



School of
Electrical &
Computer
Engineering

Ενσωματωμένα Συστήματα Μικροεπεξεργαστών

MILESTONE 2-3 ΠΑΙΧΝΙΔΙ ΟΤHELLO

Μποκαλίδης Αναστάσιος – 2014030069

Καδίτης Μανώλης – 2014030136

LAB41140623

Σε αυτό το μέρος του πρότζεκτ υλοποιήθηκε μια βελτιστοποιημένη έκδοση του 2^{ου} μέρους του πρότζεκτ καθώς και κάποιες τεχνικές που θα χρησιμοποιήσουμε στο 3^ο και τελευταίο μέρος. Ουσιαστικά στον κώδικα του προηγούμενου milestone βελτιώσαμε συναρτήσεις-τεχνικές που μας εμπόδιζαν στην περάτωση του τελευταίου milestone.

A. Λειτουργίες Του Συστήματος

- ✓ Αναγνωρίζει όλες τις εντολές από το μενού επιλογών που δίνεται στην εκφώνηση.
- ✓ Μπορεί να απαντήσει κατάλληλα σε όλες αυτές τις εντολές.
- ✓ Ξεκινώντας το παιχνίδι με την εντολή NG(new game) γίνεται αρχικοποίηση του board μας όπως απαιτεί το παιχνίδι. Σε αυτό το μέρος αλλάξαμε την αρχική υλοποίηση του(από 2-διάστατο το μετατρέψαμε σε 1-διάστατο) διότι μας βολεύει παραπάνω σε συγκεκριμένους υπολογισμούς. Το είδος του board είναι int.
- ✓ Συνεχίζοντας , μέσω της SP αναθέτεται το χρώμα παίχτη στο AVR αλλά και στον αντίπαλο του. Αν επιλεγθεί ο AVR να έχει το μαύρο χρώμα τότε ξεκινά αμέσως και τις κινήσεις του.
- ✓ Έπειτα από την πραγματοποίηση της πρώτης κίνησης το σύστημα καταλαβαίνει ποιος παίχτης έχει σειρά για να κάνει την κίνηση του.
- ✓ Όταν γίνει κάποια κίνηση από οποιοδήποτε παίχτη , αν αυτή είναι έγκυρη αποτυπώνεται στο board και ενημερώνεται κάθε φορά ο αντίπαλος του.
- ✓ Σε περίπτωση λάθος κίνησης νικάει ο αντίπαλος παίχτης αυτουνού που έκανε την λάθος κίνηση.
- ✓ Σε περίπτωση που ο αντίπαλος ξεπεράσει τον χρόνο του νικάει το AVR.
- ✓ Αναγνωρίζει το σύστημα μας πότε κάποιος παίχτης έχει διαθέσιμη κίνηση και αν πρέπει να πάει πάσο ή όχι.
- ✓ Για να κάνει κίνηση το AVR ακολουθεί την εξής στρατηγική : ερευνά αν έχει διαθέσιμη νόμιμη κίνηση -> αποθηκεύει όσες νόμιμες κινήσεις έχει βρεί -> χρησιμοποιεί μια τυχαία από αυτές.
- ✓ Τέλος αν έρθει η εντολή EG(end game) αυτόματα μετρούνται πόσα πούλια έχει κάθε παίχτης και έτσι κρίνεται ο νικητής ή ακόμα και η

ισοπαλία. Όποιο και να είναι το αποτέλεσμα ανάβει το κατάλληλο LED ως ένδειξη του αποτελέσματος.

