

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος
Αντικειμενοστρεφής Ανάπτυξη Εφαρμογών

<i>Αριθμός εργασίας – Τίτλος εργασίας</i>	<i>Ατομική Εργασία 2 Μονάδων</i>
Όνομα φοιτητή	Κωνσταντίνος Καλογερόπουλος
Αρ. Μητρώου	Π19057
Ημερομηνία παράδοσης	15/01/2021



Εκφώνηση εργασίας

Καλείσθε να αναπτύξετε μια παραθυρική εφαρμογή που να υποστηρίζει κάποιες λειτουργίες ζωγραφικής. Βασικές λειτουργίες σχεδίασης που θα πρέπει να υποστηρίζει η εφαρμογή σας είναι οι ακόλουθες:

- Ευθύγραμμα τμήματα
- Ελεύθερο σχέδιο
- Ορθογώνια σχήματα
- Ελλείψεις
- Κύκλοι

Επιπλέον θα πρέπει να δίνετε τη δυνατότητα στον χρήστη να μπορεί κατά τη λειτουργία του προγράμματός σας να αλλάξει τόσο το χρώμα, όσο και το πάχος της γραμμής, από κατάλληλο μενού με επιλογές που θα έχετε δώσει εσείς μέσα στην εφαρμογή σας (π.χ. με τη χρήση ComboBox).

Η εφαρμογή σας θα πρέπει να διαθέτει και μενού (menustrip) για τουλάχιστον κάποιες από τις λειτουργίες που υποστηρίζει.

Τα όσα/όποια σχήματα έχει χρησιμοποιήσει ο χρήστης θα πρέπει να αποθηκεύονται σε ΒΔ (της επιλογής σας), μαζί με χρονοσφραγίδα (timestamp). Δεν χρειάζεται να αποθηκεύετε τα στοιχεία του σχήματος (points) παρά μόνο το είδος (ευθύγραμμο τμήμα, κύκλος, κ.λπ).

Τέλος αξιοποιώντας τη χρήση των timers, η εφαρμογή σας θα διαθέτει έως 4 κουμπιά, στο πάτημα των οποίων θα σχεδιάζεται κάτι πάνω στη φόρμα (σχέδια της επιλογής σας, όπως ένα σπίτι, ή ακόμα και μια φράση) σταδιακά, γραμμή-γραμμή. Δηλαδή το τελικό σχήμα θα έχει προκύψει από τα επιμέρους ευθύγραμμα τμήματα από τα οποία αποτελείται, με μια μικρή χρονική καθυστέρηση (π.χ. 500 milliseconds), ώστε να δώσετε την αίσθηση ότι το πρόγραμμά σας ζωγραφίζει εκείνη τη στιγμή.

Είστε ελεύθεροι να «επεκτείνετε» τις δυνατότητες της εφαρμογής σας και με δικές σας προσθήκες λειτουργιών.

Καλή επιτυχία!



1 Εισαγωγή

Το θέμα της εργασίας αποτελεί η δημιουργία ενός προγράμματος ζωγραφικής με ορισμένες λειτουργίες. Το πρόγραμμα υλοποιήθηκε με χρήση C# στο WinForms framework του .NET.

2 Περιγραφή του προγράμματος

Το βασικό σκέλος το προγράμματος υλοποιήθηκε με την χρήση μιας λίστας η οποία αποθηκεύει τα σημεία (Points) που ζωγραφίζει ο χρήστης, το χρώμα που έχει επιλέξει και το μέγεθος της πέννας. Η λίστα αυτή παίρνει σαν όρισμα την κλάση **PenSettings** η οποία της παρέχει όλες τις παραπάνω ιδιότητες. Ακόμα η κλάση **PenSettings** περιέχει ένα αντικείμενο της κλάσης **Shapes**, η οποία αναφέρεται στην δημιουργία των σχημάτων και περιέχει όλες τις συντεταγμένες(X,Y) που θα δοθούν αργότερα σαν παράμετροι σε συναρτήσεις όπως είναι οι:

`DrawEllipse()`, `DrawLine()` , `DrawRectangle()`.

