

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΚΑΤΑΣΚΕΥΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ ΜΝΗΜΗΣ ΓΙΑ ΠΑΙΔΙΑ ΣΕ C#

Μαρία Μπουραντά

Επιβλέπων: Βασίλειος Χαριλόγης τίτλος, βαθμίδα

Άρτα, Φεβρουάριος,2020

CREATING AN EDUCATIONAL MEMORY GAME FOR CHILDREN WITH C# PROGRAMMING LANGUAGE

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Τόπος, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

- Επιβλέπων καθηγητής
 Όνομα Επίθετο,
- 2. Μέλος επιτροπής

Όνομα Επίθετο,

3. Μέλος επιτροπής

Όνομα Επίθετο,

© Μπουραντά Μαρία, 2020

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα πτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Μπουραντά Μαρία

Υπογραφή

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου Βασίλειο Χαριλόγη για την πολύτιμη βοήθεια του και την κατανόηση του. Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου και τον σύντροφο μου που με στήριξαν με το δικό τους ξεχωριστό τρόπο.

ΠΕΡΙΛΗΨΗ

Η πτυχιακή εργασία την οποία θα υλοποιήσουμε θα απαρτίζεται από δύο μέρη. Στο θεωρητικό μέρος θα αναφερθούμε στη γλώσσα C# με την οποία θα υλοποιήσουμε το προγραμματιστικό κομμάτι του παιχνιδιού. Η εφαρμογή θα είναι ένα εκπαιδευτικό παιχνίδι για παιδιά στο οποίο ο χρήστης (παιδί) θα έχει την δυνατότητα να επιλέξει ζευγάρια από ίδιες εικόνες προσπαθώντας να απομνημονεύσει τη θέση τους κάθε φορά. Το παιχνίδι θα αρχίζει με όλες τις εικόνες να μην είναι ορατές και όταν το παιδί επιλέξει δύο ίδιες εικόνες τότε μένουν ανοιχτές αν είναι ίδιες, αλλιώς αποκρύπτονται. Στη συνέχεια ο χρήστης – παιδί προσπαθεί να απομνημονεύσει την θέση της κάθε εικόνας με σκοπό να βρει όλα τα κοινά ζεύγη εικόνων. Αν το παιδί κερδίζει στο προβλεπόμενο χρόνο τότε εμφανίζεται κάποια επιβράβευση!

Λέζεις-κλειδιά: παιχνίδι, κάρτες, εκπαίδευση, παιδιά, μνήμη, προγραμματισμός καθοδηγούμενος από γεγονότα.

ABSTRACT

The dissertation that we are going to implement, it will be consist of two parts. The firs one is a theoretical part were we are going to describe the features of C# and the second one is the implementation part of the game application. The application will be an educational game for kids, whose are going to have the chance to select pairs from same images. The game will start with all the icons hidden (not visible to the player) and will only be visible if the player finds two pair of the same icon . If the images doesn't match then they will be hidden again in order to find a perfect match again. After that, the player is going to try and memorize every image that he finds and try to pair it with the right one in order to complete the game successfully. If the player (kid) wins inside the time limit then a reward will pop up.

Keywords: game, cards, education, kids, memory, event-driven programming

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	6
ABSTRACT	7
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	8
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	9
ΕΙΣΑΓΩΓΗ	11
1.Προγραμματισμός καθοδηγούμενος από γεγονότα	12
1.1 Ορισμός	12
1.2 Πλεονεκτήματα –μειονεκτήματα	13
1.3 H C#	14
1.3.1 Ιστορία ,	14
1.3.2 Χαρακτηριστικά	15
1.3.3 .Net Framework	15
2.Το περιβάλλον ανάπτυξης Visual Studio 2019	16
2.1 Λήψη και εγκατάσταση	16
3-Δημιουργία Εφαρμογής	20
3.1 Περιγραφή της εφαρμογής-Αλγόριθμος	20
3.2 Δημιουργία εφαρμογής	22
3.2.1 Δημιουργία νέου project	22
3.2.2 Εισαγωγή όλων των απαραίτητων στοιχείων	23
3.2.3 Δημιουργία λίστας κατηγοριών	28
3.2.4 Αρχικοποίηση παιχνιδιού	35
3.2.5 Εισαγωγή εικόνας υποβάθρου φόρμας (Background image)	37
3.2.6 Εισαγωγή μπάρας κύλισης χρόνου	39
3.2.7 Εμφάνιση αρχείου- πολυμέσου επιβράβευσης	40
3.3 Συγγραφή πηγαίου κώδικα	43
3.3.1 Προσθήκη λειτουργικότητας για την μπάρα κύλισης χρόνου	43
3.3.2 Προσθήκη λειτουργικότητας σε κάθε επιλογή της εικόνας-Label	44
3.4 Δημιουργία αρχείου εγκατάστασης της εφαρμογής	44
Κεφάλαιο 4- Συμπεράσματα	52
4. ΒΙΒΛΙΟΓΡΑΦΙΑ	54
Ελληνόγλωσση	54
Ξενόγλωσση	54

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1.	1- Εκδόσεις της C# (C# History, χ.η)	14
Εικόνα 2.	1-Αρχική σελίδα VS 2019 (Visual Studio, 2019)	16
Εικόνα 2.	2-Επιλογή έκδοσης	16
Εικόνα 2.	3-Λήψη VS (Visual Studio, 2019)	17
Εικόνα 2.	4-Εμφάνιση διαθέσιμων στοιχείων εγκατάστασης	17
Εικόνα 2.	5-Επιλογή .Net Desktop Development	18
Εικόνα 2.	6-Εμφάνιση προόδου εγκατάστασης	18
Εικόνα 3.	1-Επιλογή νέου έργου	22
Εικόνα 3.	2-Επιλογή project	22
Εικόνα 3.	3-Όνομα project	23
Εικόνα 3.	4-Eισαγωγή TableLayoutPanel	24
Εικόνα 3.	5-Ορισμός γραμμών και στηλών(1)	24
Εικόνα 3.	6-Ορισμός γραμμών και στηλών (2)	25
Εικόνα 3.	7-Ορισμός πλάτους στήλης	25
Εικόνα 3.	8-Εισαγωγή Labels	26
Εικόνα 3.	9 -Ιδιότητα Dock σε Label	26
Εικόνα 3.	10- Αλλαγή στυλ περιγράμματος	27
Εικόνα 3.	11-Εμφάνιση αποτελέσματος label panel	27
Εικόνα 3.	12-Ορισμός συνάρτησης που θα καλείται σε κάθε event	28
Εικόνα 3.	13-Προσθήκη νέου item	29
Εικόνα 3.	14-Επιλογή φόρμας ως νέου item	29
Εικόνα 3.	15-Εισαγωγή Image List	30
Εικόνα 3.	16-Εισαγωγή στοιχείου ListView	31
Εικόνα 3.	17-Εισαγωγή εικόνων σε Image List	31
Εικόνα 3.	18-Επιλογή αρχείων εικόνων Image list	32
Εικόνα 3.	19-Ορισμός ListView item	32
Εικόνα 3.	20- Προσθήκη εικόνων σε ListView	33
Εικόνα 3.	21- Προσθήκη text στις κατηγορίες	33
Εικόνα 3.	22-Κλήση συνάρτησης selectcategory()	34
Εικόνα 3.	23-Επιβεβαίωση ορθής κλήσης	34
Εικόνα 3.	24-Εμφάνιση πάνελ με κρυφές εικόνες	35
Εικόνα 3.	25-Αρίθμηση Label	36
Εικόνα 3.	26-Πίνακας παρασκηνίου με αρίθμηση Labels	37
Εικόνα 3.	27-Ορισμός εικόνας background	37
Εικόνα 3.	28-Επιλογή εικόνας	38
Εικόνα 3.	29-Ιδιότητα Stretch	38
Εικόνα 3.	30-Εισαγωγή μπάρας προόδου	39
Εικόνα 3.	31-Ορισμός παραμέτρων progress bar	40
Εικόνα 3.	32-Εισαγωγή timer	40
Εικόνα 3.	33-Επιλογή νέου item για την εργαλειοθήκη	41
Εικόνα 3.	34-Επιλογή Windows Media Player	42
Εικόνα 3.	35-Ορισμός ιδιοτήτων media player	43
Εικόνα 3.	36-Επιλογή setup wizard(1)	45
Εικόνα 3.	37- Επιλογή setup wizard(2)	46

