Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

HUNGER GAMES

MAY THE ODDS BE EVER IN YOUR FAVOR

ΜΕΡΟΣ Δ

Φωτεινή Σαββίδου
(9657, sfoteini@ece.auth.gr)

Δομές Δεδομένων

Ακαδημαϊκό έτος
2019 - 2020

Πίνακας Περιεχομένων

Περιγραφή του παιχνιδιού Hunger Games

Το Hunger Games είναι ένα τηλεοπτικό παιχνίδι επιβίωσης. Οι παίκτες διαγωνίζονται σε ένα νησί του οποίου η διάμετρος μειώνεται με την πάροδο του χρόνου. Νικητής του παιχνιδιού αναδεικνύεται ο παίκτης που θα καταφέρει να παραμείνει ζωντανός και να επιβιώσει από τους κινδύνους που εγκυμονεί το νησί. Προκειμένου να το πετύχει αυτό πρέπει να αναζητήσει τροφή και όπλα, τα οποία βρίσκονται στο κέντρο του νησιού. Τα όπλα είναι απαραίτητα για την αντιμετώπιση των παγίδων που είναι κρυμμένες στο νησί, αλλά και για την εξόντωση των αντιπάλων.

Στην παρούσα εργασία υλοποιούμε μια απλουστευμένη προσομοίωση του παιχνιδιού Hunger Games. Συγκεκριμένα, δύο παίκτες διαγωνίζονται σε ένα ταμπλό αρχικής διάστασης 20×20. Σε κάθε γύρο του παιχνιδιού οι παίκτες μετακινούνται τυχαία κατά μία θέση (δεξιά, αριστερά, πάνω, κάτω, διαγώνια). Σκοπός τους είναι να φτάσουν στο κέντρο του ταμπλό, όπου είναι τοποθετημένα τα τρόφιμα και τα όπλα. Για να το πετύχουν, όμως, αυτό θα πρέπει να αποφύγουν τις διάφορες παγίδες που είναι τοποθετημένες περιμετρικά των εφοδίων. Παράλληλα, η διάσταση του ταμπλό μικραίνει και το παιχνίδι λήγει όταν το ταμπλό αποκτήσει διάσταση 4×4. Νικητής του παιχνιδιού αναδεικνύεται ο παίκτης που θα παραμείνει ζωντανός ή αυτός που θα συγκεντρώσει τους περισσότερους πόντους (σε περίπτωση που και οι δύο παίκτες επιβιώσουν).

Ένας παίκτης συγκεντρώνει πόντους όταν συλλέγει ένα εφόδιο, όταν δηλαδή μετακινηθεί σε ένα πλακίδιο του ταμπλό που περιλαμβάνει ένα εφόδιο. Αντίθετα, οι παίκτες χάνουν πόντους όταν συναντούν μια παγίδα και δεν διαθέτουν το κατάλληλο όπλο για να την αντιμετωπίσουν. Υπάρχουν τρία είδη όπλων: τόξο, σπαθί και πιστόλι. Το τόξο χρησιμοποιείται για την αντιμετώπιση των ζώων, το σπαθί για την αντιμετώπιση των παγίδων που περιλαμβάνουν σχοινιά, ενώ το πιστόλι για την εξόντωση του αντιπάλου.

Υλοποίηση του παιχνιδιού σε Java

Η προσομοίωση του παιχνιδιού Hunger Games βασίζεται στη δημιουργία ενός ταμπλό μεγέθους 20×20, που θα περιλαμβάνει συγκεκριμένο αριθμό όπλων, εφοδίων και παγίδων και δύο παικτών, οι οποίοι θα έχουν τη δυνατότητα να μετακινούνται, να συλλέγουν όπλα και τρόφιμα και να αντιμετωπίζουν παγίδες. Το τέταρτο μέρος της εργασίας αφορά την υλοποίηση των γραφικών του παιχνιδιού. Για τη δημιουργία του γραφικού περιβάλλοντος χρησιμοποιήθηκε η βιβλιοθήκη Swing, δημιουργήθηκαν τέσσερις επιπλέον κλάσεις, οι οποίες αναπαριστούν τα γραφικά στοιχεία και τροποποιήθηκαν οι συναρτήσεις των υπαρχουσών κλάσεων ώστε να ανταποκρίνονται στα νέα δεδομένα.

