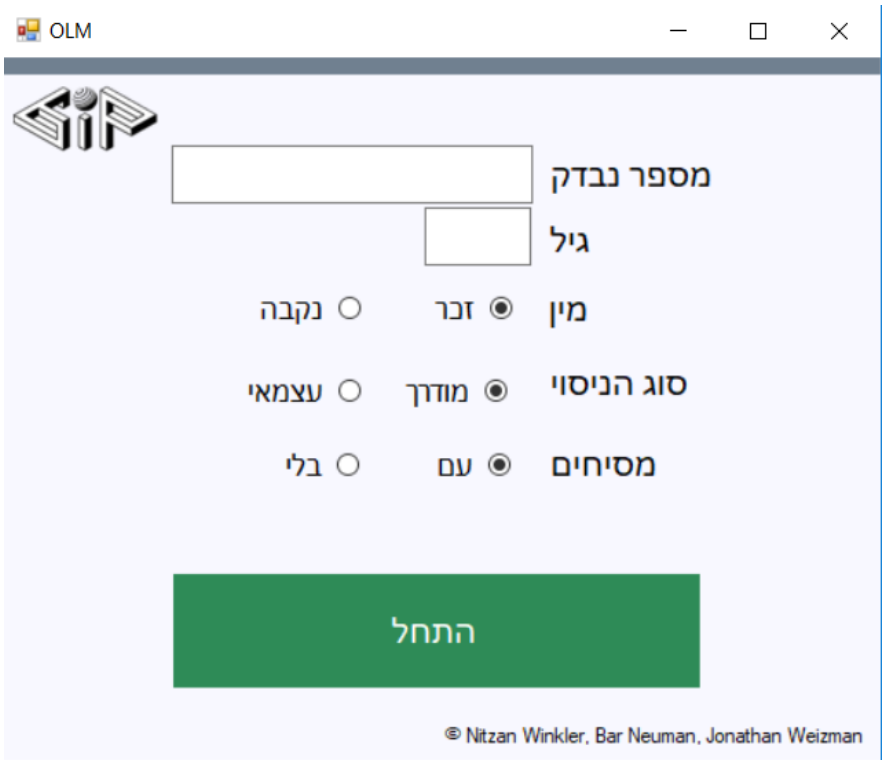
The application includes all the data we want to save on a subject including the movement method and experiment's settings (distractions). The data will be saved in folders. The hierarchy of the folders is explained in the application activation section.

Application parts:

1. Home page
2. Experiment flow

HOME PAGE

The home page includes the subject's info form: id, age and gender as well as the experiment's settings including movement method and distractions settings. After the information is entered, a new folder will be created for the subject and the experiment may begin.



EXPERIMENT FLOW

The experiment flow includes a button representing each of the 3 stages (introduction, experiment and summery).

When clicking one of the buttons the relevant scene will be launched and a timer will appear. If the stage had an output, it will be saved in the folder and you can also open it and view it through the application.

When done with the experiment you can close the experiment flow window in order to start a new experiment with a new subject.

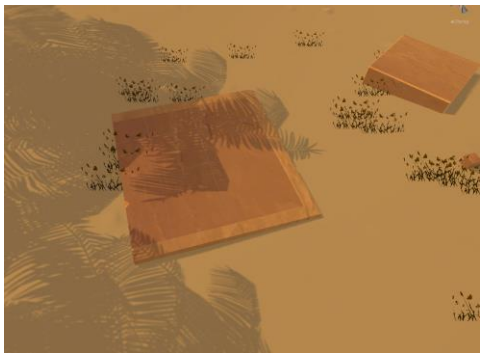
DEVELOPMENT PROCESS

To accomplish our final application, we used unity 2018.1.2. We searched online for ways to make unity work with our equipment and adjust it to our goals. We found VRTK- virtual reality tool kit, a productive VR Toolkit for rapidly building VR solutions in Unity3d.
Our first milestone was to create 6 different movement types to show the clients so they can select the one most suitable for their needs in the experiment.

We created a demo environment in order to demonstrate all the movement types and give the client the ability to explore them for themselves.