Εικόνα 3. 38-Επιλογή τύπου αρχείου εγκατάστασης	
Εικόνα 3. 39-Δυνατότητα προσθήκης αρχείων	
Εικόνα 3. 40-Επιλογή του Application Folder για προσθήκη αρχείων	
Εικόνα 3. 41-Επιλογή αρχείων για προσθήκη στο setup	
Εικόνα 3. 42-Εικόνα μετά την προσθήκη	
Εικόνα 3. 43-Επιλογή δημιουργίας συντόμευσης	
Εικόνα 3. 44-Ορισμός αρχείου προορισμού συντόμευσης	
Εικόνα 3. 45-Εκκίνηση δοκιμαστικής εγκατάστασης	
Εικόνα 3. 46-Επιλογή φακέλου εγκατάστασης	
Εικόνα 3. 47-Ολοκλήρωση εγκατάστασης	
Εικόνα 3. 48-Εμφάνιση συντόμευσης και αρχείου εγκατάστασης	

ΕΙΣΑΓΩΓΗ

Σε αυτή την εργασία θα δούμε πως φτιάχνεται από την αρχή (βηματικά) ένα παιχνίδι μνήμης για παιδιά σε C# χρησιμοποιώντας το περιβάλλον ανάπτυξης (IDE) Microsoft Visual Studio. Αρχικά θα μιλήσουμε για τον ορισμό του καθοδηγούμενου προγραμματισμού από γεγονότα ,τα πλεονεκτήματα και τα μειονεκτήματα του. Ακόμα θα δούμε την ιστορία, την δημιουργία και τα χαρακτηριστικά της γλώσσας C#. Επίσης δίνονται αναλυτικές οδηγίες λήψης και εγκατάστασης του Microsoft Visual Studio. Έπειτα εμφανίζεται η δημιουργία project στο πρόγραμμα και ο πηγαίος κώδικας της εργασίας. Ολοκληρώνοντας την εφαρμογή μας θα προσθέσουμε και ένα έργο (project) το οποίο θα δημιουργεί το τελικό αρχείο εγκατάστασης της εφαρμογής μας. Στο τέλος της παρούσας εργασίας παραθέτουμε τα συμπεράσματα από την ενασχόληση μας με την δημιουργία αυτής της εκπαιδευτικής εφαρμογής.

1.Προγραμματισμός καθοδηγούμενος από γεγονότα.

1.1 Ορισμός

Σύμφωνα με έναν ορισμό ο προγραμματισμός καθοδηγούμενος από γεγονότα είναι μια σύγχρονη μέθοδος προγραμματισμού η οποία επωφελείται από τα πλεονεκτήματα του αντικειμενοστραφούς προγραμματισμού επεκτείνοντας τις προγραμματιστικές μας δυνατότητες. Στα σύγχρονα περιβάλλοντα ανάπτυξης (IDE) χρησιμοποιούμε έτοιμα εργαλεία που μας βοηθάνε να δημιουργήσουμε εύκολα και γρήγορα το δικό μας πρόγραμμα. Η διαφορά με τον δομημένο προγραμματισμό είναι ότι η ροή του προγράμματος κατά τη διάρκεια εκτέλεσης εξαρτάται από τις επιλογές του χρήστη (Πολίτης & Γιαννόπουλος, 1997).

Τα προγράμματα που βασίζονται σε κείμενο (κλασσικά προγράμματα) το πρόγραμμα ελέγχει τη ροή και την ακολουθία εισόδου δεδομένων. Τα προγράμματα γραφικού περιβάλλοντος (GUI) βασίζονται στο χειρισμό γεγονότων (event driven). Σε αυτά τα προγράμματα :

- Η είσοδος καθορίζει τη ροή της εκτέλεσης του προγράμματος. Αυτός είναι ένας λόγος που η ροή του προγράμματος αναπαρίσταται δύσκολα σε ένα διάγραμμα ροής.
- Η ακολουθία της εισόδου δεδομένων ελέγχει την εκτέλεση του προγράμματος.
- Ο κώδικας του χρήστη καλείται από τον κώδικα του συστήματος (Συμβώνης, χ.η)
- Η ροή ελέγχεται με βάση τη διαδραστικότητα του χρήστη με το περιβάλλον (Γραφικά Περιβάλλοντα, χ.η)

1.2 Πλεονεκτήματα -μειονεκτήματα

Ο προγραμματισμός καθοδηγούμενος από γεγονότα με μία πρώτη ματιά φαίνεται να παρουσιάζει μόνο πλεονεκτήματα και να είναι η ιδανικότερη προσέγγιση στην ανάπτυξη λογισμικού.

- Τα προγράμματα μπορούν εύκολα να τροποποιηθούν. Η αλλαγές στον πηγαίο κώδικα γίνονται με τη χαρακτηριστική ευκολία που παρέχει ο οποιοδήποτε επεξεργαστής κειμένου.
- Ιδανικός για προγράμματα με γραφική διεπαφή (graphical interface). Ο χρήστης έχει μία εργαλειοθήκη με ένα μεγάλο πλήθος εργαλείων που μπορεί να εισάγει στην εφαρμογή του.
- Απλότητα. Ο προγραμματιστής με ένα διπλό κλικ μπορεί να επεξεργαστεί τον κώδικα από το αντικείμενο που επέλεξε. Αυτό το είδαμε πρακτικά στην εφαρμογή μας όπου με αυτόν τον τρόπο προσθέταμε λειτουργικότητα στα στοιχεία της εφαρμογής μας.
- Ευκολία σε μοναδιαίους ελέγχους (unit tests). Τα τμήματα του κώδικα είναι ευδιάκριτα χωρισμένα και αυτό διευκολύνει τη δημιουργία ελέγχων σε αυτούς.
- Διαδραστικότητα. Σχεδόν όλα τα σύγχρονα προγράμματα χρησιμοποιούν προγραμματισμό καθοδηγούμενο από γεγονότα και υλοποιούνται σε γραφικό περιβάλλον.
- Προσαρμογή αισθητήρων. Μπορούμε να προσαρμόσουμε αισθητήρες και γενικότερα άλλο υλικό (hardware).

Ας προσπαθήσουμε να δώσουμε μία καλύτερη εικόνα παραθέτοντας και τα μειονεκτήματα του

 Είναι σύνθετος για απλά προγράμματα. Σε κάποιες περιπτώσεις όπου το πρόγραμμα είναι μικρό και απλό δεν είναι ιδανική επιλογή μιας και μπορούμε να ακολουθήσουμε άλλο δρόμο ευκολότερης υλοποίησης όπως για παράδειγμα με την κατασκευή ενός απλού έργου κονσόλας (console application).

- Η ροή είναι λιγότερο λογική και προφανής μιας και εξαρτάται από τα γεγονότα (events).
- Δύσκολος στον εντοπισμό λάθους. Η δυσκολία έγκεται στη δυσκολία κατανόησης της ροής του προγράμματος και όχι στα συντακτικά λάθη τα οποία εντοπίζει ο μεταγλωττιστής.
- Τα προγράμματα είναι πιο αργά.
- Ο αποκλεισμός (blocking) είναι πολύ πιο σύνθετος (Advantages and Disadvantages of Event-Driven Programming, 2019). Η έννοια του αποκλεισμού εδώ αναφέρεται σε πιθανή ανάγκη παραλληλοποίησης τμημάτων κώδικα που ωστόσο δεν χρειαστήκαμε στην εφαρμογή μας.

1.3 H C#

1.3.1 Ιστορία

Η ιστορία της C# ,προφέρεται ως «c sharp», ξεκινάει κάπου στο 2002 με ιδρυτή τον Anders Hejlsberg. Είναι βασισμένη στη C++ και τη Java αλλά έχει πολλές πρόσθετες επεκτάσεις. Ας ρίξουμε μία ματιά στις εκδόσεις της.