B. Νέες Συναρτήσεις Του Συστήματος

- *int opponent (int player)*
Αυτή η συνάρτηση παίρνει σαν όρισμα το χρώμα ενός παίχτη και μας επιστρέφει το χρώμα του αντιπάλου. Αυτό χρειάζεται καθώς σε επόμενες συναρτήσεις κάθε φορά χρειάζεται διαφορετικός παίχτης για διαφορετική αναζήτηση.
- *int validp (int move)*
Αυτή η συνάρτηση ελέγχει αν η κίνηση που έγινε-θα γίνει είναι μέσα στο πλαίσιο του board.
- *int findbracketingpiece (int square, int player, int dir)*
Αυτή η συνάρτηση ψάχνει προς κάποια κατεύθυνση από κάποιο δοθέν πούλι αν υπάρχει κάποιο πούλι ίδιου χρώματος μετά από ακολουθία αντίπαλων πούλιων και επιστρέφει αυτό το πούλι.
- *int wouldflip (int move, int player, int dir)*
Αυτή η συνάρτηση ετοιμάζει την θέση από την οποία θα αρχίσει την έρευνα η προηγούμενη συνάρτηση.
- *int legalp (int move, int player)*
Αυτή η συνάρτηση ψάχνει να βρει αν η κίνηση που θα γίνει ή έγινε είναι νόμιμη. Ως κριτήριο έχει η κίνηση να βρίσκεται εντός του πίνακα , να μην είναι κατειλημμένη από κάποιο άλλο πούλι και να έχει προς κάποια κατεύθυνση μετά από κάποια γειτονικά πούλια ένα πούλι ίδιο με αυτό του παίχτη που έκανε την κίνηση.
- *void makeflips (int move, int player, int dir)*
Αυτή η συνάρτηση κάνει τις αλλαγές των πούλιων που περικυκλώνονται από πούλια αντιπάλου.
- *void makemove (int move, int player)*
Αυτή η συνάρτηση παίρνει ως όρισμα τον παίχτη και την κίνηση που έκανε και ετοιμάζει την διαδικασία να βρει πόσα και ποια πούλια πρέπει να αντιστρέψει.
- *int anylegalmove (int player)*

Αυτή η συνάρτηση ελέγχει αν κάποιος παίχτης έχει τουλάχιστον 1 νόμιμη κίνηση όπου και επιστρέφει 1, αλλιώς 0.

- *int legalmoves (int player)*

Αυτή η συνάρτηση μαζεύει όλες τις νόμιμες κινήσεις που μπορεί να κάνει ο AVR και έπειτα επιλέγει μία τυχαία από όλες αυτές.

- *int returnposition (unsigned char x,unsigned char y)*

Αυτή η συνάρτηση παίρνει σαν όρισμα τα γράμματα από την θέση-κίνηση που θα λάβει το AVR και θα την μετατρέψει σε ακέραιο αριθμό integer που θα αντιστοιχεί σε κάποια θέση του 1-διάστατου πίνακα μας.

- *void printposition(int x)*

Αυτή η συνάρτηση όταν το AVR ολοκληρώσει την νόμιμη κίνηση του τότε μετατρέπει την θέση του πίνακα σε ένα γράμμα και ένα αριθμό ώστε να μπορέσει να στείλει στο αντίπαλό του την κίνηση MM<SP>A-H,1-9<CR>.

- *int count (int player)*

Αυτή η συνάρτηση μετράει τα πούλια ενός παίχτη πάνω στο board.

- *void CountAndFinish()*

Αυτή η συνάρτηση κρίνει σύμφωνα με τους υπολογισμούς της προηγούμενης συνάρτησης ποιο θα είναι το αποτέλεσμα του παιχνιδιού.

C. Βελτιώσεις Και Μελέτη Στο Επόμενο Milestone

- ❖ Θα βγάλουμε την στρατηγική ώστε να επιλέγει το σύστημα μας μια τυχαία κίνηση και θα χρησιμοποιήσουμε τον αλγόριθμο minmax (Alpha-Beta pruning). Έχει μελετηθεί το σχέδιο της στρατηγικής και ελάχιστα σε κώδικα αλλά δεν μπορεί να ανταπεξέλθει ακόμα στο σύστημα γιατί πρέπει να μελετήσουμε την κατανάλωση μνήμης που θα υποστεί. Θα προσπαθήσουμε να βρούμε την βέλτιστη κίνηση έως και 2 κινήσεις μπροστά αν βέβαια δεν μας παρουσιαστεί κάποιο πρόβλημα χώρου της μνήμης.

- ❖ Έχουμε σκεφτεί να υλοποιήσουμε και κάποιες λειτουργίες που ίσως έχει το interface του υπολογιστή που θα χρησιμοποιηθεί για τον διαγωνισμό.
- ❖ Το υποσύστημα με το ρολόι για τον χρόνο κίνησης το έχουμε τεστάρει απομονωμένα και λειτουργεί αλλά σε αυτό το μέρος το αφαιρέσαμε. Θα το προσθέσουμε στο τελικό interface του συστήματος.
- ❖ Επίσης στο τελικό interface του συστήματος θα προσθέσουμε και το εξωτερικό κρύσταλλο.

~~Σας έχουμε ενημερώσει ότι το τελικό πρότζεκτ θα το παρουσιάσουμε για την εξεταστική του Σεπτεμβρίου καθώς θέλουμε να φτάσουμε και να τελειοποιήσουμε ό,τι σκέψεις και σχέδια έχουμε κάνει και να μην πάει χαμένος ο χρόνος που αφιερώσαμε.~~