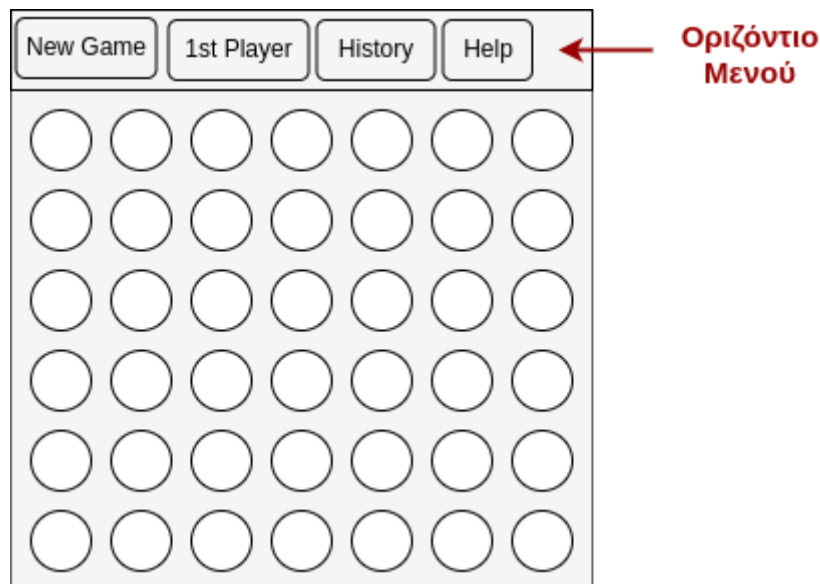


Connect 4 - Παιχνίδι Στρατηγικής

Ημερομηνία παράδοσης: Τρίτη 16 Μαΐου 2023

Σε αυτή την εργασία θα γράψετε ένα πρόγραμμα με τίτλο που υλοποιεί το παιχνίδι στρατηγικής Connect-4 (Σκορ 4). Το πρόγραμμα σας θα πρέπει να υλοποιεί ένα έξυπνο αλγόριθμο επίλυσης του παιχνιδιού με τη βοήθεια του αλγορίθμου τεχνητής νοημοσύνης που συνιστάται για παιχνίδια στρατηγικής [MinMax with alpha-beta pruning](#) (σημείωση: είναι υποχρεωτικό να χρησιμοποιήσετε alpha-beta pruning).

Το πρόγραμμα θα βρίσκεται στο πακέτο `ce326.hw3` και η κλάση εντός του πακέτου αυτού που θα περιέχει τη στατική μέθοδο `main` θα ονομάζεται `connect4`. Το πρόγραμμα υλοποιεί ένα GUI σε Java Swing το οποίο στο κύριο παράθυρο του απεικονίζει τον καμβά του παιχνιδιού που αποτελείται από έξι γραμμές και 7 στήλες, μέσω του οποίου ο χρήστης θα μπορεί να παίξει το παιχνίδι. Ένα σχεδιάγραμμα του παραθύρου φαίνεται στο παρακάτω σχήμα. Τα κουμπιά που εμφανίζονται στην κορυφή του παραθύρου αντιστοιχούν στις επιλογές του οριζόντιου μενού και εξηγούνται στη συνέχεια.



Εφόσον ξεκινήσει μία παρτίδα (παιχνίδι), ο χρήστης παίζει εναντίον του AI module της εφαρμογής προσπαθώντας πάντα να κερδίσει τον υπολογιστή.

Το οριζόντιο μενού που φαίνεται στο σχήμα έχει τις εξής επιλογές.

- **New Game:** Το μενού αυτό είναι το σημείο έναρξης ενός νέου παιχνιδιού. Επιλέγοντας το, εμφανίζεται μία λίστα με τα επίπεδα δυσκολίας που μπορεί να επιλέξει ο παίκτης. Τα επίπεδα αυτά εμφανίζονται με τη μορφή JMenuItems και είναι τα εξής:
 - **Trivial:** Το AI module κοιτάζει σε βάθος μίας κίνησης.
 - **Medium:** Το AI module κοιτάζει σε βάθος τριών κινήσεων.
 - **Hard:** Το AI module κοιτάζει σε βάθος πέντε κινήσεων.

Επιλέγοντας από το μενού το επίπεδο δυσκολίας το παιχνίδι ξεκινά. Εάν υπάρχει παιχνίδι σε εξέλιξη το τρέχον παιχνίδι διακόπτεται χωρίς να καταγραφεί (δες [παρακάτω](#)) και ξεκινάει νέο από την αρχή.