Εικόνα 1. 1- Εκδόσεις της C# (C# History, χ.η)

1.3.2 Χαρακτηριστικά

Η C# είναι μία γλώσσα αντικειμενοστραφούς προγραμματισμού. Τα κυριότερα χαρακτηριστικά της είναι:

- Είναι απλή, υπό την έννοια ότι έχει μία δομημένη προσέγγιση, μεγάλο σύνολο βιβλιοθηκών και τύπων δεδομένων.
- Είναι μία αντικειμενοστραφής γλώσσα.
- Παρέχει ασφάλεια κώδικα (type safe). Ο κώδικας μπορεί να έχει πρόσβαση μόνο σε εκείνο το μέρος της μνήμης που του επιτρέπεται να εκτελέσει.
- Χαρακτηρίζεται από διαλειτουργικότητα. Μπορεί δηλαδή να κάνει ότι και οποιαδήποτέ γηγενής σε C++ (native) εφαρμογή μπορεί να κάνει.
- Πλούσια βιβλιοθήκη από συναρτήσεις
- Η μεταγλώττιση και η εκτέλεση γίνονται γρήγορα.
- Ακολουθεί τον δομημένο προγραμματισμό (δομές ακολουθίας, ελέγχου και επανάληψης). Μπορούμε επίσης να δομήσουμε τον κώδικα υπό μορφή συναρτήσεων, πράγμα που μας δίνει μεγαλύτερο έλεγχο. (C# Features, χ.η)

1.3.3 .Net Framework

Η C# από μόνη της δεν μπορεί να κάνει πάρα πολλά αν δεν υπάρχει μία κατάλληλη δομή πάνω στην οποία θα τρέξει. Το ρόλο αυτής της δομής ανέλαβε μία τεχνολογία που ονομάζεται .Net Framework και κατά μία έννοια είναι σαν μία εικονική μηχανή (Virtual Machine) όπου εκεί γίνεται η μεταγλώττιση και η εκτέλεση ενός προγράμματος γραμμένου σε C#. Στην πραγματικότητα σε ένα .Net Framework τρέχουν πάνω από εξήντα γλώσσες , εκ των οποίων οι έντεκα είναι της microsoft. Η έκδοση .Net Framework 1.0 εμφανίστηκε το 2002 και έκτοτε νέες εκδόσεις ακολουθούν και το δημοφιλές περιβάλλον ανάπτυξης Microsoft Visual Studio (Aggarval, χ.η). Στην εργασία μας χρησιμοποιήσαμε την έκδοση .Net Framework 4.7.2 η οποία εγκαταστάθηκε αυτόματα με το περιβάλλον ανάπτυξης Visual Studio 2019 της microsoft.

2.Το περιβάλλον ανάπτυξης Visual Studio 2019

2.1 Λήψη και εγκατάσταση

Επισκεπτόμαστε την ιστοσελίδα <u>https://visualstudio.microsoft.com/vs/</u> και επιλέγουμε Download Visual Studio



Εικόνα 2. 1-Αρχική σελίδα VS 2019 (Visual Studio, 2019)

Έπειτα κάνουμε λήψη της έκδοσης Community 2019



Εικόνα 2. 2-Επιλογή έκδοσης

Και αναμένουμε για την λήψη του αρχείου

Vi	isual Studio Vis	ual Studio 2019	Features 🗸	Editions 🗸 Downl	oads Support	✓ More √	Free	e Visual Studio	
	V	'is	Visual Stud	dio Installer)19	
	Code f	aster.	Just a mon	nent Fetching your fil	les.	1,04 MB/sec		in-class ID	E.
						Cancel			
View	Project Build Del	bug Test A ebug - Any	nalyze Tools CPU + S	Extensions Window martHotel360.PublicWeb	Help Search V	isual Studio	ዖ # 💷	SmartHotel360.Pub ₌	licWeb
: 7 60.Publ	CuestRequestTest.cs # X - 02.PublicWeb.Tests * * * * SmartHotel360.PublicWeb.Tests.GuestReque * * GoGreenInitTest()								irr • ≒
⊟ name { : :	<pre>espace SmartHotel [TestClass] 0 references kendrahave public class Gue</pre>	360.Publick	eb.Tests 2 authors, 12 char st	rges			Î	Solution 'Smar Solution Ite Solution Ite B B DoWork B B LargeTestPr	tHotel360.P ms oject

Εικόνα 2. 3-Λήψη VS (Visual Studio, 2019)

Μόλις ολοκληρωθεί η λήψη εμφανίζονται οι παρακάτω επιλογές:



Εικόνα 2. 4-Εμφάνιση διαθέσιμων στοιχείων εγκατάστασης

Εμείς επιλέγουμε .NET desktop development και έπειτα install

Work	oads Individual components Language page	cks	Installation locations	
Web &	Cloud (4)			
	ASP.NET and web development Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker support.	Δ	Azure development Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET Core and .NET	
2	Python development Editing, debugging, interactive development and source control for Python.	(3)	Node.js development Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.	
Window	vs (3)			
	.NET desktop development Build WPF, Windows Forms, and console applications using C=, Visual Basic, and F= with .NET Core and .NET	ţ	Desktop development with C++ Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.	

Εικόνα 2. 5-Επιλογή .Net Desktop Development

Περιμένουμε να ολοκληρωθεί η λήψη

÷	Visual Studio Installer	
	Visual Studio Community 2017 15.9.12 Update available 15.9.16 View details	Update Launch More 🔻
	Visual Studio Community 2019 Downloading and verifying: 54 MB of 1.12 GB 4% Installing: package 12 of 311 1% Microsoft.VisualStudio.Initializer Start after installation	Pause (2 MB/sec)

Εικόνα 2. 6-Εμφάνιση προόδου εγκατάστασης

Μόλις ολοκληρωθεί η λήψη μας ζητάει να κάνουμε είσοδο στον λογαριασμό μας στην Microsoft. Επιλέγουμε Not now, maybe later.



Εικόνα 2. 7-Σύνδεση στο λογαριασμό της Microsoft



Εικόνα 2.8-Επιλογή εμφάνισης

3-Δημιουργία Εφαρμογής

3.1 Περιγραφή της εφαρμογής-Αλγόριθμος

Το παιχνίδι θα ξεκινά με επιλογή της κατηγορίας των εικόνων που θα χρησιμοποιηθούν στο παιχνίδι μας. Έπειτα θα εμφανίζεται ένα πάνελ με κρυφές εικόνες από την κατηγορία που επιλέχθηκε. Ο χρήστης θα επιλέγει δύο τυχαίες εικόνες κάθε φορά. Οι εικόνες αν είναι ίδιες θα απενεργοποιούνται και θα μένουν ανοιχτές, ενώ αν δεν είναι ίδιες θα μένουν για μια χρονική στιγμή ανοιχτές και στη συνέχεια θα κρύβονται. Το σημείο αυτό βοηθάει το παιδί-χρήστη να απομνημονεύσει την εικόνα έτσι ώστε σε επόμενο κύκλο να κάνει πιο σωστή επιλογή. Τα ίδια βήματα επαναλαμβάνονται μέχρι να αποκαλυφθούν όλες οι εικόνες ή μέχρι να τελειώσει ο χρόνος που έχει στη διάθεσή του ο παίκτης. Με το τέλος του παιχνιδιού εμφανίζουμε ανάλογα κάποια έξοδο. Πιο συγκεκριμένα αν βρεθούν όλες οι εικόνες θα εμφανιστεί ένα σύντομο αρχείο πολυμέσων , ενώ σε διαφορετική περίπτωση θα εμφανίσουμε μόνο ένα ενημερωτικό μήνυμα για το τέλος του χρόνου.

Ακολουθεί ένα διάγραμμα ροής (What is a DFD, χ.η) του προγράμματος.



3.2 Δημιουργία εφαρμογής

3.2.1 Δημιουργία νέου project





Επιλέγουμε Windows Forms App όπως φαίνεται στην επόμενη εικόνα. Αν η επιλογή δεν εμφανίζεται προσέχουμε να είναι τα πρώτα στοιχεία (κυκλωμένα) επιλεγμένα. Το πρόγραμμα θα δημιουργήσει αυτόματα την κλάση Form1 όπου θα υπάρχει το κυρίως παιχνίδι.

i i i	Create a new project	Search for templates (Alt+5)
	Recent project templates	C# • Windows
	Ti Windows Forms App (.NET Core) C#	WPF App (.NET Core) Windows Presentation Foundation client applicatio C# Windows Desktop
		WPF Custom Control Library (.NET Core) Windows Presentation Foundation custom control C# Windows Desktop Library
		WPF User Control Library (.NET Core) Windows Presentation Foundation user control libr C# Windows Desktop Library
		Windows Forms App (.NET Framework) A project for creating an application with a Window C# Windows Destron
		E ^{C®} Windows Service (.NET Framework) A project for creating Windows Services

Εικόνα 3. 2-Επιλογή project

Δίνουμε όνομα και επιλέγουμε create.

