Πανεπιστήμιο Πειραιώς - Μεταπτυχιακό

Προηγμένα Συστήματα Πληροφορικής - Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης

Μεταπτυχιακή Διατριβή

Τίτλος Μεταπτυχιακής Διατριβής	**KakuroAI: Διαδικαστική Παραγωγή Περιεχομένου με Προσαρμοστική Δυσκολία Χρησιμοποιώντας Ενισχυτική Μάθηση** **KakuroAI: Procedural Content Generation with Adaptive Difficulty Using Reinforcement Learning**
Ονοματεπώνυμο Φοιτητή	**Μουρλάς Παναγιώτης**
Πατρώνυμο	**Παναγιώτης**
Αριθμός Μητρώου	**ΜΠΣΠ2324**
Επιβλέπων	**Αλέπης Ευθύμιος, Καθηγητής**

Μάιος 2025

Τριμελής Εξεταστική Επιτροπή

Ευθύμιος Αλέπης Καθηγητής	Μαρία Βίρβου Καθηγήτρια	Κωνσταντίνα Χρυσαφιάδη Επίκουρος καθηγήτρια

Πίνακας Περιεχομένων

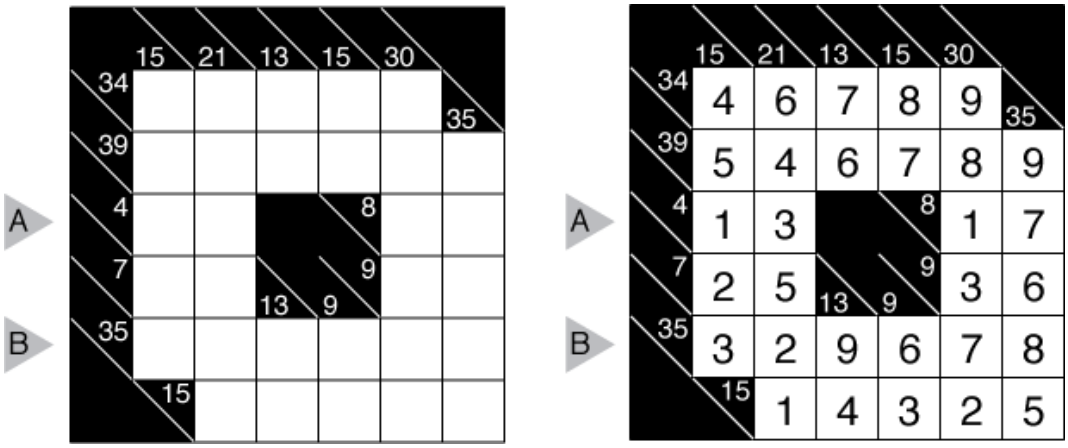
1. Εισαγωγή

1.1. Πρόλογος

Τα παζλ Κακούρο, γνωστά και ως "Σταυρόλεξα Αθροισμάτων", αποτελούν μια πρόκληση που συνδυάζει αριθμητική και λογική σκέψη. Παρόλο που οι κανόνες τους είναι απλοί, η δημιουργία ενός παζλ με μοναδική λύση και κατάλληλο επίπεδο δυσκολίας απαιτεί ιδιαίτερη προσοχή. Παραδοσιακά, ο σχεδιασμός τέτοιων παζλ βασίζεται σε εμπειρικές και ευρετικές μεθόδους, οι οποίες συχνά οδηγούν σε παζλ με ασύμβατη δυσκολία για διαφορετικούς παίκτες.

Η παρούσα εργασία προτείνει μια σύγχρονη προσέγγιση στη δημιουργία παζλ Κακούρο, αξιοποιώντας αλγορίθμους Διαδικαστικής Παραγωγής Περιεχομένου (PCG) και Ενισχυτικής Μάθησης (Reinforcement Learning), ώστε να παραχθούν παζλ που προσαρμόζονται αυτόματα στο επίπεδο του χρήστη. Βασιζόμενοι στη θεωρία της "ροής" του Csikszentmihalyi, στόχος μας είναι να προσφέρουμε παζλ που είναι αρκετά απαιτητικά για να προκαλούν ενδιαφέρον, χωρίς όμως να δημιουργούν απογοήτευση ή ανία.

Μέσω της πλατφόρμας Unity και της χρήσης των ML-Agents, αναπτύχθηκε ένα σύστημα που συλλέγει δεδομένα απόδοσης του χρήστη (όπως χρόνος επίλυσης, λάθη, χρήση βοηθειών) και εκπαιδεύει έναν πράκτορα (agent) να παράγει παζλ με δυναμικά ρυθμιζόμενη δυσκολία. Το σύστημα αυτό αποτελεί ένα βήμα προς την εξατομικευμένη εμπειρία παιχνιδιού σε λογικά παζλ, αναδεικνύοντας πώς η τεχνητή νοημοσύνη μπορεί να ενισχύσει τη διαδικασία μάθησης και διασκέδασης.



Εικόνα 1.1: Παράδειγμα Παζλ Kakuro

1.2. Περίληψη

Η παρούσα πτυχιακή εργασία παρουσιάζει το KakuroAI, ένα σύστημα που συνδυάζει τη Διαδικαστική Παραγωγή Περιεχομένου (PCG) με Ενισχυτική Μάθηση (RL) για τη δημιουργία παζλ Κακούρο προσαρμοστικής δυσκολίας. Τα παζλ Κακούρο είναι λογικά παζλ με αριθμούς που απαιτούν από τους παίκτες να συμπληρώσουν ένα πλέγμα ακολουθώντας συγκεκριμένους περιορισμούς αθροίσματος. Παρά τη δημοτικότητά τους, η δημιουργία παζλ Κακούρο με συγκεκριμένο επίπεδο δυσκολίας παραμένει μια πρόκληση. Το σύστημά μας αντιμετωπίζει αυτό το πρόβλημα χρησιμοποιώντας έναν πράκτορα RL που εκπαιδεύεται να προσαρμόζει τις παραμέτρους παραγωγής παζλ με βάση τις μετρήσεις απόδοσης του παίκτη.

Η εργασία περιγράφει την αρχιτεκτονική του συστήματος, τον αλγόριθμο παραγωγής παζλ, και το μοντέλο RL που χρησιμοποιείται για την προσαρμογή της δυσκολίας. Υλοποιήσαμε το σύστημα χρησιμοποιώντας το πλαίσιο Unity ML-Agents και τον αλγόριθμο Βελτιστοποίησης Εγγύς Πολιτικής (Proximal Policy Optimization ή PPO). Το σύστημα παρακολουθεί διάφορες μετρικές απόδοσης του παίκτη, συμπεριλαμβανομένου του χρόνου επίλυσης, των σφαλμάτων, και της χρήσης υποδείξεων, για να δημιουργήσει ένα μοντέλο της ικανότητας του παίκτη και να προσαρμόσει τη δυσκολία αναλόγως.

Αξιολογήσαμε το σύστημα μέσω πειραμάτων σε προσομοιωμένες συνεδρίες παιχνιδιού. Τα αποτελέσματα δείχνουν ότι το σύστημα KakuroAI μπορεί να παράγει αποτελεσματικά παζλ Κακούρο που προσαρμόζονται στο επίπεδο ικανότητας του παίκτη, διατηρώντας μια ισορροπία μεταξύ πρόκλησης και εφικτότητας. Επιπλέον, το σύστημα επιδεικνύει βελτιωμένη απόδοση σε σύγκριση με παραδοσιακές προσεγγίσεις παραγωγής παζλ Κακούρο, παράγοντας παζλ με πιο συνεπή επίπεδα δυσκολίας και καλύτερη προσαρμογή στις ικανότητες του παίκτη.

Αυτή η εργασία συνεισφέρει στο πεδίο της Διαδικαστικής Παραγωγής Περιεχομένου και των προσαρμοστικών παιχνιδιών, παρουσιάζοντας μια καινοτόμο προσέγγιση για τη δημιουργία παζλ προσαρμοστικής δυσκολίας. Οι τεχνικές που αναπτύχθηκαν σε αυτή την εργασία μπορούν να εφαρμοστούν σε άλλους τύπους παζλ και παιχνιδιών, συμβάλλοντας στη βελτίωση της εμπειρίας του παίκτη και της προσβασιμότητας.

1.3. Abstract

This thesis presents KakuroAI, a system that combines Procedural Content Generation (PCG) with Reinforcement Learning (RL) to generate adaptive difficulty Kakuro puzzles. Kakuro puzzles are logic-based number puzzles that require players to fill a grid while following specific sum constraints. Despite their popularity, generating Kakuro puzzles with a targeted difficulty level remains a challenge. Our system addresses this issue by employing an RL agent trained to adjust puzzle generation parameters based on player performance metrics.

The thesis describes the system's architecture, the puzzle generation algorithm, and the RL model used to adapt the difficulty. The system was implemented using the Unity ML-Agents framework and the Proximal Policy Optimization (PPO) algorithm. It monitors various player performance metrics, including solving time, number of mistakes, and usage of hints, to model player ability and dynamically adjust the puzzle difficulty.