Στην αρχική οθόνη ο χρήστης επιλέγει το είδος των δύο παικτών (Random, Heuristic, Min Max player) και με το πάτημα του κουμπιού Play ξεκινάει το παιχνίδι και εμφανίζεται το ταμπλό του παιχνιδιού καθώς και πληροφορίες σχετικά με την κατάσταση των δύο παικτών (θέση, πόντοι, κατοχή όπλων). Έπειτα, με το πάτημα του κουμπιού Next round αρχίζει ο επόμενος γύρος του παιχνιδιού, ανανεώνεται το ταμπλό (μετακίνηση παικτών, διαγραφή όπλων και τροφίμων που έχουν πάρει οι παίκτες) και οι πληροφορίες των παικτών και εμφανίζεται ενημερωτικό μήνυμα σχετικά με την εξέλιξη του γύρου. Στο τέλος του παιχνιδιού εμφανίζεται κατάλληλο μήνυμα για την ανάδειξη του νικητή.

Οι κλάσεις Food, Trap, Weapon

Στις κλάσεις Food, Trap και Weapon, που αντιπροσωπεύουν τα τρόφιμα, τις παγίδες και τα όπλα αντίστοιχα, προστίθεται η μεταβλητή imgSrc, στην οποία αποθηκεύεται η διεύθυνση της εικόνας του αντικειμένου που αναπαριστά η κλάση και ορίζονται οι κατάλληλες συναρτήσεις get και set. Ακόμα, η συνάρτηση *loadImgSrc()* υπολογίζει και επιστρέφει την διεύθυνση της εικόνας.

Οι κλάσεις Player, HeuristicPlayer, MinMaxPlayer

Στις κλάσεις Player, HeuristicPlayer και MinMaxPlayer, που αντιπροσωπεύουν τους τρεις δυνατούς τύπους παικτών, προστίθεται η μεταβλητή imgSrc, στην οποία, όπως αναφέρθηκε παραπάνω, αποθηκεύεται η διεύθυνση της εικόνας που αναπαριστά τον παίκτη και η μεταβλητή dead, η οποία μπορεί να πάρει τις τιμές false ή true ανάλογα με το αν ο παίκτης είναι ζωντανός ή νεκρός αντίστοιχα. Ορίζονται οι συναρτήσεις get και set για τις νέες μεταβλητές και η συνάρτηση *loadImgSrc()*, η οποία υπολογίζει και επιστρέφει τη διεύθυνση της εικόνας του παίκτη.

Ακόμα προσθέτουμε στην κλάση Player τις συναρτήσεις *playersDistance()* και *kill()*, οι οποίες υπολογίζουν την απόσταση μεταξύ δύο παικτών και ελέγχουν αν ο παίκτης μπορεί να σκοτώσει τον αντίπαλό του. Αλλάζουμε τη συνάρτηση statistics() των κλάσεων έτσι ώστε να επιστρέφει μία συμβολοσειρά με πληροφορίες σχετικά με την κίνηση του παίκτη στον τρέχων γύρο (αριθμός κίνησης που επέλεξε, νέα θέση, πόντοι που κέρδισε, αριθμός εφοδίων, παγίδων και όπλων -και το είδος τους- που συνέλεξε).

Η κλάση Game

Η κλάση Game περιέχει την υλοποίηση του παιχνιδιού. Το παιχνίδι χαρακτηρίζεται από τον τρέχων γύρο, το ταμπλό (αντικείμενο τύπου Board), τους δύο παίκτες (πίνακας τύπου Player), τον αριθμό του παίκτη που παίζει πρώτος (turn), μία συμβολοσειρά που περιέχει τις πληροφορίες για τις κινήσεις των παικτών στον τρέχων γύρο (gameInfo) και τη μεταβλητή gameOver, η οποία παίρνει την τιμή true όταν το παιχνίδι έχει λήξει.

Υλοποιούμε δύο συναρτήσεις αρχικοποίησης (constructors). Η πρώτη συνάρτηση δεν έχει ορίσματα, θέτει τον γύρο στην τιμή 0 και τις υπόλοιπες μεταβλητές σε null και η δεύτερη δέχεται ως όρισμα έναν ακέραιο αριθμό, οποίος εκχωρείται στη μεταβλητή round.

