ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΤΡΙΣΔΙΑΣΤΑΤΗ ΠΡΟΣΟΜΟΙΩΣΗ ΤΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΥΠΟΛΟΓΙΣΤΗ ΤΩΝ ΑΝΤΙΚΥΘΗΡΩΝ ΚΑΙ JAVA ΜΙΚΡΟΕΦΑΡΜΟΓΗ ΥΠΟΛΟΓΙΣΜΟΥ ΤΩΝ ΕΝΔΕΙΞΕΩΝ ΤΗΣ ΟΠΙΣΘΙΑΣ ΟΨΗΣ ΤΟΥ ΥΠΟΛΟΓΙΣΤΗ ΤΩΝ ΑΝΤΙΚΥΘΗΡΩΝ

Μουστάκας Ιωάννης
Α. Μ. 4/06

Επιβλέπων Καθηγητής
Ρουμελιώτης Μάνος

Εξεταστής Καθηγητής
Γεωργίου Ανδρέας

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή…………………………………………………………....................3
2. Η τρισδιάστατη προσομοίωση του υπολογιστή των Αντικυθήρων.
    2.1 Εισαγωγή………………………………………………...................11
    2.2 Οι νέες θέσεις, διαστάσεις και ο αριθμός των δοντιών στα γρανάζια…………..…………………………………...................12
    2.3 Το γρανάζι του Ίππαρχου γίνεται διαφορικό…………................…14
    2.4 Πρόσθεση του δείκτη με την ένδειξη του κύκλου του Μέτωνα στο καντράν του οπίσθιου μέρους……………..............................18
    2.5 Η επικόλληση των εικόνων που αναγράφουν τις ενδείξεις του μηχανισμού…………………………….................………....20

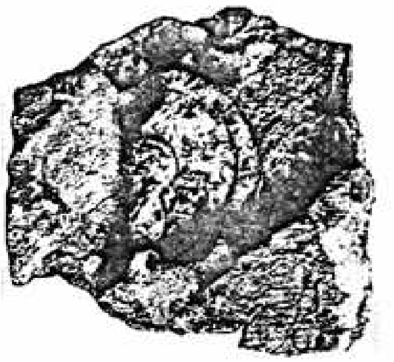
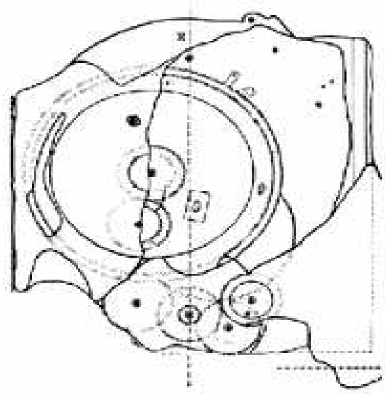
3. Η Java μικροεφαρμογή υπολογισμού των ενδείξεων της οπίσθιας όψης του μηχανισμού.
    3.1 Εισαγωγή………………………………………..................…..22
    3.2 Πρώτη περιοχή…………………………..................……...…25
    3.3 Δεύτερη περιοχή………………………..................……...27
    3.4 Τρίτη περιοχή………………………………...............…...……30

4. Συμπεράσματα ..……………………………….................…………....32
5. Βιβλιογραφία…………………………………….................……...…..33
6. Παράρτημα - ο κώδικας του applet …………………..................……...……34

1. ΕΙΣΑΓΩΓΗ

Ο Μηχανισμός των Αντικυθήρων είναι μια συνδεσμολογία από πολλά μεταλλικά γρανάζια τοποθετημένα έτσι, ώστε να εξομοιώνεται η κίνηση ορισμένων πλανητών. Με λίγα λόγια πρόκειται για έναν αρχαίο αστρονομικό μηχανικό υπολογιστή.

Η συσκευή αυτή αρχικά βρίσκονταν τοποθετημένη σε ένα ξύλινο πλαίσιο με δυο πόρτες πάνω στις οποίες αναγραφόταν οι οδηγίες χρήσης της. Στο μπροστινό μέρος υπήρχε ένας πίνακας που απεικόνιζε τον ελληνικό ζωδιακό κύκλο και το αιγυπτιακό ημερολόγιο σε ομόκεντρους κύκλους. Στο πίσω μέρος, δύο ακόμα καντράν έδειχναν τους κύκλους και τις εκλείψεις της Σελήνης.

Η ιστορία του υπολογιστή των Αντικυθήρων είναι λίγο πολύ γνωστή στους ακαδημαϊκούς κύκλους αλλά και στο ευρύ κοινό. Ο μηχανισμός αυτός ανασύρθηκε από τα συντρίμμια ενός μεγάλου ελληνικού εμπορικού πλοίου από σφουγγαράδες της Σύμης το 1900 και είχε τη μορφή μιας σκουριασμένης μάζας μετάλλων, μορφή που αποκτήθηκε ύστερα από την πολυετή παραμονή του σε συνθήκες διάβρωσης στα βαθιά νερά της θάλασσας των Αντικυθήρων, λίγες δεκάδες μέτρα από την ακτή.

Πολύ αργότερα, την δεκαετία του ’70, με τη συνδρομή του μεγάλου ερευνητή των βυθών Ζακ Υβ Κουστώ και των έμπειρων δυτών της ομάδας του, ανασύρθηκαν νέα θραύσματα από το ναυάγιο συμπληρώνοντας με αυτόν τον τρόπο κάποια από τα κομμάτια που έλειπαν για την ολοκλήρωση της εικόνας του θαυμαστού αυτού αντικειμένου.

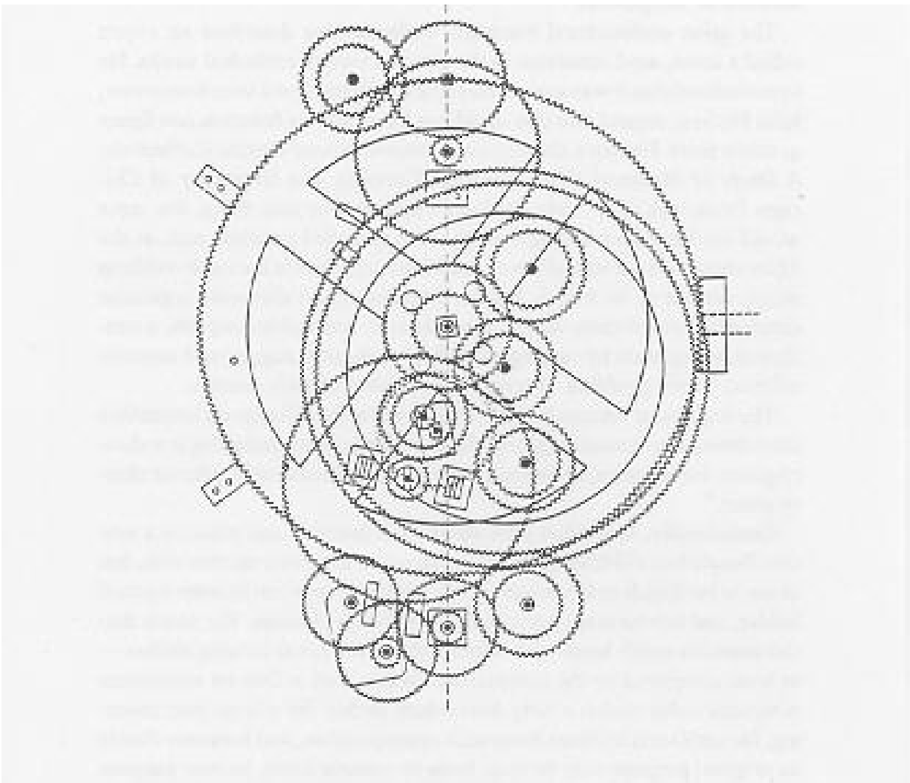
Τα κομμάτια του μηχανισμού που βρίσκονται στα χέρια επιστημόνων εδώ και περισσότερο από εκατό χρόνια κίνησαν από πολύ νωρίς το ενδιαφέρον κάθε είδους ερευνητών της αρχαιότητας .

Έγιναν πολλές μελέτες των θραυσμάτων του μηχανισμού από έλληνες και ξένους ειδικούς που προσπάθησαν να δώσουν επιστημονικές ερμηνείες για την προέλευση, την χρήση, τον τρόπο λειτουργίας και τον χρονολογικό εντοπισμό της κατασκευής του Υπολογιστή.

Αρχικά η πληρέστερη επιστημονική έρευνα για τον Υπολογιστή έγινε από τον

ο Derek De Solla Price

Derek De Solla Price σε συνεργασία με τον ειδικό επιγραφολόγο του Ινστιτούτου Προχωρημένων Μελετών του Princeton Γεώργιου Σταμίρη και τον πυρηνικό φυσικό του ελληνικού κέντρου Πυρηνικής Ενέργειας "Δημόκριτος" Χαράλαμπο Καράκαλο. Η έρευνα ξεκίνησε το 1958 και ολοκληρώθηκε το 1974 με την δημοσίευση της μελέτης «Gears from the Greeks: the Antikythera mechanism — a calendar computer from ca. 80 B.C..». Σε αυτό το μοντέλο για την επεξήγηση της λειτουργίας του μηχανισμού, χρησιμοποιούνται διαφορικά γρανάζια, τα οποία του προσδίδουν τη δυνατότητα πρόσθεσης και αφαίρεσης των γωνιακών τους ταχυτήτων (των γραναζιών). Το διαφορικό χρησιμοποιήθηκε για τον υπολογισμό του σεληνιακού συνοδικού κύκλου, για την αφαίρεση των αποτελεσμάτων της κίνησης του ήλιου από αυτά της αστρικής κίνησης της σελήνης.



Το μοντέλο του Derek De Solla Price

Παρ’ όλη την –φαινομενική- λειτουργικότητα του πρώτου μοντέλου του υπολογιστή, υπήρχαν κάποια κενά στη λειτουργία του συγκεκριμένου μοντέλου. Ο Michael Wright έφορος του London Science Museum και τώρα του Imperial College στο Λονδίνο, μαζί με τον Allan George Bromley, έκανε μια εντελώς νέα μελέτη των αρχικών θραυσμάτων του μηχανισμού με τα εφόδια που του παρείχε η μοντέρνα τεχνολογία, με την τεχνική της γραμμικής τομογραφίας ακτινών Χ. Πέρα από τις διαφορές που προέκυψαν στον αριθμό των δοντιών σε ορισμένα γρανάζια, η αρχική σκέψη του Price για συμπερίληψη διαφορικών γραναζιών στον μηχανισμό απεδείχθη λανθασμένη. Προσετέθησαν καινούρια γρανάζια και αντικατεστάθησαν οι δυο ομόκεντροι κύκλοι, που προέβλεπε ο Price στην πίσω πλευρά, με σπειροειδείς έλικες.