TYPES OF MOVEMENTS:

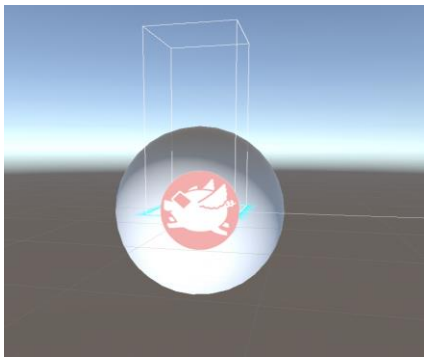
1. Free movement **#selected for the independent mode**



2. Touchpad – movement control with the touchpad

3. Move in place – creative movement method by swinging the controllers up and down

4. Straight movement – guided tour with no rotation of the head
5. Rotation movement – guided tour with rotation of the head
   #selected for the guided tour
6. Capture movement – guided tour but with pre-recorded 360° environment



After the first milestone we chose assets for the different environments and an asset for the distraction.

We started building the final environments (savanna and office) and started working on the wrapping application and the logic behind the actions in the scene.

At this stage the development was coordinated directly with the client with small milestones so the product will achieve all their goals and even more.

The work on the project at this stage was done on four different stages simultaneously.

OFFICE ENVIRONMENT

The first mission was to build the office environment according to the demands of the experiment. We build the office with a lot of objects in it to give it a good feel. When we tried to implement the guided tour in the crowded environment we needed to improve the algorithm of the movement to make it more precise and easier to manipulate on turns, we added to the script the option to determine the rotation speed, tick's counter, forward speed of each section, to make the feeling of the tour more pleasant.

```
void FixedUpdate()
{
    if (i < vec.Length){
        if (vec[i].y > 0){
            rig.position += transform.forward * Time.deltaTime * vec[i].z;
            x += Time.deltaTime * initRotationSpeed;
            transform.rotation = Quaternion.Euler(0, x, 0);
            if (vec[i].y > temp/2)
            {
                initRotationSpeed += vec[i].x;
            }else
            {
                initRotationSpeed -= vec[i].x;
            }
            vec[i].y--;
        }
        else
        {
            initRotationSpeed = 1;
            ++i;
            if (i < vec.Length)
            {
                temp = vec[i].y;
            }
        }
    }
}
```

After finishing the environment and compiling it we realized that the objects in the scene were vibrating when we moved around the environment. We discovered that the problem was caused by too many objects in the scene so we needed to clean some of the objects.

MAIN ENVEIROMENT

This environment needed to be smaller and with just a few objects so the focus of the subject will be on the bushes. Because of the strict limitation it was hard to build an environment that will be good for the free movement and the guided tour. For the free movement we needed all the bushes to be close to the center and for the guided we needed them to spread out so we will be able to do the tour around them without to many Sharpe angels. The buses needed to be compatible for the end scene where the subject will need to click them.
After we finished building the environment we needed to start working on the bushes logic (how the subject will choose what he thinks changed), for this part we decided to use a constant ray out of the controller that will turn red when the object is not clickable, blue when its pointed at a bush and green when the bush is already selected. The pointer we used is based on VRTK pointer with a little changes to add the 3rd color.

DISTRACTIONS

There are two types of distraction, one for the guided tour and one for the free movement. The logic for the foxes couldn't be the same because on the guided tour the movement is always the same therefore, we could place and time the foxes hard coded. In the free movement we decided to locate number of foxes in the environment, we created a logic which coordinate the appearances of the fox with the field of view of the subject.
isVisable() – each fox updates the fox manager when the subjects can see it.

```
void Update()
{
    if(fox.activeSelf == false)
    {
        fox.transform.position = origPosition;
        fox.GetComponent<AnimalAIControl>().SetTarget(waypoint.transform);
    }
    Vector3 viewPos = cam.WorldToViewportPoint(fox.transform.position);
    if (viewPos.x >= 0 && viewPos.x <= 0.8 && viewPos.y >= 0 && viewPos.y <= 1 && viewPos.z > 0)
    {
        if(isLastFox == false || VisableFox.instance.foxNumber == -1)
        {
            isLastFox = true;
            VisableFox.instance.foxNumber = foxID;
        }


    }
    else
    {
        if(isLastFox == true)
        {
            isLastFox = false;
            VisableFox.instance.foxNumber = -1;
        }
    }


}
```