Στις μεθόδους της κλάσης περιλαμβάνονται ακόμα οι συναρτήσεις εκχώρησης τιμής και επιστροφής της τιμής των μεταβλητών της κλάσης (συναρτήσεις get και set).

Η συνάρτηση *setTurns()* υπολογίζει τον αριθμό του παίκτη που θα παίξει πρώτος με χρήση της συνάρτησης *Math.random()* και ενημερώνει τη μεταβλητή turn της κλάσης.

Η συνάρτηση *start()* είναι η συνάρτηση έναρξης του παιχνιδιού. Δημιουργεί ένα ταμπλό διάστασης 20×20 με 6 όπλα, 10 εφόδια και 8 παγίδες, τυχαία κατανεμημένα στο ταμπλό και ορίζει δύο παίκτες με βάσει τον τύπο παικτών που επιλέγει ο χρήστης από το αρχικό menu (οι τύποι των παικτών μεταβιβάζονται ως ορίσματα στη συνάρτηση). Οι δύο παίκτες βρίσκονται αρχικά στο κάτω δεξιά πλακίδιο του ταμπλό και έχουν σκορ ίσο με 15. Καλείται η συνάρτηση *setTurns()* για τον ορισμό του παίκτη που θα παίξει πρώτος.

Η συνάρτηση *run()* υλοποιεί την εκτέλεση ενός νέου γύρου. Ανάλογα με τον τύπο του πρώτου παίκτη (αυτού που ορίζεται από τη μεταβλητή turn[1]) καλείται η αντίστοιχη συνάρτηση μετακίνησης του παίκτη και ενημερώνεται η μεταβλητή gameInfo από τη

[1] Οι τιμές που μπορεί να πάρει η μεταβλητή turn είναι 0 ή 1. Οι παίκτες αποθηκεύονται στον πίνακα Player[] players. Έτσι, ο πρώτος παίκτης είναι ο players[turn] και ο δεύτερος ο players[1 – turn].

συνάρτηση *statistics()*. Έπειτα ελέγχεται αν ο παίκτης μπορεί να σκοτώσει τον αντίπαλό του. Σε αυτήν την περίπτωση, τίθεται η μεταβλητή dead του αντίπαλου παίκτη στην τιμή true, όπως και η μεταβλητή gameOver, ενημερώνεται η μεταβλητή gameInfo με το κατάλληλο μήνυμα και τερματίζεται το παιχνίδι. Η διαδικασία μετακίνησης του δεύτερου παίκτη υλοποιείται με παρόμοιο τρόπο. Μετά τη μετακίνηση των δύο παικτών ελέγχεται αν ο γύρος του παιχνιδιού είναι ακέραιο πολλαπλάσιο του τρία, οπότε και καλείται η συνάρτηση *resizeBoard()* της κλάσης Board. Τέλος, ελέγχεται αν οι διαστάσεις του ταμπλό είναι μικρότερες ή ίσες του τέσσερα ή αν κάποιος παίκτης έχει αρνητικό σκορ, έτσι ώστε να λήξει το παιχνίδι.

Η συνάρτηση *check()* καλείται μετά τη λήξη του παιχνιδιού για την εύρεση του νικητή. Αν κάποιος παίκτης έχει σκοτωθεί ή έχει αρνητικό σκορ, τότε νικητής είναι ο αντίπαλος παίκτης. Αν δεν ισχύει καμία από τις παραπάνω συνθήκες, τότε νικητής είναι ο παίκτης που έχει τους περισσότερους πόντους, αλλιώς το παιχνίδι αξιολογείται ως ισόπαλο.

Η κλάση Menu

Η κλάση Menu (επεκτείνει το JFrame) δημιουργεί την αρχική σελίδα του παιχνιδιού, η οποία δίνει στον χρήστη τη δυνατότητα να επιλέξει το είδος των δύο παικτών. Η μεταβλητές της κλάσης είναι ένα αντικείμενο JPanel, δύο ομάδες κουμπιών (ButtonGroup), ένα κουμπί (Button) play και δύο συμβολοσειρές στις οποίες αποθηκεύεται το είδος των δύο παικτών.

