ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΛΗ 412: Αυτόνομοι Πράκτορες 2025-2026

Ανάπτυξη και Σύγκριση Αυτόνομων Πρακτόρων στο 2048

Από Ευριστικές Μεθόδους σε Βαθιά Ενισχυτική Μάθηση

Από

Μιχαήλ Οικονόμου

Α.Μ. : 2023030017

Εικόνα 1: Τυχαία θέση στο παιχνίδι 2048.

1. Περίληψη

Στην παρούσα εργασία μελετάται η ανάπτυξη και η συγκριτική αξιολόγηση αυτόνομων πρακτόρων (autonomous agents) στο περιβάλλον του παιχνιδιού παζλ 2048. Στόχος είναι η διερεύνηση διαφορετικών προσεγγίσεων της Τεχνητής Νοημοσύνης για τη λήψη αποφάσεων υπό συνθήκες αβεβαιότητας. Συγκεκριμένα, υλοποιήθηκαν και συγκρίθηκαν τέσσερις διακριτοί πράκτορες: ένας τυχαίος πράκτορας (Random) ως μέτρο αναφοράς, ένας ευριστικός άπληστος πράκτορας (Greedy), ένας ευριστικός διορατικός πράκτορας (Expectimax), και ένας πράκτορας Βαθιάς Ενισχυτικής Μάθησης (Deep Q-Network - DQN).

2. Εισαγωγή

2.1 Το Περιβάλλον του 2048

Το 2048 είναι ένα παιχνίδι ενός παίκτη που διεξάγεται σε πλέγμα 4x4. Σκοπός είναι η συγχώνευση πλακιδίων με την ίδια τιμή ώστε να δημιουργηθούν πλακίδια μεγαλύτερης αξίας, με απώτερο στόχο το πλακίδιο «2048».

Σύμφωνα με τη θεωρία των Αυτόνομων Πρακτόρων *(Διάλεξη 02, "Περιβάλλοντα")*, το περιβάλλον του 2048 χαρακτηρίζεται ως:

- **Πλήρως Παρατηρήσιμο (Fully Observable):** Ο πράκτορας έχει ανά πάσα στιγμή πλήρη πρόσβαση στην κατάσταση του πλέγματος.
- **Στοχαστικό (Stochastic):** Η εμφάνιση νέων πλακιδίων (2 ή 4) σε τυχαίες θέσεις εισάγει τον παράγοντα της αβεβαιότητας, καθιστώντας το αποτέλεσμα των ενεργειών μη πλήρως προβλέψιμο.
- **Διακριτό (Discrete):** Τόσο οι καταστάσεις (συνδυασμοί πλακιδίων) όσο και οι ενέργειες (4 κατευθύνσεις) είναι πεπερασμένες και διακριτές.
- **Ακολουθιακό (Sequential):** Η τρέχουσα απόφαση επηρεάζει όλες τις μελλοντικές καταστάσεις.

2.2 Ορισμός του Πράκτορα

Ο αυτόνομος πράκτορας που αναπτύχθηκε αλληλεπιδρά με το περιβάλλον μέσω:

- **Αισθητήρων (Sensors):** Λαμβάνει ως είσοδο την τρέχουσα διάταξη του πίνακα (Grid State).
- **Επενεργητών (Actuators):** Εκτελεί μία από τις τέσσερις δυνατές κινήσεις (Πάνω, Κάτω, Αριστερά, Δεξιά).

- **Μέτρο Απόδοσης (Performance Measure):** Η αξιολόγηση γίνεται βάσει του τελικού σκορ (Score) και της μέγιστης τιμής πλακιδίου (Max Tile) που επιτεύχθηκε.

3. Μεθοδολογία: Ανάπτυξη Πρακτόρων

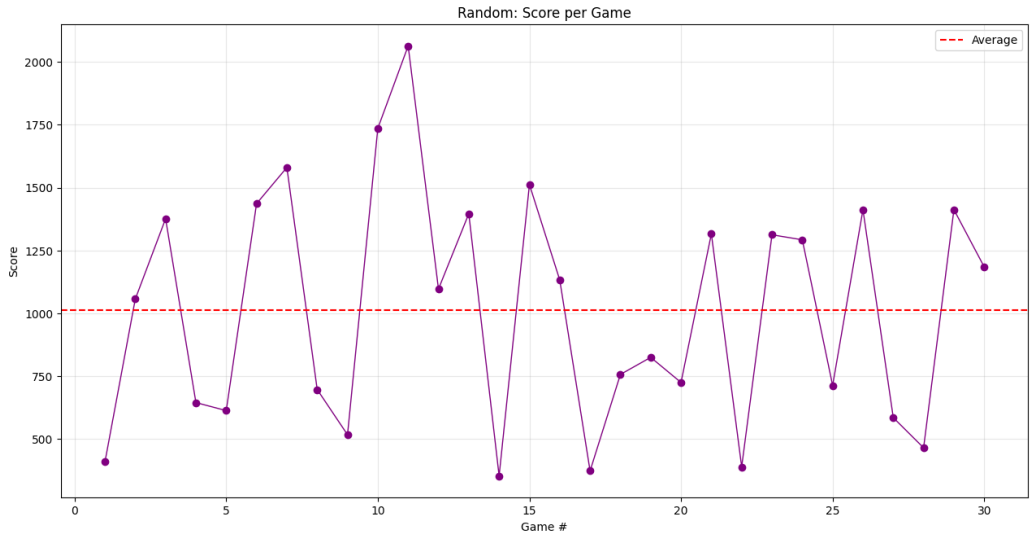
3.1 Τυχαίος Πράκτορας (Random Agent)

Ο Τυχαίος Πράκτορας αποτελεί την απλούστερη μορφή αυτονομίας και λειτουργεί ως σημείο αναφοράς (Baseline) για την αξιολόγηση των υπόλοιπων αλγορίθμων. Ο πράκτορας αυτός δεν διαθέτει καμία γνώση για τους κανόνες ή τη στρατηγική του παιχνιδιού. Σε κάθε χρονική στιγμή, επιλέγει μία κίνηση a από το σύνολο των έγκυρων κινήσεων A_{valid} με ομοιόμορφη πιθανότητα:

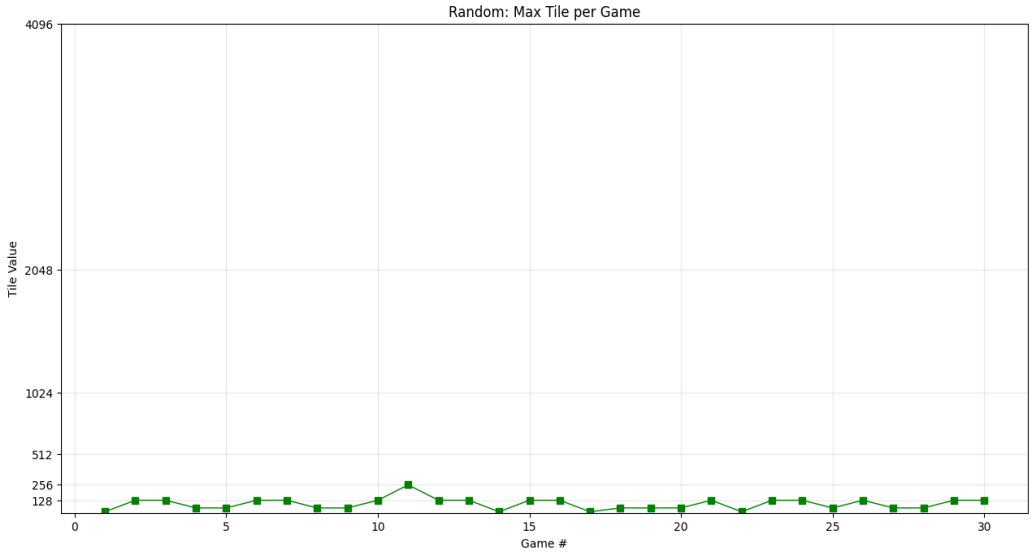
$$P(a) = \frac{1}{| A_{valid} |}$$

Η συμπεριφορά του αναδεικνύει τη στοχαστική φύση του 2048, καθώς επιτυγχάνει χαμηλά σκορ και σπάνια ξεπερνά το πλακίδιο «128» ή «256», αποδεικνύοντας ότι η τυχαία περιήγηση στον χώρο καταστάσεων δεν επαρκεί για την επίλυση του προβλήματος.

Ο πράκτορας αγνοεί πλήρως την αβεβαιότητα. Δεν διαθέτει μοντέλο πρόβλεψης, επομένως δεν τον "ενδιαφέρει" αν θα εμφανιστεί 2 ή 4, ούτε πού θα εμφανιστεί. Η επιβίωσή του είναι καθαρά θέμα τύχης.



Εικόνα 2: Score του Random Agent σε simulation 30 παιχνιδιών



Εικόνα 3: Max Tile του Random Agent σε simulation 30 παιχνιδιών

3.2 Ευριστικός Άπληστος Πράκτορας (Greedy Agent)

Ο Greedy Agent εισάγει την έννοια της **Συνάρτησης Χρησιμότητας (Utility Function)**. Σε αντίθεση με έναν πράκτορα που απλώς μεγιστοποιεί το άμεσο σκορ (συγχωνεύσεις), ο συγκεκριμένος πράκτορας αξιολογεί την **ποιότητα της θέσης** (Positional Heuristic).

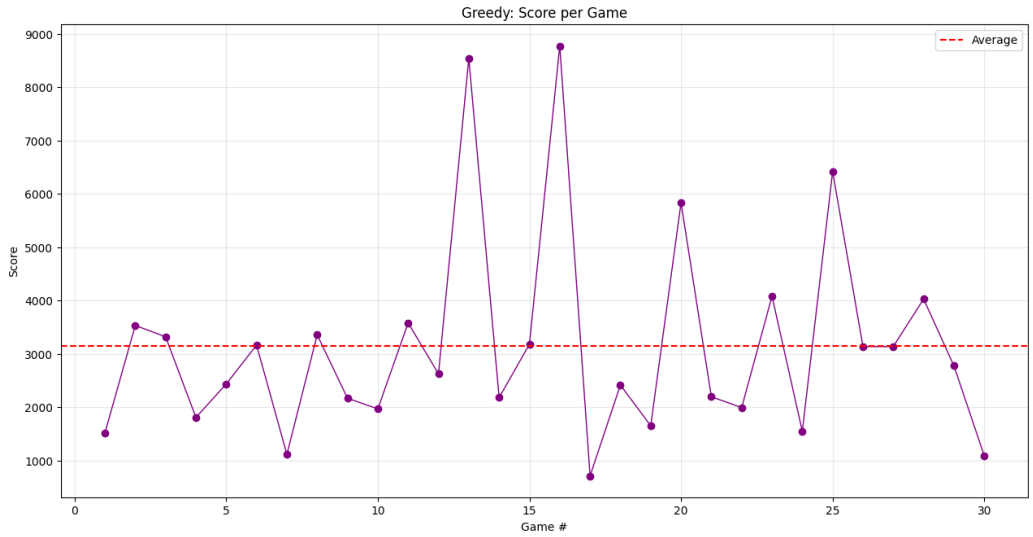
Χρησιμοποιήσαμε μια στρατηγική "Snake" (Μονοτονίας), εφαρμόζοντας έναν πίνακα βαρών (Weight Matrix) στο πλέγμα, όπου τα βάρη αυξάνονται εκθετικά προς μία γωνία. Η αξία V μιας κατάστασης s ορίζεται ως:

$$V(s) = \sum_{i,j} (Tile_{i,j} \times Weight_{i,j}) + (EmptyCells \times Bonus)$$

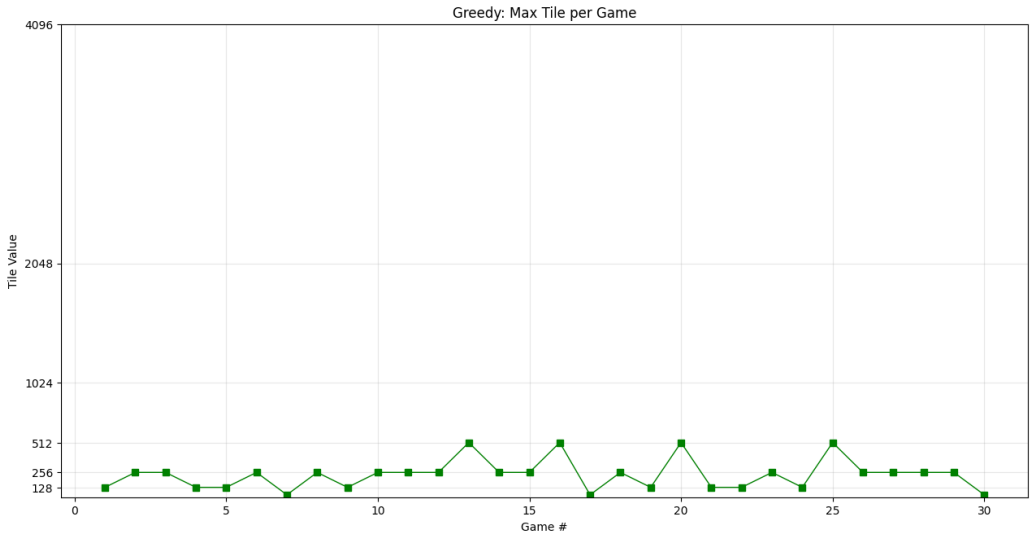
Ο πράκτορας προσομοιώνει όλες τις πιθανές κινήσεις για ένα βήμα μπροστά (1-step lookahead) και επιλέγει εκείνη που μεγιστοποιεί την $V(s)$. Αν και βελτιώνει δραματικά την απόδοση σε σχέση με τον Random, ο Greedy Agent παραμένει «μυωπικός», καθώς αδυνατεί να προβλέψει παγίδες που εμφανίζονται μετά από δύο ή περισσότερες κινήσεις.