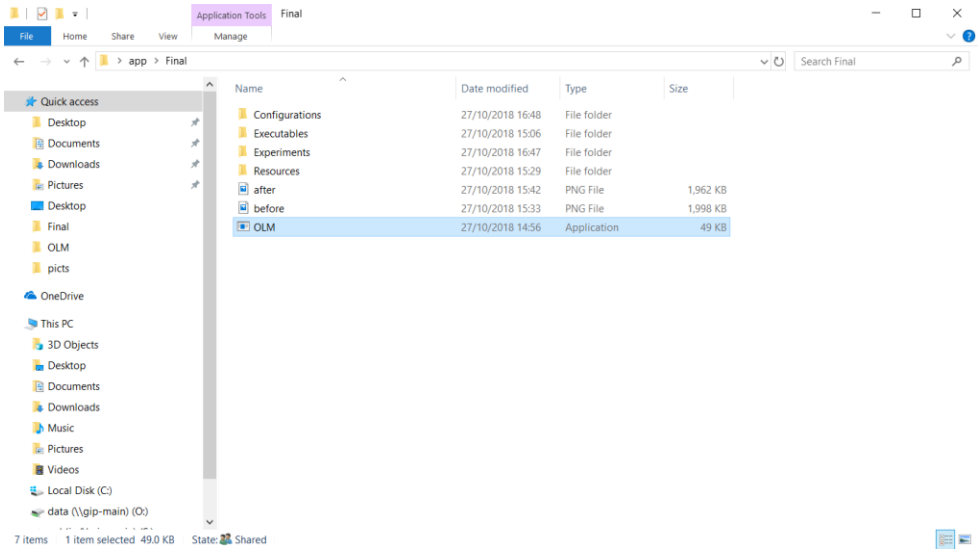
FoxController – run the last visable fox in pre-defined times.

```
public class FoxController : MonoBehaviour {

    public GameObject[] foxes;
    public int[] times;
    private int i = 0;
    private int tmp = 0;

    // Use this for initialization
    void Start () {
    }

    // Update is called once per frame
    void Update () {

        int timer = (int)Time.time;

        if (i<times.Length && VisableFox.instance.foxNumber!=-1 && timer!=0 && (timer % times[i]) == 0)
        {
            VisableFox.instance.foxInstance = i;
            foxes[VisableFox.instance.foxNumber].SetActive(true);
            VisableFox.instance.isFoxRunning = true;
            i++;
        }

    }

    void SendMessage(string s)
    {
        VisableFox.instance.isFoxRunning = false;
    }
}
```

APPLICATION USAGE

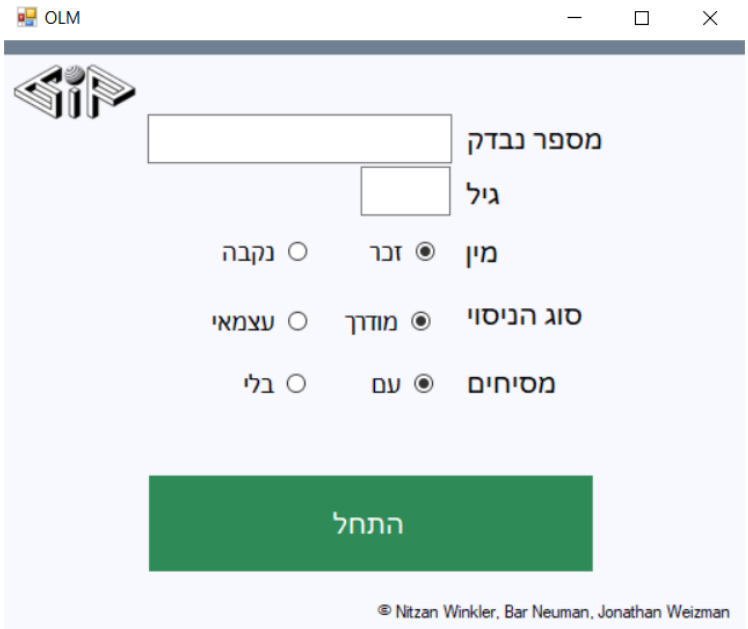
On this section we will step-by-step guide you to activate the OLM application and use its components.

1. **Launch** the OLM application by clicking the 'OLM' exe file.



The screen you will see now is a form for a new subject for the experiment. The details you fill there are saved and will be shown in the output files.