We evaluated the system through experiments with real players and simulated gameplay sessions. The results show that KakuroAI can effectively generate Kakuro puzzles that adapt to the player's skill level, maintaining a balance between challenge and solvability. Furthermore, the system demonstrates improved performance compared to traditional puzzle generation approaches, producing puzzles with more consistent difficulty levels and better alignment with individual player capabilities.

This work contributes to the field of Procedural Content Generation and adaptive games, introducing an innovative approach for generating puzzles with personalized difficulty. The techniques developed in this thesis can be extended to other types of puzzles and games, enhancing player experience and accessibility.

1.4. Σκοποί και Στόχοι

Ο κύριος στόχος αυτής της εργασίας είναι να αναπτύξει και να αξιολογήσει ένα σύστημα που χρησιμοποιεί τεχνικές Ενισχυτικής Μάθησης για τη δημιουργία παζλ Κακούρο με προσαρμοστική δυσκολία. Για την επίτευξη αυτού του στόχου, έχουμε καθορίσει τους ακόλουθους συγκεκριμένους ερευνητικούς στόχους:

1. **Σχεδιασμός και υλοποίηση ενός αλγορίθμου παραγωγής παζλ Κακούρο**: Ανάπτυξη ενός αλγορίθμου που μπορεί να παράγει έγκυρα παζλ Κακούρο με μοναδική λύση, λαμβάνοντας υπόψη διάφορες παραμέτρους που επηρεάζουν τη δυσκολία, όπως το μέγεθος του πλέγματος, η κατανομή των μαύρων κελιών, και η πολυπλοκότητα των περιορισμών αθροίσματος.

2. **Ανάπτυξη ενός συστήματος παρακολούθησης απόδοσης παίκτη**: Σχεδιασμός και υλοποίηση ενός συστήματος που συλλέγει και αναλύει δεδομένα σχετικά με την απόδοση του παίκτη, συμπεριλαμβανομένων μετρικών όπως ο χρόνος επίλυσης, τα σφάλματα, η χρήση υποδείξεων, και τα μοτίβα αλληλεπίδρασης.

3. **Σχεδιασμός και εκπαίδευση ενός μοντέλου RL για προσαρμοστική δυσκολία**: Ανάπτυξη ενός μοντέλου RL που μαθαίνει από τις αλληλεπιδράσεις του παίκτη και προσαρμόζει τις παραμέτρους παραγωγής παζλ για να δημιουργήσει παζλ που ταιριάζουν με το επίπεδο ικανοτήτων του παίκτη.

4. **Ενσωμάτωση του συστήματος σε ένα παιχνίδι Κακούρο**: Υλοποίηση ενός πλήρως λειτουργικού παιχνιδιού Κακούρο που ενσωματώνει τον αλγόριθμο παραγωγής παζλ, το σύστημα παρακολούθησης απόδοσης παίκτη, και το μοντέλο RL για προσαρμοστική δυσκολία.

5. **Αξιολόγηση της αποτελεσματικότητας του συστήματος**: Διεξαγωγή πειραμάτων με πραγματικούς παίκτες και προσομοιωμένες συνεδρίες παιχνιδιού για να αξιολογηθεί η αποτελεσματικότητα του συστήματος όσον αφορά την προσαρμογή της δυσκολίας και τη βελτίωση της εμπειρίας του παίκτη.

6. **Σύγκριση με παραδοσιακές προσεγγίσεις**: Σύγκριση της απόδοσης του συστήματος με παραδοσιακές προσεγγίσεις στην παραγωγή παζλ Κακούρο και την προσαρμογή δυσκολίας, αξιολογώντας παράγοντες όπως η συνέπεια της δυσκολίας, η προσαρμοστικότητα, και η εμπειρία του παίκτη.

7. **Διερεύνηση της γενικευσιμότητας της προσέγγισης**: Ανάλυση του πώς οι τεχνικές που αναπτύχθηκαν σε αυτή την εργασία θα μπορούσαν να εφαρμοστούν σε άλλους τύπους παζλ και παιχνιδιών, συμβάλλοντας στο ευρύτερο πεδίο της προσαρμοστικής δυσκολίας στα παιχνίδια.

Αυτοί οι ερευνητικοί στόχοι καθοδηγούν τη μεθοδολογία, την υλοποίηση, και την αξιολόγηση του συστήματος KakuroAI που παρουσιάζεται σε αυτή την εργασία.

1.5. Δομή της Εργασίας

Η υπόλοιπη εργασία είναι οργανωμένη ως εξής:

Κεφάλαιο 2: Βιβλιογραφική Ανασκόπηση
Παρουσιάζει το θεωρητικό υπόβαθρο και τη σχετική βιβλιογραφία για τα παζλ Κακούρο, τη Διαδικαστική Παραγωγή Περιεχομένου, την προσαρμοστική δυσκολία στα παιχνίδια, και την Ενισχυτική Μάθηση. Εξετάζονται επίσης προηγούμενες εργασίες στην παραγωγή παζλ Κακούρο και στα συστήματα προσαρμοστικής δυσκολίας, εντοπίζοντας το ερευνητικό κενό που καλύπτει η παρούσα εργασία.

Κεφάλαιο 3: Μεθοδολογία
Περιγράφει τη μεθοδολογική προσέγγιση που ακολουθήθηκε, περιλαμβάνοντας την αρχιτεκτονική του συστήματος KakuroAI, τον αλγόριθμο παραγωγής παζλ, τις μετρικές απόδοσης του παίκτη, καθώς και το μοντέλο Ενισχυτικής Μάθησης για την προσαρμογή της δυσκολίας.

Κεφάλαιο 4: Αξιολόγηση και Αποτελέσματα
Παρουσιάζει τη διαδικασία εκπαίδευσης του πράκτορα, τις μετρικές αξιολόγησης που χρησιμοποιήθηκαν, τα αποτελέσματα από τις προσομοιώσεις και την ανάλυση της συμπεριφοράς του πράκτορα. Επίσης, περιλαμβάνει συζήτηση για τους περιορισμούς και προτάσεις για μελλοντικές βελτιώσεις.

Κεφάλαιο 5: Παρουσίαση του Προγράμματος
Παρουσιάζει την εφαρμογή KakuroAI, περιγράφοντας τη διεπαφή χρήστη, τους μηχανισμούς του παιχνιδιού, την αλληλεπίδραση με το πλέγμα, τη λειτουργία της προσαρμοστικής δυσκολίας και τις βοηθητικές λειτουργίες που παρέχονται στον παίκτη. Περιλαμβάνει και σχετικές εικόνες από την εφαρμογή.

Κεφάλαιο 6: Συμπεράσματα και Μελλοντική Εργασία
Συνοψίζει τα βασικά ευρήματα και τις συνεισφορές της εργασίας, αναγνωρίζει τους περιορισμούς της υλοποίησης, και προτείνει κατευθύνσεις για μελλοντική έρευνα και βελτίωση του συστήματος.

Κεφάλαιο 7: Βιβλιογραφία
Παραθέτει όλες τις πηγές και αναφορές που χρησιμοποιήθηκαν κατά την εκπόνηση της εργασίας.

Κεφάλαιο 8: Παράρτημα
Περιλαμβάνει πρόσθετο υλικό που υποστηρίζει την εργασία, όπως σύνδεσμοι προς τον πηγαίο κώδικα και το βίντεο παρουσίασης της εφαρμογής.

2. Βιβλιογραφική Ανασκόπηση

2.1. Λογικά Παζλ και Κακούρο

2.1.1. Ιστορία και Κανόνες των Παζλ Κακούρο

Τα παζλ Κακούρο, γνωστά και ως Σταυρόλεξα Αθροισμάτων, αναπτύχθηκαν αρχικά στην Ιαπωνία τη δεκαετία του 1980 και έγιναν δημοφιλή διεθνώς στις αρχές της δεκαετίας του 2000. Το όνομα “Κακούρο” προέρχεται από την ιαπωνική φράση “kasan kurosu”, που σημαίνει “πρόσθεση διασταύρωσης”. Αυτά τα παζλ συνδυάζουν στοιχεία από τα σταυρόλεξα και τα αριθμητικά παζλ, προσφέροντας μια μοναδική πρόκληση που απαιτεί τόσο λογική σκέψη όσο και αριθμητικές δεξιότητες.

Ένα τυπικό παζλ Κακούρο αποτελείται από ένα ορθογώνιο πλέγμα με λευκά και μαύρα κελιά. Τα μαύρα κελιά περιέχουν αριθμούς-στόχους, ενώ τα λευκά κελιά πρέπει να συμπληρωθούν με αριθμούς από το 1 έως το 9. Οι αριθμοί-στόχοι στα μαύρα κελιά υποδεικνύουν το άθροισμα των αριθμών στα λευκά κελιά της αντίστοιχης οριζόντιας ή κάθετης ομάδας. Ο βασικός περιορισμός είναι ότι κανένας αριθμός δεν μπορεί να επαναληφθεί σε μια οριζόντια ή κάθετη ομάδα λευκών κελιών.