Ο Michael Wright

Το μοντέλο του Wright ήταν πιο λειτουργικό και ακριβές, ανέπτυξε την αρχική ιδέα του μηχανισμού σαν «πλανητάριο», με την διαφορά όμως ότι δεν μοντελοποιούσε την κίνηση του ήλιου και της σελήνης μόνο, αλλά και άλλων πλανητών όπως ο Άρης η Αφροδίτη ο Ερμής ο Δίας και ο Κρόνος.

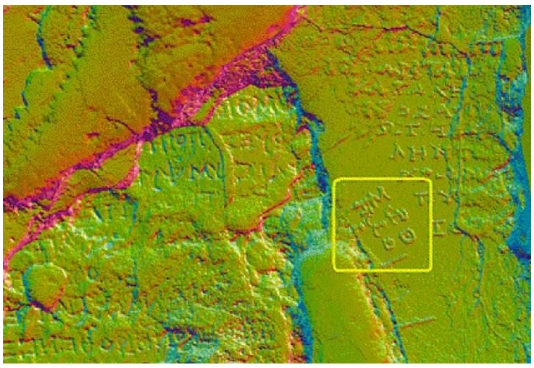
Ο Allan George Bromley

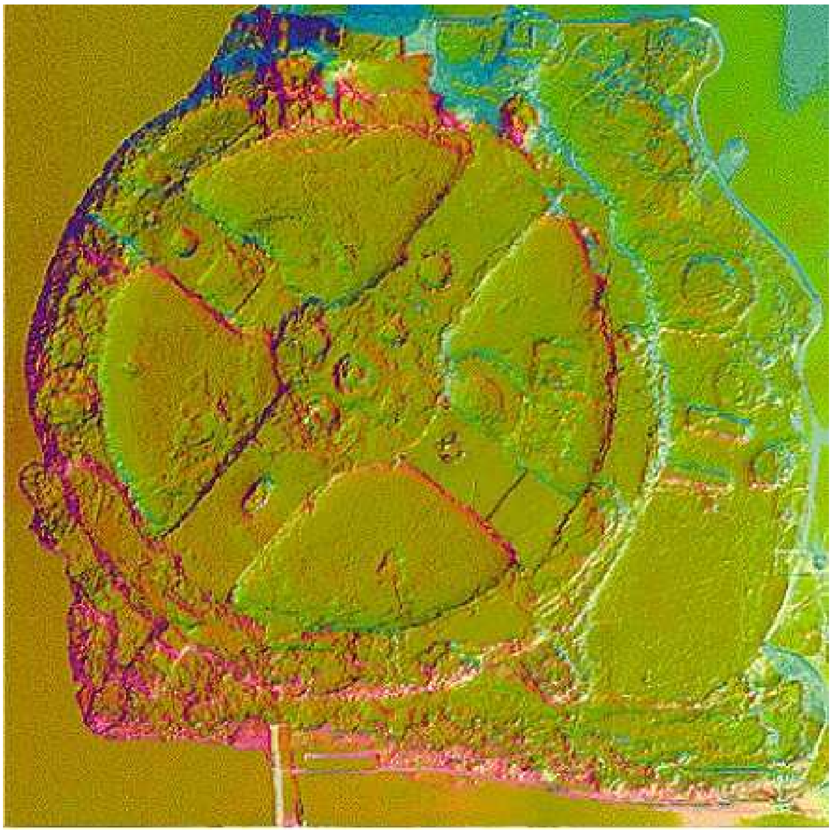
Το μοντέλο του Wright και η ανακατασκευή του

Την άνοιξη του 2005 δόθηκε η άδεια για τη μελέτη του Αστρολάβου και ειδικά μηχανήματα μεταφέρθηκαν από την Αγγλία με χορηγία του ιδρύματος Leverhulm. Η ερευνητική ομάδα άρχισε τη μελέτη με φωτογράφηση των θραυσμάτων σε μεγάλη ανάλυση. Επόμενο στάδιο ήταν η φωτογράφηση με τη «διαφορική» μέθοδο της Hewlett Packard. Πρόκειται για μια διαδικασία φωτογράφησης που γίνεται υπό 50 διαφορετικούς φωτισμούς. Τα στοιχεία καταγράφονται και αναλύονται σε ηλεκτρονικό υπολογιστή.

Η επιστημονική ομάδα της HP επί τω έργω.

Το τρίτο στάδιο έγινε με ένα μοναδικό μηχάνημα στον κόσμο, που ζυγίζει 8,5 τόνους και μεταφέρθηκε από την Αγγλία για τη μελέτη του Αστρολάβου. Είναι ένας ειδικός τομογράφος πολύ μεγάλης ενέργειας για να διαπερνά το θραύσμα των 16 εκατοστών από πέτρα και μπρούντζο. Ο τομογράφος έχει εξαιρετική μικροεστίαση και καταγράφει τα δεδομένα με ακρίβεια. Ένα εκατοστό του χιλιοστού, σε κάθε τομή.





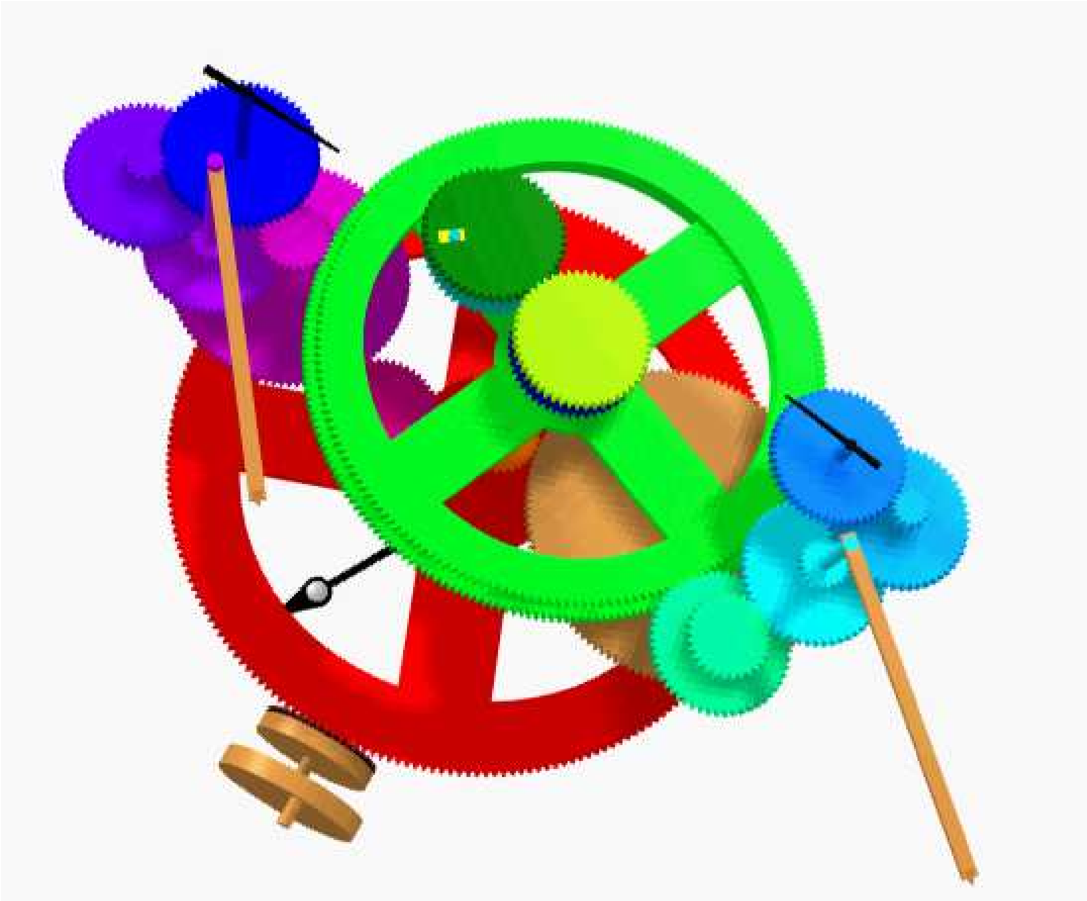
Άποψη του Υπολογιστή των Αντικυθήρων πριν και μετά τη σάρωση από τον ειδικό τομογράφο.

Στην ομάδα που άρχισε τη μελέτη του μηχανισμού με την εφαρμογή μεθόδων σύγχρονης τεχνολογίας μετείχαν οι καθηγητές: από το Παν. Αθηνών, Γιάννης Πιτσάκης και Ξενοφών Μουσάς, από το Κάρντιφ της Αγγλίας οι Mike Edmunts και Antony Freeth και από το ΑΠΘ ο Γ. Σειραδάκης και αποτελούν μαζί με άλλους την 'Ομάδα του προγράμματος για τη μελέτη του μηχανισμού των Αντικυθήρων '.

Η Ομάδα του προγράμματος για τη μελέτη του μηχανισμού των Αντικυθήρων

Τα αποτελέσματα που έδωσε η νέα αυτή έρευνα ήταν αφενός μεν αποκάλυψη ενός μέρους των οδηγιών χρήσεως του υπολογιστή τα οποία μας βοηθούν στην περεταίρω κατανόηση της λειτουργίας του, αφετέρου δε η τελική(;) συνδεσμολογία των γραναζιών που το αποτελούν. Η διάφορα του από το προηγούμενο μοντέλο είναι ότι το λεγόμενο 'γρανάζι του Ιππάρχου' (Κ1-Κ2) αντί να είναι στατικό, είναι στερεωμένο πάνω στο μεγάλο γρανάζι (Ε4) και κινείται μαζί του.

.



Άποψη του τελικού μοντέλου από το animation του καθ. Μάνου Ρουμελιώτη.