2. **Enter** the details about the subject and **choose** the experiment relevant mode.



After filling the info press on the Start button.

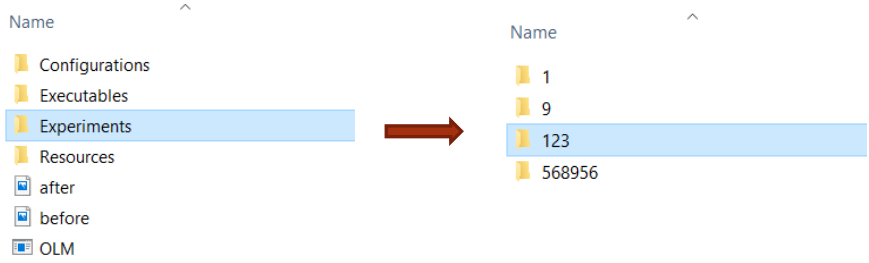
3. On this page you have 3 buttons pre-defined by your experiment type (independent or guided and with or without distractions). If you click the "preparation" ("הכנה") button the office environment will be launched.

If you click "experiment" ("ניסוי") the main savanna scene will start, then click the "results" ("תוצאות") to launch the summery environment.

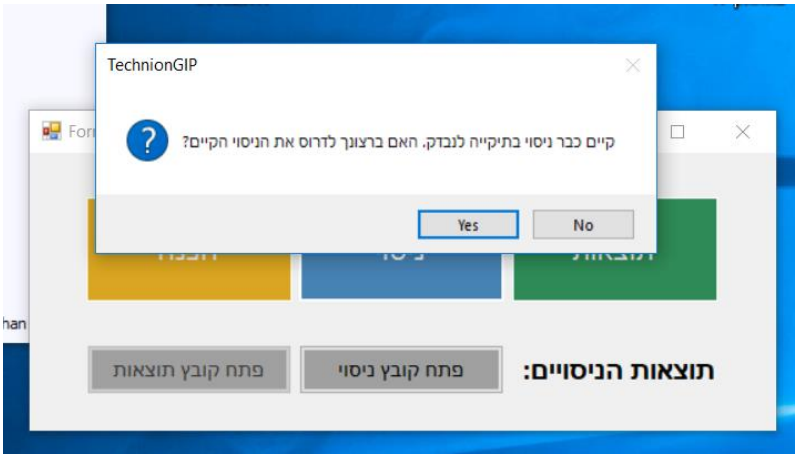
4. Whenever an environment is being clicked, you will see a timer that will tell you how much time left from the moment the environment was launched.

*Explanations on how to change the timers is at the end of this file

5. The "experiment" and "results" scene have outcome, the outcome will be saved in the a folder opened for the subject, the folder will be located at "OLM/Experiments ", the folder will have the subject number as a name:



6. If a file for the same subject number exists the following error will pop:



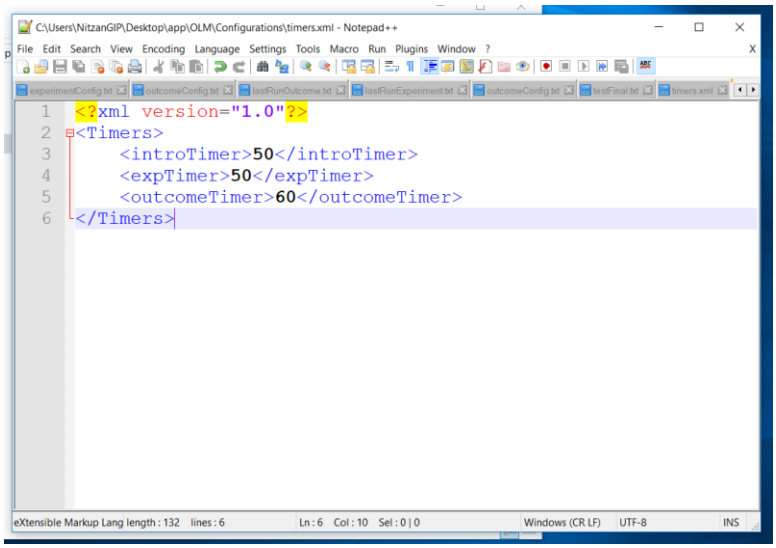
if you click "yes" the last outcome of subject with this number will be lost, so if you

do need the results make sure to create a backup of the information in a different location.