Όλη ζωγραφική γίνεται «εμφανή» με την χρήση του Paint Event του control πάνω στο οποίο ζωγραφίζουμε και το οποίο καλείται με την `Invalidate()` συνάρτηση του ίδιου control.

Η λίστα εμπλουτίζεται στο `mouseUp` Event του control στο οποίο ζωγραφίζουμε.

Το «preview» των σχημάτων πραγματοποιείται στο Paint Event σε συνεργασία με το `mouseMove` Event του control το οποίο μας ανανεώνει τις συντεταγμένες X,Y καθώς κινείται ο κέρσορας επάνω στο control.



2.1 PenSettings Class

```
//pen settings
[Serializable]
13 references | Konstantinos Kalogeropoulos, 19 hours ago | 1 author, 4 changes
public class PenSettings
{
    3 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public List<Point> Points { get; set; }
    7 references | Konstantinos Kalogeropoulos, 23 hours ago | 1 author, 1 change
    public Color PenColor { get; set; }
    7 references | Konstantinos Kalogeropoulos, 23 hours ago | 1 author, 1 change
    public float PenWidth { get; set; }
    21 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public Shapes Shapes { get; set; }
    [NonSerialized]
    public Pen Pen;
    5 references | Konstantinos Kalogeropoulos, 19 hours ago | 1 author, 2 changes
    public PenSettings(Color penColor, float penWidth, List<Point> points, Shapes shape)
    {
        PenColor = penColor;
        PenWidth = penWidth;
        Pen = new Pen(PenColor, PenWidth);
        Pen.DashCap = System.Drawing.Drawing2D.DashCap.Flat;
        Points = points;
        Shapes = new Shapes( shape.MouseX, shape.MouseY, shape.MouseX1, shape.MouseY1);
    }
}
```

2.2 Shapes Class

```
[Serializable]
10 references | Konstantinos Kalogeropoulos, 18 hours ago | 1 author, 4 changes
public class Shapes //holds the coordinates of the shapes
{
    13 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public int MouseX { get; set; }
    15 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public int MouseY { get; set; }
    11 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public int MouseX1 { get; set; }
    13 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public int MouseY1 { get; set; }

    14 references | Konstantinos Kalogeropoulos, 19 hours ago | 1 author, 1 change
    public Shapes(int mouseX, int mouseY, int mouseX1, int mouseY1)
    {
        MouseX = mouseX;
        MouseY = mouseY;
        MouseX1 = mouseX1;
        MouseY1 = mouseY1;
    }
}
```



2.3 Περιγραφή της δημιουργίας του timelapse.