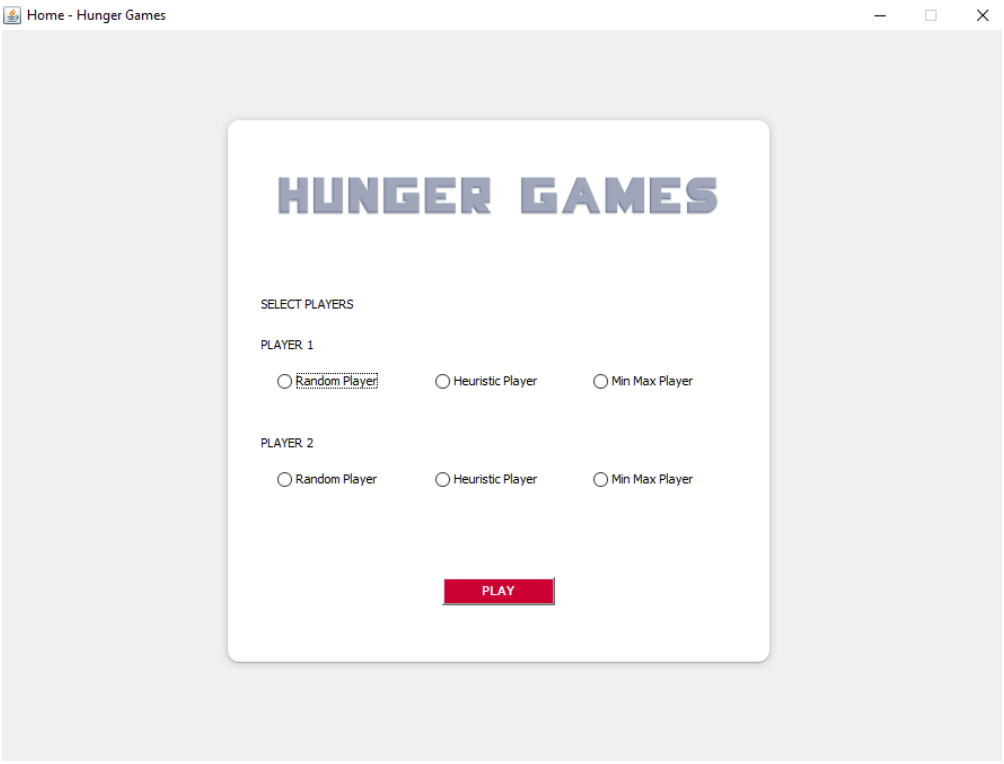
Στην συνάρτηση αρχικοποίησης της κλάσης ορίζονται για κάθε παίκτη τρία κουμπιά JRadioButton που αντιστοιχούν στους τρεις διαφορετικούς τύπους παικτών. Τα τρία αυτά κουμπιά εισάγονται σε ένα ButtonGroup έτσι ώστε να είναι η δυνατή η επιλογή ενός μόνο από αυτά. Εισάγεται ακόμα ένα κουμπί (Button), με το πάτημα του οποίου ξεκινάει το παιχνίδι, αν ο χρήστης έχει επιλέξει δύο παίκτες, αλλιώς εμφανίζεται μήνυμα λάθους. Προσαρμόζονται ακόμα τα χρώματα και το μέγεθος του παραθύρου και προστίθεται μία εικόνα φόντου.

Η συνάρτηση *buttonSelected()* δέχεται ως όρισμα ένα αντικείμενο ButtonGroup και επιστρέφει true αν κάποιο κουμπί από την ομάδα έχει ενεργοποιηθεί. Τα κουμπιά τοποθετούνται σε μία λίστα (ArrayList) και με έναν βρόχο for ελέγχεται διαδοχικά αν κάποιο από τα κουμπιά της λίστας έχει ενεργοποιηθεί.

Η συνάρτηση *calculatePlayerType()* δέχεται ως όρισμα ένα ButtonGroup και επιστρέφει τον τύπο παίκτη που επέλεξε ο χρήστης. Τα κουμπιά τοποθετούνται σε μία λίστα και διαδοχικά ελέγχεται αν κάποιο από αυτά είναι ενεργοποιημένο, οπότε επιστέφεται και ο αντίστοιχος τύπος παίκτη. Αν κανένα κουμπί δεν έχει επιλεχθεί, τότε επιστρέφεται null.