Ο Greedy αξιολογεί την κατάσταση του πλέγματος αμέσως μετά την κίνησή του (afterstate), **αλλά πριν** το περιβάλλον τοποθετήσει το νέο τυχαίο πλακίδιο.

Το πρόβλημα: Υποθέτει σιωπηρά ότι η κατάσταση είναι στατική. Δεν λαμβάνει υπόψη το ρίσκο ότι ένα νέο πλακίδιο (π.χ. ένα 4 σε λάθος θέση) μπορεί να μπλοκάρει μια σημαντική συγχώνευση. Γι' αυτό συχνά "παγιδεύεται".



Εικόνα 4: Score του Greedy Agent σε simulation 30 παιχνιδιών



Εικόνα 5: Max Tile του Greedy Agent σε simulation 30 παιχνιδιών

3.3 Πράκτορας Monte Carlo

Ο Πράκτορας Monte Carlo βασίζεται στη στατιστική δειγματοληψία και δεν απαιτεί εξειδικευμένη γνώση (domain knowledge) ή ευριστικές συναρτήσεις. Για κάθε δυνατή κίνηση στην τρέχουσα κατάσταση, ο πράκτορας εκτελεί έναν αριθμό N τυχαίων προσομοιώσεων (rollouts) μέχρι τον τερματισμό του παιχνιδιού.

Η αξία κάθε κίνησης υπολογίζεται ως ο μέσος όρος των τελικών σκορ που προέκυψαν από τις προσομοιώσεις που ξεκίνησαν με αυτήν.

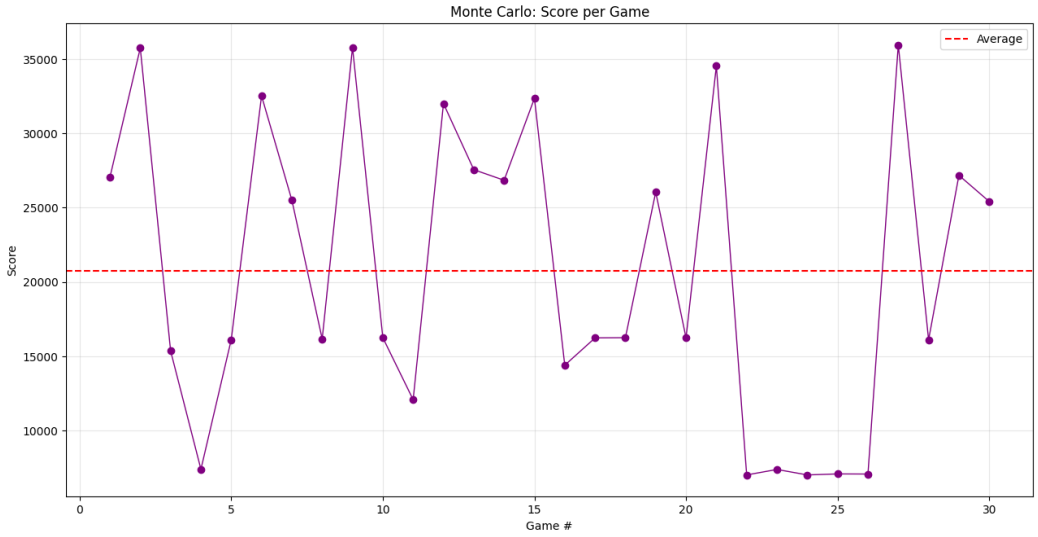
$$Value(move) = \frac{1}{N}\sum_{i=1}^{N} FinalScor\, e_i$$

Η μέθοδος αυτή είναι ιδιαίτερα ισχυρή καθώς «ανακαλύπτει» ποιες κινήσεις οδηγούν σε επιβίωση και υψηλό σκορ μακροπρόθεσμα, χωρίς να του έχουμε διδάξει τη στρατηγική του

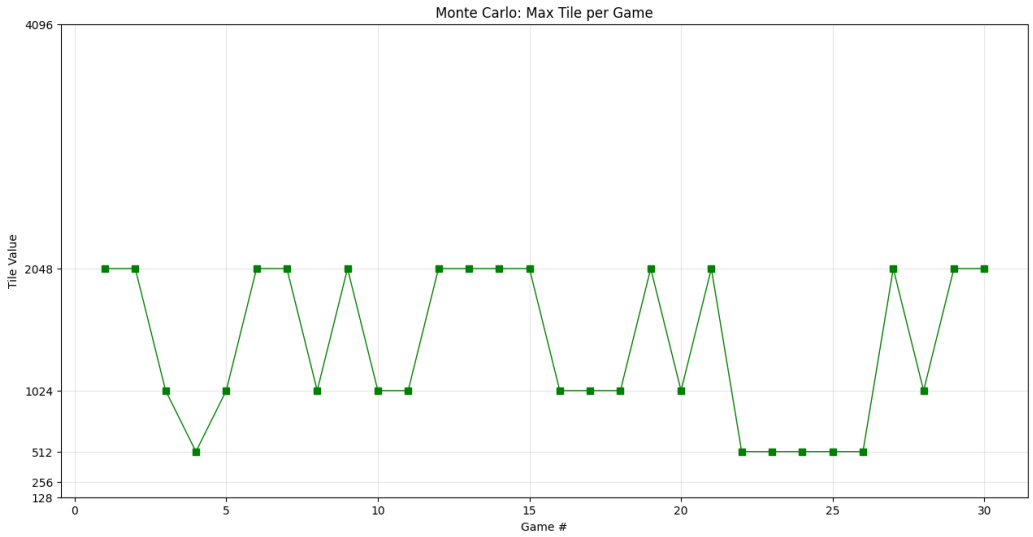
φιδιού. Ωστόσο, το υπολογιστικό κόστος είναι υψηλό, καθώς απαιτεί την εκτέλεση χιλιάδων τυχαίων κινήσεων ανά βήμα.

Ο πράκτορας δεν γνωρίζει ρητά τις πιθανότητες (90% για 2, 10% για 4), αλλά τις μαθαίνει **εμμέσως** μέσω του Νόμου των Μεγάλων Αριθμών.