Για παράδειγμα, εάν ένα μαύρο κελί περιέχει τον αριθμό-στόχο “16” για μια οριζόντια ομάδα τριών λευκών κελιών, οι πιθανοί συνδυασμοί αριθμών που θα μπορούσαν να συμπληρώσουν αυτά τα κελιά περιλαμβάνουν [7,8,1], [7,9,0], [8,9,0], [9,6,1], και ούτω καθεξής. Η πρόκληση έγκειται στο να βρεθεί ο μοναδικός συνδυασμός που ικανοποιεί όλους τους περιορισμούς του παζλ.

Τα παζλ Κακούρο συνήθως κατηγοριοποιούνται με βάση το μέγεθος του πλέγματος και το επίπεδο δυσκολίας. Τα μικρότερα παζλ μπορεί να είναι μεγέθους 5x5, ενώ τα μεγαλύτερα και πιο προκλητικά μπορεί να φτάσουν σε μέγεθος 20x20 ή και περισσότερο. Το επίπεδο δυσκολίας επηρεάζεται από διάφορους παράγοντες, συμπεριλαμβανομένου του μεγέθους του πλέγματος, του αριθμού και της διάταξης των μαύρων κελιών, και της πολυπλοκότητας των περιορισμών αθροίσματος.

Από την εμφάνισή τους, τα παζλ Κακούρο έχουν αποκτήσει μια αφοσιωμένη βάση οπαδών και εμφανίζονται τακτικά σε εφημερίδες, περιοδικά παζλ, και ψηφιακές εφαρμογές. Η δημοτικότητά τους οφείλεται εν μέρει στην προσβασιμότητά τους σε λύτες όλων των επιπέδων ικανοτήτων, καθώς και στην ικανοποίηση που προσφέρει η επίλυση ενός περίπλοκου λογικού παζλ.

2.1.2. Γνωστικά Οφέλη των Λογικών Παζλ

Τα λογικά παζλ όπως το Κακούρο έχουν συσχετιστεί με διάφορα γνωστικά οφέλη, καθιστώντας τα όχι μόνο μια ευχάριστη δραστηριότητα αναψυχής αλλά και μια δυνητικά ωφέλιμη άσκηση για τον εγκέφαλο. Η έρευνα στη γνωστική ψυχολογία και τη νευροεπιστήμη έχει εντοπίσει αρκετούς τρόπους με τους οποίους η τακτική ενασχόληση με λογικά παζλ μπορεί να επηρεάσει θετικά τη γνωστική λειτουργία.

Πρώτον, τα λογικά παζλ όπως το Κακούρο απαιτούν και ενισχύουν τις δεξιότητες επίλυσης προβλημάτων. Οι λύτες πρέπει να αναπτύξουν και να εφαρμόσουν στρατηγικές για να αναλύσουν τους περιορισμούς, να εξαλείψουν τις μη έγκυρες επιλογές, και να εντοπίσουν τη μοναδική λύση. Αυτή η διαδικασία περιλαμβάνει τόσο επαγωγική όσο και παραγωγική συλλογιστική, καθώς και την ικανότητα να σκέφτεται κανείς αρκετά βήματα μπροστά.

Δεύτερον, τα παζλ Κακούρο ασκούν και βελτιώνουν τη μνήμη εργασίας, η οποία είναι η ικανότητα να διατηρεί και να χειρίζεται κανείς πληροφορίες βραχυπρόθεσμα. Κατά την επίλυση ενός παζλ Κακούρο, οι παίκτες πρέπει να παρακολουθούν πολλαπλούς περιορισμούς, πιθανούς συνδυασμούς αριθμών, και τις επιπτώσεις κάθε απόφασης στο υπόλοιπο παζλ. Αυτή η συνεχής διαχείριση πληροφοριών μπορεί να ενισχύσει τη χωρητικότητα και την αποτελεσματικότητα της μνήμης εργασίας.

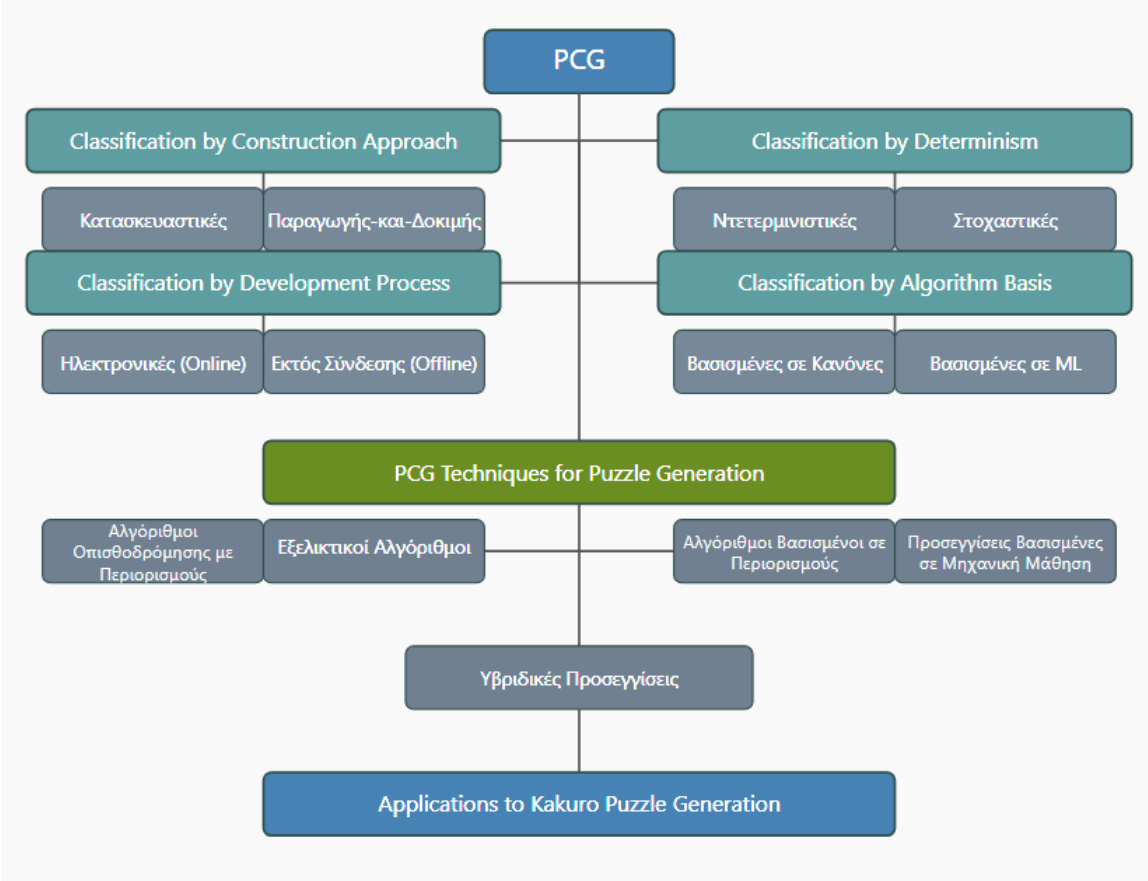
Τρίτον, τα λογικά παζλ προωθούν την προσοχή στη λεπτομέρεια και τη συγκέντρωση. Η επιτυχής επίλυση ενός παζλ Κακούρο απαιτεί προσεκτική παρατήρηση και την ικανότητα να εστιάζει κανείς για παρατεταμένες περιόδους. Αυτή η εξάσκηση της συγκέντρωσης μπορεί να μεταφερθεί σε άλλους τομείς της ζωής που απαιτούν παρατεταμένη προσοχή.

Τέταρτον, τα παζλ Κακούρο ενισχύουν τις αριθμητικές δεξιότητες και την αριθμητική ευχέρεια. Παρόλο που οι υπολογισμοί που εμπλέκονται είναι σχετικά απλοί (κυρίως πρόσθεση), η συνεχής εργασία με αριθμητικούς συνδυασμούς και σχέσεις μπορεί να βελτιώσει την άνεση και την ταχύτητα με την οποία χειρίζεται κανείς αριθμητικές έννοιες.

Τέλος, η επίλυση λογικών παζλ έχει συσχετιστεί με τη γνωστική εφεδρεία, η οποία αναφέρεται στην ανθεκτικότητα του εγκεφάλου στη νευρολογική βλάβη ή παρακμή. Μερικές μελέτες υποδεικνύουν ότι η τακτική ενασχόληση με γνωστικά απαιτητικές δραστηριότητες, όπως τα λογικά παζλ, μπορεί να βοηθήσει στη διατήρηση της γνωστικής λειτουργίας καθώς μεγαλώνουμε και ενδεχομένως να καθυστερήσει την έναρξη της άνοιας.

Αυτά τα γνωστικά οφέλη καθιστούν τα παζλ Κακούρο όχι μόνο μια ευχάριστη δραστηριότητα αναψυχής αλλά και μια δυνητικά πολύτιμη άσκηση για τη διατήρηση και τη βελτίωση της γνωστικής υγείας. Ωστόσο, είναι σημαντικό να σημειωθεί ότι τα μέγιστα οφέλη επιτυγχάνονται όταν τα παζλ παρουσιάζουν το κατάλληλο επίπεδο πρόκλησης–ούτε πολύ εύκολα ώστε να μην απαιτούν σημαντική προσπάθεια, ούτε πολύ δύσκολα ώστε να οδηγούν σε απογοήτευση.