2. Η τρισδιάστατη προσομοίωση του υπολογιστή των Αντικυθήρων.

2.1 Εισαγωγή

Στο πρώτο μέρος θα περιγραφεί η διαδικασία που ακολουθήθηκε για την κατασκευή της τρισδιάστατης προσομοίωσης του υπολογιστή των Αντικυθήρων, βάσει των ευρημάτων της 'Ομάδας του προγράμματος για τη μελέτη του μηχανισμού των Αντικυθήρων ' .

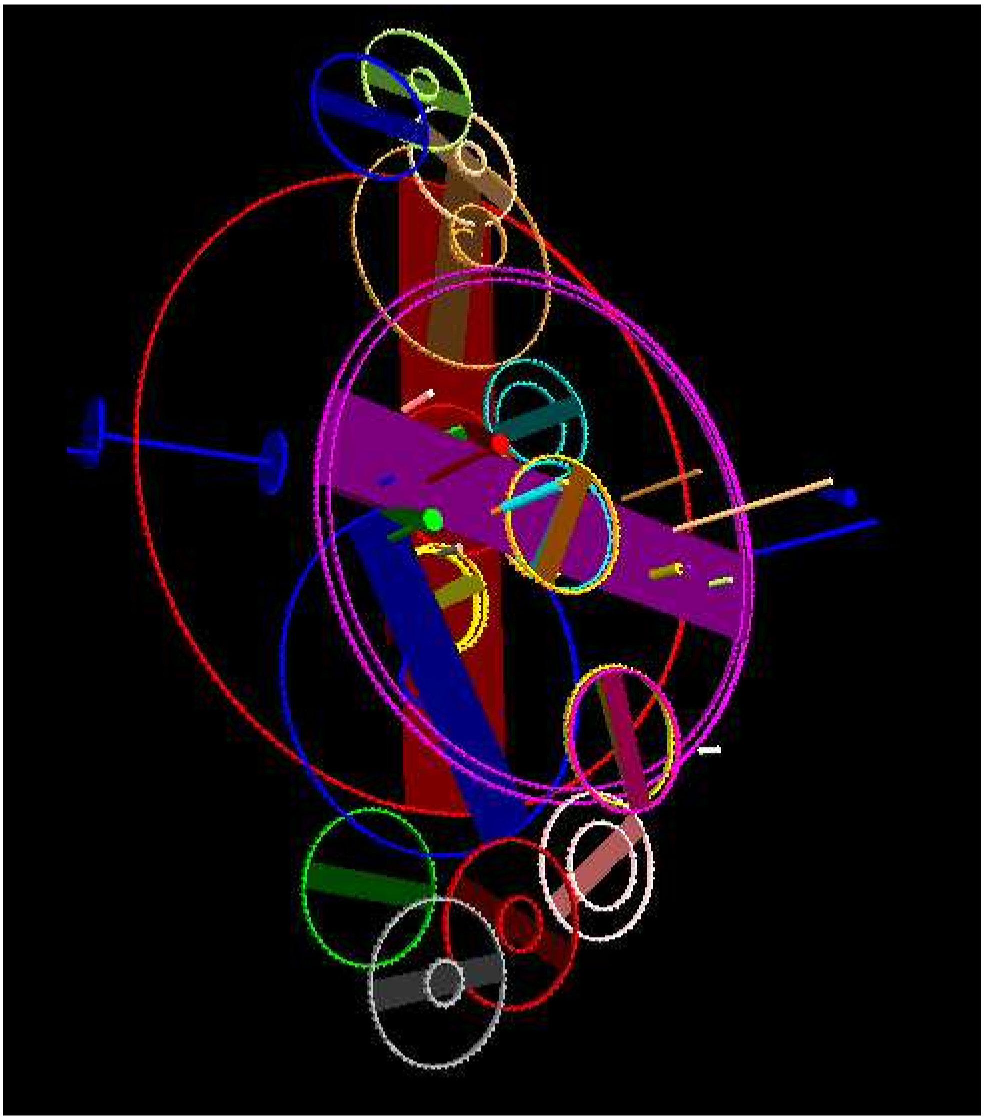
Η νέα προσομοίωση βασίζεται κατά κύριο λόγο στην προηγούμενη, που είχε κατασκευαστεί από τον Φοίβο Ασημακοπουλο φοιτητή του Τμήματος Εφαρμοσμένης Πληροφορικής του Πανεπιστήμιου Μακεδονίας ως πτυχιακή εργασία. είναι γραμμένη σε γλώσσα προγραμματισμού C ++ και συνδυάστηκε με την OpenGL για τα γραφικά.

Τα σημεία στα οποία απαντώνται οι διάφορες της νέας από την προηγούμενη είναι τα εξής:

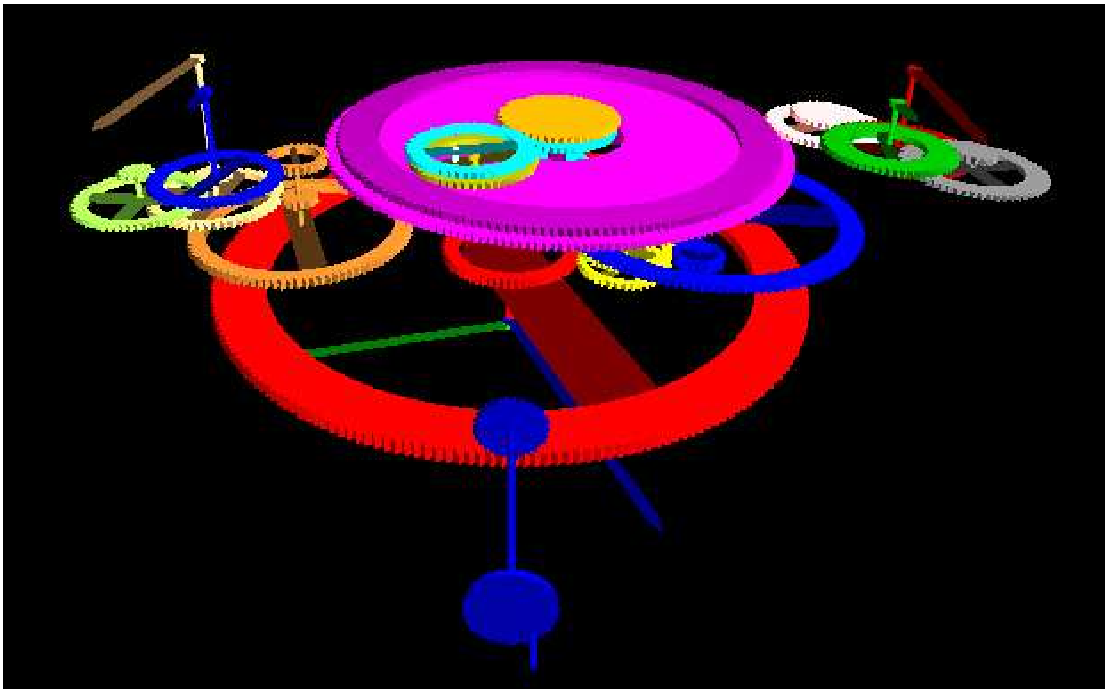
1. οι νέες θέσεις, διαστάσεις και ο αριθμός των δοντιών στα γρανάζια
2. το γρανάζι του Ίππαρχου γίνεται διαφορικό
3. προστίθεται ο δείκτης με την ένδειξη του κύκλου του Μέτονα στο καντράν του οπίσθιου μέρους
4. η επικόλληση των εικόνων που αναγράφουν τις ενδείξεις του μηχανισμού.

2.2 Οι νέες θέσεις και διαστάσεις των γραναζιών.

Όταν τοποθετήθηκαν τα νέα δεδομένα για τις θέσεις των γραναζιών διαπιστώθηκε ότι οι διαστάσεις x και y ήταν αντεστραμμένες από αυτές του προγράμματος. Επίσης τα δεδομένα που στο πρόγραμμα εννοούνταν εσωτερική και εξωτερική ακτίνα (για τον σχεδιασμό), στα στοιχεία που λήφθηκαν από την επιστημονική ομάδα εννοούνταν ως η απόσταση της κορυφής του δοντιού ενός γραναζιού από τη βάση του. Με αυτά τα δεδομένα σχηματίστηκε η παρακάτω εικόνα.



Αφού ήρθαν στη θέση τους οι διαστάσεις x και y, με υπόδειξη του επιβλέποντος καθηγητή μεγάλωσε η απόσταση μεταξύ των γραναζιών (διάσταση z) για να υπάρχει μια καλύτερη άποψη αυτών, όπως επίσης και για λόγους αισθητικής.

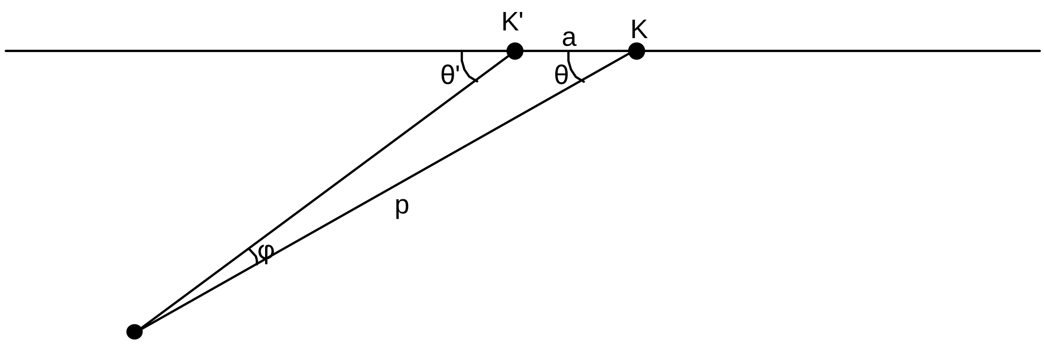


Εικόνα από την τελική προσομοίωση

Επιπλέον έγινε απαραίτητη και η αλλαγή στις αρχικές γωνίες των γραναζιών για να 'δένουν' μεταξύ τους, χωρίς να συμπίπτουν τα δόντια από διαφορετικά γρανάζια και να φαίνεται η κίνηση όσο το δυνατόν πιο φυσική. Το κομμάτι αυτό ήταν αρκετά δύσκολο γιατί δεν υπήρχε συγκεκριμένος αλγόριθμος για την επίτευξη του, αλλά έπρεπε να γίνει ξεχωριστά για το κάθε γρανάζι, αρχικά αλλάζοντας τις αρχικές τιμές της θέσης και της γωνίας του καθενός και μετά τρέχοντας το πρόγραμμα και εστιάζοντας στο εκάστοτε γρανάζι.