- **1st Player:** Εμφανίζεται ένα μενού δύο επιλογών μέσω menu radio button με τις επιλογές "AI" και "You". Η επιλογή AI είναι προεπιλεγμένη κατά την εκκίνηση του προγράμματος και δίνει την πρώτη

κίνηση κάθε νέου παιχνιδιού στο AI module, ενώ η επιλογή You στον χρήστη της εφαρμογής. Η επιλογή "AI" ή "You" δεν έχει καμία επίδραση στο τρέχον παιχνίδι και αφορά την έναρξη του επόμενου παιχνιδιού που θα γίνει μέσω του μενού "New Game".

- **History:** Από το παράθυρο "εξαφανίζεται" ο καμβάς του παιχνιδιού και εμφανίζεται μία scrollable λίστα (μέσω [JList](#)) με τα παιχνίδια που έχουν ολοκληρωθεί στο παρελθόν. Για κάθε παιχνίδι που έχει ολοκληρωθεί πρέπει να εμφανίζεται η εξής πληροφορία α) ημερομηνία και ώρα έναρξης, β) το επίπεδο δυσκολίας του AI module και γ) αν κέρδισε το AI module ή ο τελικός χρήστης. Ενδεικτικά αναφέρεται η παρακάτω μορφή απεικόνισης των ολοκληρωμένων παιχνιδιών:

- 2023.03.28 - 16:21 L: Hard W: AI
- 2023.03.28 - 14:55 L: Medium W: P
- 2023.03.22 - 10:43 L: Trivial W: AI

Τα παιχνίδια κατατάσσονται στη λίστα με χρονολογικά φθίνουσα σειρά, όπου το πιο πρόσφατο είναι στην κορυφή της λίστας. Πατώντας διπλό αριστερό κλικ πάνω σε οποιοδήποτε παιχνίδι, εμφανίζεται και πάλι ο καμβάς. Στον καμβά αυτό εξελίσσεται το ιστορικό του επιλεγμένου παιχνιδιού. Κατά την αναπαράσταση, κάθε κίνηση πρέπει απέχει από την προηγούμενη τρία (3) δευτερόλεπτα. ~~Σε κάθε νέα κίνηση το πούλι που τοποθετείται πρέπει να φαίνεται ότι πέφτει στην επιλεγμένη στήλη ή εναλλακτικά πρέπει αρχικά να εμφανίζεται με διαφοροποιημένο χρώμα (διαφορετικό κόκκινο ή διαφορετικό κίτρινο), ώστε να είναι ξεκάθαρο στον τελικό χρήστη σε ποια στήλη αυτό τοποθετήθηκε.~~ Ο χρήστης παίζει πάντα με τα κόκκινα πούλια και το AI module με τα κίτρινα.

- **Help:** Το μενού είναι απενεργοποιημένο ή δεν εμφανίζεται καθόλου.

Σημείωση: Οποτεδήποτε ο χρήστης επιλέξει "New Game" από το οριζόντιο μενού εμφανίζεται και πάλι ο καμβάς και ξεκινά ένα νέο παιχνίδι. Οποτεδήποτε ο χρήστης επιλέξει "History" από το οριζόντιο μενού εμφανίζεται η λίστα με τα παιχνίδια που έχουν ολοκληρωθεί στο παρελθόν. Και στις δύο περιπτώσεις το τρέχον παιχνίδι διακόπτεται, κάτι που δε συμβαίνει εάν ο χρήστης πατήσει το μενού 1st Player και αλλάξει τη σχετική επιλογή.

Εξέλιξη του παιχνιδιού

Ο χρήστης παίζει πάντοτε με τα κόκκινα πούλια και το AI module με τα κίτρινα. Ο χρήστης μπορεί να επιλέξει την επόμενη κίνηση του με δύο τρόπους:

- Πατώντας έναν αριθμό από το πληκτρολόγιο μεταξύ 0 και 6. Ο αριθμός 0 αντιστοιχεί στην αριστερότερη στήλη και ο αριθμός 6 στη δεξιότερη.
- Κάνοντας διπλό αριστερό κλικ σε οποιοδήποτε σημείο στη στήλη στην οποία θέλει να ρίξει το πούλι του.