3	comigate year new project		
	Windows Forms App (.NET Core) C# Windows Desktop		
Ē	Project name		
	EducationalGame		
	Location		
	C:\Users\user\source\repos		
	Solution name 🕕		
	✓ Place solution and project in the same directory		
		×	

Εικόνα 3. 3-Όνομα project

3.2.2 Εισαγωγή όλων των απαραίτητων στοιχείων

Βάζουμε ένα αντικείμενο TableLayoutPanel (TableLayouPanel Class, χ.η) για την απαραίτητη στοίχιση των labels με τις εικόνες με σύρσιμο και εναπόθεση (drag and drop) ή με διπλό κλικ (εικόνα 3).

×	File	Edit View	Proje	t Build	Debug	Test	Analyze	Tools	Extension	s Window	Help	Search (Ctrl+C		
3	0 - 0	18 - 🖕 🖌		9.0	- Release	•	Any CPU	•	► Start -	# = 1 +	I= ±	二 〒 + 山		
s					_					-	•			
erve	Toolbox				▼ +⊐ X									
Ēx	Search	Toolbox			ρ-									
plor	×	PropertyGrid			-						_			
ę	0	RadioButton												
Too		RichTextBox												
box		SaveFileDialog												
D	-	SerialPort												
ata	*œ	ServiceControl	ServiceController											
our		SplitContainer												
ES.	€ ≯	Splitter												
	here	StatusStrip												
	Bb .	TabControl												
		TableLayoutPa	nel											
	abl	TextBox		TableLayo	utPanel	licros	oft Corporati		1					
	٢	Timer		.NET Com	ponent	neros	on corporati	on						
		ToolStrip												
		ToolStripConta	iner	Handles t	he layout of mat of a tab	its co e auto	mponents ar matically.	nd arrang	ges them					
	1	ToolTip							_					
	⊡—	TrackBar												
	12	TreeView												
		VScrollBar												
	9	WebBrowser												
	⊿ Com	mon Controls												

Εικόνα 3. 4-Εισαγωγή TableLayoutPanel

Εμφανίζεται ένα tablelayout 2x2 που είναι το προεπιλεγμένο (default) και εμείς πρέπει να το κάνουμε 4x4. Αυτό γίνεται από αντίστοιχες επιλογές ColumnCount και RowCount στις ιδιότητες (Properties) όπου ορίζονται και τα 2 ίσα με 4

ew	Project	Build	Debug	Test	Analyze	Tools	Extensions	Window	Help	Search (Ctrl+C	2)	P	Educational	Game
•	P 9	• (" •	Release	•	Any CPU	•	▶ Start -	* •	= +	= 〒 ++ 山	* <u>T</u> ^{K #} 1 <u>0</u>	() =:	u ⁿ % =	
For	n1.cs [Desi	gn]* ⊹⊧	×											- Sol
						000000000		0.0	E					G See
								O						Pro tal

Εικόνα 3. 5-Ορισμός γραμμών και στηλών(1)

Στην ίδια εικόνα κάνουμε δεξί κλικ και επιλέγουμε Edit Rows and Columns

×	File	Edit	View	Project	Build	Debug	Test	Analyze	Tools	Exten	sions	Window	Help	Searc
	- (o 🏠 ·	- 👛 💾	19	• 🥲 •	Release	/	Any CPU	• 1	Star	t - ,	- * *	I= ±	= 11
Serve	Form	1.cs*	Form	m1.cs [Desi	gn]* -⊧	×								
r Explorer		- Form1												, 0
Toolbox														
Data Sources							Q	•	O			Þo		
							q	I		<>	View	Code		
										С. %	Bring Send	To Front To Back		
										+-	Align	To Grid		
										₿	Lock	Controls		
											Row			- •
											Colu	mn	1	•
											Selec	t	oiumns	•

Εικόνα 3. 6-Ορισμός γραμμών και στηλών (2)

Όπου ορίζουμε το πλάτος κάθε στήλης και κάθε γραμμής να είναι το 25% του TableLayoutPanel, όπως φαίνεται στην επόμενη εικόνα.

Project Build Debug	Format	Test Analyz	e Tools	Extensions	Window	Help	Search (Ctrl+Q)
📲 🔈 🗸 🖓 🖌 Release		PU 👻	► Start -	🏓 🗸 🕴 🕇	= = =		<u> </u> ★ <u>T</u> 23 =
m1.cs [Design]* 😐 🗙							
	Column	and Row Styles					
	Show:	Columns				~	Size Type
	Memb	Columns					O Absolute
	Colum Colum Colum	nn1 Percent nn2 Percent nn3 Percent	25,00% 25,00% 25,00%				Percent AutoSize
	Colum	nn4 Percent	25,00%				
							 Column and row span If you want a control to <u>columns</u>, set the RowSp properties on the contr
							 Alignment and stretch If you want to <u>align a co</u> you want a control to s control's Anchor prope

Εικόνα 3. 7-Ορισμός πλάτους στήλης

Στο επόμενο βήμα θα εισάγουμε τα 16 Labels στο TableLayoutPanel που μόλις φτιάξαμε.

X	File Edit View Project Build Debug ⊙ - ⊙ ∰ - ☆ 💾 🖓 ♡ - ♡ - Releas.	Test Analyze Tools Extensions Window Help → Any CPU → Start → 🗗 _ 🗄 + 🗠 ±
Server Explorer Toolbox Data Source	O Image: Control of the second of the sec	Any CPU ► Start - = = = = = = = = = = = = = = = = = =
	Containers FlowLayoutPanel TableLayoutPanel	

Εικόνα 3. 8-Εισαγωγή Labels

Για να καλυφθεί όλο το κελί του TableLayoutPanel από το κάθε Label φροντίζω να αλλάξουμε την κατάλληλη ιδιότητα όπως φαίνεται παρακάτω:



Εικόνα 3. 9 - Ιδιότητα Dock σε Label



Αλλάζουμε και το περίγραμμα του Label από την ιδιότητα BorderStyle

Εικόνα 3. 10- Αλλαγή στυλ περιγράμματος

και αντιγράφουμε το Label του TableLayout έτσι ώστε προκύπτει το επόμενο αποτέλεσμα.



Εικόνα 3. 11-Εμφάνιση αποτελέσματος label panel

Στην ίδια εικόνα θα αφαιρέσουμε τις ετικέτες Label1-Label16.Επειδή θέλουμε κάθε label να ανταποκρίνεται σε κάθε κλικ που γίνεται κατά την εκτέλεση του παιχνιδιού θα πρέπει να ορίσουμε ποια συνάρτηση θα καλείται με το γεγονός (event) αυτό. Στην επόμενη εικόνα ορίσαμε ότι θα καλείται η συνάρτηση ForEveryLabelClick. Όλα τα labels θα καλούνε την ίδια αυτή συνάρτηση (C# Label Control, 2019). Αυτό έγινε στο παράθυρο properties και στην ιδιότητα Click



Εικόνα 3. 12-Ορισμός συνάρτησης που θα καλείται σε κάθε event

3.2.3 Δημιουργία λίστας κατηγοριών

Καλό θα ήταν στην εκκίνηση του παιχνιδιού να εμφανίζεται μία λίστα από εικόνες όπου θα επιλέγουμε την κατηγορία του παιχνιδιού. Ανάλογα με την κατηγορία που θα επιλέγεται θα «φορτώνονται» και οι ανάλογες εικόνες στο παρασκήνιο του παιχνιδιού.

Για να γίνει αυτό θα πρέπει να εμφανίσουμε μια ξεχωριστή φόρμα με την εκκίνηση.



Εικόνα 3. 13-Προσθήκη νέου item

και επιλέγω windows form όπου θα δημιουργήσουμε μία Form2 όπου θα εμφανίζονται οι κατηγορίες.

oject	Build Debug Test Analyze Tool	Is Extensions Window Help Search (Ctrl+Q)	
19	- 🤍 - Release - Any CPU -	🔹 🕨 Start • 🎜 🚽 🔚 🌾 🗐 🖆 🗍 🕷 🐐	Ŧ
cs	Add New Item - EducationalGame		
/*	Installed	Sort by: Default	
h epc ton p * */	✓ Visual C# Items Code	Ciass Class	Vis
0 refere	Data General	••• Interface	Vis
priva { f	Windows Forms WPF	E Form (Windows Forms)	Vis
{	▷ Online	User Control (Windows Forms)	Vis
		Component Class	Vis
		User Control (WPF)	Vis
		About Box (Windows Forms)	Vis
		ADO.NET Entity Data Model	Vis
		Application Configuration File	Vis
found		Application Manifest File	Vis

Εικόνα 3. 14-Επιλογή φόρμας ως νέου item

Από την εργαλειοθήκη toolbox επιλέγουμε με τον ίδιο τρόπο, όπως προηγουμένως, ένα αντικείμενο ImageList το οποίο εμφανίζεται κάτω από την φόρμα όπως φαίνεται στην επόμενη εικόνα.