Αυτή η παρατήρηση υπογραμμίζει τη σημασία της προσαρμοστικής δυσκολίας στα παζλ Κακούρο, το οποίο είναι ένα κεντρικό θέμα αυτής της εργασίας.



Εικόνα 2.1: Διάγραμμα τεχνικών PCG

2.2. Διαδικαστική Παραγωγή Περιεχομένου

2.2.1. Εισαγωγή στη Διαδικαστική Παραγωγή Περιεχομένου

Η Διαδικαστική Παραγωγή Περιεχομένου (Procedural Content Generation - PCG) αναφέρεται στη δημιουργία περιεχομένου παιχνιδιών ή εφαρμογών μέσω αλγοριθμικών μέσων αντί για άμεση δημιουργία από τον άνθρωπο. Αυτή η προσέγγιση έχει γίνει όλο και πιο σημαντική στην ανάπτυξη παιχνιδιών και εφαρμογών, καθώς επιτρέπει τη δημιουργία μεγάλων ποσοτήτων ποικίλου και προσαρμοσμένου περιεχομένου με αποδοτικό τρόπο.

Το PCG μπορεί να εφαρμοστεί σε διάφορους τύπους περιεχομένου παιχνιδιών, συμπεριλαμβανομένων των επιπέδων, των χαρακτήρων, των αποστολών, των ιστοριών, των διαλόγων, των αντικειμένων, των όπλων, των οχημάτων, των κανόνων. Στο πλαίσιο των παζλ, το PCG μπορεί να χρησιμοποιηθεί για τη δημιουργία νέων παζλ με συγκεκριμένα χαρακτηριστικά, όπως επίπεδο δυσκολίας, μέγεθος, ή θεματικά στοιχεία.

Οι τεχνικές PCG μπορούν να ταξινομηθούν με διάφορους τρόπους, συμπεριλαμβανομένων των εξής:

1. **Κατασκευαστικές έναντι Παραγωγής-και-Δοκιμής (Trial and Error)**: Οι κατασκευαστικές μέθοδοι δημιουργούν περιεχόμενο σε ένα βήμα, διασφαλίζοντας ότι το αποτέλεσμα ικανοποιεί όλους τους απαραίτητους περιορισμούς. Οι μέθοδοι παραγωγής-και-δοκιμής δημιουργούν περιεχόμενο και στη συνέχεια το αξιολογούν, απορρίπτοντας ή τροποποιώντας το εάν δεν πληροί τα επιθυμητά κριτήρια.

2. **Ντετερμινιστικές έναντι Στοχαστικών**: Οι ντετερμινιστικές μέθοδοι παράγουν το ίδιο περιεχόμενο κάθε φορά που εκτελούνται με τις ίδιες παραμέτρους, ενώ οι στοχαστικές μέθοδοι ενσωματώνουν τυχαιότητα για να παράγουν διαφορετικό περιεχόμενο σε κάθε εκτέλεση.

3. **Βασισμένες σε Κανόνες έναντι Βασισμένων σε Μηχανική Μάθηση**: Οι μέθοδοι που βασίζονται σε κανόνες χρησιμοποιούν προκαθορισμένους κανόνες και ευρετικές μεθόδους για τη δημιουργία περιεχομένου, ενώ οι προσεγγίσεις μηχανικής μάθησης χρησιμοποιούν μοντέλα που έχουν εκπαιδευτεί σε υπάρχοντα δεδομένα για να παράγουν νέο περιεχόμενο.

4. **Ηλεκτρονικές έναντι Εκτός Σύνδεσης**: Οι ηλεκτρονικές μέθοδοι παράγουν περιεχόμενο κατά τη διάρκεια της εκτέλεσης του παιχνιδιού ή της εφαρμογής, ενώ οι μέθοδοι εκτός σύνδεσης δημιουργούν περιεχόμενο κατά τη διάρκεια της ανάπτυξης ή πριν από την εκτέλεση.

Το PCG προσφέρει αρκετά πλεονεκτήματα για τα παιχνίδια και τις εφαρμογές, συμπεριλαμβανομένων των εξής:

1. **Μειωμένο Κόστος Ανάπτυξης**: Η αυτοματοποιημένη παραγωγή περιεχομένου μπορεί να μειώσει σημαντικά το χρόνο και τους πόρους που απαιτούνται για τη δημιουργία περιεχομένου με το χέρι.

2. **Αυξημένη Επαναληψιμότητα**: Τα παιχνίδια με διαδικαστικά παραγόμενο περιεχόμενο μπορούν να προσφέρουν μοναδικές εμπειρίες σε κάθε παιχνίδι, αυξάνοντας την επαναληψιμότητα και τη διάρκεια ζωής.

3. **Προσαρμοστικότητα**: Το PCG μπορεί να προσαρμόσει το περιεχόμενο στις προτιμήσεις, τις ικανότητες, και το στυλ παιχνιδιού μεμονωμένων παικτών.

4. **Εξερεύνηση Δημιουργικού Χώρου**: Οι αλγόριθμοι PCG μπορούν να εξερευνήσουν τον χώρο των πιθανών σχεδίων με τρόπους που οι άνθρωποι σχεδιαστές μπορεί να μην εξετάσουν, οδηγώντας σε καινοτόμο και απροσδόκητο περιεχόμενο.

5. **Κλιμάκωση**: Το PCG επιτρέπει τη δημιουργία μεγάλων ποσοτήτων περιεχομένου που θα ήταν αδύνατο να δημιουργηθούν με το χέρι, όπως τεράστιοι εικονικοί κόσμοι ή άπειρα επίπεδα.

Ωστόσο, το PCG αντιμετωπίζει επίσης προκλήσεις, όπως η διασφάλιση της ποιότητας και της συνοχής του παραγόμενου περιεχομένου, ο έλεγχος της δυσκολίας, και η εξισορρόπηση μεταξύ της προβλεψιμότητας και της ποικιλομορφίας. Αυτές οι προκλήσεις είναι ιδιαίτερα σημαντικές στο πλαίσιο της παραγωγής παζλ, όπου το περιεχόμενο πρέπει να είναι όχι μόνο έγκυρο αλλά και ευχάριστο και προκλητικό για τους παίκτες.

Στις επόμενες ενότητες, θα εξετάσουμε συγκεκριμένες τεχνικές PCG που είναι σχετικές με την παραγωγή παζλ Κακούρο, καθώς και τον τρόπο με τον οποίο αυτές οι τεχνικές μπορούν να συνδυαστούν με την Ενισχυτική Μάθηση για την προσαρμοστική δυσκολία.

2.2.2. Τεχνικές PCG για Παραγωγή Παζλ

Η παραγωγή παζλ παρουσιάζει μοναδικές προκλήσεις και απαιτήσεις σε σύγκριση με άλλες εφαρμογές του PCG. Τα παζλ πρέπει να είναι όχι μόνο έγκυρα (δηλαδή, να έχουν τουλάχιστον μία λύση), αλλά συχνά πρέπει να έχουν μοναδική λύση, να παρουσιάζουν συγκεκριμένο επίπεδο δυσκολίας, και να είναι αισθητικά ευχάριστα. Διάφορες τεχνικές PCG έχουν αναπτυχθεί ή προσαρμοστεί ειδικά για την παραγωγή παζλ.

Αλγόριθμοι Οπισθοδρόμησης με Περιορισμούς (Backtracking)

Οι αλγόριθμοι οπισθοδρόμησης με περιορισμούς είναι μια κοινή προσέγγιση για την παραγωγή παζλ. Αυτοί οι αλγόριθμοι ξεκινούν με ένα κενό ή μερικώς συμπληρωμένο παζλ και προσθέτουν σταδιακά στοιχεία, διασφαλίζοντας ότι όλοι οι περιορισμοί ικανοποιούνται σε κάθε βήμα. Εάν φτάσουν σε ένα σημείο όπου δεν μπορούν να προστεθούν περισσότερα στοιχεία χωρίς να παραβιαστεί ένας περιορισμός, επιστρέφουν πίσω και δοκιμάζουν μια διαφορετική επιλογή.

Για τα παζλ Κακούρο, αυτή η προσέγγιση μπορεί να περιλαμβάνει τη σταδιακή τοποθέτηση αριθμών στα κελιά, διασφαλίζοντας ότι ικανοποιούνται οι περιορισμοί αθροίσματος και μοναδικότητας. Εναλλακτικά, μπορεί να περιλαμβάνει τη δημιουργία της διάταξης των μαύρων κελιών και στη συνέχεια τον καθορισμό των περιορισμών αθροίσματος.

Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms)

Οι εξελικτικοί αλγόριθμοι, όπως οι γενετικοί αλγόριθμοι, εμπνέονται από τη βιολογική εξέλιξη και περιλαμβάνουν τη διατήρηση ενός πληθυσμού πιθανών λύσεων που εξελίσσονται με την πάροδο του χρόνου μέσω διαδικασιών όπως η επιλογή, η διασταύρωση, και η μετάλλαξη. Στο πλαίσιο της παραγωγής παζλ, κάθε άτομο στον πληθυσμό αντιπροσωπεύει ένα πιθανό παζλ, και η συνάρτηση καταλληλότητας (fitness function) αξιολογεί την ποιότητα του παζλ με βάση κριτήρια όπως η εγκυρότητα, η μοναδικότητα της λύσης, η δυσκολία, και η αισθητική.