Αν μια κίνηση (π.χ. UP) αφήνει το ταμπλό ευάλωτο σε "κακές" τυχαίες γεννήσεις πλακιδίων, τότε η πλειοψηφία των τυχαίων προσομοιώσεων θα οδηγήσει σε γρήγορη ήττα. Έτσι, ο μέσος όρος του σκορ για την κίνηση αυτή θα πέσει. Ο πράκτορας επιλέγει κινήσεις που είναι στατιστικά ανθεκτικές (robust) απέναντι στην αβεβαιότητα, χωρίς να υπολογίζει τύπους.



Εικόνα 6: Score του Monte Carlo Agent σε simulation 30 παιχνιδιών με N=30.



Εικόνα 7:Max Tile του Monte Carlo Agent σε simulation 30 παιχνιδιών με N=30.

3.4 Πράκτορας Expectimax (Model-Based)

Ο αλγόριθμος Expectimax αποτελεί τη βέλτιστη προσέγγιση για παιχνίδια με στοχαστικότητα (όπως το 2048). Αντιμετωπίζει το παιχνίδι ως ένα δέντρο αναζήτησης με δύο είδη κόμβων:

1. **Max Nodes (Πράκτορας):** Επιλέγει την κίνηση που μεγιστοποιεί την αξία.
2. **Chance Nodes (Περιβάλλον):** Υπολογίζουν την Αναμενόμενη Τιμή (Expected Value) λαμβάνοντας υπόψη όλες τις πιθανές θέσεις και τιμές (2 ή 4) των νέων πλακιδίων.

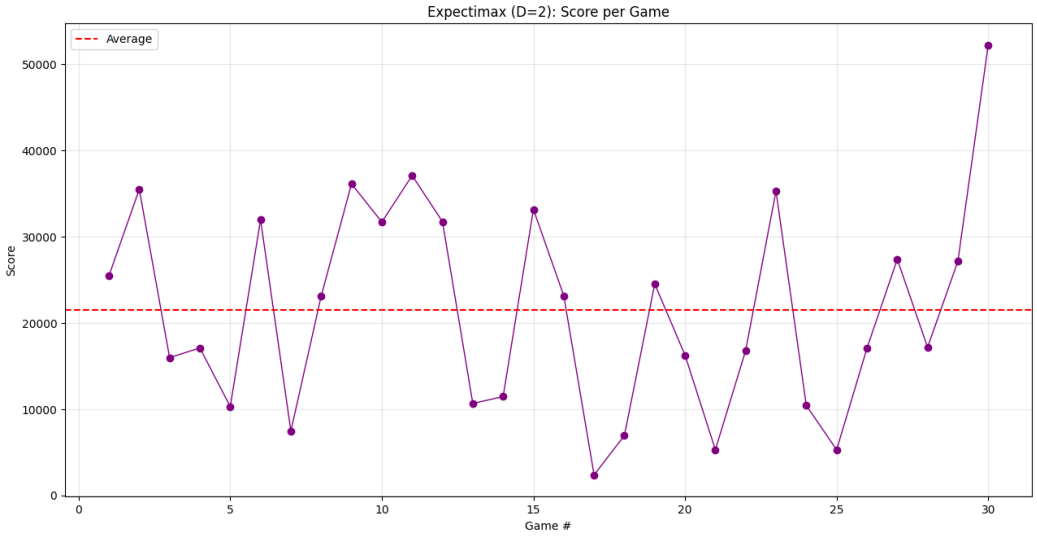
Η αναμενόμενη τιμή σε έναν κόμβο τύχης υπολογίζεται ως:

$$V(s) = \sum_{s'} P(s' \mid s, a) \cdot V(s')$$

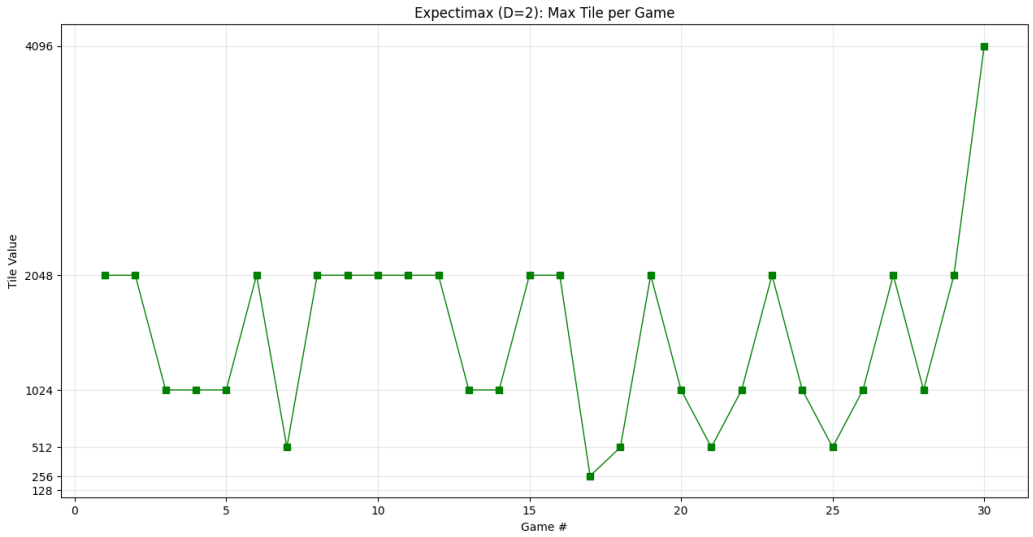
Όπου

$$P(s' \mid s, a)$$

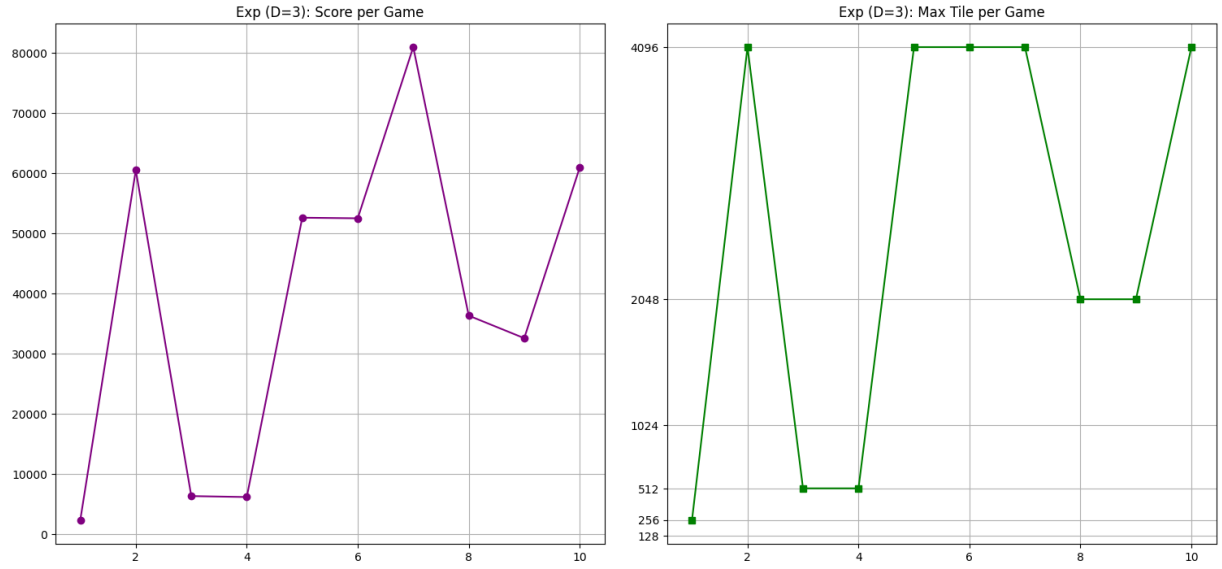
είναι η πιθανότητα μετάβασης (90% για 2, 10% για 4). Ο πράκτορας χρησιμοποιεί την ίδια ευριστική συνάρτηση (Snake) με τον Greedy, αλλά την εφαρμόζει στα φύλλα του δέντρου σε βάθος 2 ή 3 (Depth). Αυτό του επιτρέπει να σχεδιάζει στρατηγικά και να διαχειρίζεται το ρίσκο, επιτυγχάνοντας τα υψηλότερα σκορ.



Εικόνα 8: Score του Expectimax Agent σε simulation 30 παιχνιδιών με depth=2.



Εικόνα 9: Max Tile του Excpectimax Agent σε simulation 30 παιχνιδιών με depth=2.



Εικόνα 10 : Max Tile και Score του Excpectimax Agent σε simulation 10 παιχνιδιών με depth=3.

3.5 Πράκτορας Deep Q-Network (Model-Free RL)

Ο πράκτορας DQN αντιπροσωπεύει την προσέγγιση της Βαθιάς Ενισχυτικής Μάθησης (Deep Reinforcement Learning). Είναι **Model-Free**, δηλαδή δεν γνωρίζει τους κανόνες του παιχνιδιού ούτε τις πιθανότητες εμφάνισης των αριθμών. Μαθαίνει προσεγγίζοντας τη συνάρτηση αξίας $Q(s, a)$ μέσω ενός Νευρωνικού Δικτύου.

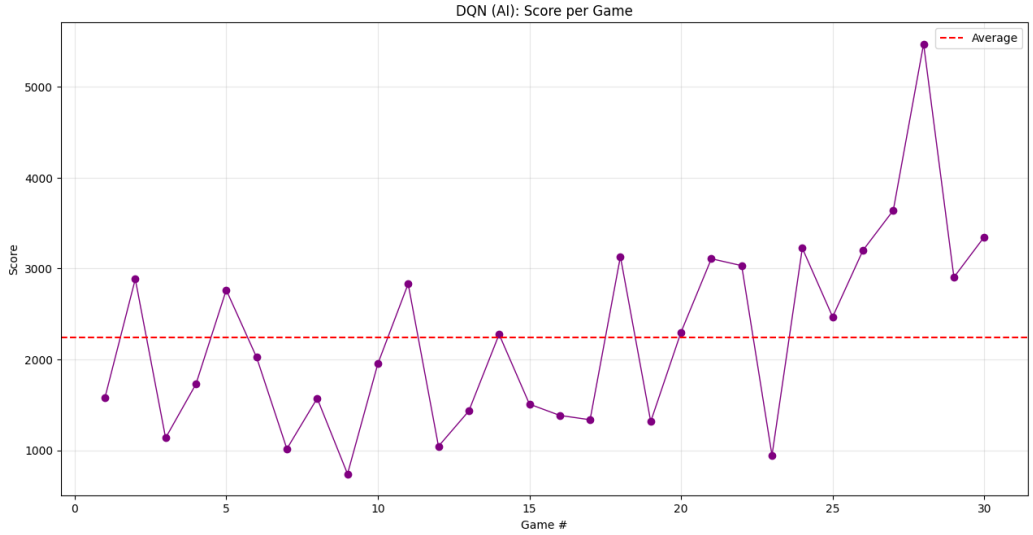
Τεχνικά Χαρακτηριστικά:

- **Είσοδος (Input):** Το πλέγμα μετατρέπεται σε One-Hot Encoding διάνυσμα ($4 \times 4 \times 16$), επιτρέποντας στο δίκτυο να αναγνωρίζει γεωμετρικά μοτίβα.
- **Δίκτυο (Network):** Χρησιμοποιήθηκε ένα πυκνό δίκτυο (Dense Network) με επίπεδα 512 και 256 νευρώνων.
- **Εκπαίδευση:** Ο πράκτορας εκπαιδεύτηκε για 10.000 επεισόδια. Χρησιμοποιήθηκε μηχανισμός **Experience Replay** για την αποθήκευση και τυχαία δειγματοληψία παρελθοντικών κινήσεων, μειώνοντας τη συσχέτιση των δεδομένων.
- **Συνάρτηση Ανταμοιβής (Reward Shaping):** Για να επιταχυνθεί η μάθηση, ορίστηκαν αυστηρές ποινές για τον τερματισμό του παιχνιδιού και επιβραβεύσεις για τη διατήρηση κενών κελιών.

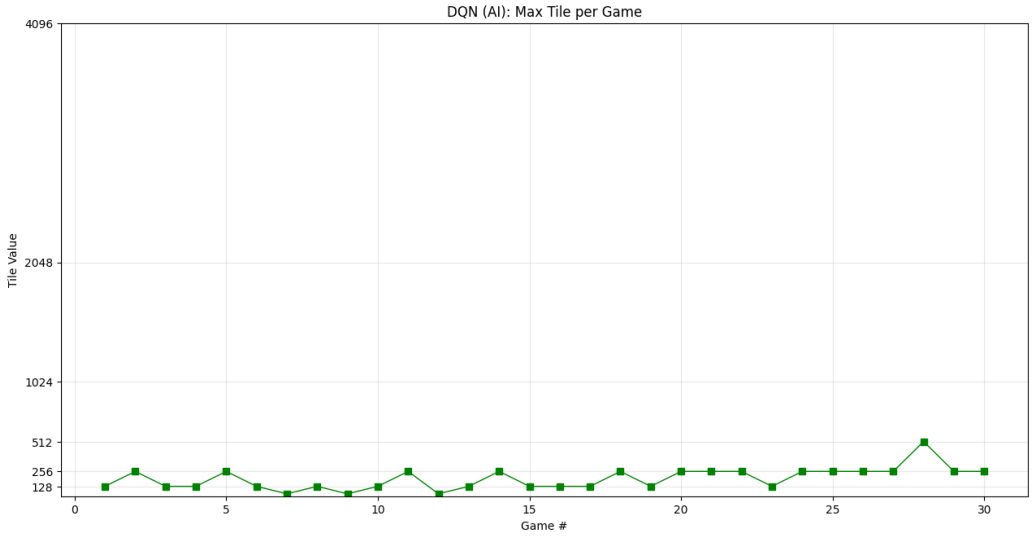
Η καμπύλη εκμάθησης δείχνει τη μετάβαση του πράκτορα από τυχαίες κινήσεις (Exploration) σε στρατηγικό παιχνίδι (Exploitation), καταφέρνοντας να ανακαλύψει μόνος του στρατηγικές όπως η διατήρηση των μεγάλων αριθμών στις άκρες