2.3 Το γρανάζι του Ίππαρχου γίνεται διαφορικό

Η αλλαγή αυτή ήταν θεωρητικά σχετικά εύκολη, στον πίνακα differential[] του προγράμματος, στις θέσεις των Κ1, Κ2 θα τοποθετούνταν συντελεστές για την ταχύτητα περιστροφής διάφορες του μηδενός που τα καθιστούσαν διαφορικά. Στη συνέχεια στο πρόγραμμα έπρεπε να γίνουν αλλαγές στα ορίσματα ορισμένων εντολών για τον λόγο ότι υπήρχαν πλέον μεταβλητά κέντρα περιστροφής τα οποία υπολογίζονταν με τον εξής τρόπο



από τον νόμο των ημιτόνων:

$$\frac{p}{\sin(180-\vartheta')} = \frac{a}{\sin\varphi} \Rightarrow \sin\varphi = \frac{a}{p}\sin(180-\vartheta') = \frac{a}{p}\sin(180-\vartheta-\varphi)$$

χρησιμοποιούμε τον τύπο του αθροίσματος

$$\sin\varphi = \frac{a}{p}(\sin(180-\vartheta)\cos\varphi - \sin\varphi\cos(180-\vartheta) \Rightarrow$$

$$\sin\varphi(1+\frac{a}{p}\cos(180-\vartheta) = \frac{a}{p}\sin(180-\vartheta)\cos\varphi$$

θέτουμε:

$$x = 1 + \frac{a}{p}\cos(180 - \vartheta)$$
$$y = \frac{a}{p}\sin(180 - \vartheta)$$

και λαμβάνουμε

$$\sin^2 \varphi\, x^2 = y^2 (1 - \sin^2 \varphi)$$

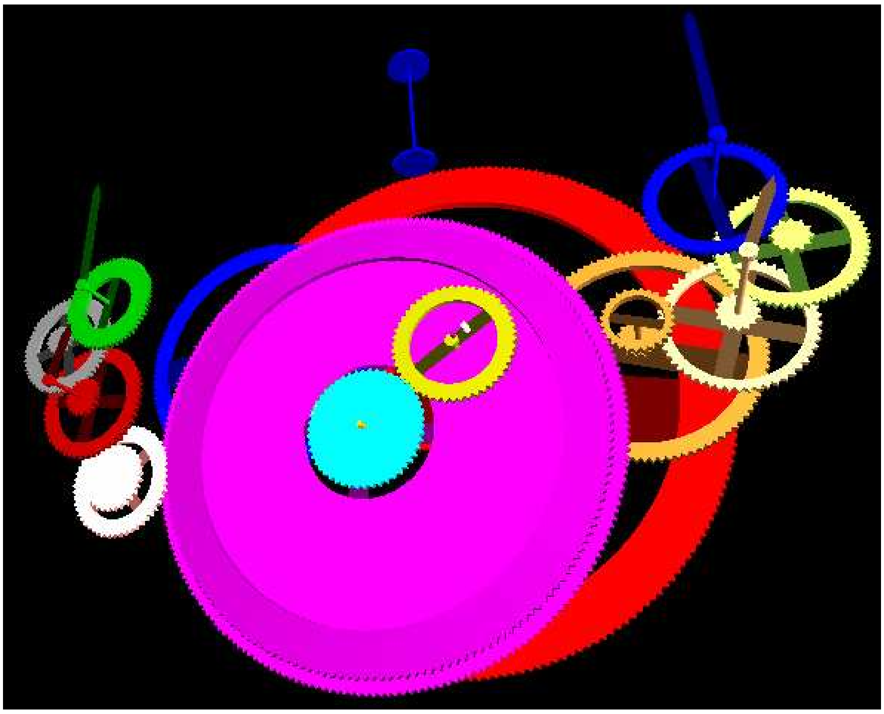
από το οποίο έχουμε:

$$\sin\varphi = \frac{y}{\sqrt{x^2 + y^2}} \text{ ή αλλιώς } \varphi = \arcsin(\frac{y}{\sqrt{x^2 + y^2}})$$

Συνεχίζοντας, το πέρασμα των νέων δεδομένων (a και p) ως μεταβλητές που λαμβάνουν τιμές από τις συντεταγμένες των αρχικών θέσεων των γραναζιών που δίνονται στην αρχή του προγράμματος

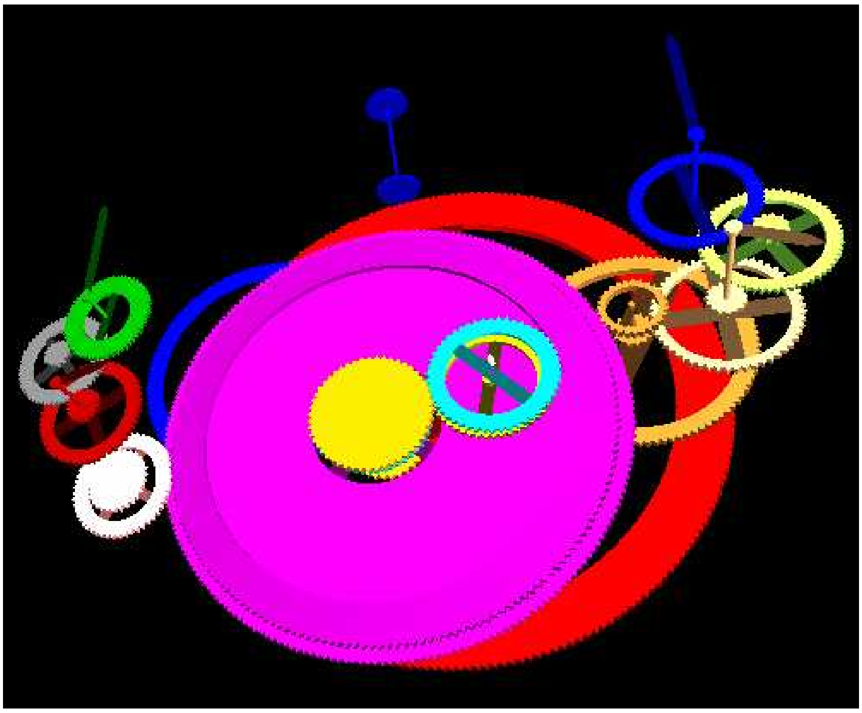
```
float a= sqrt((gearpos[21][0] - gearpos[20][0])*(gearpos[21][0] -
gearpos[20][0])+(gearpos[21][1] - gearpos[20][1])*(gearpos[21][1] –
gearpos[20][1]));

float p= sqrt((gearpos[20][0]-axlepos[10][0])*(gearpos[20][0]-
axlepos[10][0])+(gearpos[20][1]-axlepos[10][1])*(gearpos[20][1]-axlepos[10][1]));
```

είχε ως αποτέλεσμα να εξαφανιστούν τα γρανάζια K2 και E6:



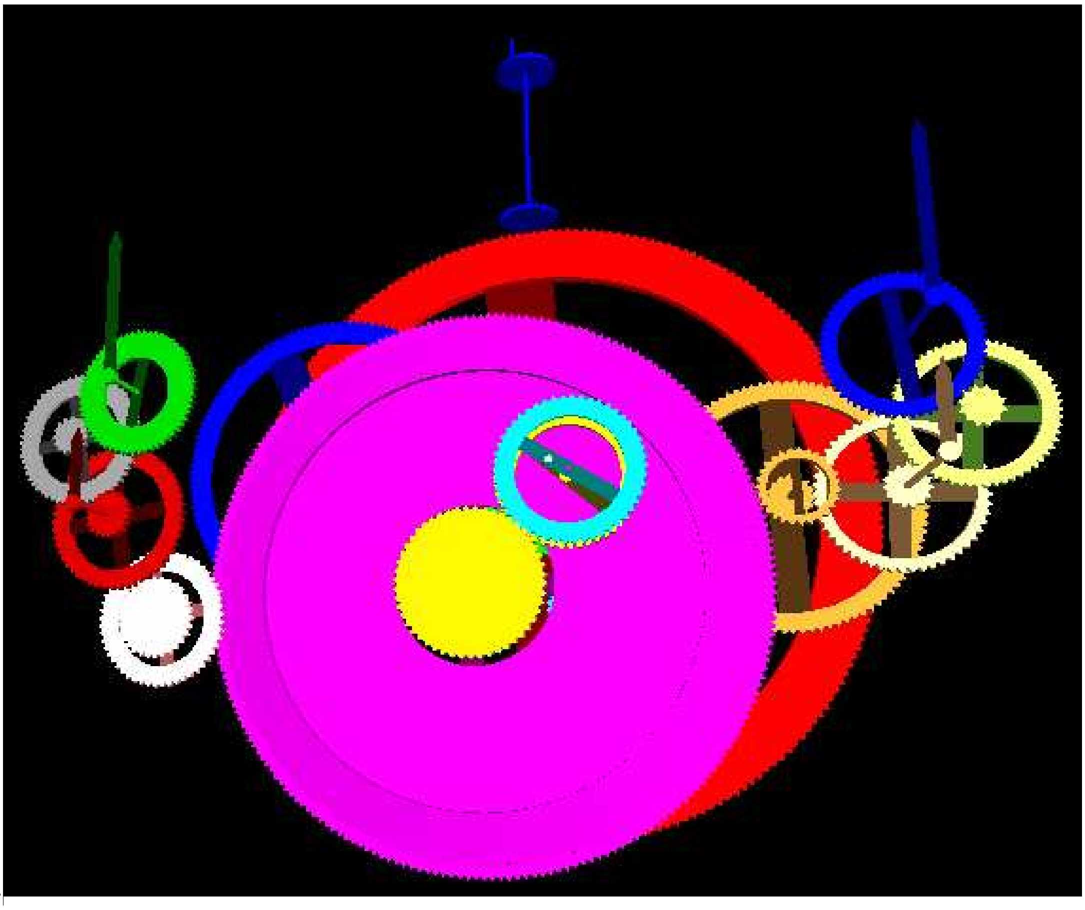
Εικόνα από την προσομοίωση test 1

Με εκ των πρότερων υπολογισμό των a και p εκτός προγράμματος και κατόπιν πέρασμα των τιμών που ευρέθησαν στον κώδικα, το αποτέλεσμα ήταν να στρέφεται το K2 με ασυνάρτητη ταχύτητα.