Οι συναρτήσεις *getPlayer1Type()* και *getPlayer2type()* επιστρέφουν τον τύπο των παικτών 1 και 2 αντίστοιχα.

Η συνάρτηση *getLookAndFeelClassname()* είναι βοηθητική και χρησιμοποιείται στην επιλογή του σχεδιασμού του παραθύρου.



Εικόνα 1: Η αρχική οθόνη του παιχνιδιού, στην οποία ο χρήστης επιλέγει το είδος των δύο παικτών.

Η κλάση PlayerInfo

Η κλάση PlayerInfo επεκτείνει το JPanel και υλοποιεί το πλαίσιο στο οποίο αναγράφονται σε κάθε γύρο του παιχνιδιού οι πληροφορίες σχετικά με την κατάσταση των παικτών (θέση, πόντοι, κατοχή όπλων). Χαρακτηρίζεται από ένα αντικείμενο τύπου JLabel, στο οποίο τοποθετείται το εικονίδιο του παίκτη και από μία περιοχή κειμένου (JTextArea), στην οποία καταγράφονται οι πληροφορίες για τον παίκτη.

Η συνάρτηση αρχικοποίησης της κλάσης τοποθετεί τα παραπάνω αντικείμενα στο πλαίσιο και ορίζει το μέγεθος και το χρώμα τους καθώς και μία εικόνα φόντου.

Η συνάρτηση *setTextArea()* δέχεται ως όρισμα έναν παίκτη (Player) και με βάσει τα χαρακτηριστικά του ενημερώνει τις πληροφορίες που αναγράφονται στο πλαίσιο.

Η κλάση GBoard

Η κλάση GBoard υλοποιεί τα γραφικά του ταμπλό του παιχνιδιού. Χαρακτηρίζεται από έναν δισδιάστατο πίνακα τύπου JLabel, ο οποίος αναπαριστά τα τετράγωνα του ταμπλό και από έναν ακέραιο αριθμός που δηλώνει το μέγεθος κάθε τετραγώνου.

Η συνάρτηση αρχικοποίησης της κλάσης δέχεται ως ορίσματα ένα αντικείμενο τύπου Board και έναν πίνακα τύπου Player. Ορίζει το μέγεθος κάθε τετραγώνου σε 20 και το μέγεθος του ταμπλό στην τιμή:

$$20 \times \text{board.getN()}.$$

Καλείται η συνάρτηση *createTiles()* για τη δημιουργία του δισδιάστατου πίνακα των τετραγώνων και η συνάρτηση *drawBoard()* για την προσθήκη των εικόνων και των χρωμάτων σε κάθε τετράγωνο.

Η συνάρτηση *createTiles()* δέχεται ως όρισμα ένα αντικείμενο τύπου Board και επιστρέφει τον δισδιάστατο πίνακα JLabel. Το αρχικό μέγεθος του πίνακα προκύπτει από τις διαστάσεις του ταμπλό. Σε κάθε κελί του πίνακα δημιουργείται ένα αντικείμενο JLabel, ορίζονται κατάλληλα οι συντεταγμένες και το μέγεθός του και προστίθεται μαύρο περίγραμμα (για ευκολότερη οριοθέτηση των τετραγώνων στο ταμπλό).

Η συνάρτηση *tileAsIcon()* δέχεται ως ορίσματα ένα αντικείμενο JLabel, τις συντεταγμένες του x, y και το ταμπλό του παιχνιδιού (Board). Ελέγχει αν στο δοθέν τετράγωνο έχει τοποθετηθεί κάποιο αντικείμενο (όπλο, τρόφιμο, παγίδα) ώστε να εμφανίζεται η κατάλληλη εικόνα. Σε περίπτωση που υπάρχει όπλο, τρόφιμο ή παγίδα καλείται η συνάρτηση *getImgSrc()* για την εύρεση της διεύθυνσης της εικόνας που θα τοποθετηθεί στο τετράγωνο.