Ο DQN είναι Model-Free, άρα δεν ξέρει ότι υπάρχει 10% πιθανότητα για '4'. Όμως, μέσα από χιλιάδες επεισόδια εκπαίδευσης, έχει "τιμωρηθεί" (μέσω αρνητικών rewards) όταν έκανε κινήσεις που άφηναν το ταμπλό σε επισφαλή κατάσταση.

Αυτό έχει ως αποτέλεσμα το Νευρωνικό Δίκτυο έχει μάθει να αποδίδει χαμηλά Q-Values σε καταστάσεις που είναι ευάλωτες στην τύχη. Έχει αναπτύξει ένα "ένστικτο" αποφυγής ρίσκου, μαθαίνοντας να κρατάει το ταμπλό ανοιχτό ώστε, ό,τι κι αν φέρει η τύχη (2 ή 4, σε οποιαδήποτε θέση), να μπορεί να αντιδράσει.



Εικόνα 11 : Score του DQN Agent σε simulation 30 παιχνιδιών.



Εικόνα 12 : Max Tile του DQN Agent σε simulation 30 παιχνιδιών.

4. Υλοποίηση Συστήματος

4.1 Αρχιτεκτονική Λογισμικού

Το σύστημα αναπτύχθηκε σε γλώσσα **Python 3.12**, αξιοποιώντας την αντικειμενοστραφή δομή της για τη διαχείριση της πολυπλοκότητας. Η αρχιτεκτονική του κώδικα χωρίζεται σε τρία διακριτά επίπεδα (modules), προάγοντας την επεκτασιμότητα και τη συντηρησιμότητα:

- **Επίπεδο Παιχνιδιού (Game Logic):** Το αρχείο board.py ενθυλακώνει τους κανόνες του 2048, τη διαχείριση του πίνακα (state management) και την υλοποίηση των κινήσεων (merge, compress). Το ui.py υλοποιεί τη γραφική διεπαφή χρήστη (GUI) χρησιμοποιώντας τη βιβλιοθήκη **Tkinter**.
- **Επίπεδο Πρακτόρων (Agents):** Κάθε αλγόριθμος (Random, Greedy, Expectimax, DQN) υλοποιείται ως ξεχωριστή κλάση στον φάκελο ai_agents/, ακολουθώντας κοινή διεπαφή (select_move(board)). Αυτό επιτρέπει την εύκολη εναλλαγή πρακτόρων κατά τη διάρκεια του παιχνιδιού.
- **Επίπεδο Εκτέλεσης (Main & Training):** Το αρχείο main.py συντονίζει τη ροή του παιχνιδιού και τη σύνδεση πρακτόρων-περιβάλλοντος, ενώ το train_dqn.py είναι υπεύθυνο για την εκπαιδευτική διαδικασία του νευρωνικού δικτύου.

4.2 Τεχνολογίες και Βιβλιοθήκες

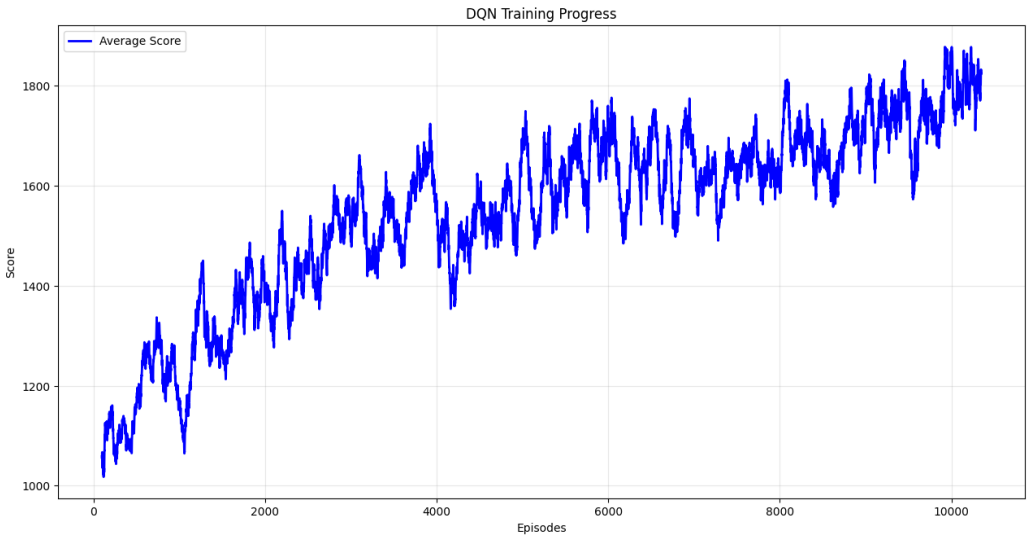
Για την υλοποίηση του πράκτορα Deep Q-Network χρησιμοποιήθηκε η βιβλιοθήκη **PyTorch**, λόγω της ευελιξίας της στη δημιουργία δυναμικών υπολογιστικών γράφων. Η επεξεργασία και κανονικοποίηση των δεδομένων του πλέγματος έγινε με τη χρήση της **NumPy**. Τέλος, για την εξαγωγή των συμπερασμάτων και την οπτικοποίηση των αποτελεσμάτων (learning curves, benchmarks) χρησιμοποιήθηκαν οι βιβλιοθήκες **Pandas** και **Matplotlib**.