7. After the scenes with an outcome was over, you will see that a link to the outcome file is enabled at the bottom of the window. When you click it you will see the subject's results.

TIMERS CONFIGURATION

The timer's configuration file is located in "OLM\Configurations", to edit timers you need to open the "timers" file with any text editor and change the numbers written in black.

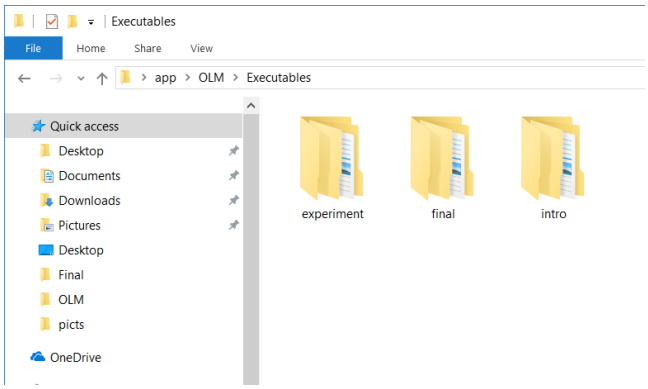


DIRECT ACTIVATION

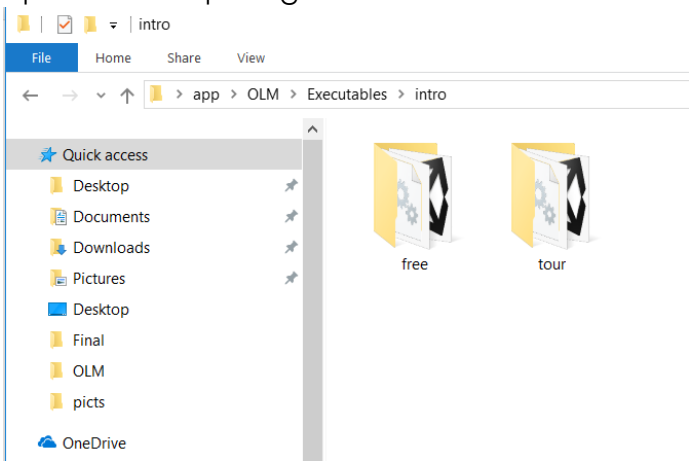
In this section we will explain how to run the experiment without the application:

ACTIVATE THE SCENES:

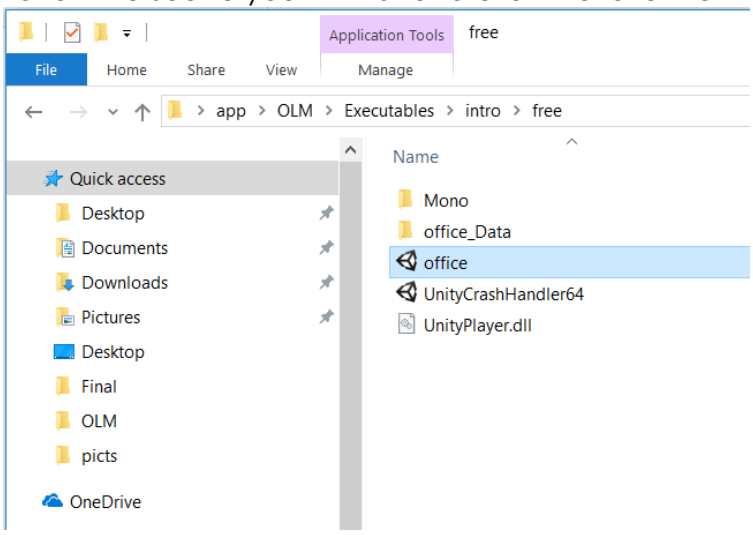
All the scenes will be in "OLM\Executables":