Εικόνα 3. 15-Εισαγωγή Image List

Τώρα θα πρέπει να εισάγουμε και ένα στοιχείο ImageList (Chand, 2011)

i G	••• 🔁 • 🖕 🖬 📲 🤊 • 🤆	• Release • Any C	CPU 🔹 🕨 Start s
Serve	Form2.cs [Design]* + X Form1.cs	Program.cs	Form1.Designer.cs
r Explorer	P Form2	- • ×	
Toolbox	Q	Po	
Data Sources	0		

Εικόνα 3. 16-Εισαγωγή στοιχείου ListView

Πάμε να εισάγουμε τις εικόνες.



Εικόνα 3. 17-Εισαγωγή εικόνων σε Image List

Με την επιλογή add της επόμενης εικόνας επιλέγουμε τις 3 εικόνες για τις κατηγορίες.

Program.cs	Form1.Designer.cs	Form1.cs [Design]		
	Form1.Designer.cs	Form1.cs [Design]	her B >	oes.jfif properties:

Εικόνα 3. 18-Επιλογή αρχείων εικόνων Image list

Θα πάμε στο ListView να προσθέσουμε τις εικόνες που έχουμε τοποθετήσει στο ImageList (Adding Images and Listview, χ.η). Πάμε στο αντικείμενο ListView στο βελάκι πάνω δεξιά (smart tag) και επιλέγουμε όπως παρακάτω.

R	File Edit View Project	Build Debug	Format	Test Analyze	Tools Extensions
	3 - 0 🕅 - 🍰 💾 🗳 🤊	• C • Release	 Any C 	:PU 🔹 🕨	• Start • 🔎 🚽 🕇 🕇
Serve	Form2.cs [Design]* 😐 🗙 Form1	l.cs Program	n.cs	Form1.Designer	.cs Form1.cs [D
er Explorer	Form2		• 🛛		
Toolbox	o	ListView Tasks			
Data Sour	0	Edit Columns Edit Groups	List		ī
Ces	0	Small ImageList:	imageList	1 v	
		Dock in Parent Co	ontainer		

Εικόνα 3. 19-Ορισμός ListView item

Επιλέγουμε Edit Items από το ListView και στο παράθυρο ListViewItem Collection Editor κάνουμε Add όπου αυτόματα μπαίνει το 1° στοιχείο της λίστας. Στο δεξιό υποπαράθυρο στην ιδιότητα Behavior ->ImageIrdex επιλέγουμε την εικόνα. Κάνουμε τα ίδια βήματα από το Add και κάτω και για τις άλλες δύο εικόνες.



Εικόνα 3. 20- Προσθήκη εικόνων σε ListView

Θα προσθέσουμε και τα ονόματα των κατηγοριών δίπλα από τις εικόνες τους.



Εικόνα 3. 21- Προσθήκη text στις κατηγορίες

Θα πάμε στο αρχείο Form1.cs και θα φτιάξουμε μια συνάρτηση που θα μας εμφανίζει την φόρμα επιλογής κατηγοριών.

```
private void selectcategory()
{
    Form2 formaMeKatigories = new Form2();
    if (formaMeKatigories.ShowDialog(this)==DialogResult.OK)
    {
       ListViewItem epilegmenoitem =
formaMeKatigories.listView1.Items[formaMeKatigories.listView1.SelectedIndices[0]];
       MessageBox.Show(epilegmenoitem.Text);
    }
}
```

Και θα την καλέσουμε μέσα στον constructor της form1

3.001		
100	1.1	
100	100	
1.1	1.1	
	1.1	
100		
198	1.0	
		//constuctor
12.5	100	// conscuccor
		4
199		Treference
212		1.1.1
1213	1.1	Dublic Forml()
T 11	110	
1.1	- 191	

Εικόνα 3. 22-Κλήση συνάρτησης selectcategory()

Τρέχουμε δοκιμαστικά το πρόγραμμα για να δούμε εάν η επιλογή μας μεταφέρεται στην form1 μετά την επιλογή κατηγορίας και το ΟΚ από την form2.

👻 Form2	-	 X		
-	ου Ζωάκια Μι Κούκλες		×	
*	Нрыес		Ηρωες	
	Ok			

Εικόνα 3. 23-Επιβεβαίωση ορθής κλήσης

3.2.4 Αρχικοποίηση παιχνιδιού

Στην συνέχεια θα εμφανίσουμε το αρχικό σημείο του παιχνιδιού όπου όλες οι εικόνες είναι κρυφές ακόμα. Για να μην εμφανίζεται ένα γκρί υπόβαθρο για κάθε label, αποφασίσαμε να γεμίσουμε τα labels με το ίδιο εικονίδιο από την κατηγορία που επιλέξαμε. Για παράδειγμα αν έχει επιλεγεί η κατηγορία ζωάκια όλα τα labels θα πάρουν την εικόνα animals.gfif. Αυτό φαίνεται στο παρακάτω στιγμιότυπο κώδικα.

```
private void arxikopoihshLabels()
{
    foreach (Control control in tableLayoutPanel1.Controls)
    {
        Label templabel = control as Label;//kano to control se label
        if (selectedCategory == 0) //an exei epilexthei h kathgoria zwakia
        {
            Image eikona =
        Image.FromFile(@"C:\Users\\user\\Desktop\\EducationalGame\\animals.jfif");
        Bitmap bmpImage = new Bitmap(eikona, new Size(100, 100));
        templabel.Image = bmpImage;
    }
}
```

```
File
         Edit
              View
                     Project Build Debug Test Analyze
                                                         Tools Extensions Window
                                                                                  Help
🖸 🗸 🗇 🔯 📲 💾 🏓 ಶ 🗸 🖓 🖌 Release 🕞 Any CPU
                                                            🔸 🕨 Continue 👻 🎵 📮 🔢 🔳 🔕
Process: [6004] EducationalGame.exe 🔹 💽 Lifecycle Events 👻 Thread:
            Form1
Form2.cs
C# Educationa
    42
    43
    44
    45
    46
    47
    48
    49
    50
    51
    52
    53
    54
    55
    56
    57
    58
    59
    60
    61
    62
    63
```

Εικόνα 3. 24-Εμφάνιση πάνελ με κρυφές εικόνες

Ο χρήστης θα βλέπει αρχικά 16 ίδια εικονίδια από την κατηγορία που επέλεξε. Κάνοντας κλικ σε κάποιο label ουσιαστικά ξεκινάει το παιχνίδι όπου θα πρέπει να φορτώνεται κάποια εικόνα, έπειτα με το επόμενο κλικ να φορτώνεται η επόμενη εικόνα και αν είναι ίδιες τότε θα μένουν ανοιχτές αλλιώς «θα κλείνουν». Για να φορτώνεται η αντίστοιχη εικόνα από την κατηγορία που επιλέχθηκε θα βάλουμε tags σε κάθε label. Τα tags θα έχουν τιμές από 0..15. Κάνοντας κλικ σε ένα label κρατάω το αντίστοιχο tag. Στη επόμενη εικόνα επιλέξαμε το label στην θέση (4,1) και εμφανίστηκε το tag=12 που χρειάζομαι για την συνέχεια.



Εικόνα 3. 25-Αρίθμηση Label

Με τον ακέραιο αυτόν θα πάω σε μία λίστα ακεραίων (mylist) στην tag θέση. Η τιμή από αυτή τη θέση στην mylist περιέχει έναν ακέραιο από 0..7, γιατί σε κάθε παιχνίδι φορτώνονται 8 εικόνες. Προφανώς η mylist περιέχει 8 ζευγάρια ίδιων τιμών στο διάστημα 0..7. Για να γίνει πιο κατανοητό αυτό δείχνουμε ένα στιγμιότυπο της mylist.