5. Πειραματικά Αποτελέσματα

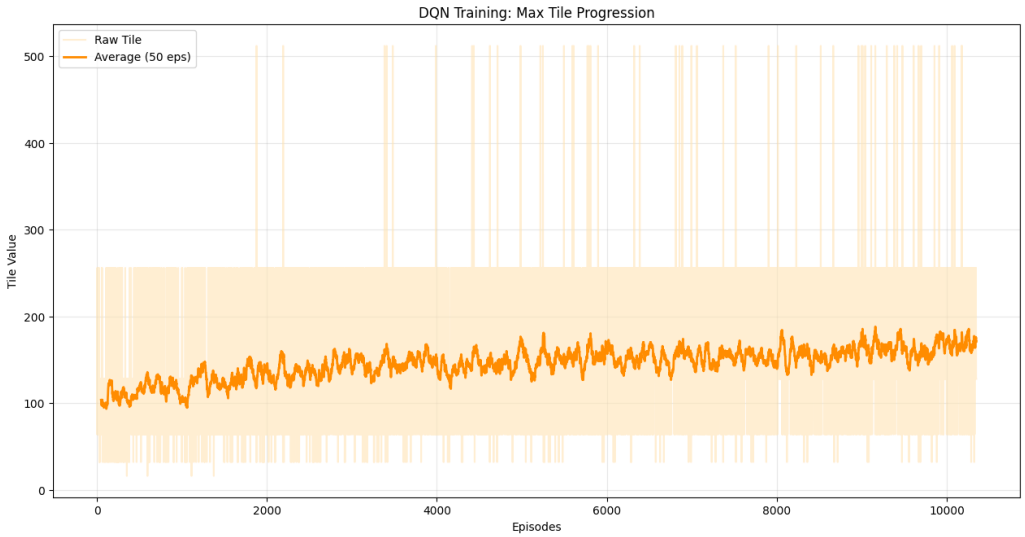
5.1 Ανάλυση Εκπαίδευσης DQN

Η διαδικασία εκπαίδευσης του πράκτορα DQN διήρκεσε 10.000 επεισόδια. Όπως φαίνεται στο Διάγραμμα 1 (DQN Learning Curve - Score), η απόδοση του πράκτορα παρουσίασε σημαντική βελτίωση στα πρώτα 3.000 επεισόδια (όπως φαίνεται στην Εικόνα 13), καθώς η τιμή του Epsilon μειωνόταν και ο πράκτορας μετέβαινε από τη φάση της εξερεύνησης (exploration) στην εκμετάλλευση (exploitation).

Παρατηρούμε ότι το μέσο σκορ σταθεροποιήθηκε μετά το επεισόδιο 5.000, υποδεικνύοντας ότι το πυκνό δίκτυο (Dense Network) έφτασε στα όρια της χωρητικότητάς του για την αναγνώριση μοτίβων στο συγκεκριμένο πρόβλημα. Παρ' όλα αυτά, ο πράκτορας έμαθε επιτυχώς να αποφεύγει τις άκυρες κινήσεις και να επιδιώκει συγχωνεύσεις.



Εικόνα 13 :Average Score ανά 50 επεισόδια του DQN Agent κατά την διάρκεια των 10.000 επεισοδίων εκπαίδευσης του .

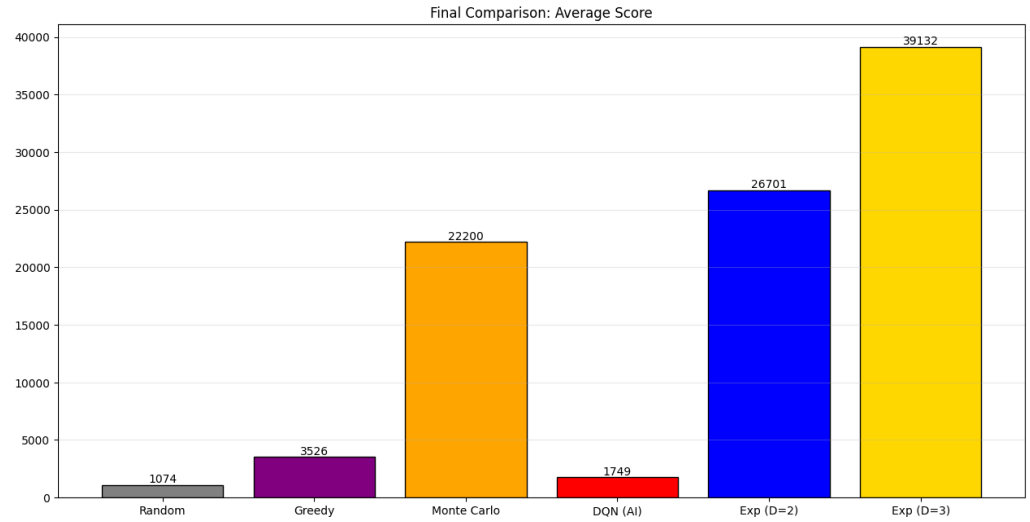


Εικόνα 14 :Average Max Tile ανά 50 επεισόδια του DQN Agent κατά την διάρκεια των 10.000 επεισοδίων εκπαίδευσης του.

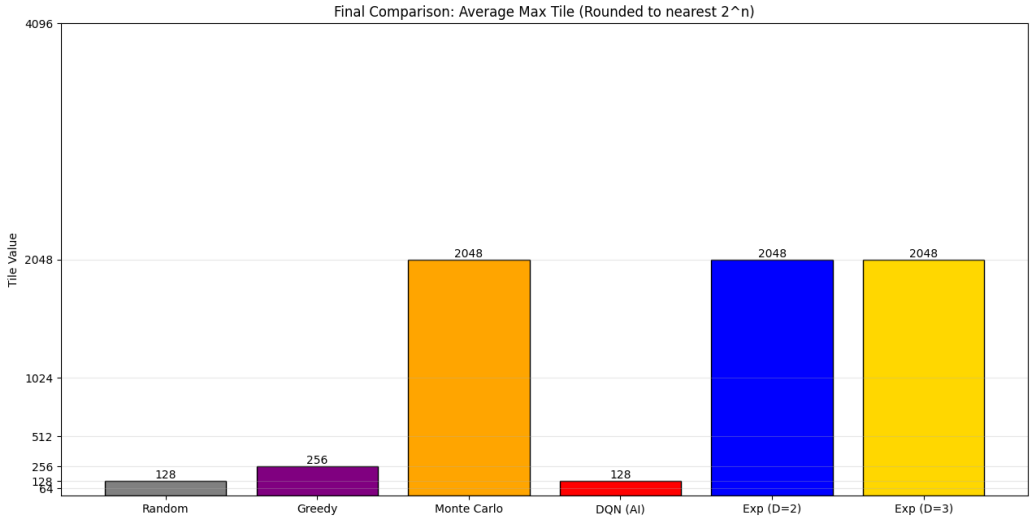
5.2 Συγκριτική Αξιολόγηση (Benchmarks)

Για την αντικειμενική σύγκριση των αλγορίθμων, εκτελέστηκαν 30 παιχνίδια για κάθε "γρήγορο" πράκτορα και 10 παιχνίδια για τον Expectimax (Depth=3). Τα αποτελέσματα συνοψίζονται στο Διάγραμμα 2 (Average Score Comparison).