Εικόνα από την προσομοίωση test 2

Υπήρξαν πολλές ακόμα προσπάθειες για να βρεθεί η κατάλληλη ταχύτητα για το K2, χωρίς το απολύτως επιθυμητό αποτέλεσμα. Έτσι έγινε συμβιβασμός με αυτό που χρησιμοποιείται, το οποίο χάνει τη σωστή του κίνηση ύστερα από έναν σχετικά μεγάλο αριθμό περιστροφών.



Εικόνα από την τελική προσομοίωση

2.4 Πρόσθεση του δείκτη με την ένδειξη του κύκλου του Μέτονα στο καντράν του οπίσθιου μέρους.

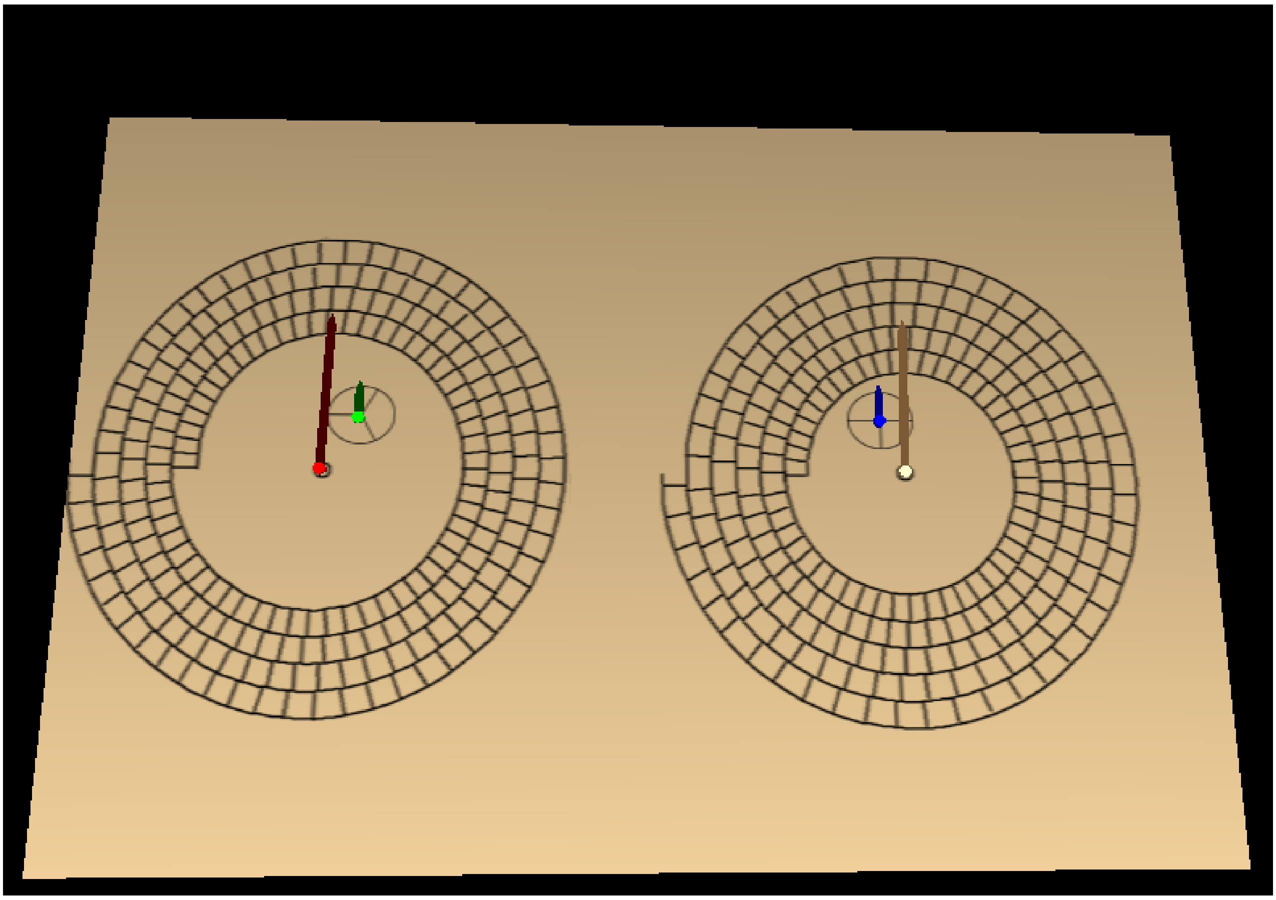
Για τη διαδικασία αυτή έγιναν οι απαραίτητες αλλαγές στο πρόγραμμα. Ορίστηκε ο αριθμός των δεικτών κατά ένα περισσότερο, ομοίως και αυτός των αξόνων διότι ο δείκτης αυτός έπρεπε κάπου να στηριχθεί.

```
#define num_pointers 6
#define num_axles 21
```

Στη συνέχεια συμπληρώθηκαν τα νέα δεδομένα που απαιτούνταν στους πίνακες που περιείχαν τα χαρακτηριστικά των δεικτών και των αξόνων:

```
 static float axlepos[num_axles][5]
static float axledata[num_axles][4]
static bool axle_visibility[num_axles]
static float axle_differential[num_axles]
static float pointer_pos[num_pointers][5]
static float pointer_len[num_pointers]
static float pointer_angle[num_axles]
static bool pointer_visibility[num_axles]
```

Τέλος, για λόγους συμμετρίας των δεικτών στην οπίσθια όψη, μεταφέρθηκαν ο δείκτης, με τον άξονα και το γρανάζι που τα έφερε, αλλάζοντας ορισμένους πίνακες από αυτούς που προαναφέρθηκαν.

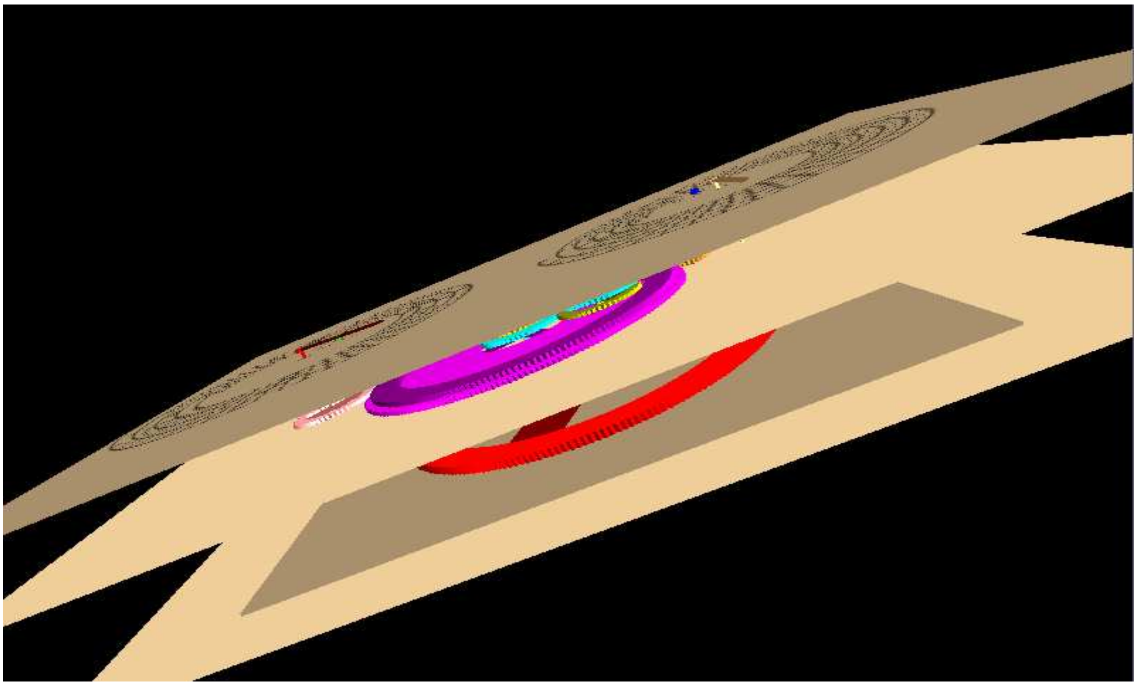


εικόνα της οπίσθιας όψης του υπολογιστή από την τελική προσομοίωση

2.5 Η επικόλληση των εικόνων που αναγράφουν τις ενδείξεις του μηχανισμού.

Η επίτευξη αυτού του θέματος ήταν απαραίτητη για την τελική πρακτική και αισθητική του όλου project. Θα έπρεπε να υπάρχει η εικόνα στην οποία θα περιγράφονται οι ενδείξεις και θα κινούνται οι δείκτες. Στις προηγούμενες προσομοιώσεις δεν έγινε εφικτή η ζητούμενη επικόλληση. Το πρόβλημα εντοπιζόταν στο ότι ενώ η πρόσθια εικόνα επικολλείτο κανονικά, η οπίσθια δεν εμφανιζόταν καν στην τελική προσομοίωση, παρ' ολο που το πρόγραμμα την ελάμβανε κανονικά, κατά παράβαση των δοθέντων εντολών. Η πέρα από τη λογική αυτή η απροθυμία του προγράμματος να προβάλλει την δεύτερη εικόνα, ξεπεράστηκε με την συνεργασία του Θέμη Κότσιαλου, διδακτορικού φοιτητή του Τμήματος Εφαρμοσμένης Πληροφορικής του Πανεπιστήμιου Μακεδονίας.

Η λύση βρέθηκε με το να τοποθετηθεί μια τρίτη πλάκα ανάμεσα στις δυο που περιέκλειαν τα γρανάζια.