Για την δημιουργία του timelapse κρίθηκε αναγκαία η δημιουργία της κλάσης **TimelapseSettings** η οποία περιέχει τα εξής property: ένα όνομα, μια λίστα ορίσματος *PenSettings*, τα δευτερόλεπτα που θέλουμε να παρακολουθήσουμε το timelapse και ένα αντικείμενο της κλάσης *Shapes* τα οποία δίνονται στο Constructor της κλάσης αυτής. Ακόμα η κλάση περιέχει το Tick Event ενός timer και μια μέθοδο εκκίνησης του Timer. Μέσα στο Tick Event προσπελαύνουμε κάθε στοιχείο της λίστας και καλούμε το *Invalidate()* του control το οποίο καλεί το Paint Event του. (Στην ουσία των πραγμάτων ξαναζωγραφίζουμε «ένα-ένα» τα στοιχεία της λίστας).

```
//timelapseSettings
[Serializable]
8 references | Konstantinos Kalogeropoulos, 20 hours ago | 1 author, 4 changes
public class TimelapseSettings
{
    2 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public string Name { get; set; }
    9 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public List<PenSettings> AllcurvesTl { get; set; }
    3 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public int ListIndex { get; set; } = 0;
    6 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public int SecondsAnimation { get; set; } = 0;

    21 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public Shapes Shapes { get; set; }
    [NonSerialized]
    Form1 form;
    [NonSerialized]
    public System.Windows.Forms.Timer timer = new System.Windows.Forms.Timer();

    7 references | Konstantinos Kalogeropoulos, 20 hours ago | 1 author, 3 changes
    public TimelapseSettings(string name, List<PenSettings> allcurvesTimelapse, int secondsAnimation, Shapes shape)
    {
        Name = name;
        AllcurvesTl = allcurvesTimelapse.ToList();
        SecondsAnimation = secondsAnimation;
        Shapes = new Shapes( shape.MouseX, shape.MouseY, shape.MouseX1, shape.MouseY1);
        // Shapes = new Shapes(allcurvesTimelapse, shape.MouseX, shape.MouseY, shape.MouseX1, shape.MouseY1);

        //adding the tick event to the timer
        timer.Tick += new EventHandler(timelapseTimer_Tick);
        if (AllcurvesTl.Count > 0) {
            //finding the interval based on the desired animation's duration of it.
            timer.Interval = (secondsAnimation * 1000) / AllcurvesTl.Count;
        }
    }
}
```



```
1 reference | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
private void timerlapseTimer_Tick(object sender, EventArgs e)
{
    try
    {
        Allcurves.Add(AllcurvesTl[ListIndex]);
        form.drawingBox.Invalidate();
        ListIndex++;
    }
    catch (ArgumentOutOfRangeException)
    {
        ListIndex = 0;
        timer.Stop();
    }
}

6 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
public void timerStarter()
{
    form = (Form1)Application.OpenForms[0];
    timer.Start(); //starts the timer
}
}
```

2.4 Περιγραφή της εισαγωγής στοιχείων σε βάση δεδομένων SQLite.

Για την ανάγκη της εισαγωγής στοιχείων στην βάση δεδομένων δημιουργήθηκε η κλάση **DataEntry** η οποία περιέχει σαν property τα στοιχεία που θα καταχωρηθούν δηλαδή το όνομα της λειτουργίας(Freestyle,Circle,Ellipse,Rectangle) και την ώρα. Στο mouseUp Event του control καλείται η συνάρτηση *InsertData(String Name)* η οποία δημιουργεί ένα αντικείμενο της DataEntry το οποίο περιέχει την ώρα δημιουργίας της λειτουργίας και το όνομα της. Τέλος εισάγει τα δεδομένα στην βάση δεδομένων.

```
//data entry for sqlite db
5 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 2 changes
public void InsertingData(String shape)
{
    conn.Open();
    newDataEntry = new DataEntry(shape);
    String insertQuery = "Insert into Shapes(name,Time) values('" + newDataEntry.Name + "','" + newDataEntry.timeStamp + "')";
    SQLiteCommand cmd = new SQLiteCommand(insertQuery, conn);
    cmd.ExecuteNonQuery();
    conn.Close();
}
```



```
2 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
public class DataEntry
{
    2 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public String Name { get; set; }
    2 references | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public String timeStamp { get; set; }
    1 reference | Konstantinos Kalogeropoulos, 1 day ago | 1 author, 1 change
    public DataEntry(String name)
    {
        Name = name;
        timeStamp = DateTime.Now.ToString();
    }
}
```