~~Κατά την κίνηση του χρήστη ή του AI module το πούλι που τοποθετείται πρέπει να φαίνεται ότι πέφτει στην επιλεγμένη στήλη ή εναλλακτικά πρέπει αρχικά να εμφανίζεται με διαφοροποιημένο χρώμα (διαφορετικό κόκκινο ή διαφορετικό κίτρινο) για ένα δευτερόλεπτο (msec), ώστε να είναι σαφής η επόμενη κίνηση.~~

Καθώς το παιχνίδι εξελίσσεται, το AI module προσπαθεί να ανιχνεύσει σε κάθε κίνηση κατά πόσο το ίδιο ή ο χρήστης έχουν καταφέρει να κερδίσουν το παιχνίδι. Εφόσον, ανιχνευθεί μία κατάσταση νίκης - δηλαδή 4 πούλια ίδιου χρώματος οριζόντια, κάθετα ή διαγώνια - το παιχνίδι σταματά και εμφανίζεται μέσω modal box ένα μήνυμα που πληροφορεί τον χρήστη για τον νικητή ("You won!" αν είναι ο νικητής ή "You lost!" αν είναι ο ηττημένος). Όταν το παιχνίδι τελειώσει, ο χρήστης μπορεί να επιλέξει να ξαναπαίξει, επιλέγοντας από το οριζόντιο μενού το επίπεδο δυσκολίας του νέου παιχνιδιού ή μπορεί να πάει στο ιστορικό των παιχνιδιών για να δει πως εξελίχθηκε ένα παιχνίδι.

Καταγραφή των παιχνιδιών

Μέσα στο κατάλογο του χρήστη (home directory) του χρήστη πρέπει να υπάρχει ένας κατάλογος (directory) με το όνομα `connect4`, μέσα στον οποίο θα υπάρχουν ένα ή περισσότερα αρχεία καταγραφής (η επιλογή αφήνεται σε εσάς) για τις κινήσεις που έχουν γίνει για τα παιχνίδια που ολοκληρώθηκαν στο παρελθόν. Εάν ο κατάλογος `connect4` δεν υπάρχει θα δημιουργείται κατά την εκκίνηση της εφαρμογής. Η μορφή της καταγραφής είναι ελεύθερη, αρκεί να περιέχει όλη την πληροφορία ώστε να μπορούν να αναζητηθούν τα παιχνίδια που έχουν ολοκληρωθεί και οι κινήσεις που έγιναν με τη σειρά στα παιχνίδια αυτά. Προτείνεται η μορφή καταγραφής να είναι JSON (για την ανάγνωση και την εγγραφή χρησιμοποιείτε τη βιβλιοθήκη `org.json`) ή XML. Καταγράφονται μόνο ολοκληρωμένα παιχνίδια και όχι παιχνίδια τα οποία δεν κατέληξαν σε νικητή ή δεν ήρθαν ισόπαλα.

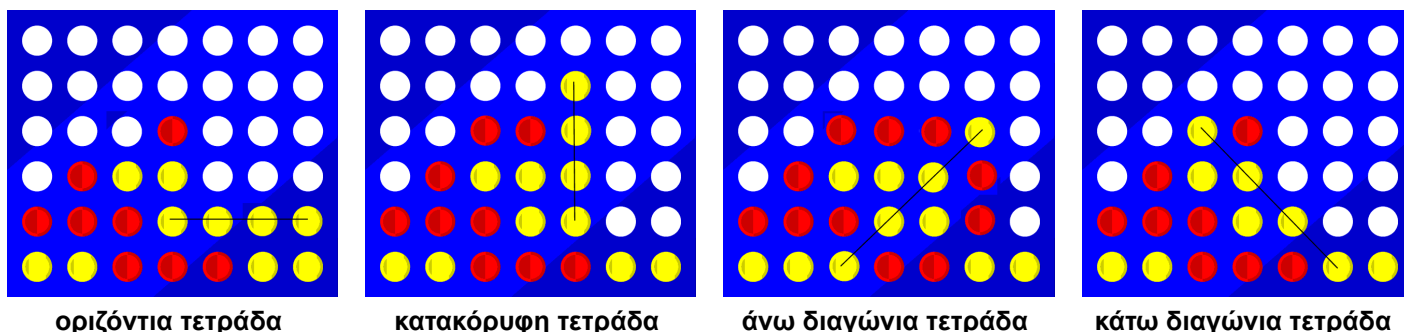
Μπορείτε να έχετε πρόσβαση στο όνομα του καταλόγου του τρέχοντος χρήστη μέσω της εντολής:

```
System.getProperty("user.home")
```