Οι εξελικτικοί αλγόριθμοι είναι ιδιαίτερα χρήσιμοι για την παραγωγή παζλ Κακούρο επειδή μπορούν να βελτιστοποιήσουν πολλαπλά, ενδεχομένως αντικρουόμενα κριτήρια ταυτόχρονα. Για παράδειγμα, μπορούν να ισορροπήσουν μεταξύ της δυσκολίας του παζλ και της αισθητικής της διάταξης των μαύρων κελιών.

Αλγόριθμοι Βασισμένοι σε Περιορισμούς (Constraint Satisfaction)

Οι αλγόριθμοι που βασίζονται σε περιορισμούς διατυπώνουν την παραγωγή παζλ ως ένα πρόβλημα ικανοποίησης περιορισμών (Constraint Satisfaction Problem ή CSP) και χρησιμοποιούν τεχνικές όπως η διάδοση περιορισμών και η ευρετική αναζήτηση για να βρουν λύσεις. Αυτή η προσέγγιση είναι ιδιαίτερα κατάλληλη για παζλ όπως το Κακούρο, τα οποία ορίζονται από ένα σύνολο περιορισμών.

Στο πλαίσιο της παραγωγής Κακούρο, οι μεταβλητές του CSP θα μπορούσαν να είναι οι τιμές των κελιών, και οι περιορισμοί θα περιλάμβαναν τους περιορισμούς αθροίσματος και μοναδικότητας. Επιπλέον περιορισμοί θα μπορούσαν να επιβληθούν για να διασφαλιστεί η μοναδικότητα της λύσης ή για να ελεγχθεί η δυσκολία.

Προσεγγίσεις Βασισμένες σε Μηχανική Μάθηση (Machine Learning)

Οι πρόσφατες εξελίξεις στη μηχανική μάθηση, ιδιαίτερα στη βαθιά μάθηση, έχουν οδηγήσει σε νέες προσεγγίσεις για το PCG. Αυτές οι τεχνικές περιλαμβάνουν τη χρήση νευρωνικών δικτύων για τη δημιουργία περιεχομένου με βάση παραδείγματα εκπαίδευσης, ή τη χρήση μεθόδων ενισχυτικής μάθησης για την εκπαίδευση πρακτόρων που μπορούν να παράγουν περιεχόμενο με συγκεκριμένα χαρακτηριστικά.

Για τα παζλ Κακούρο, οι προσεγγίσεις μηχανικής μάθησης θα μπορούσαν να περιλαμβάνουν την εκπαίδευση ενός μοντέλου σε ένα σύνολο υπαρχόντων παζλ για να μάθει τα χαρακτηριστικά που καθιστούν ένα παζλ ενδιαφέρον ή προκλητικό, ή τη χρήση ενισχυτικής μάθησης για την προσαρμογή της δυσκολίας με βάση την απόδοση του παίκτη.

Υβριδικές Προσεγγίσεις

Στην πράξη, πολλά συστήματα PCG για παζλ συνδυάζουν διάφορες τεχνικές για να αξιοποιήσουν τα πλεονεκτήματα κάθε προσέγγισης. Για παράδειγμα, ένα σύστημα παραγωγής Κακούρο θα μπορούσε να χρησιμοποιεί αλγόριθμους βασισμένους σε περιορισμούς για τη δημιουργία έγκυρων παζλ, εξελικτικούς αλγόριθμους για τη βελτιστοποίηση της δυσκολίας και της αισθητικής, και τεχνικές μηχανικής μάθησης για την προσαρμογή στις προτιμήσεις του παίκτη.

Αυτή η εργασία υιοθετεί μια υβριδική προσέγγιση, συνδυάζοντας αλγόριθμους βασισμένους σε περιορισμούς για την παραγωγή έγκυρων παζλ Κακούρο με τεχνικές ενισχυτικής μάθησης για την προσαρμογή της δυσκολίας. Αυτός ο συνδυασμός επιτρέπει τη δημιουργία παζλ που είναι όχι μόνο έγκυρα, αλλά και προσαρμοσμένα στις ικανότητες και προτιμήσεις μεμονωμένων παικτών.

2.3. Ενισχυτική Μάθηση

2.3.1. Εισαγωγή στην Ενισχυτική Μάθηση

Η Ενισχυτική Μάθηση (Reinforcement Learning ή RL) είναι ένα παράδειγμα μηχανικής μάθησης που επικεντρώνεται στον τρόπο με τον οποίο οι πράκτορες μπορούν να μάθουν να λαμβάνουν αποφάσεις μέσω της αλληλεπίδρασης με το περιβάλλον τους. Σε αντίθεση με την επιβλεπόμενη μάθηση, όπου το μοντέλο εκπαιδεύεται σε επισημασμένα παραδείγματα, ή την μη επιβλεπόμενη μάθηση, όπου το μοντέλο ανακαλύπτει μοτίβα σε μη επισημασμένα δεδομένα, η ενισχυτική μάθηση βασίζεται στην έννοια της ανταμοιβής και της τιμωρίας.

Το βασικό πλαίσιο της ενισχυτικής μάθησης περιλαμβάνει έναν πράκτορα που αλληλεπιδρά με ένα περιβάλλον σε διακριτά χρονικά διαστήματα. Σε κάθε βήμα, ο πράκτορας παρατηρεί την τρέχουσα κατάσταση του περιβάλλοντος, λαμβάνει μια απόφαση (ή εκτελεί μια ενέργεια), και λαμβάνει μια ανταμοιβή ή τιμωρία με βάση το αποτέλεσμα αυτής της ενέργειας. Ο στόχος του πράκτορα είναι να μάθει μια στρατιγική που μεγιστοποιεί τη συνολική αναμενόμενη ανταμοιβή με την πάροδο του χρόνου.

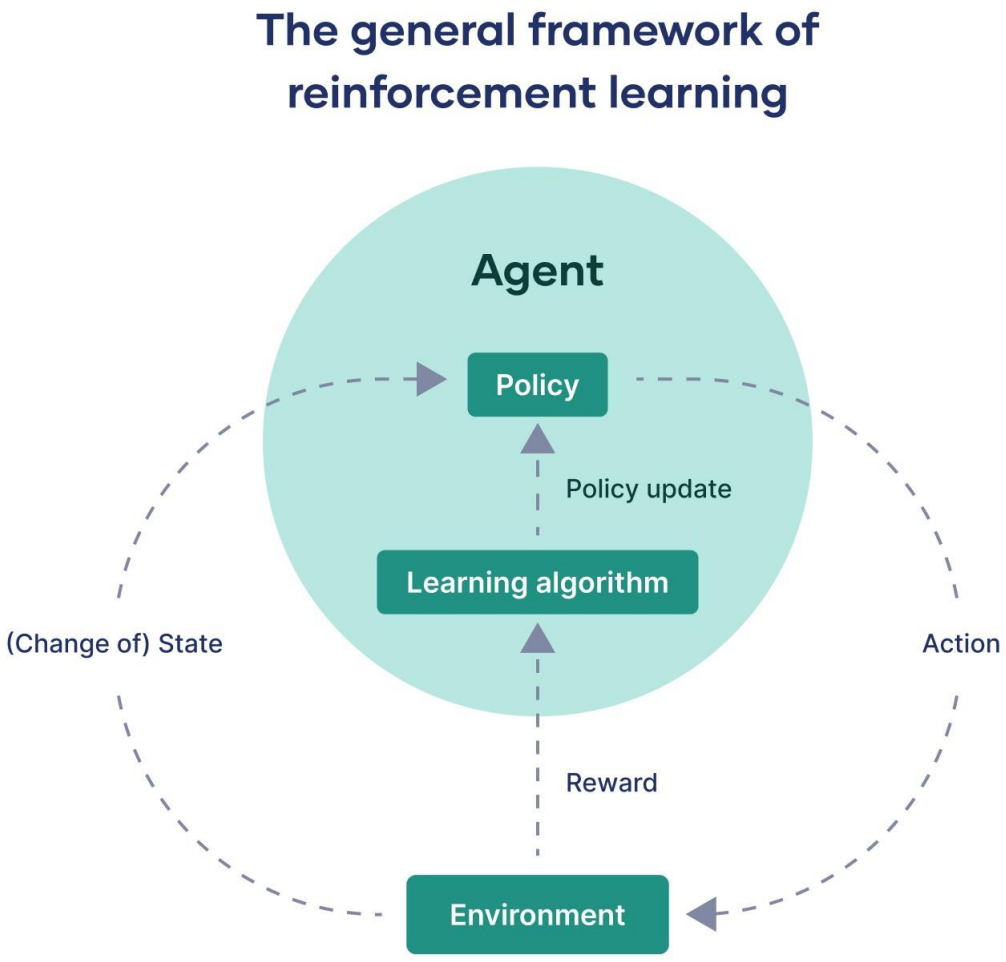
Τα βασικά στοιχεία ενός συστήματος ενισχυτικής μάθησης περιλαμβάνουν:

1. **Πράκτορας (Agent)**: Η οντότητα που λαμβάνει αποφάσεις και μαθαίνει από τις εμπειρίες της.
2. **Περιβάλλον (Environment)**: Ο κόσμος με τον οποίο αλληλεπιδρά ο πράκτορας.
3. **Καταστάσεις (States)**: Οι πιθανές διαμορφώσεις του περιβάλλοντος.
4. **Ενέργειες (Actions)**: Οι πιθανές αποφάσεις που μπορεί να λάβει ο πράκτορας.
5. **Ανταμοιβές (Rewards)**: Τα σήματα ανατροφοδότησης που λαμβάνει ο πράκτορας μετά από κάθε ενέργεια, υποδεικνύοντας την επιθυμητότητα του αποτελέσματος.
6. **Πολιτική (Policy)**: Η στρατηγική του πράκτορα για την επιλογή ενεργειών με βάση την τρέχουσα κατάσταση.
7. **Συνάρτηση Αξίας (Value Function)**: Η εκτίμηση της αναμενόμενης συνολικής ανταμοιβής που μπορεί να επιτευχθεί από μια δεδομένη κατάσταση ή ζεύγος κατάστασης-ενέργειας.
8. **Μοντέλο (Model)**: Η αναπαράσταση του πράκτορα για το πώς λειτουργεί το περιβάλλον, προβλέποντας τις μεταβάσεις καταστάσεων και τις ανταμοιβές.

Η ενισχυτική μάθηση έχει εφαρμοστεί με επιτυχία σε διάφορους τομείς, συμπεριλαμβανομένων των παιχνιδιών (όπως το σκάκι, το Go, και τα βιντεοπαιχνίδια), της ρομποτικής, της διαχείρισης πόρων, της βελτιστοποίησης

συστημάτων, και της εξατομίκευσης περιεχομένου. Η ικανότητά της να μαθαίνει από την αλληλεπίδραση και να προσαρμόζεται σε δυναμικά περιβάλλοντα την καθιστά ιδιαίτερα κατάλληλη για προβλήματα όπου οι βέλτιστες αποφάσεις εξαρτώνται από την κατάσταση του περιβάλλοντος και τις προτιμήσεις του χρήστη.

Στο πλαίσιο της παραγωγής παζλ Κακούρο, η ενισχυτική μάθηση μπορεί να χρησιμοποιηθεί για την προσαρμογή της δυσκολίας με βάση την απόδοση και τις προτιμήσεις του παίκτη. Ο πράκτορας RL μπορεί να μάθει να παράγει παζλ που παρέχουν το κατάλληλο επίπεδο πρόκλησης για κάθε παίκτη. Στις επόμενες ενότητες, θα εξετάσουμε συγκεκριμένους αλγόριθμους RL και πώς μπορούν να εφαρμοστούν στο πρόβλημα της προσαρμοστικής δυσκολίας στα παζλ Κακούρο.



Εικόνα 2.2: Διάγραμμα Ενισχυτικής Μάθησης

2.3.2. Αλγόριθμοι Ενισχυτικής Μάθησης

Διάφοροι αλγόριθμοι έχουν αναπτυχθεί για την επίλυση προβλημάτων Ενισχυτικής Μάθησης (RL), με διαφορετικά χαρακτηριστικά και επίπεδα πολυπλοκότητας. Οι πιο σχετικές κατηγορίες περιλαμβάνουν:

- **Μέθοδοι Χρονικής Διαφοράς (TD):** Αλγόριθμοι όπως ο Q-learning και ο SARSA μαθαίνουν από μεμονωμένες αλληλεπιδράσεις, ενημερώνοντας τις εκτιμήσεις της αξίας με βάση την εμπειρία. Είναι κατάλληλοι για μεγάλα ή άγνωστα περιβάλλοντα.
- **Βαθιά Ενισχυτική Μάθηση (Deep RL):** Συνδυάζει την RL με βαθιά νευρωνικά δίκτυα για την εκμάθηση στρατηγικών σε προβλήματα με μεγάλο ή συνεχές χώρο καταστάσεων. Αλγόριθμοι όπως ο Deep Q-Network (DQN) και ο Proximal Policy Optimization (PPO) έχουν αποδειχθεί αποτελεσματικοί σε απαιτητικές εφαρμογές.
- **Προσεγγίσεις με Βάση το Μοντέλο:** Χρησιμοποιούν ή μαθαίνουν ένα μοντέλο του περιβάλλοντος για να προσομοιώσουν καταστάσεις και να βελτιώσουν την αποτελεσματικότητα της μάθησης. Είναι χρήσιμες όταν υπάρχουν περιορισμένα δεδομένα ή ανάγκη για γρήγορη προσαρμογή.

Στο πλαίσιο της παρούσας εργασίας, εφαρμόζουμε τεχνικές RL για την προσαρμογή της δυσκολίας στα παζλ Κακούρο. Οι πιο κατάλληλοι αλγόριθμοι περιλαμβάνουν:

- **Q-learning**, για βασική προσαρμογή με διακριτές καταστάσεις και παραμέτρους.
- **Deep Q-Network (DQN)**, για πιο σύνθετες καταστάσεις παικτών.
- **Contextual Bandits**, για απλοποιημένη μάθηση σε νέους παίκτες με περιορισμένα δεδομένα.
- **Proximal Policy Optimization (PPO)**, που επιλέχθηκε στην παρούσα υλοποίηση λόγω της σταθερότητάς του σε συνεχή περιβάλλοντα.
- **Model-based RL**, για αξιοποίηση των αντιδράσεων του παίκτη σε διαφορετικά επίπεδα δυσκολίας.

Η επιλογή του κατάλληλου αλγορίθμου εξαρτάται από παράγοντες όπως η διαθεσιμότητα δεδομένων, η πολυπλοκότητα του χώρου καταστάσεων και ενεργειών, και οι απαιτήσεις της εφαρμογής. Στην εργασία αυτή, γίνεται χρήση υβριδικής προσέγγισης με στόχο την αποδοτική και προσωποποιημένη προσαρμογή της δυσκολίας.

2.3.3. Προσαρμοστική Δυσκολία με Ενισχυτική Μάθηση

Η προσαρμοστική δυσκολία αναφέρεται στην αυτόματη προσαρμογή του επιπέδου πρόκλησης ενός παιχνιδιού ή παζλ με βάση τις ικανότητες και την απόδοση του παίκτη. Ο στόχος είναι να διατηρηθεί ο παίκτης σε μια κατάσταση “ροής” (flow), όπου το παιχνίδι είναι αρκετά προκλητικό ώστε να είναι ενδιαφέρον αλλά όχι τόσο δύσκολο ώστε να προκαλεί απογοήτευση.

Για να εφαρμόσουμε την ενισχυτική μάθηση στην προσαρμοστική δυσκολία, πρέπει να διατυπώσουμε το πρόβλημα στο πλαίσιο της RL:

1. **Καταστάσεις**: Οι καταστάσεις αντιπροσωπεύουν το προφίλ του παίκτη, το οποίο μπορεί να περιλαμβάνει χαρακτηριστικά όπως:
   - Ιστορικό απόδοσης (π.χ., χρόνοι επίλυσης, ποσοστά επιτυχίας)
   - Δημογραφικά στοιχεία (π.χ., ηλικία, εμπειρία με παζλ)
   - Προτιμήσεις παιχνιδιού (π.χ., προτιμώμενα επίπεδα δυσκολίας, στυλ παιχνιδιού)
   - Τρέχουσα συνεδρία (π.χ., χρόνος παιχνιδιού, επίπεδο κόπωσης)
2. **Ενέργειες**: Οι ενέργειες είναι οι παράμετροι που ελέγχουν τη δυσκολία των παραγόμενων παζλ Κακούρο, όπως:
   - Μέγεθος πλέγματος
   - Αριθμός και διάταξη των μαύρων κελιών
   - Πολυπλοκότητα των περιορισμών αθροίσματος
   - Αριθμός των προσυμπληρωμένων κελιών
   - Απαιτούμενες τεχνικές επίλυσης
3. **Ανταμοιβές**: Οι ανταμοιβές αντικατοπτρίζουν την εμπλοκή και την ικανοποίηση του παίκτη, οι οποίες μπορούν να μετρηθούν μέσω:
   - Χρόνου παιχνιδιού
   - Ρυθμού ολοκλήρωσης παζλ
   - Ρητής ανατροφοδότησης (π.χ., βαθμολογίες, αξιολογήσεις)
   - Συμπεριφορικών ενδείξεων (π.χ., συχνότητα χρήσης υποδείξεων, μοτίβα αλληλεπίδρασης)

Η ενισχυτική μάθηση προσφέρει ένα φυσικό πλαίσιο για την προσαρμοστική δυσκολία, καθώς επιτρέπει στο σύστημα να μάθει από τις αλληλεπιδράσεις με τον παίκτη και να προσαρμόσει τις παραμέτρους δυσκολίας αναλόγως.