Η συνάρτηση *clearTiles()* χρησιμοποιείται για τον επανακαθορισμό των ιδιοτήτων των τετραγώνων του δισδιάστατου πίνακα JLabel. Από κάθε κελί του πίνακα αφαιρείται η εικόνα που περιέχει, τίθεται το χρώμα του σε *RGB(248, 195, 191)*, δηλαδή στο χρώμα της κόκκινης περιοχής του ταμπλό και ορίζεται ως μη ορατό.

Η συνάρτηση *isInside()* ελέγχει αν οι δοθείσες x, y συντεταγμένες βρίσκονται στο εσωτερικό μιας περιοχής του ταμπλό. Οι συντεταγμένες των ορίων της περιοχής, δηλαδή οι συντεταγμένες των κορυφών του τετραγώνου, μεταβιβάζονται ως ορίσματα στη συνάρτηση.

Αντίστοιχα, η συνάρτηση *isOnMargin()* ελέγχει αν οι δοθείσες x, y συντεταγμένες βρίσκονται στην περίμετρο μιας περιοχής του ταμπλό. Οι συντεταγμένες των ορίων της περιοχής, δηλαδή οι συντεταγμένες των κορυφών του τετραγώνου, μεταβιβάζονται ως ορίσματα στη συνάρτηση.

Η συνάρτηση *tileAsColor()* δέχεται ως ορίσματα ένα αντικείμενο JLabel, τις συντεταγμένες του x, y και το ταμπλό του παιχνιδιού (Board). Με βάση τις δοθείσες συντεταγμένες υπολογίζεται η περιοχή του ταμπλό στην οποία ανήκει το τετράγωνο. Η άσπρη περιοχή (περιοχή των όπλων) αντιστοιχεί στο λευκό, η γκρι περιοχή (περιοχή τροφίμων) αντιστοιχεί στο χρώμα *RGB(243, 243, 243)*, η μπλε περιοχή (περιοχή παγίδων) αντιστοιχεί στο χρώμα *RGB(204, 237, 252)* και η κόκκινη περιοχή αντιστοιχεί στο χρώμα *RGB(248, 195, 191)*. Για τον έλεγχο της περιοχής των όπλων χρησιμοποιείται η συνάρτηση *isInside()* και ο πίνακας weponAreaLimits της κλάσης Board, ενώ για τον έλεγχο των περιοχών τροφίμων και παγίδων χρησιμοποιούνται οι συναρτήσεις *isInside()* και

isOnMargin() με τους πίνακες foodAreaLimits και trapAreaLimits αντίστοιχα της κλάσης Board.

Η συνάρτηση *drawBoard()* καλεί τη συνάρτηση *clearTiles()* για τον καθαρισμό των τετραγώνων του ταμπλό και διαδοχικά θέτει σε κάθε τετράγωνο την κατάλληλη εικόνα και χρώμα με τις συναρτήσεις *tileAsIcon()* και *tileAsColor()*. Στο ταμπλό τοποθετούνται επίσης και οι θέσεις των δύο παικτών. Αν και οι δύο παίκτες βρίσκονται στο ίδιο τετράγωνο, τότε χρησιμοποιείται κατάλληλη εικόνα. Ο παίκτης Player συμβολίζεται με R, ο Heuristic Player με H και ο MinMax Player με M. Επίσης για τον παίκτη 1 χρησιμοποιείται μπλε χρώμα και για τον παίκτη 2 κόκκινο χρώμα.