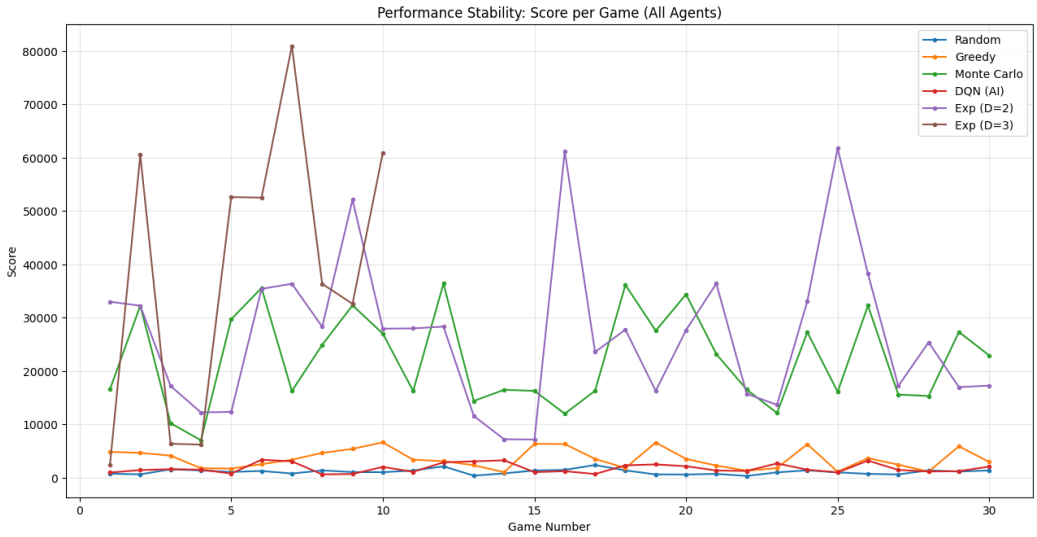
- Ο **Expectimax (D=3)** κυριάρχησε απόλυτα, επιτυγχάνοντας τον στόχο του 2048 στην πλειοψηφία των περιπτώσεων.
- Ο **Monte Carlo** επέδειξε αξιοσημείωτη σταθερότητα, ξεπερνώντας τον Greedy Agent, αποδεικνύοντας ότι η στατιστική δειγματοληψία είναι ανώτερη από τις απλές ευριστικές μεθόδους.
- Ο **DQN** πράκτορας, αν και δεν ξεπέρασε τον Expectimax, κατάφερε να αποδώσει σταθερά καλύτερα από τον Random Agent, αποδεικνύοντας ότι έμαθε μια βασική στρατηγική επιβίωσης χωρίς να γνωρίζει τους κανόνες του παιχνιδιού.



Εικόνα 15 :Average Score Bar Chart όλων των πρακτόρων σε simulation 30 παιχνιδιών οπού φαίνεται ξεκάθαρα η επικράτηση του Excpectimax με βάθος 3.



Εικόνα 16 :Average Max Tile Bar Chart όλων των πρακτόρων σε simulation 30 παιχνιδιών



Εικόνα 17 :Score per game Line Chart όλων των πρακτόρων σε simulation 30 παιχνιδιών

5.3 Ανάλυση Χρόνου Εκτέλεσης

Ένα σημαντικό εύρημα αφορά το κόστος υπολογισμού. Ο αλγόριθμος Expectimax (D=3), παρότι βέλτιστος, απαιτούσε σημαντικά περισσότερο χρόνο ανά κίνηση λόγω της εκθετικής αύξησης του δέντρου αναζήτησης. Αντιθέτως, ο πράκτορας DQN, μετά την ολοκλήρωση της εκπαίδευσης, λαμβάνει αποφάσεις σχεδόν ακαριαία (σε χιλιοστά του δευτερολέπτου), καθώς απαιτεί μόνο ένα πέρασμα (forward pass) μέσα από το νευρωνικό δίκτυο.

6. Συμπεράσματα και Μελλοντικές Επεκτάσεις

6.1 Συμπεράσματα

Η παρούσα εργασία κατέδειξε ότι η επιλογή του κατάλληλου αλγορίθμου για το 2048 εξαρτάται από τη διαθεσιμότητα γνώσης για το περιβάλλον.

1. **Υπεροχή των Model-Based Μεθόδων:** Σε περιβάλλοντα όπου οι κανόνες και οι πιθανότητες μετάβασης είναι γνωστές (όπως το 2048), οι αλγόριθμοι αναζήτησης όπως ο Expectimax παραμένουν ανίκητοι, καθώς μπορούν να υπολογίσουν με ακρίβεια το ρίσκο.
2. **Η Πρόκληση της Ενισχυτικής Μάθησης:** Η προσέγγιση Model-Free (DQN) απέδειξε ότι είναι δυνατή η εκμάθηση στρατηγικής από το μηδέν. Ωστόσο, η έλλειψη ρητής γνώσης του μοντέλου καθιστά δύσκολη την επίτευξη του τέλειου σκορ (2048) χωρίς τεράστιο χρόνο εκπαίδευσης.
3. **Διαχείριση Αβεβαιότητας:** Ο Expectimax διαχειρίζεται την αβεβαιότητα μαθηματικά (υπολογισμός μέσης τιμής), ενώ ο DQN και ο Monte Carlo τη διαχειρίζονται εμπειρικά και στατιστικά αντίστοιχα.

6.2 Μελλοντικές Επεκτάσεις

Για τη βελτίωση της απόδοσης του DQN πράκτορα, προτείνεται η χρήση **Συνελικτικών Νευρωνικών Δικτύων (CNN)**. Τα CNNs είναι σχεδιασμένα να αναγνωρίζουν χωρικά μοτίβα (spatial patterns) σε πλέγματα, κάτι που ταιριάζει απόλυτα στη φύση του 2048 (π.χ. αναγνώριση "φιδιού" ή γωνιακών δομών), σε αντίθεση με τα Πυκνά Δίκτυα (Dense Networks) που χρησιμοποιήθηκαν εδώ και αγνοούν τη γεωμετρία του πίνακα.

7. Βιβλιογραφία

Λαγουδάκης, Μ. Γ. (2025). *ΠΛΗ 412: Αυτόνομοι Πράκτορες*. Σημειώσεις Διαλέξεων, Πολυτεχνείο Κρήτης.

- Διάλεξη 02: Πράκτορες & Περιβάλλοντα.
- Διάλεξη 07: Αβεβαιότητα.
- Διάλεξη 19-20: Λήψη Αποφάσεων (MDPs).
- Διάλεξη 21-22: Ενισχυτική Μάθηση (Reinforcement Learning).