Εικόνα από την προσομοίωση – η ενδιάμεση πλάκα

Με τον τρόπο αυτό αποτυπώθηκαν και η δυο εικόνες πάνω στις εξωτερικές πλάκες, με τη μόνη διάφορα ότι η εικόνα με τις πρόσθιες ενδείξεις του μηχανισμού προβάλλονταν στην πίσω πλάκα ενώ η εικόνα με τις οπίσθιες ενδείξεις στην μπροστινή. Υστέρα από όλα αυτά, επειδή δεν μπορούσε να γίνει τίποτα παραπάνω, αλλάχθηκε το όνομα των εικόνων που εισαγαγονταν στο πρόγραμμα, με την πρόσθια να μετονομάζεται σε rear.bmp και την οπίσθια σε front.bmp το οποίο και επέφερε το επιθυμητό αποτέλεσμα.

Αξίζει να σημειωθεί η απάντηση του επιβλέποντος καθηγητή Μάνου Ρουμελιώτη, Όταν του εξήγησα τον τρόπο με τον οποίο έγινε εφικτό το ζητούμενο, που έλεγε ότι με αυτά που συμβαίνουν σε αυτό το πρόγραμμα θα μάθει ότι ο προγραμματισμός είναι μια non-deterministic διαδικασία!

3 Η Java μικροεφαρμογή υπολογισμού των ενδείξεων της οπίσθιας όψης του μηχανισμού.

3.1 Εισαγωγή

Αρχικά θα δοθούν κάποιοι ορισμοί και έννοιες αστρονομικής φύσεως που θα επεξηγήσουν τα στοιχεία που κατασκευάστηκαν στην μικροεφαρμογή αυτή.

Μετωνικός κύκλος

Ο Μέτων ο Αθηναίος αστρονόμος της αρχαιότητας, παρατήρησε ότι μια περίοδος 19 ηλιακών ετών είναι σχεδόν ίση με 235 σεληνιακούς μήνες Το σφάλμα αυτό του μετωνικού κύκλου ανάμεσα στα 19 έτη (= 6939,602 ημέρες) και τους 235 μήνες (= 6939,688 ήμερες) είναι μόνο 2 ώρες η αλλιώς μια ολόκληρη ημέρα κάθε 219 έτη Οι παρατηρήσεις αυτές προσδίδουν στο έτος το μέγεθος των 365 + 1/4 + 1/76 ημερών = 365 ημέρες 6 ώρες 18 λεπτά και 56 δευτερόλεπτα

Κύκλος του Καλλίπου

Σχεδόν έναν αιώνα μετά τον Μέτωνα, ο Κάλλιπος πρότεινε έναν κύκλο με περίοδο 76 ετών, ως βελτίωση της εννεαδεκαετηρίδος. Πίστευε ότι η πραγματική τιμή του μεγέθους του έτους ήταν προσεγγιστικά 365 + 1/4 ήμερες = 365 ημέρες και 6 ώρες Έτσι πολλαπλασίασε τον 19ετη κύκλο του Μέτωνα με το 4 για να αποφέρει ακέραιο αριθμό ημερών και στη συνέχεια αφαίρεσε μια ολόκληρη ημέρα από την τελευταία εννεαδεκαετηρίδα. Με αυτόν τον τρόπο κατασκεύασε έναν κύκλο 76 ετών που περιέχει 940 σεληνιακούς μήνες και 27,759 ημέρες.

Κύκλος Σάρος των εκλείψεων

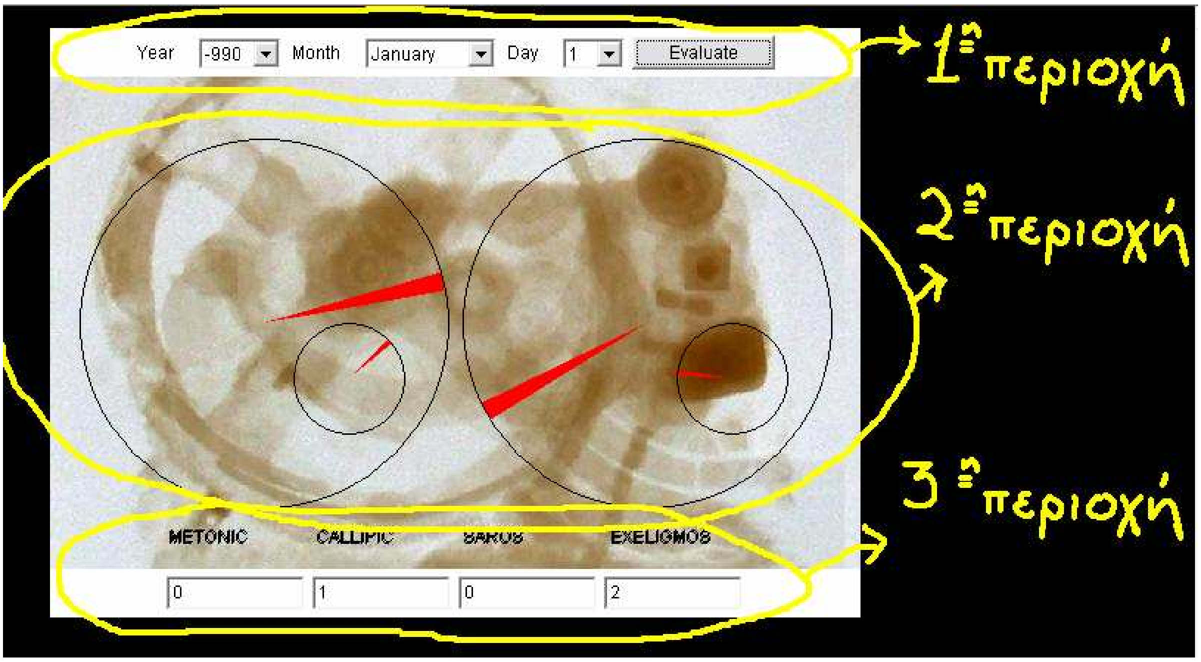
Ο κύκλος Σάρος ανακαλύφθηκε από τους Χαλδαίους μερικούς αιώνες πριν από την γέννηση του Χριστού και αποτελεί μια περίοδο περίπου 18 ετών 11 ημερών και 8 ωρών (περίπου 6585 + 1/3 ημέρες) και χρησιμοποιήθηκε για την πρόβλεψη των εκλείψεων του ήλιου και της σελήνης Ουσιαστικά μετά την έλευση ενός κύκλου Σάρος ύστερα από μια έκλειψη ο ήλιος η γη και σελήνη θα επανέλθουν σχεδόν στην ίδια σχετική γεωμετρία δημιουργώντας μια σχεδόν πανομοιότυπη έκλειψη

Εξελιγμος

Ο Εξελιγμος αποτελεί ένα άθροισμα τριών κύκλων Σάρος έχοντας το πλεονέκτημα ότι περιέχει έναν σχεδόν ακέραιο αριθμό ημερών. Έτσι η επόμενη έκλειψη θα είναι ορατή σε κοντινές με αυτές περιοχές στις οποίες εμφανίστηκε μια έκλειψη έναν Εξελιγμο πριν, σε αντίθεση με τον Σάρος στον οποίο η έκλειψη εμφανίζεται 8 ώρες αργότερα στην ημέρα, η περίπου 120 μοίρες δυτικότερα από εκείνη που εμφανίστηκε έναν Σάρος πριν.

Τους κύκλους αυτούς (Μετωνικός, του Καλλίππου, Σάρος και Εξελιγμού) και τις φάσεις στις οποίες βρισκόταν ο καθένας υπολόγιζε και απεικόνιζε ο μηχανισμός των Αντικυθήρων.

Στο δεύτερο αυτό κομμάτι λοιπόν θα περιγραφεί το java applet που κατασκευάστηκε για να υπολογίζει τις ενδείξεις του καντράν του οπίσθιου μέρους του μηχανισμού των Αντικυθήρων, βάσει δοθείσης ημερομηνίας.



Οι περιοχές που χωρίζεται το applet.

Με μια γρήγορη ματιά μπορούμε να δούμε ότι αποτελείται από τρεις περιοχές. Την πρώτη που τοποθετείται στο πάνω μέρος του applet, όπου εντοπίζονται με τη σειρά οι επιλογείς του έτους, του μήνα και της ημέρας με τις default τιμές τους, όπως Επίσης και το κουμπί 'Evaluate' που όταν το επιλέξουμε ενεργοποιείται το applet. Στη δεύτερη περιοχή βρίσκεται το γραφικά κατασκευασμένο καντράν της οπίσθιας όψης του μηχανισμού. Οι δείκτες κινούνται κατά τη διάρκεια του υπολογισμού των ενδείξεων και σταματούν στο σημείο στο οποίο θα έπρεπε θεωρητικά να σταματούσε και ο Μηχανισμός κατά την λειτουργία του. Τέλος στην τρίτη περιοχή εντοπίζονται τα πεδία στα οποία εμφανίζονται αριθμητικά οι τελικές ενδείξεις.

3.2 Πρώτη περιοχή

Αρχικά ορίστηκαν τα labels για το έτος τον μήνα και την ημέρα, και στη συνέχεια κατασκευάστηκαν τα choices για το κάθε ένα, για το έτος από το –990 έως το 2999

```
ch1=new Choice();
for (g=-990;g<0;g++)
        {ch1.add(String.valueOf(g));
        }
for (g=1;g<3000;g++)
        {ch1.add(String.valueOf(g));
        }
this.add(ch1);
```

, για τον μήνα

```
  ch2=new Choice();
        ch2.add("January");
        ……….
        ch2.add("December");

        this.add(ch2);
```

και την ημέρα

```
for (int w=1;w<32;w++)
        {ch3.add(String.valueOf(w));
        }
        this.add(ch3);
```

Επίσης δημιουργήθηκε ένα button στο οποίο προσετέθη και το actionlistener,

```
b1=new Button(str); // οπου str  είναι το evaluate
        add(b1);
        b1.addActionListener(this);
```

με την επιλογή του οποίου ξεκινά να τρέχει η μικροεφαρμογή. Τα στοιχεία αυτά προσετέθησαν με την κατάλληλη διαδοχή σε ένα panel το οποίο με τη σειρά του τοποθετήθηκε με την εντολή

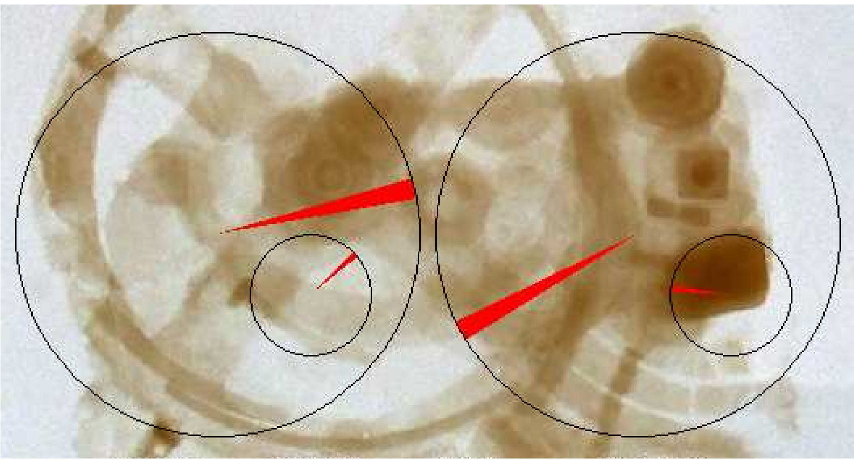
```
this.add(p1,BorderLayout.NORTH);
```

στο πάνω μέρος του applet.