Database Structure Browse Data Edit Pragmas Execute SQL			
Table: Shapes			
	ID	name	Time
	Filter	Filter	Filter
1	23	ellipse	1/5/2021 1:07:16 AM
2	24	freestyle	1/5/2021 1:07:16 AM
3	25	ellipse	1/5/2021 1:07:17 AM
4	26	freestyle	1/5/2021 1:07:17 AM
5	27	ellipse	1/5/2021 1:07:17 AM
6	28	freestyle	1/5/2021 1:07:17 AM
7	29	ellipse	1/5/2021 1:07:18 AM
8	30	freestyle	1/5/2021 1:07:18 AM
9	31	ellipse	1/5/2021 1:07:18 AM
10	32	freestyle	1/5/2021 1:07:18 AM
11	33	freestyle	1/5/2021 1:07:23 AM
12	34	freestyle	1/5/2021 1:07:24 AM
13	35	freestyle	1/5/2021 1:10:17 AM
14	36	freestyle	1/5/2021 1:10:17 AM
15	37	line	1/5/2021 1:10:19 AM
16	38	ellipse	1/5/2021 1:10:21 AM
17	39	circle	1/5/2021 1:10:23 AM



2.5 Λοιπές Λειτουργίες

Επιπλέον λειτουργίες εφαρμογής που υλοποιήθηκαν πέρα από τις παραπάνω:

- Αποθήκευση εικόνας σε .jpg, .png, .bmp, .gif formats

(File -> Save As

- Άνοιγμα εικόνας με βάση τα παραπάνω formats.

(File -> Open)

- Αποθήκευση και δημιουργία .timelapse αρχείων από τον ίδιο των χρήστη

(File-> Save As Timelapse, Open A Timelapse)

- Προβολή ενός timelapse της εικόνας που δημιούργησε ο χρήστης με ένα κουμπί

(Tools -> Watch a Timelapse)

- Εκτύπωση εικόνας

(File -> Print)

- Προεπισκόπηση εκτύπωσης εικόνας

(File -> Print Preview)

- Drag and Drop λειτουργία εισαγωγής εικόνας απευθείας στην εφαρμογή

- Αντιγραφή(Copy) σε Clipboard της εικόνας του χρήστη από την εφαρμογή για επικόλληση σε τρίτες εφαρμογές που υποστηρίζουν αυτή την λειτουργία (Συνήθως chatting εφαρμογές όπως Messenger,Discord,Viber,Microsoft Teams Chat).

(Edit -> Copy)

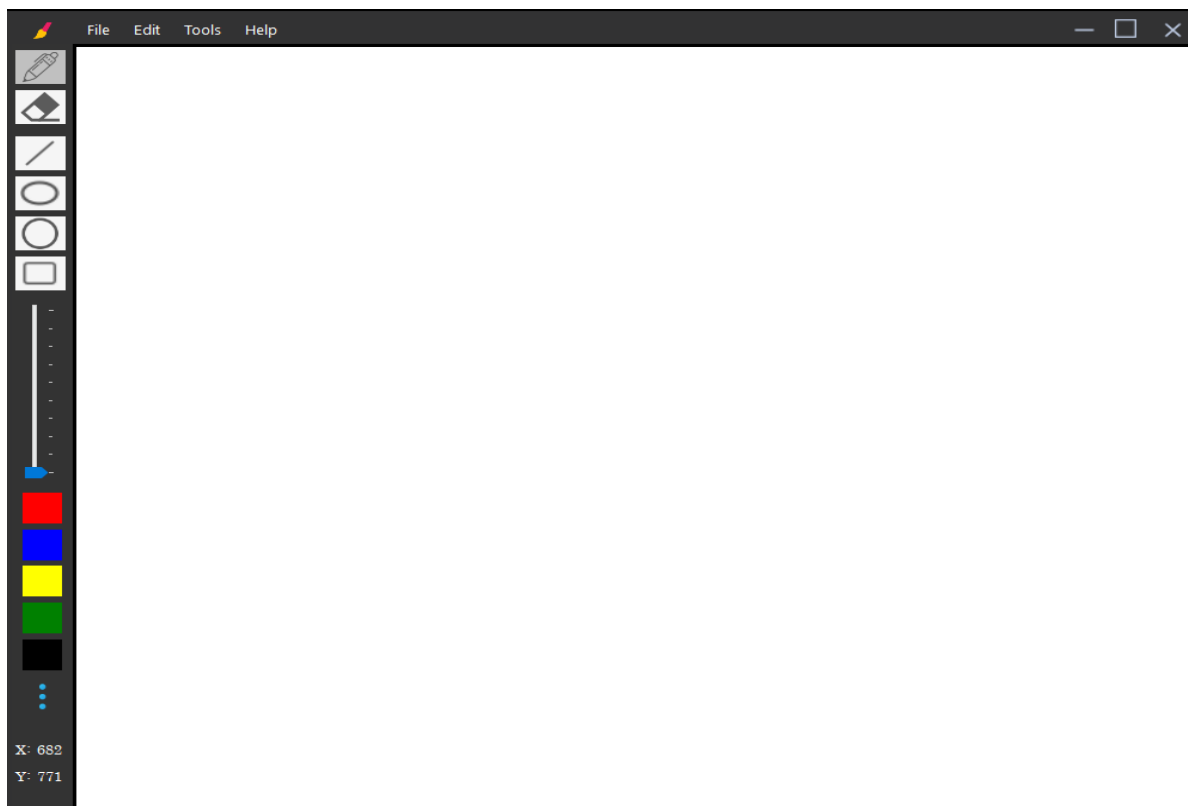
- Επικόλληση(Paste) εικόνας από Clipboard απευθείας μέσα στην εφαρμογή για επεξεργασία. (Edit -> Paste)