Εικόνα 3. 26-Πίνακας παρασκηνίου με αρίθμηση Labels

Στο 1° και στο 2° κελί της δεύτερης γραμμής του tableLayoutPanel με τα Labels έχουν αντιστοιχηθεί οι εικόνες με id=5, που σημαίνει ότι θα τις «αντλήσω» από τον πίνακα ImagesFolderPath από το κελί 4.Ο πίνακας αυτός περιέχει τις διαδρομές αρχείων (paths) για τις εικόνες του παιχνιδιού. Άρα ίδιο path ίδιες εικόνες.

3.2.5 Εισαγωγή εικόνας υποβάθρου φόρμας (Background image)

Με επιλεγμένη την φόρμα εντοπίζω από τις ιδιότητες στο στοιχείο BackgroundImage όπως φαίνεται στην εικόνα



Εικόνα 3. 27-Ορισμός εικόνας background



Επιλέγουμε Local Resourse και import και στο νέο παράθυρο επιλέγουμε την εικόνα

Εικόνα 3. 28-Επιλογή εικόνας

Για να προσαρμοστεί η εικόνα που επιλέξαμε στο μέγεθος της φόρμας επιλέγουμε από την ιδιότητα BackgroundImageLayout την τιμή Stretch. Μετά από αυτά εμφανίζεται το παρακάτω αποτέλεσμα.



Εικόνα 3. 29-Ιδιότητα Stretch

3.2.6 Εισαγωγή μπάρας κύλισης χρόνου

Θα ήταν καλό να εμφανίζεται και ο υπολοιπόμενος χρόνος λήξης του παιχνιδιού όχι σε μορφή αριθμών αλλά με έναν πιο γραφικό τρόπο. Την λύση αυτή μας την δίνει το εργαλείο progressBar από την εργαλειοθήκη Toolbox του VisualStudio (C# ProgressBar Control, χ.η).



Εικόνα 3. 30-Εισαγωγή μπάρας προόδου.

Στην συνέχεια θα ορίσουμε την μέγιστη τιμή και το βήμα με το οποίο θα μετακινείται η μπάρα κύλισης χρόνου. Θέλοντας να δώσουμε χρονικό περιθώριο 2 λεπτών για κάθε παιχνίδι θέσαμε maximum=120, minimum=0 και step=1.



Εικόνα 3. 31-Ορισμός παραμέτρων progress bar

Για να μπορεί να φαίνεται η πρόοδος σε κάθε ένα δευτερόλεπτο θα προσθέσουμε και έναν χρονομετρητή (timer) στην εφαρμογή μας. Ο χρονομετρητής αυτός θα εκτελεί έναν «χτύπο» κάθε 1000ms και θα ενημερώνει την μετακίνηση της μπάρας. Η εισαγωγή του timer γίνεται από την ίδια εργαλειοθήκη και θα αλλάξουμε και την ιδιότητα Interval του timer.



Εικόνα 3. 32-Εισαγωγή timer

Παρακάτω στην ενότητα που αφορά την συγγραφή κώδικα θα δώσουμε τις εντολές που θα εκτελούνται έτσι ώστε να αποκτήσει την συγκεκριμένη λειτουργικότητα η μπάρα χρόνου.

3.2.7 Εμφάνιση αρχείου- πολυμέσου επιβράβευσης.

Δημιουργούμε μία φόρμα έτσι όπως ξέρουμε και θα προσθέσουμε ένα στοιχείο ελέγχου που ονομάζεται Windows Media Player το οποίο παίρνουμε από την εργαλειοθήκη (toolbox). Σε περίπτωση που δεν εμφανίζεται το εργαλείο αυτό θα το προσθέσουμε σύμφωνα με τις εικόνες που ακολουθούν.



Εικόνα 3. 33-Επιλογή νέου item για την εργαλειοθήκη

Build Debug Test	Analyze Tools Extensions	Window	Help	Search (Ctrl+Q)	P
	Any CPU - Start -	🟓 = 🖁 = 1	= +	コ 〒 井 山 1*1 工 55 風	11 2
3.cs Form3.cs [Design] 🗢 🗙 Form2.cs	Form2.cs [Desig	in]	Form1.cs* Form1.cs [D	esign]
1	r		-		
n Forms	Choose Toolbox Items				
	.NET Framework Components	COM Compon	ents	Universal Windows Components W	/PF Co
	Name		Path		Libr
	System Monitor Control		C:\WINDOWS\System32\sysmon.ocx C:\Windows\SysWOW64\tdc.ocx		
	TaskSymbol Class		C:\WI	NDOWS\system32\mmcndmgr.dll ndows\SysWOW64\qdvd.dll	Noo
	VocabCtl Class		C:\Pro	ogram Files (x86)\Common Files\M ogram Files (x86)\Common Files\M	pkn
	VSTO WinFormsHost Cont	trol	c:\Pro C:\Wii	gram Files (x86)\Common Files\M ndows\SysWOW64\inetcomm.dll	Mic
	Windows Media Player		C:\WI	NDOWS\system32\wmp.dll	Win
	Windows Store Remote De	esktop Client	C:\WI	NDOWS\system32\mstscax.dll	Mic
	WorkspaceBrokerAx Class		C:\WI	NDOWS\system32\wkspbrokerAx.dll	Wo
	Γραμμή συνδέσεων		C:\PR	OGRA~2\MICROS~2\Office10\FPD	FPC
	:-) VideoSoft FlexArray Contr Language: Langu	ol uage Neutral			

Εικόνα 3. 34-Επιλογή Windows Media Player

Τώρα εμφανίζεται κανονικά στην εργαλειοθήκη μας από όπου την επιλέγουμε και την εναποθέτουμε στην Form3 που δημιουργήσαμε. Μένει μόνο να παραμετροποιήσουμε την φόρμα μας και να ορίσουμε ποιο αρχείο πολυμέσων (βίντεο) θα αναπαράγεται από αυτή.

ew	Project	Build	Debug	Format	Test	Analyze	Tools	Extensions	Window	Help	Search (Ctr
•	1 9 5	• • •	Release	e 🕶 Any	CPU	•	Start -	- 🞜 🚽 🖁 🕇	= + -	=	<u> </u> * <u> </u>
- џ	× Fo	orm3.cs*	For	m3.cs [Desi	gn]* -⊧	× Form	n2.cs	Form2.cs	[Design]	Form	11.cs*
×	-	- Form	3		Win	dows Med	ia Player F	Properties			
ia Play	/er				Ger	neral Adva	anced				
		0				Source					
						File Name	e or URL:	C:\Users\us	er\Desktop\f	ile_examp	le_WMV_480_
					-	Controls la	ayout				
						Select a n	node:	None	~		
						Player is e displayed	mbeddeo	d without cont	rols, and with	n only the	video or visua
		Ċ			-	Playback o	options		Volume sett	ings	
						Auto st	tart		Mute		
						Stretch	to fit		Volume:	Min	
						Play fu	ll screen		Palance		
						Play coun	t	· ·	Datafice:	L	1.1.1.1.1.1

Εικόνα 3. 35-Ορισμός ιδιοτήτων media player

3.3 Συγγραφή πηγαίου κώδικα

3.3.1 Προσθήκη λειτουργικότητας για την μπάρα κύλισης χρόνου

Θα παραθέσουμε ένα τμήμα από τον πηγαίο κώδικα που γράψαμε για την εφαρμογή μας. Ένα μεγάλο τμήμα κώδικα προστίθεται έτοιμο με τη δημιουργία του έργου και εμείς θα προσθέσουμε το κεντρικότερο και ουσιαστικότερο κομμάτι που αφορά την λειτουργικότητα του. Επίσης θα δούμε και πρακτικά ομοιότητες που έχει ο κώδικας της C# με τον κώδικα της Java, της C και της C++.