Αξιολόγηση κάθε κίνησης κατά τη διάρκεια του παιχνιδιού

Το AI module θα πρέπει να είναι σε θέση να επιλέξει τις δικές του βέλτιστες κινήσεις. Η προτεινόμενη στρατηγική είναι η εξής. Για την εκτίμηση κάθε κίνησης του AI module, αθροίζονται οι θέσεις από τα κίτρινα πούλια που συμβάλουν στη νίκη του AI module και αφαιρούνται οι θέσεις από τα κόκκινα που συμβάλλουν στην ήττα του.

Ένα παιχνίδι μπορεί να κερδηθεί ή να χαθεί όταν υπάρχουν τέσσερα πούλια ίδιου χρώματος οριζόντια ή κάθετα ή διαγώνια προς τα κάτω ή διαγώνια προς τα πάνω, όπως φαίνεται στα σχήματα που ακολουθούν.



Σε κάθε κίνηση εξετάζουμε τα δυνατά παράθυρα στα οποία μπορούν να σχηματιστούν τετράδες ίδιου χρώματος οριζόντια, κάθετα ή κατά τις δύο διαγώνιες κατευθύνσεις. Τα δυνητικά παράθυρα έχουν εξής:

- 6 γραμμές, από 4 δυνητικές οριζόντιες τετράδες σε κάθε γραμμή → 24 δυνητικές τετράδες
- 7 στήλες, από 3 δυνητικές κάθετες τετράδες σε κάθε στήλη → 21 δυνητικές τετράδες
- 3 γραμμές, από 4 δυνητικές άνω διαγώνιες τετράδες σε κάθε γραμμή → 12 δυνητικές τετράδες
- 3 γραμμές, από 4 δυνητικές κάτω διαγώνιες τετράδες σε κάθε γραμμή → 12 δυνητικές τετράδες

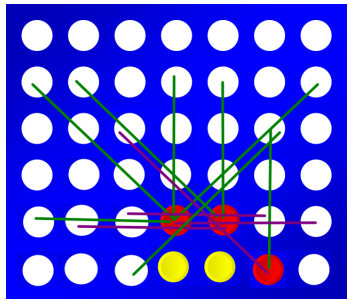
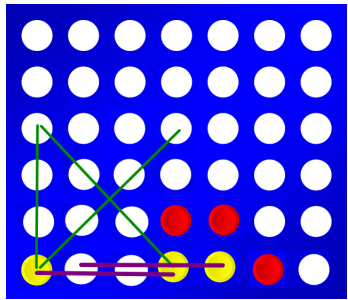
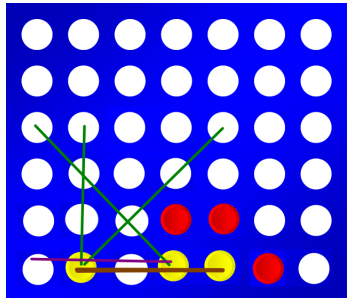
Κάθε παράθυρο ή δυνητική τετράδα αποτιμάται ως εξής:

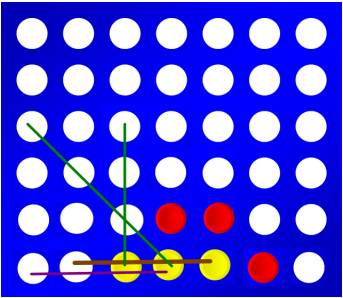
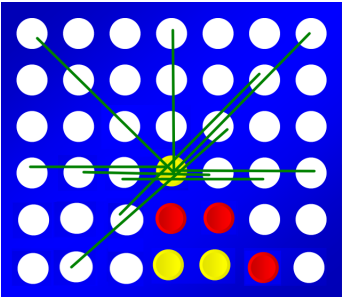
- Εάν σε μία δυνητική τετράδα έχουμε τέσσερα κενά ή έχουμε πούλια διαφορετικού χρώματος (και πιθανόν κάποια κενά) η αποτίμηση της είναι ίση με 0.
- Εάν σε μία δυνητική τετράδα έχουμε μόνο ένα πούλι και τρία κενά, τότε η αποτίμηση αυτής της τετράδας είναι ίση με 1.
- Εάν σε μία δυνητική τετράδα έχουμε δύο πούλια ίδιου χρώματος, και δύο κενά τότε η αποτίμηση της τετράδας είναι ίση με 4.
- Εάν σε μία δυνητική τετράδα έχουμε τρία πούλια ίδιου χρώματος, και ένα κενό τότε η αποτίμηση της τετράδας είναι ίση με 16.