- Αναίρεση (Undo) (Edit -> Undo)

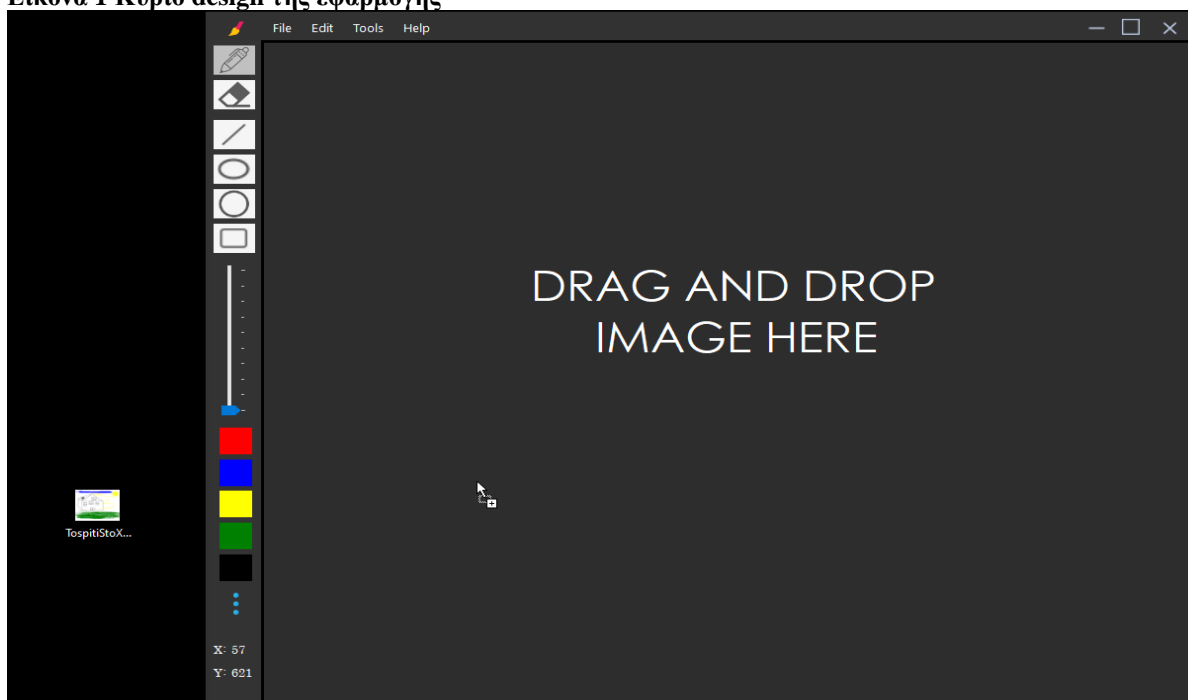
- Επανάληψη (Redo) (Edit -> Redo)