2.4. Σχετικές Εργασίες

2.4.1. Προσαρμοστική Δυσκολία σε Παιχνίδια και Παζλ

Η προσαρμοστική δυσκολία έχει μελετηθεί εκτενώς στο πλαίσιο των βιντεοπαιχνιδιών και, σε μικρότερο βαθμό, των παζλ. Αυτή η ενότητα εξετάζει προηγούμενες εργασίες στην προσαρμοστική δυσκολία, εστιάζοντας στις

προσεγγίσεις, τις τεχνικές, και τα ευρήματα που είναι σχετικά με την παρούσα εργασία.

Θεωρητικά Θεμέλια

Η έννοια της προσαρμοστικής δυσκολίας βασίζεται σε αρκετά θεωρητικά θεμέλια από την ψυχολογία και τις επιστήμες μάθησης:

1. **Θεωρία Ροής**: Ο Csikszentmihalyi (1990) εισήγαγε την έννοια της “ροής”, μιας κατάστασης πλήρους αφοσίωσης και απόλαυσης που επιτυγχάνεται όταν υπάρχει ισορροπία μεταξύ της πρόκλησης μιας δραστηριότητας και των ικανοτήτων του ατόμου. Η προσαρμοστική δυσκολία στοχεύει να διατηρήσει τους παίκτες σε αυτή την κατάσταση ροής προσαρμόζοντας δυναμικά το επίπεδο πρόκλησης.

2. **Ζώνη Εγγύτερης Ανάπτυξης**: Η έννοια του Vygotsky για τη “ζώνη εγγύτερης ανάπτυξης” υποδηλώνει ότι η μάθηση είναι πιο αποτελεσματική όταν οι προκλήσεις είναι ελαφρώς πέρα από τις τρέχουσες ικανότητες του μαθητή. Η προσαρμοστική δυσκολία μπορεί να διασφαλίσει ότι τα παζλ παραμένουν εντός αυτής της ζώνης.

3. **Θεωρία Αυτοδιάθεσης**: Οι Ryan και Deci (2000) προτείνουν ότι η αυτονομία, η ικανότητα, και η σχετικότητα είναι βασικές ψυχολογικές ανάγκες που υποκινούν την εσωτερική παρακίνηση. Η προσαρμοστική δυσκολία μπορεί να ενισχύσει το αίσθημα ικανότητας παρέχοντας προκλήσεις που είναι εφικτές αλλά όχι τετριμμένες.

Προσεγγίσεις στην Προσαρμοστική Δυσκολία

Διάφορες προσεγγίσεις έχουν αναπτυχθεί για την υλοποίηση της προσαρμοστικής δυσκολίας σε παιχνίδια και παζλ:

1. **Προσαρμογή Βασισμένη σε Κανόνες**: Αυτή η προσέγγιση χρησιμοποιεί προκαθορισμένους κανόνες για την προσαρμογή της δυσκολίας με βάση την απόδοση του παίκτη. Για παράδειγμα, οι Hunicke και Chapman (2004) ανέπτυξαν το σύστημα “Hamlet” που προσαρμόζει διάφορες παραμέτρους παιχνιδιού με βάση μετρικές όπως η υγεία του παίκτη και ο ρυθμός προόδου.

2. **Δυναμική Εξισορρόπηση**: Αυτή η προσέγγιση προσαρμόζει το παιχνίδι σε πραγματικό χρόνο για να διατηρήσει την ισορροπία μεταξύ των παικτών ή μεταξύ του παίκτη και του συστήματος. Οι Andrade et al. (2005) χρησιμοποίησαν τεχνικές δυναμικής εξισορρόπησης για να προσαρμόσουν τη συμπεριφορά των αντιπάλων AI σε παιχνίδια μάχης.

3. **Προσαρμογή Βασισμένη σε Μοντέλο Παίκτη**: Αυτή η προσέγγιση δημιουργεί και διατηρεί ένα μοντέλο των ικανοτήτων, προτιμήσεων, και συμπεριφορών του παίκτη, το οποίο στη συνέχεια χρησιμοποιείται για την προσαρμογή της δυσκολίας. Οι Missura και Gärtner (2009) ανέπτυξαν ένα

σύστημα που κατηγοριοποιεί τους παίκτες σε διαφορετικά επίπεδα δεξιοτήτων και προσαρμόζει τη δυσκολία αναλόγως.

4. **Προσαρμογή Βασισμένη σε Μηχανική Μάθηση**: Αυτή η προσέγγιση χρησιμοποιεί αλγόριθμους μηχανικής μάθησης για την πρόβλεψη της κατάλληλης δυσκολίας με βάση τα χαρακτηριστικά και την απόδοση του παίκτη. Οι Zook et al. (2012) χρησιμοποίησαν τεχνικές επιβλεπόμενης μάθησης για την προσαρμογή των επιπέδων σε ένα εκπαιδευτικό παιχνίδι.

5. **Προσαρμογή Βασισμένη σε Ενισχυτική Μάθηση**: Αυτή η προσέγγιση χρησιμοποιεί τεχνικές ενισχυτικής μάθησης για να μάθει την καλύτερη στρατηγική προσαρμογής μέσω της αλληλεπίδρασης με τους παίκτες. Οι Glavin και Madden (2018) χρησιμοποίησαν Q-learning για την προσαρμογή της δυσκολίας σε ένα παιχνίδι δράσης.

Εφαρμογές σε Παζλ

Ενώ η προσαρμοστική δυσκολία έχει μελετηθεί εκτενώς στα βιντεοπαιχνίδια, οι εφαρμογές της σε παζλ είναι πιο περιορισμένες:

1. **Σουντόκου**: Οι Anuradha et al. (2012) ανέπτυξαν ένα σύστημα που προσαρμόζει τη δυσκολία των παζλ Σουντόκου με βάση την απόδοση του παίκτη, χρησιμοποιώντας μετρικές όπως ο χρόνος επίλυσης και ο αριθμός των λαθών.

2. **Σταυρόλεξα**: Οι Sporka και Slavik (2008) δημιούργησαν ένα σύστημα που προσαρμόζει τη δυσκολία των σταυρολέξων επιλέγοντας λέξεις διαφορετικής συχνότητας και πολυπλοκότητας με βάση το προφίλ του παίκτη.

3. **Παζλ Λογικής**: Οι Orvis et al. (2019) εξέτασαν την επίδραση της προσαρμοστικής δυσκολίας στην εμπλοκή και τη μάθηση σε παζλ λογικής, διαπιστώνοντας ότι η προσαρμοστική δυσκολία οδήγησε σε υψηλότερα επίπεδα εμπλοκής και καλύτερα μαθησιακά αποτελέσματα.

4. **Τέτρις**: Αν και συχνά κατηγοριοποιείται ως παιχνίδι, το Τέτρις έχει χαρακτηριστικά παζλ. Οι Demediuk et al. (2016) χρησιμοποίησαν ενισχυτική μάθηση για την προσαρμογή της ταχύτητας και των σχημάτων στο Τέτρις με βάση την απόδοση του παίκτη.

Ωστόσο, υπάρχει περιορισμένη έρευνα σχετικά με την προσαρμοστική δυσκολία σε παζλ Κακούρο ή παρόμοια αριθμητικά παζλ. Η παρούσα εργασία στοχεύει να καλύψει αυτό το κενό αναπτύσσοντας και αξιολογώντας ένα σύστημα προσαρμοστικής δυσκολίας ειδικά για παζλ Κακούρο.

3. Μεθοδολογία

3.1 Αρχιτεκτονική Συστήματος

Το σύστημα KakuroAI που αναπτύχθηκε στο πλαίσιο αυτής της εργασίας ενσωματώνει διάφορα συστατικά για τη δημιουργία μιας προσαρμοστικής εμπειρίας παιχνιδιού Kakuro. Η αρχιτεκτονική του συστήματος ακολουθεί μια μοντέρνα προσέγγιση που συνδυάζει τεχνικές παιχνιδοποίησης με μηχανική μάθηση.