Η πρώτη περιοχή του applet.

3.3 Δεύτερη περιοχή

Για την κατασκευή αυτού του κομματιού του applet έγινε χρήση νημάτων (threads) η αλλιώς 'ελαφριών διεργασιών' και της μεθόδου repaint() δημιουργώντας έτσι την κίνηση των δεικτών, σύμφωνα με τον τρόπο που θα περιγραφεί.



Η δεύτερη περιοχή του applet.

Οπίσθια όψη ανακατασκευής του υπολογιστή.

Για τον υπολογισμό της ενδείξεως του Μετονικού κύκλου, έχει αρχικά υπολογιστεί από τους λόγους των περιστροφών των γραναζιών, ότι για μια πλήρη περιστροφή του b1 - κατά την οποία παρέρχεται ένα ολόκληρο έτος – αντιστοιχούν 0.2631579 περιστροφές του n1. Αναλυτικότερα:

b2-l1+l2-m1+m2-*n1*:

$$\frac{64}{38}\times\frac{53}{96}\times\frac{15}{53} = \frac{2\times 32}{2\times 19}\times\frac{53}{3\times 32}\times\frac{3\times 5}{53} = \frac{5}{19} = 0.2631579$$

Τα έτη οι μήνες και οι ημέρες που έχουν επιλεγεί από τα choices στο πρώτο μέρος μετατρέπονται αρχικά σε ημέρες και στη συνέχεια σε έτη. Οι μήνες έχουν ήδη τιμές μέσα στο πρόγραμμα από τις παρελθούσες μέρες (t2) ,

Π. χ.

```
if(a=="September")
    t2=243;mn=30;
```

και σε αυτές απλά προστίθενται οι ημέρες που έχουν επιλεχθεί στο τρίτο choice. Συνοψίζοντας, όλα αυτά που αναφέρθησαν μεταφράζονται βάσει του λόγου περιστροφής σε γωνία στροφής με τις εντολές

```
calc_days=(float)(((t1+990)*(365.24219)+t2+t3));
calc_years=(float)((calc_days/365.24219));
calc_met_rot=(float)((calc_years*0.2631579)%360);
```

στη συνέχεια, οι εντολές της run()

```
m=0;                                    //m ένας counter
            while(m<=calc_met_rot) {
            e1+=1;                      // e1 η αρχική γωνία του δείκτη
            m+=1;
            repaint();
            try {             thr.sleep(30);}}
```

σε συνδυασμό με τις εντολές της paint()

```
    g.setColor(Color.red);
```

```
g.fillArc(20,75,250,250,e1,6);
g.drawOval(20,75,250,250);
```

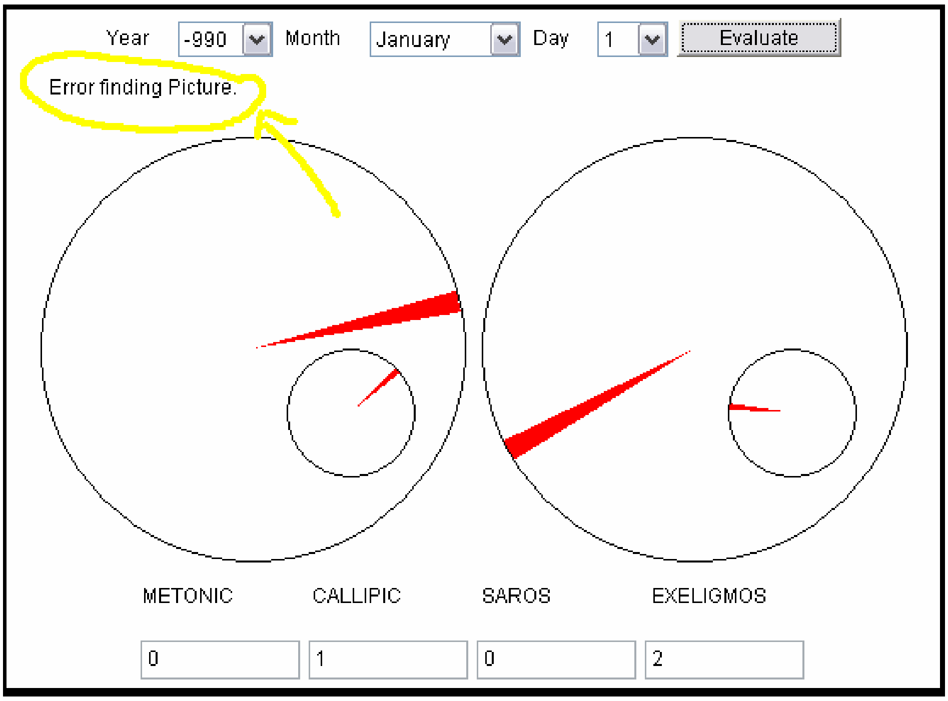
σχεδιάζουν ένα κύκλο και ένα δείκτη ο οποίος διαγράφεται και ξαναδημιουργείται σε διαδοχικές θέσεις, μέχρι να φτάσει στην τελική θέση που προκύπτει από τους παραπάνω υπολογισμούς.

Επίσης για λόγους καλαισθησίας, φορτώθηκε μια εικόνα για φόντο στους λιτούς σχηματισμούς που κατασκευάστηκαν. Η εικόνα αυτή ονομάζεται anti.gif και πρέπει να βρίσκεται στο ίδιο folder με τα αρχεία .class και .html του applet. Η κλήση της εικόνας γίνεται στην init(), ταυτόχρονα με τον ορισμό της

```
this.picture = this.getImage(this.getDocumentBase(), "anti.jpg");
```

ενώ εάν ο σχεδιασμός της στο applet αποτύχει είτε γιατί η εικόνα απουσιάζει είτε γιατί το αρχείο είναι φθαρμένο εμφανίζεται το μήνυμα λάθους Error finding Picture, με τις εντολές της paint()

```
if(!g.drawImage(this.picture, 0, 0, this)) {
g.drawString("Error finding Picture.", 25, 50);}
```



3.4 Τρίτη περιοχή

Η τρίτη και τελευταία περιοχή του applet αποτελείται από τα πεδία στα οποία εμφανίζονται αριθμητικά οι τελικές ενδείξεις με τα αντίστοιχα labels τους. Αφού ορίστηκαν τα στοιχεία αυτά τοποθετήθηκαν σε ένα δεύτερο panel το οποίο με τη σειρά του τοποθετήθηκε με την εντολή

```
this.add(p2,BorderLayout.SOUTH);
```

στο κάτω μέρος του applet.

Η τρίτη περιοχή του applet

Τα textfields συμπληρώνονται με τις τιμές που υπολογίζονται στο πρόγραμμα, για παράδειγμα για την συμπλήρωση του tf1 του πεδίου που εμφανίζει τον αριθμό του μήνα του μετονικου κύκλου στον οποίο ανήκει η δοθείσα από τον χρήστη ημερομηνία, εκτελείται η εντολή

```
tf1.setText(String.valueOf( (int)((calc_days%6939.68865)/29.53059)));
```

όπου το ακέραιο υπόλοιπο των υπολογιζόμενων ημερών με τον αριθμό 6939.68865 (αριθμός ημερών του μετονικου κύκλου) , διαιρείται με τον αριθμό 29.53059 (αριθμός ημερών του συνοδικού μήνα) και αποδίδεται έτσι η ζητούμενη τιμή. Κατά αντίστοιχο τρόπο γίνεται και ο υπολογισμός της τιμής του τρίτου πεδίου για την ένδειξη του κύκλου Σάρος των εκλείψεων:

```
tf3.setText(String.valueOf((int)((calc_days%6585.32)/29.53059)));
```

Στην περίπτωση όμως του κύκλου του Καλλίππου υπολογίζεται απλά το τεταρτημόριο στο οποίο βρίσκεται ο δείκτης μετά τους υπολογισμούς με την εξής ακολουθία εντολών

```
deg1=(int)(36+(calc_cal_rot));
 //deg1 η αρχική τιμή της γωνίας του δείκτη
if(0<=deg1&&90>deg1)
            d=1;
        if(90<=deg1&&180>deg1)
            d=2;
        if(180<=deg1&&270>deg1)
            d=3;
        if(270<=deg1&&360>deg1)
            d=4;
            tf2.setText(String.valueOf(d));
```

Με τον ίδιο τρόπο υπολογίζεται και η εγγραφή του πεδίου του Εξελιγμού, μόνο που εδώ ο κύκλος χωρίζεται σε τριμόρια.

4. ΣΥΜΠΕΡΑΣΜΑΤΑ

Αυτή η εργασία βασίζεται στα δεδομενα και τη γνώση που αποκαλύφθηκε με τις τελευταίες μελέτες των θραυσμάτων του Υπολογιστή των Αντικυθήρων. Για τις μελέτες αυτές οι επιστήμονες ήταν εξοπλισμένοι με ο, τι καλύτερο είχε να επιδείξει η σύγχρονη τεχνολογία. Η τεχνολογία όμως δε σταματά ποτέ να εξελίσσεται, οπότε δεν μπορούμε να γνωρίζουμε τι μπορεί να μας επιφυλάσσει μια μελλοντική μελέτη του μηχανισμού όπου θα χρησιμοποιούνται οι τότε μοντέρνες τεχνικές .