- Γρήγορη εισαγωγή σχεδίου από Tools->Quick Add όπου ο χρήστης μπορεί να επιλέξει διάφορες έτοιμες εικόνες οι οποίες εισάγονται με την μορφή timelapse.

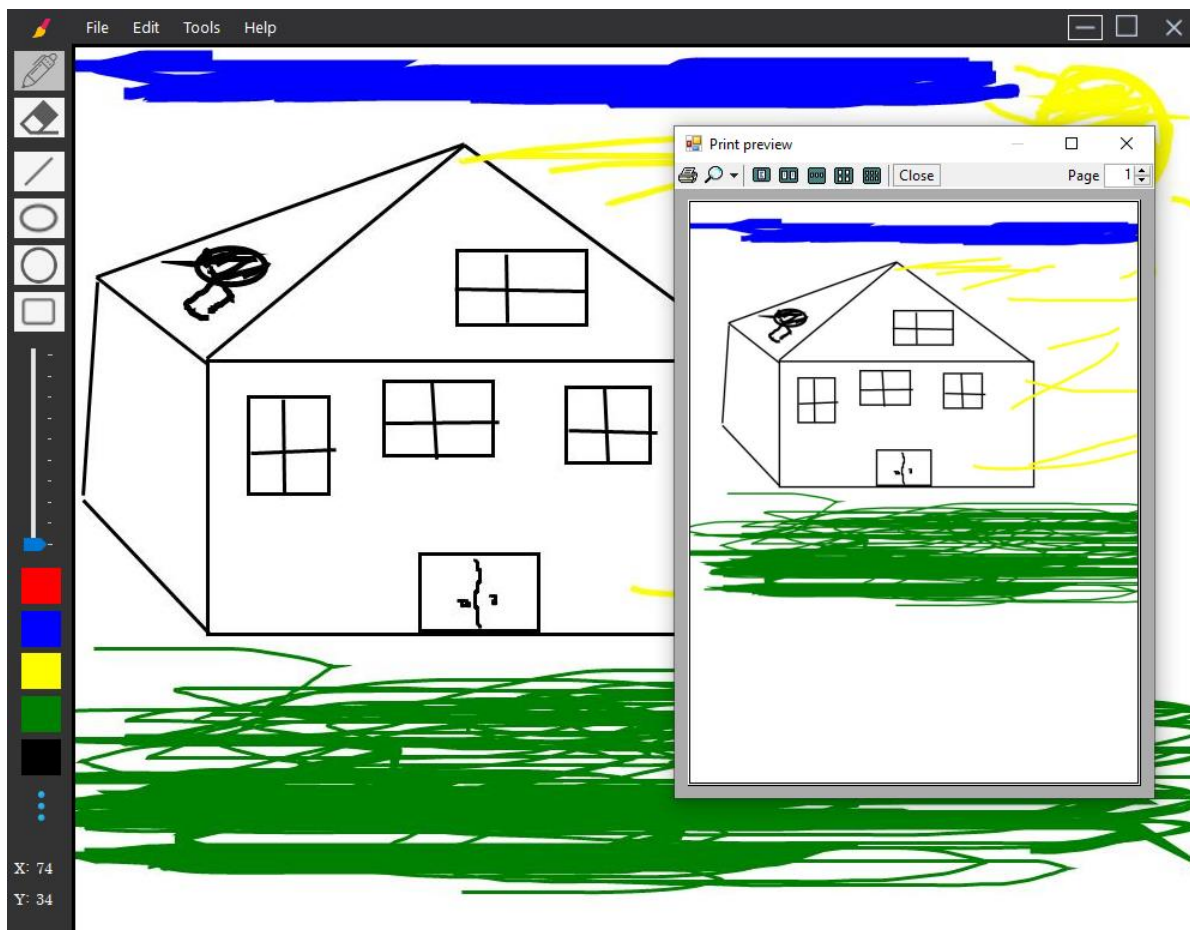
3 Επίδειξη της λύσης



Εικόνα 1 Κύριο design της εφαρμογής



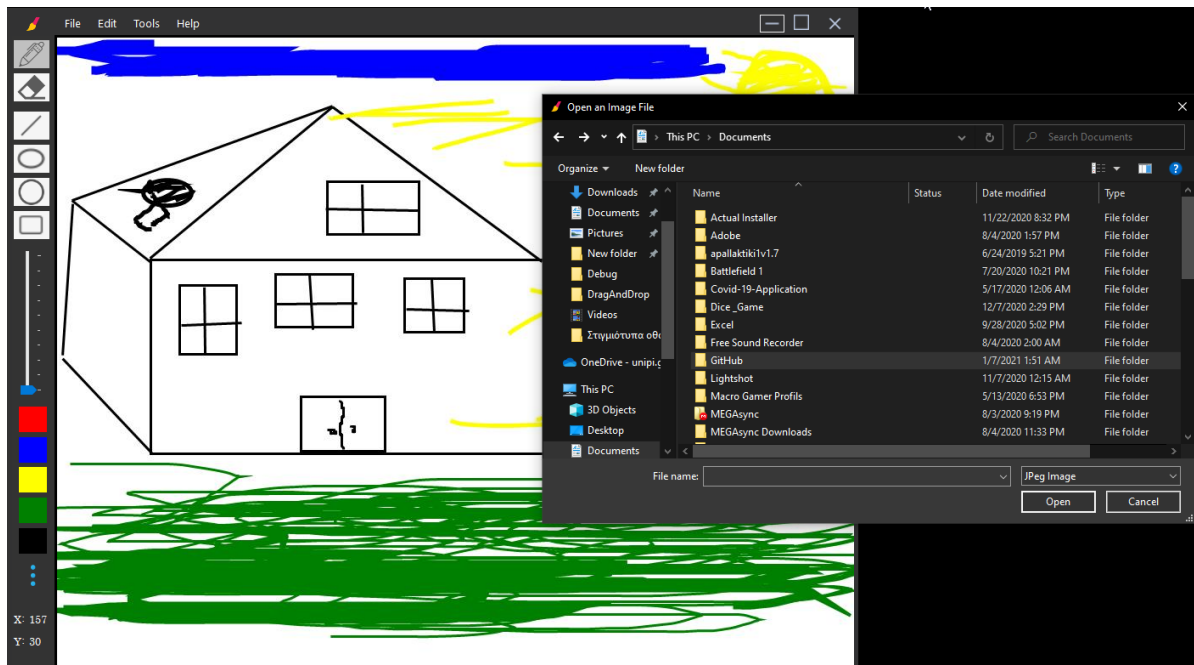
Εικόνα 2 Drag and Drop αρχείο εικόνας



Εικόνα 3 Print Preview



Εικόνα 4 Αποθήκευση Εικόνας



Εικόνα 5 Άνοιγμα Εικόνας

Οι λειτουργίες timelapse δεν μπορούν να γίνουν εισαγωγή σαν εικόνα καθώς δεν θα είχε κάποιο νόημα διότι η εισαγωγή .Gif images από το Word δεν υποστηρίζεται. Όπως και να έχει, μπορείτε να τις δείτε απευθείας μέσα από την εφαρμογή «ζωντανά».

Για περισσότερες πληροφορίες ανατρέξτε στο github.com/CostasCF.