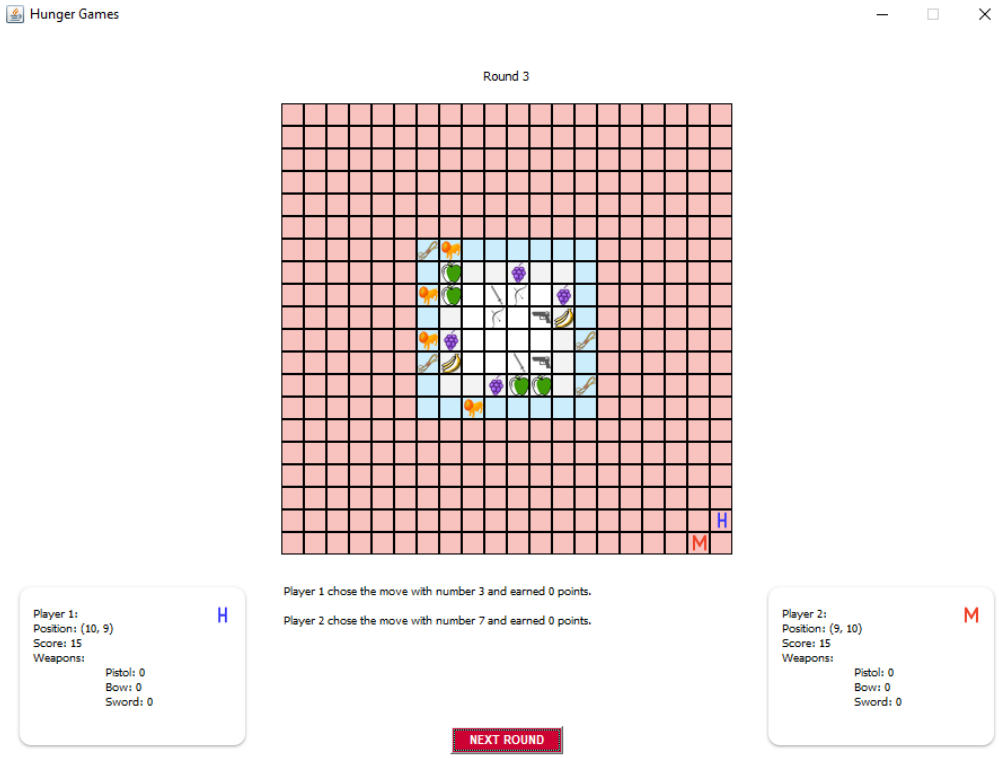
Η κλάση GameFrame

Η κλάση GameFrame επεκτείνει το JFrame και υλοποιεί το κύριο παράθυρο του παιχνιδιού. Αποτελείται από δύο περιοχές PlayerInfo, ένα ταμπλό GBoard, μία ετικέτα στην οποία αναγράφεται ο τρέχων γύρος του παιχνιδιού, μία περιοχή JTextArea στην οποία εμφανίζονται πληροφορίες σε κάθε γύρο, ένα κουμπί για την εκτέλεση του επόμενου γύρου και από την υλοποίηση του παιχνιδιού (αντικείμενο Game).

Η συνάρτηση αρχικοποίησης της κλάσης δέχεται ως παράμετρο τον τύπο των δύο παικτών (που προκύπτει από την επιλογή του χρήστη). Ορίζει το μέγεθος του παραθύρου, τοποθετεί τα πλαίσια PlayerInfo στο κάτω μέρος της οθόνης και δημιουργεί την ετικέτα για την αναγραφή του γύρου του παιχνιδιού και το κουμπί επόμενου γύρου. Ορίζεται ένα νέο παιχνίδι Game, καλείται η συνάρτηση *start()* για την αρχικοποίηση του ταμπλό και των δύο παικτών και δημιουργείται το αντικείμενο GBoard. Τέλος καλείται η συνάρτηση *nextRoundEvent()*.

Η συνάρτηση *nextRoundEvent()* προσθέτει μία ενέργεια στο κουμπί επόμενου γύρου. Συγκεκριμένα, όταν ο χρήστης πατάει το κουμπί καλείται η συνάρτηση *run()* της κλάσης Game για την εκτέλεση του επόμενου γύρου και η συνάρτηση *drawBoard()* της κλάσης GBoard για την ενημέρωση των γραφικών στοιχείων του ταμπλό. Επίσης, ανανεώνεται ο γύρος του παιχνιδιού και οι πληροφορίες σχετικά με τον γύρο που εκτελέστηκε και την κατάσταση των παικτών. Τέλος, αν το παιχνίδι έχει λήξει, καλείται η συνάρτηση *check()* της κλάσης Game για την εύρεση του νικητή και εμφανίζεται αντίστοιχο ενημερωτικό μήνυμα.

Υλοποιούμε ακόμη συναρτήσεις επιστροφής της τιμής των μεταβλητών της κλάσης (συναρτήσεις get).



Εικόνα 2: Στιγμιότυπο από την εκτέλεση του παιχνιδιού. Απεικονίζονται το ταμπλό και ο γύρος του παιχνιδιού, οι πληροφορίες σχετικά με τον γύρο που εκτελέστηκε και τους παίκτες και το κουμπί επόμενου γύρου.