} }

3.3.2 Προσθήκη λειτουργικότητας σε κάθε επιλογή της εικόνας-Label

```
private void GiakatheLabelClick(object sender, EventArgs e)
        {
            if(timer2started! = true)
            {
                timer2.Start();
                timer2started = true;
            }
            Label clickedLabel = sender as Label;
            if (prwtoClick == -1 && deuteroClick == -1)
            {
                FirstLabel = clickedLabel;
               int position = Int32.Parse(clickedLabel.Tag.ToString());
                prwtoClick = mylist[position];
                string A = ImagesFolderPath[prwtoClick];
                Image eikona = Image.FromFile(@A);
                Bitmap bmpImage = new Bitmap(eikona, new Size(100, 100));
                FirstLabel.Image = bmpImage;
            }
            else if (prwtoClick != -1)
            {
                SecondLabel = clickedLabel;
                int position = Int32.Parse(clickedLabel.Tag.ToString());
deuteroClick = mylist[position];
                string A = ImagesFolderPath[deuteroClick];
                Image eikona = Image.FromFile(@A);
                Bitmap bmpImage = new Bitmap(eikona, new Size(100, 100));
                SecondLabel.Image = bmpImage;
                if (prwtoClick == deuteroClick)
                {
                    FirstLabel.Enabled = false;
                    SecondLabel.Enabled = false;
                    prwtoClick = -1;
                    deuteroClick = -1;
                }
                else
                {
                    timer1.Start();
                }
            }
        }
```

3.4 Δημιουργία αρχείου εγκατάστασης της εφαρμογής

Αφού κατασκευάσαμε την εφαρμογή μας θα πρέπει να μπορούμε να την διαμοιράσουμε με εύκολο τρόπο. Ως εδώ αν θα θέλαμε να δώσουμε την εφαρμογή μας σε κάποιον άλλον για να τη δοκιμάσει στον υπολογιστή του θα έπρεπε πέρα από το εκτελέσιμο αρχείο να τοποθετήσουμε και όλα τα υπόλοιπα αρχεία πολυμέσων στις σωστές

διαδρομές και με τα σωστά ονόματα, πράγμα που θα καθιστούσε δύσχρηστη την εφαρμογή. Εμείς θέλουμε μία εφαρμογή που να εγκαθίσταται εύκολα.

Τη λύση αυτή μας δίνει το visual studio με την κατασκευή ενός setup project το οποίο θα εσωκλείει σε ένα αρχείο εγκατάστασης (setup) όλη την εφαρμογή και μέσω βημάτων θα την εγκαθιστά στον υπολογιστή και θα δημιουργεί μία συντόμευση στην επιφάνεια εργασίας (Deployment Examles, 2016).

Visual Studio ct Debug Team Tools Test Analyze Window Help 19-9-1 🕨 Attach... - 👂 -New Project ▶ Recent • # E Sort by: Default ▲ Installed Setup Project 1 Visual Studio Installer arte Visual C# Visual Basic Web Setup Project Visual Studio Installer ♦ Visual C++ in 5 min JavaScript Merge Module Project Visual Studio Installer 1 ductivity ▲ Other Project Types Visual Studio Installer the newe Visual Studio Solutions tes CAB Project Visual Studio Installer fully-native D Online t Not finding what you are looking for? hal\sour Open Visual Studio Installer alat Name Setup1

Θα δημιουργήσουμε αρχικά ένα setup wizard project

Εικόνα 3. 36-Επιλογή setup wizard(1)



Εικόνα 3. 37- Επιλογή setup wizard(2)



Εικόνα 3. 38-Επιλογή τύπου αρχείου εγκατάστασης.

Στο επόμενο βήμα δεν θα προσθέσουμε κανένα αρχείο. Μόλις δημιουργηθούν οι φάκελοι του project θα προσθέσουμε αρχεία και φακέλους με σύρσιμο και εναπόθεση (drag and drop)

feam Tools Test	t Analyze Window Help
•	- 🕨 Attach 📁 📮
	Setup Wizard (3 of 4)
	Choose files to include
	You can add files such as Readme files or HTML pages to the setup.
	Which additional files do you want to include?
ese tips and tricks fo	
iologies to deploy be	
oid and iOS apps	
s\Setup1	

Εικόνα 3. 39-Δυνατότητα προσθήκης αρχείων



Εικόνα 3. 40-Επιλογή του Application Folder για προσθήκη αρχείων

Στο σημείο αυτό θα ανοίξουμε το φάκελο Application Folder όπου και θα επικολλήσουμε όλα τα αρχεία και τους φακέλους της εφαρμογής μας.

					PDF			
ManaP	i i i i i i i i i i i i i i i i i i i	EducationalGame Share View	- C	× ^ 🔞	Test	Action Analyze	Window	Help
0	Pin to Quick Copy	Paste	New Open	ielect f	ault	Attach	•• • • • • • •	Type
Reeye	Clipboz ← → ∽ ↑	ard Organise	√ ⊘ Se	arch Ed 🔎				.,,,,,
Dyna	^ ≹ Quick	Name		Date m				
Sen	OneD	bin bin		06/03/2				
	Picto	Properties		06/03/2				
Micro Ed	💻 This P	App.config		31/10/2 21/10/2				
	3D C	ARROW.WAV BOMB.WAV		10/09/2				
	Doc	CHIMES.WAV		10/09/2				
Chro	Dow Mus	C# EducationalGame.csproj		24/02/2	_			
	Picti	Form1.cs		06/03/2		• 2 2 3	- <u>1</u>	
Katan		Form1.Designer.cs		21/02/2				

Εικόνα 3. 41-Επιλογή αρχείων για προσθήκη στο setup

Οι φάκελοι και τα αρχεία προστέθηκαν όπως φαίνεται στην επόμενη εικόνα.

🗐 ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ2.de	oc - Microsoft Word		
<u>Α</u> ρχείο Επεξερ <u>ι</u>			<u>'0 D '0</u>
🖸 (🔤 🗹 🚽 🚽	EducationalGame	- 🗆 X	
L File Home	Share View	^ 🕐	Test Action Analyze Wi
Pin to Quick Copy	Paste	New Open Select	Jlt → Attach → j
access			Name
Clipboa	rd Organise		CHIMES.W
$\leftarrow \rightarrow \bullet \uparrow$	→ Mari → Education	V 🖸 Search Ed 🔎	
	~		
- Ouick	Name	Date m. "	i animals ifi
A Color	.VS	06/03/2	
ConeD	📙 bin	06/03/2	bin
· Doc	📙 obj	06/03/2	🚞 obj
- Pict	Properties	06/03/2	🚞 Properties
	animals.jfif	31/10/2	COIN.WAV
📜 📃 This P	App.config	21/10/2	📄 Education
🗍 🧊 3D C	ARROW.WAV	10/09/2	📄 Education
- Desl	BOMB.WAV	10/09/2	Form1.cs
- B Doc	CHIMES.WAV	10/09/2	Form1.Des
· Doc	COIN.WAV	10/09/2	Form1.res
· • • • • • • •	C# EducationalGame.csproi	24/02/2	
Mus	EducationalGame.sln	21/10/2	
E Picti	Form1.cs	06/03/2	

Εικόνα 3. 42-Εικόνα μετά την προσθήκη

Στη συνέχεια θα πούμε στο πρόγραμμα να δημιουργεί και μία συντόμευση της εφαρμογής μας στην επιφάνεια εργασίας του υπολογιστή στον οποίο εγκαθιστούμε την εφαρμογή μας.

🛃 ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ2.α	doc - Microsoft	Word			
Δρχείο <u>Ε</u> πεξερ Δ 😅 🖬 🔒 File	MyGameSetu Edit View	p - Microsoft Visual S Project Build	tudio Debug Team C - Debug	Tools Test	Action An
Server Explorer Toolbox	File System (My File System > Derical Userical Userical Userical Contemportation Con	(GameSetup) + × n on Target Machine ation Folder Deskton Add Create Shortcut to U Cut Paste Delete Properties Window	Jser's Desktop	Ctrl+X Ctrl+V Del F4	N

Εικόνα 3. 43-Επιλογή δημιουργίας συντόμευσης

Εδώ θα πρέπει να ορίσουμε το αρχείο προς το οποίο θα οδηγεί η συντόμευση μας. Αυτό το αρχείο προορισμός θα είναι προφανώς το εκτελέσιμο αρχείο EducationaGame.exe που δημιουργείται μέσα στο φάκελο debug.

soft Visual Studio tt Build Debug Team	Tools Test Act	ion Analyze Window	v Help	
up) → ×	Derault			
t Machine ler Menu	Select Item in Pro	Name oject Debug /MPLib.dll Game.exe Game.exe.config Game.pdb PLib.dll	Type	
ild	Source path: Files of type:	C:\Users\mihal\Deskt	op\MariaPtyxiaki\Educati	ona V

Εικόνα 3. 44-Ορισμός αρχείου προορισμού συντόμευσης

Κάνουμε built στο έργο μας και το αρχείο εγκατάστασης msi δημιουργείται στο φάκελο που ορίσαμε. Θα τον αντιγράψουμε στην επιφάνεια εργασίας και θα επιλέξουμε να το εγκαταστήσουμε.