- Εάν σε μία δυνητική τετράδα έχουμε τέσσερα πούλια ίδιου χρώματος, και κανένα κενό τότε η αποτίμηση της τετράδας είναι ίση με 10000 ή περισσότερο.
- ~~Ανάμεσα σε δύο κινήσεις που έχουν το ίδιο βέλτιστο όφελος/κόστος επιλέγουμε εκείνη που είναι πιο κοντά στη μεσαία στήλη~~ (αγνοήστε αυτό τον περιορισμό).

Οι κινήσεις του χρήστη με τα κόκκινα πούλια υπολογίζονται με τον ίδιο τρόπο, αλλά με αρνητικό πρόσημο αντί για θετικό. Σε κάθε κίνηση αθροίζουμε το όφελος και αφαιρούμε το κόστος από όλες τις δυνητικές τετράδες, όπως αυτές περιγράφονται από τα διαφορετικά παράθυρα που ορίστηκαν παραπάνω.

Στο παράδειγμα που ακολουθεί η δυνητικές τετράδες που έχουν κόστος-όφελος 1 εμφανίζονται με **πράσινο**, οι δυνητικές τετράδες που έχουν κόστος-όφελος 4 με **φούξια - ροζ** και οι δυνητικές τετράδες με κόστος-όφελος 16 με **καφέ**.

σ τ ή λ η		<p>Στο διπλανό παράδειγμα, διερευνούμε το όφελος/κόστος το AI να τοποθετήσει το κίτρινο πούλι σε μία από τις στήλες 0 (αριστερότερη στήλη) έως 3 (μεσαία στήλη).</p> <p>Το κόστος (όφελος για τον αντίπαλο με τα κόκκινα πούλια, δηλαδή τον τελικό χρήστη) είναι:</p> <ul style="list-style-type: none"> • τρεις οριζόντιες δυνητικές τετράδες με δύο πούλια (κόστος $3 \times 4 = 12$), • μία οριζόντια δυνητική τετράδα με ένα πούλι (κόστος 1), • μία κάτω διαγώνια δυνητική τετράδα με δύο πούλια (κόστος 4), • δύο κάτω διαγώνιες δυνητικές τετράδες με ένα πούλι (κόστος $2 \times 1 = 2$), • δύο άνω διαγώνιες δυνητικές τετράδες με 1 πούλι (κόστος $1 \times 2 = 2$), • 3 κατακόρυφες δυνητικές τετράδες (κόστος $3 \times 1 = 3$). <p>Σύνολο $12 + 1 + 4 + 2 + 2 + 3 = 24$</p>
0		<p>Τοποθετώντας στη στήλη 0 δημιουργούνται στη γραμμή 0 δύο δυνητικές τετράδες με δύο πούλια η κάθε μία (όφελος $4 + 4 = 8$). Ξεκινώντας από τη στήλη 0 δημιουργείται μία κατακόρυφη και μία άνω διαγώνια δυνητική τετράδα οφέλους 1 η κάθε μία ($2 \times 1 = 2$). Τέλος, ξεκινώντας από τη στήλη 3 δημιουργείται μία κάτω διαγώνια δυνητική τετράδα οφέλους 1.</p> <p>Το τελικό όφελος είναι $8 + 2 + 1 - 24 = -13$</p>
1		<p>Τοποθετώντας στη στήλη 1 δημιουργείται μία δυνητική τετράδα με 3 πούλια με όφελος 16 και μία δυνητική τετράδα με δύο πούλια και όφελος 4. Επιπλέον, ξεκινώντας από τη στήλη 1 δημιουργείται μία κατακόρυφη και μία άνω διαγώνια δυνητική τετράδα οφέλους 1 η κάθε μία ($2 \times 1 = 2$). Τέλος, ξεκινώντας από τη στήλη 3 δημιουργείται μία κάτω διαγώνια δυνητική τετράδα οφέλους 1.</p> <p>Το τελικό όφελος είναι $16 + 4 + 2 + 1 - 24 = 23 - 24 = -1$</p>