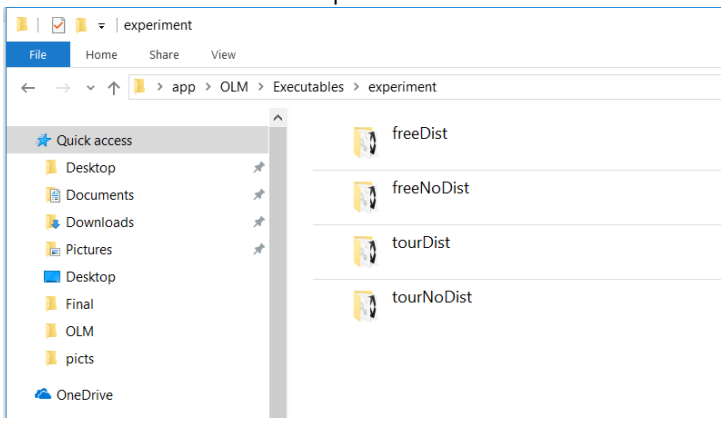
- **הכנה**- you will find the scenes in the "intro" folder, there you will have all the options of this part: guided tour and the free movement in separate folders:



To run the scene you will have to click the 'exe' file with name 'office':

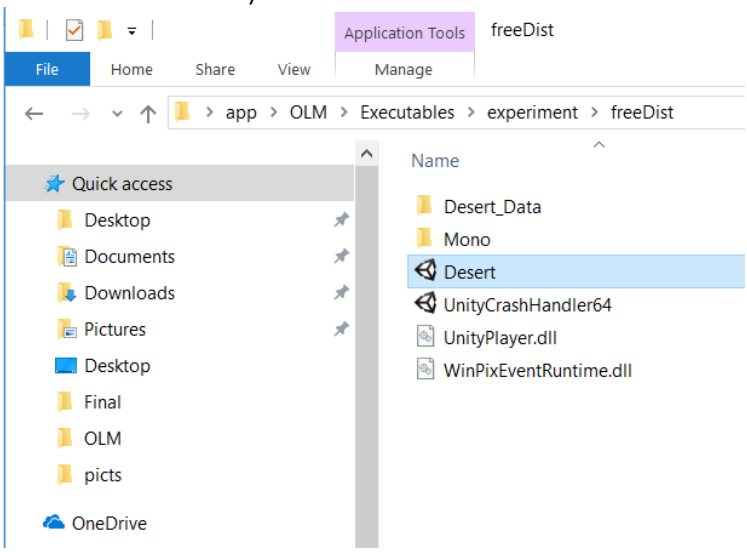


- **ניסוי** – you will find the scenes in the "experiment" folder, there you will have all the options of this part: guided tour and the free movement with or without distractions will have separate folders:



*Dist = מסיח

To run the scene you will have to click the 'exe' file with name 'Desert'



- **תוצאות**- you will find the scenes in the "final" folder, with same folders as in the intro (guided or free), To run the scene you will have to click the 'exe' file with name 'Desert'.

Important: there will be no timer in this method of activation so it's your responsibly to time it.

GET THE OUTCOME FILES:

The outcome files of the last experiment will be in "OLM\Configurations".

The outcome of the "experiment" is named "experimentConfig":

experimentConfig - Notepad

File Edit Format View Help

```
Experiment Results
=========================================
  Diversion Number      |      Clicked
=========================================
        1                            V
        2                            X
        3                            X
        4                            X
        5                            X
=========================================
Number of irrelevant clicks: 0
```

notice that the subject information is not shown, you will have to add it manually

The outcome of the "result" is named "outcomeConfig":

outcomeConfig - Notepad

File Edit Format View Help

```
Outcome Results
=============================================================================
  Object#     |  playerThoughtChanged  |   ChangedOriginally   |   Result
=============================================================================
    01                   X                       V                   X
    02                   X                       X                   V
    03                   X                       V                   X
    04                   X                       V                   X
    05                   X                       X                   V
    06                   X                       V                   X
    07                   X                       V                   X
    08                   X                       V                   X
    09                   X                       X                   V
    10                   X                       V                   X
    11                   X                       X                   V
    12                   X                       V                   X
    13                   X                       X                   V
    14                   X                       V                   X
    15                   X                       V                   X
    16                   X                       X                   V
    17                   X                       V                   X
    18                   X                       X                   V
    19                   X                       V                   X
    20                   X                       V                   X
    21                   X                       X                   V
    22                   X                       X                   V
    23                   X                       X                   V
    24                   X                       V                   X
=============================================================================
14 mistakes (1,3,4,6,7,8,10,12,14,15,17,19,20,24)
```

Important: those files will be over written by the next experiment, remember to make a copy of them after each subject.