Εικόνα 3. 45-Εκκίνηση δοκιμαστικής εγκατάστασης

	🛃 MyGameSetup
	Select Installation Folder
RALIFICED COSA- DALABABE	
an a	The installer will install MyGameSetup to the following folder.
	To install in this folder, click "Next". To install to a different folder, enter i
daster church	<u>F</u> older:
Linear Santas Alle	C:\Program Files (x86)\Common Files\
Santas Santas Santas	
Datavena, carriedor, juntor	Install MyGameSetup for yourself, or for anyone who uses this comput
B 🧏 🛸	Everyone

Εικόνα 3. 46-Επιλογή φακέλου εγκατάστασης



Εικόνα 3. 47-Ολοκλήρωση εγκατάστασης

Στην επιφάνεια εργασίας βλέπουμε το αρχείο MyGameSetup.msi και την συντόμευση που δημιουργήθηκε



Εικόνα 3. 48-Εμφάνιση συντόμευσης και αρχείου εγκατάστασης.

Κεφάλαιο 4- Συμπεράσματα

Ξεκινώντας την παρούσα εργασία με βάση τη γλώσσα C# περιμέναμε, όσον αφορά το προγραμματιστικό κομμάτι, να χρησιμοποιήσουμε όλα όσα γνωρίζαμε από τη γλώσσα C και τη C++ όπως και έγινε. Δομές επανάληψης και ελέγχου καθώς και σύνταξη συναρτήσεων και άλλων εντολών είναι σε πολλά σημεία πανόμοιες και πάντα νιώθαμε ότι η συγγραφή γινόταν σε ένα πολύ οικείο περιβάλλον παρότι είναι η πρώτη μας επαφή με τη γλώσσα C#.

Η πραγματική δύναμη όμως της γλώσσας C# φάνηκε στον τρόπο με τον οποίο χρησιμοποιείται μέσα από το περιβάλλον ανάπτυξης του Microsoft Visual Studio. Η πρώτη εικόνα που δίνει το περιβάλλον αυτό είναι μία εικόνα ευταξίας από ένα επαγγελματικό εργαλείο. Το .Net Framework που περιέχεται απογειώνει πραγματικά όποια γλώσσα το χρησιμοποιεί. Από ό,τι είδαμε κατά τη διάρκεια της εγκατάστασης , αλλά και κατά τη διάρκεια της δημιουργίας νέου έργου (project) υπάρχουν και άλλες γλώσσες , όπως η Visual Basic και η Visual C++ καθώς και ένα μεγάλο πλήθος διαφορετικών έργων που μπορούμε να δημιουργήσουμε.

Η συγγραφή κώδικα εκ μέρους μας ήταν απαραίτητη σε αυτό καθαυτό το προγραμματιστικό κομμάτι μιας και δημιουργείται αυτόματα αρκετός κώδικας για τα βασικά μίας εφαρμογής. Για παράδειγμα συναρτήσεις δημιουργίας και απαραίτητες μεταβλητές για τη δημιουργία μιας νέας φόρμας δημιουργούνται αυτόματα.

Εντυπωσιαστήκαμε από την εργαλειοθήκη (toolbox) του περιβάλλοντος όπου επίσης ανακαλύψαμε ότι τα διαθέσιμα εργαλεία δεν είναι μόνο αυτά που βλέπουμε, αλλά υπάρχει η δυνατότητα λήψης και εγκατάστασης ενός μεγάλου πλήθους άλλων. Το χαρακτηριστικό αυτό προσφέρει πραγματικά απεριόριστες δυνατότητες στην κατασκευή προγραμμάτων που να καλύπτουν οποιεσδήποτε απαιτήσεις.

Η διαχείριση πολυμέσων (φωτογραφίες, αρχεία βίντεο και ήχου) γίνεται με μεγάλη ευκολία και ένα πλήθος τύπων (format) είναι απόλυτα συμβατά για να χρησιμοποιηθούν από την εφαρμογή μας.

Ολοκληρώνοντας την κατασκευή της εφαρμογής μας θελήσαμε να βρούμε έναν κομψό τρόπο για να μπορούμε να την διανείμουμε. Τη λύση αυτή μας έδωσε το ίδιο το περιβάλλον ανάπτυξης που παρέχει και το εργαλείο αυτό. Έτσι η εφαρμογή μας είναι

πλέον σε ένα αρχείο εγκατάστασης και μπορούμε εύκολα να την εγκαταστήσουμε και σε άλλους υπολογιστές.

Οι εντυπώσεις που μας άφησε η ενασχόλησή μας με την εφαρμογή αυτή ήταν πολύ θετικές και συνειδητοποιήσαμε ότι χρησιμοποιήσαμε μία συμπαγή γλώσσα προγραμματισμού μέσα σε ένα περιβάλλον το οποίο τις έδινε απεριόριστες δυνατότητες. Ο σκοπός δε της κατασκευής ενός εκπαιδευτικού παιχνιδιού για παιδιά έδωσε και έναν πιο ευχάριστο χαρακτήρα στο σκοπό της εργασίας.

4. ΒΙΒΛΙΟΓΡΑΦΙΑ ΠΗΓΩΝ ΚΑΙ ΕΙΚΟΝΩΝ

Ελληνόγλωσση

Γραφικά Περιβάλλοντα, (χ.η). [Αρχείο pdf]. Ανακτήθηκε από MYY105, Πανεπιστήμιο Ιωαννίνων, Ιωάννινα. Online: http://ecourse.uoi.gr/pluginfile.php/180240/mod_resource/content/0/15_guis.pdf

Πολίτης, Π., &Γιαννόπουλος, Η. (1997). ΣΥΓΧΡΟΝΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ, Προγραμματισμός με τη Visual Basic 4.0 (pp.231-239). Ανακτήθηκε από <u>http://ebooks.edu.gr/modules/ebook/show.php/DSGL-C101/36/198,1055/</u>

Συμβώνης, A. (χ.η). Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #17 [Αρχείο pdf]. Εθνικό Μετσόβειο Πολυτεχνείο, Αθήνα. Online: <u>http://www.math.ntua.gr/~symvonis/teaching/intro_to_object_oriented_programming_material/lectures/handouts_OO-lect_17_GUI_1.pdf</u>

Ξενόγλωσση

Adding Images and ListView. (χ.η). Ανακτήθηκε Ιανουάριος 21, 2020 από <u>https://www.homeandlearn.co.uk/extras/picture-viewer/image-lists-and-listviews.html</u>

Advantages and Disadvantages of Event-Driven Programming. (2019). Ανακτήθηκε Φεβρουάριος 20, 2020, από <u>https://benchpartner.com/advantages-and-disadvantages-of-</u> <u>event-driven-programming/</u>

Aggarval, (χ.η). *Introduction to .NET Framework*. Ανακτήθηκε Δεκέμβριος 28, 2019 από <u>https://www.geeksforgeeks.org/introduction-to-net-framework/</u>

Chant, M. (2011). *ImageList in C#*. Ανακτήθηκε Ιανουάριος 14, 2020, από <u>https://www.c-sharpcorner.com/UploadFile/mahesh/imagelist-in-C-Sharp/</u>

C# Features. (χ.η). Ανακτήθηκε Ιανουάριος 13, 2020, από <u>https://www.javatpoint.com/csharp-features</u>

C# History.[εικόνα] (χ.η). Ανακτήθηκε Ιανουάριος 12, 2020, από <u>https://www.javatpoint.com/csharp-history</u>

C# Label Control. (2019). Ανακτήθηκε Ιανουάριος 12, 2020 από <u>http://www.tutorialspanel.com/csharp-label-control/</u>

C# ProgressBar Control. (χ.η). Ανακτήθηκε Φεβρουάριος 10, 2020 από <u>http://csharp.net-informations.com/gui/cs-progressbar.htm</u>

Deployment Examples. (2016). Ανακτήθηκε Φεβρουάριος 23, 2020 από <u>https://docs.microsoft.com/en-us/cpp/windows/deployment-examples?view=vs-2019</u>

TableLayoutPanel Class. (χ.η). Ανακτήθηκε Φεβρουάριος 16, 2020 από <u>https://docs.microsoft.com/en-</u> us/dotnet/api/system.windows.forms.tablelayoutpanel?view=netcore-3.1

Visual Studio. [Εικόνα]. (2019). Ανακτήθηκε Ιανουάριος 22, 2020 από <u>https://visualstudio.microsoft.com/vs/</u>

What is a DFD. (χ.η). Ανακτήθηκε Ιανουάριος 23, 2020 από https://www.lucidchart.com/pages/data-flow-diagram