2		<p>Τοποθετώντας στη στήλη 2 δημιουργείται μία δυνητική τετράδα με 3 πούλια με όφελος 16 και μία δυνητική τετράδα με δύο πούλια και όφελος 4. Επιπλέον, ξεκινώντας από τη στήλη 1 δημιουργείται μία κατακόρυφη δυνητική τετράδα οφέλους 1. Τέλος, ξεκινώντας από τη στήλη 3 δημιουργείται μία κάτω διαγώνια δυνητική τετράδα οφέλους 1.</p> <p>Στη συγκεκριμένη περίπτωση το όφελος για τον τελικό χρήστη (κόστος για το AI) είναι μειωμένο κατά 1, διότι με την κίνηση στη θέση 2 για τα κίτρινα, περιορίζονται οι άνω διαγώνιες τετράδες για τα κόκκινα σε μία αντί για δύο.</p> <p>Το τελικό όφελος είναι $16+4+2 - (24-1) = 22-23 = -1$</p>
3		<p>Τοποθετώντας στη στήλη 3 δημιουργούνται:</p> <p>Στη γραμμή 0:</p> <ul style="list-style-type: none"> • μία οριζόντια δυνητική τετράδα με δύο πούλια και όφελος 4, • μία οριζόντια δυνητική τετράδα με ένα πούλι και όφελος 1 • ξεκινώντας από την κατώτερη θέση της στήλης 3 δημιουργείται μία κάτω διαγώνια δυνητική τετράδα οφέλους 1. <p>Το πούλι που τοποθετήθηκε τελευταίο στη στήλη 3 συμμετέχει στις παρακάτω δυνητικές τετράδες (όλες με ένα πούλι και όφελος 1):</p> <ul style="list-style-type: none"> • μία κάτω (1) διαγώνια δυνητική τετράδα • τέσσερις (4) άνω διαγώνιες δυνητικές τετράδες, • μία (1) κατακόρυφη και • τέσσερις (4) οριζόντιες <p>Επιπλέον, το πούλι που τοποθετήθηκε τελευταίο μειώνει το όφελος του παίκτη με τα κόκκινα πούλια (κόστος για το AI), διότι μειώνει τις κατακόρυφες τετράδες με ένα πούλι κατά 1 και ακυρώνει τις δύο κάτω διαγώνιες δυνητικές τετράδες με τα δύο πούλια και με το ένα πούλι. Επομένως, το ανανεωμένο όφελος για τον χρήστη με τα κόκκινα πούλια (κόστος για το AI) είναι $24 - (1+4+1) = 18$.</p> <p>Το τελικό όφελος είναι $(4+1+1)+(1+3+1+4) - 18 = 15 - 18 = -3$</p>

Οδηγίες Αποστολής

Η αποστολή της εργασίας θα γίνει μέσω της πλατφόρμας autolab. Για την υποβολή ακολουθήστε τα εξής βήματα.

- **Υποβολή κώδικα Java:** Δημιουργήστε ένα κατάλογο με το όνομα **hw3**. Μέσα σε αυτόν τοποθετήστε:
 - όλα τα αρχεία με κατάληξη .java της εργασίας. Υπενθυμίζεται ότι όλες οι κλάσεις πρέπει να ανήκουν στο package **ce326.hw3**.
 - Εφόσον χρησιμοποιείτε εικονίδια, ένα άλλο κατάλογο με όνομα **icons** που περιέχει όλα τα εικονίδια της εργασίας σας. Ο φάκελος **icons** θεωρούμε ότι τοποθετείται στο αρχικό κατάλογο του project (εκεί που βρίσκεται και ο φάκελος **src** που περιέχει τον πηγαίο κώδικα).
 - Ένα αρχείο **README.txt** που να περιέχει οποιαδήποτε άλλη πληροφορία για το project σας.
- Συμπιέστε ~~τον κατάλογο~~ τα περιεχόμενα του καταλόγου **hw3** που δημιουργήσατε παραπάνω. Το αρχείο που προκύπτει πρέπει να έχει όνομα **hw3.zip**.
- Συνδέστε στο **autolab** και επιλέγετε το μάθημα **ECE326_2023 (s23)** και από αυτό την εργασία **HW3**.
- Για να υποβάλετε την εργασία σας κάνετε click στην επιλογή *"I affirm that I have compiled with this course academic integrity policy..."* και πατάτε submit. Στη συνέχεια επιλέγετε το αρχείο **hw3.zip** που δημιουργήσατε παραπάνω.