Ο Υπολογιστής των Αντικυθήρων δεν μας έχει αποκαλύψει όλα τα μυστικά του. Είναι γνωστό στους μελετητές ότι υπάρχουν κι' άλλα ακόμα κομμάτια του Υπολογιστή στην τάφρο πλάι στο σημείο που κείται το αρχαίο ναυάγιο. Γνωρίζουμε επίσης ότι το εύρημα αυτό δεν ήταν το μοναδικό που υπήρχε στον αρχαίο κόσμο και ότι υπάρχουν πιθανότητες να συναντήσουμε και άλλα όμοια με αυτό αντικείμενα.

Η αρχαία ελληνική τεχνολογία έχει πολλά να επιδείξει ακόμα, ποιος μπορεί να ξέρει τι ευρήματα μας επιφυλάσσει το αύριο, ποιος μπορεί να ξέρει πόσοι τέτοιοι θησαυροί αναπαύονται στην αγκαλιά της ελληνικής γης και των ελληνικών θαλασσών και περιμένουν να μας αποκαλύψουν τα μυστικά τους για την χαμένη πια επιστημονική γνώση που κατείχε ο αρχαίος κόσμος.

5. ΒΙΒΛΙΟΓΡΑΦΙΑ

Εισαγωγή στην JAVA 2, Γιώργος Λιακέας, εκδόσεις Κλειδάριθμος

Πτυχιακή εργασία με θέμα : προσομοίωση του Μηχανισμού των Αντικυθήρων, Ασημακόπουλος Φοίβος 2007

http://www.opengl.org/documentation/blue_book/

http://www.antikythera-mechanism.gr

http://en.wikipedia.org/wiki/Antikythera_mechanism

http://www.cyclesresearchinstitute.org/lunar/saros.html

http://www.hermit.org/Eclipse/gen_stats.cgi?mode=query&page=full&qtype=type&body=L&saros=59

http://tycho.usno.navy.mil/vphase.html

http://www.lachesis.blackfury.com/hellenic.html

http://rosettacalendar.com/calendars.html

http://www.hsuyun.org/Dharma/zbohy/Literature/essays/czs/antikythera-full2

http://blog.onurpay.com/images/antikythera_05

http://www.futura-sciences.com

6. ΠΑΡΑΡΤΗΜΑ

Ο κώδικας του applet

```
import java.applet.*;
import java.awt.*;
import java.math.*;
import java.awt.event.*;
import java.applet.Applet;
import javax.swing.*;

public class Moustakasv_04 extends Applet implements Runnable,ActionListener {
	String s,str,a;
	TextField tf1,tf2,tf3,tf4;
    public  int t1=0,t2=0,t3=0,mn=0;
	int x=0,y=0;
	int e1=9;//9.03
	int e2=36;//36.4515
	int e3=205;//205.2
	int e4=171;//170.99316
	int m=0,n=0;
//	int c=10;
	double
calc_days,calc_years,calc_met_rot,calc_cal_rot,calc_sar_rot,calc_exe_rot;
	int deg1=0;
	int deg2=0;
	int d=0;
//	int c1=0;
	double l,v;
	Thread thr;
    private Image picture;
    Choice ch1,ch2,ch3,ch4;
    int g=0;
    Panel p1,p2;

	public void actionPerformed(ActionEvent e) {

		s=e.getActionCommand();

	 t1=Integer.parseInt(ch1.getSelectedItem());
	 t3=Integer.parseInt(ch3.getSelectedItem());
	 a=ch2.getSelectedItem();
	if(a=="January")
	t2=0;mn=31;
	if(a=="February")
	t2=31;mn=28;
	if(a=="March")
	t2=59;mn=31;
	if(a=="April")
```

```
t2=90;mn=30;
 if(a=="May")
t2=120;mn=31;
 if(a=="June")
t2=151;mn=30;
 if(a=="July")
t2=181;mn=31;
if(a=="August")
t2=212;mn=31;
 if(a=="September")
t2=243;mn=30;
 if(a=="October")
t2=273;mn=31;
 if(a=="November")
t2=304;mn=30;
 if(a=="December")
t2=334;mn=31;

 System.out.println((String.valueOf(t1)));
 start();
}

    public void start() {
        thr=new Thread(this);
        thr.start();
    }

    public void run() {

        calc_days=(float)(((t1+990)*(365.24219)+t2+t3));
        calc_years=(float)((calc_days/365.24219));
        calc_met_rot=(float)((calc_years*0.2631579)%360);
        calc_cal_rot=(float)((calc_met_rot*0.05*360)%360);
        calc_sar_rot=(float)((calc_years*0.2218551)%360);
        calc_exe_rot=(float)((calc_sar_rot*0.08333*360)%360);
      n=(int)calc_days;

 deg1=(int)(36+(calc_cal_rot));
 deg2=(int)(calc_exe_rot+171)%360;
 System.out.println((String.valueOf(n)));
 System.out.println("met="+(String.valueOf(calc_met_rot))+"
cal="+(String.valueOf(calc_cal_rot))+"          sar="+(String.valueOf(calc_sar_rot))+"
exe="+(String.valueOf(calc_exe_rot)));
 if (s==str){

m=0;
        while(m<=calc_met_rot) {
        e1+=1;
        m+=1;
        repaint();
        try {
```

```
                thr.sleep(30);
            }
            catch (InterruptedException e) {
                showStatus("Diakopi");
            }
        }
            tf1.setText(String.valueOf(
(int)((calc_days%6939.68865)/29.53059)));

            m=0;
                while(m<=calc_cal_rot) {
            e2+=1;
            m+=1;
            repaint();
            try {
                thr.sleep(30);
            }
            catch (InterruptedException e) {
                showStatus("Diakopi");
            }
        }
        if(0<=deg1&&90>deg1)
          d=1;
        if(90<=deg1&&180>deg1)
          d=2;
        if(180<=deg1&&270>deg1)
          d=3;
        if(270<=deg1&&360>deg1)
          d=4;
          tf2.setText(String.valueOf(d));

        m=0;
            while(m<=calc_sar_rot) {
            e3+=1;
            m+=1;
            repaint();
            try {
                thr.sleep(30);
            }
            catch (InterruptedException e) {
                showStatus("Diakopi");
            }
        }
            tf3.setText(String.valueOf((int)((calc_days%6585.32)/29.53059)));

        m=0;
            while(m<=calc_exe_rot) {
            e4+=1;
            m+=1;
            repaint();
            try {
```

```
                        thr.sleep(30);
                    }
                    catch (InterruptedException e) {
                        showStatus("Diakopi");
                    }
            }int c1=9;
            if(0<=deg2&&deg2<90)
            c1=1;
            if(90<=deg2&&deg2<210)
            c1=2;
            if(210<=deg2&&deg2<330)
            c1=3;
            if(330<=deg2&&deg2<360)
            c1=1;
            tf4.setText(String.valueOf(c1));
                    repaint();
            }

}

            public void paint(Graphics g) {

        if(!g.drawImage(this.picture, 0, 0, this)) {
        g.drawString("Error finding Picture.", 25, 50);
       }

                    if (s==str){

                    g.setColor(Color.red);
//                  g.setColor(Color.orange);
                    //deiktes
                    g.fillArc(20,75,250,250,e1,6);
                    g.fillArc(165,200,75,75,e2,6);
                    g.fillArc(280,75,250,250,e3,6);
                    g.fillArc(425,200,75,75,e4,6);
                    g.setColor(Color.black);
            //kykloi
                    g.drawOval(20,75,250,250);
                    g.drawOval(165,200,75,75);
                    g.drawOval(280,75,250,250);
                    g.drawOval(425,200,75,75);
            //      for(int y=1;y<t2;y++)

                g.setColor(Color.black);
                    g.drawString("METONIC",80,350);
                    g.drawString("CALLIPIC",180,350);
                    g.drawString("SAROS",280,350);
                    g.drawString("EXELIGMOS",380,350);
 /*  fillArc(TopCornerX, TopCornerY, width, height, arcStart, arcStop);

  *
```

```
*g.drawLine(x,y,Math.round((float)(x+100*Math.sin(i))),
        Math.round((float)(y+100*Math.cos(i))));}
                g.drawLine(x,y,Math.sin(),y);*/
                g.drawString(String.valueOf(n),15,15);
         }
         }

         public void init() {

                this.picture = this.getImage(this.getDocumentBase(), "anti.jpg");

         str="      Evaluate      ";
            Button b1;
            Label lb1;
            Label lb2,lb3;
           setLayout(new BorderLayout());
            p1=new Panel();
            p2=new Panel();
            lb1=new Label("Year");
            lb2=new Label("Month");
            lb3=new Label("Day");

                this.add(lb1);
                tf1=new TextField(10);
                add(tf1);
                tf2=new TextField(10);
                add(tf2);
                tf3=new TextField(10);
                add(tf3);
                tf4=new TextField(10);
                add(tf4);

                ch1=new Choice();

                for (g=-990;g<0;g++)
                {
                        ch1.add(String.valueOf(g));
                }
                for (g=1;g<3000;g++)
                {
                        ch1.add(String.valueOf(g));
                }

                this.add(ch1);

         this.add(lb2);

                ch2=new Choice();
                ch2.add("January");
                ch2.add("February");
                ch2.add("March");
```

```
        ch2.add("April");
        ch2.add("May");
        ch2.add("June");
        ch2.add("July");
        ch2.add("August");
        ch2.add("September");
        ch2.add("October");
        ch2.add("November");
        ch2.add("December");

        this.add(ch2);

        ch3=new Choice();

        for (int w=1;w<32;w++)
        {
                ch3.add(String.valueOf(w));
        }

        this.add(lb3);

        this.add(ch3);
 //     tf2=new TextField(10);
 //   add(tf2);
//    tf2.addActionListener(this);
        b1=new Button(str);
        add(b1);
        b1.addActionListener(this);
        p1.add(lb1);
        p1.add(ch1);
        p1.add(lb2);
        p1.add(ch2);
        p1.add(lb3);
        p1.add(ch3);
        p1.add(b1);
        p2.add(tf1);
        p2.add(tf2);
        p2.add(tf3);
        p2.add(tf4);
        this.add(p1,BorderLayout.NORTH);
        this.add(p2,BorderLayout.SOUTH);

        }
}
```