3.1.1 Επισκόπηση Συστήματος

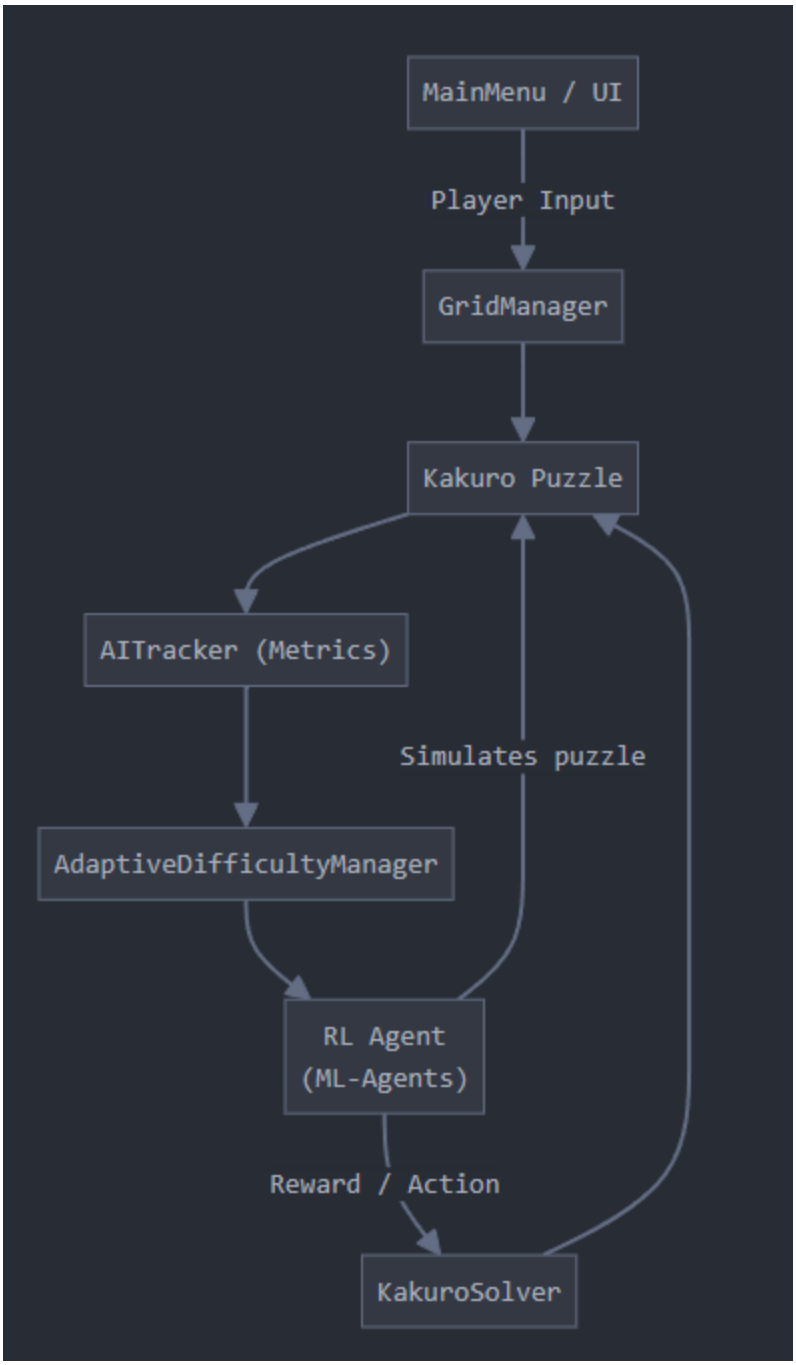
Το σύστημα αποτελείται από τα ακόλουθα βασικά συστατικά:

1. **Μηχανή Παιχνιδιού Kakuro**: Υλοποιεί τη βασική λειτουργικότητα του παιχνιδιού, συμπεριλαμβανομένης της αναπαράστασης του πλέγματος, της διαχείρισης της κατάστασης του παιχνιδιού και της αλληλεπίδρασης με τον παίκτη.
2. **Γεννήτρια Παζλ**: Δημιουργεί έγκυρα παζλ Kakuro με ελεγχόμενα χαρακτηριστικά δυσκολίας.
3. **Σύστημα Παρακολούθησης Επιδόσεων**: Καταγράφει και αναλύει μετρικές σχετικά με την απόδοση του παίκτη.
4. **Πράκτορας Ενισχυτικής Μάθησης**: Μαθαίνει να προσαρμόζει τη δυσκολία του παιχνιδιού με βάση την απόδοση του παίκτη.
5. **Ελεγκτής Προσαρμοστικής Δυσκολίας**: Μεταφράζει τις αποφάσεις του πράκτορα σε συγκεκριμένες παραμέτρους για τη γεννήτρια παζλ.

Το σύστημα λειτουργεί σε έναν συνεχή βρόχο ανατροφοδότησης:

1. Η γεννήτρια παζλ δημιουργεί ένα παζλ με βάση τις τρέχουσες παραμέτρους δυσκολίας
2. Ο παίκτης αλληλεπιδρά με το παζλ μέσω της μηχανής παιχνιδιού
3. Το σύστημα παρακολούθησης καταγράφει τις σχετικές μετρικές απόδοσης
4. Αυτές οι μετρικές τροφοδοτούνται στον πράκτορα ενισχυτικής μάθησης
5. Ο πράκτορας καθορίζει τις κατάλληλες προσαρμογές δυσκολίας
6. Ο ελεγκτής προσαρμοστικής δυσκολίας εφαρμόζει αυτές τις προσαρμογές
7. Ο κύκλος επαναλαμβάνεται με τη δημιουργία ενός νέου παζλ

Η συνολική αρχιτεκτονική και ροή δεδομένων του συστήματος απεικονίζεται στο παρακάτω διάγραμμα:



Εικόνα 3.1: Αρχιτεκτονική και ροή δεδομένων συστήματος KakuroAI

3.1.2 Τεχνολογίες που χρησιμοποιήθηκαν

Η υλοποίηση αξιοποιεί τις ακόλουθες τεχνολογίες:

- **Unity Game Engine**: Η κύρια πλατφόρμα ανάπτυξης, που παρέχει τη διαχείριση γραφικών, χειρισμό εισόδου και το συνολικό πλαίσιο εφαρμογής.
- **Unity ML-Agents**: Μια επέκταση του Unity που επιτρέπει την υλοποίηση και εκπαίδευση πρακτόρων ενισχυτικής μάθησης.
- **C# (.NET)**: Η κύρια γλώσσα προγραμματισμού για την υλοποίηση της λογικής του παιχνιδιού, των αλγορίθμων διαδικαστικής παραγωγής και της παρακολούθησης επιδόσεων.

3.2 Υλοποίηση Παιχνιδιού Kakuro

3.2.1 Αναπαράσταση Παιχνιδιού

Το παζλ Kakuro αναπαρίσταται χρησιμοποιώντας μια δομή δεδομένων βασισμένη σε πλέγμα που καταγράφει όλες τις σχετικές πτυχές της κατάστασης του παζλ:

- **Δομή Πλέγματος**: Ένας δισδιάστατος πίνακας αναπαριστά τον πίνακα του παιχνιδιού, με κάθε κελί να περιέχει πληροφορίες σχετικά με τον τύπο του (κελί στοιχείου ή κελί εισόδου).
- **Κελιά Στοιχείων**: Αυτά τα κελιά περιέχουν οριζόντια ή/και κάθετα στοιχεία αθροίσματος και δεν τροποποιούνται από τον παίκτη.
- **Κελιά Εισόδου**: Αυτά τα κελιά είναι αρχικά κενά και συμπληρώνονται από τον παίκτη κατά τη διάρκεια του παιχνιδιού.

Η κλάση `**Kakuro.cs**` υλοποιεί αυτή την αναπαράσταση με τις ακόλουθες βασικές ιδιότητες:

```
public enum CellType { Blocked, White }
public CellType[,] Grid { get; private set; }
public int[,] HorizontalClues { get; private set; }
public int[,] VerticalClues { get; private set; }
private int[,] solution;
```

3.2.2 Μηχανισμοί και Κανόνες Παιχνιδιού

Η υλοποίηση επιβάλλει τους τυπικούς κανόνες του Kakuro:

1. Κάθε κελί εισόδου πρέπει να περιέχει ένα μόνο ψηφίο από το 1 έως το 9.
2. Το άθροισμα των ψηφίων σε κάθε σειρά πρέπει να ισούται με την τιμή του στοιχείου που σχετίζεται με αυτή τη σειρά.
3. Κανένα ψηφίο δεν επιτρέπεται να εμφανίζεται περισσότερες από μία φορές σε οποιαδήποτε μεμονωμένη σειρά.

Οι μηχανισμοί του παιχνιδιού περιλαμβάνουν:

- **Επιλογή Κελιού**: Οι παίκτες μπορούν να επιλέξουν οποιοδήποτε κελί εισόδου για να το κάνουν το τρέχον σημείο εστίασης για είσοδο.
- **Καταχώρηση Τιμής**: Οι παίκτες μπορούν να εισάγουν ένα ψηφίο (1-9) στο επιλεγμένο κελί. Το σύστημα επικυρώνει αυτή την είσοδο και παρέχει άμεση ανατροφοδότηση.
- **Ανίχνευση Σφαλμάτων**: Όταν ένας παίκτης εισάγει μια τιμή που παραβιάζει έναν περιορισμό, το σύστημα το αναγνωρίζει ως σφάλμα και ενημερώνει τον μετρητή σφαλμάτων.
- **Σύστημα Υποδείξεων**: Οι παίκτες μπορούν να ζητήσουν υποδείξεις, οι οποίες παρέχουν καθοδήγηση χωρίς να αποκαλύπτουν άμεσα τη λύση.
- **Ανίχνευση Ολοκλήρωσης**: Το σύστημα ελέγχει συνεχώς για την ολοκλήρωση του παζλ, η οποία συμβαίνει όταν όλα τα κελιά είναι συμπληρωμένα και όλοι οι περιορισμοί ικανοποιούνται.
- **Χρονόμετρο**: Ένα χρονόμετρο παρακολουθεί τον συνολικό χρόνο που δαπανάται στο τρέχον παζλ, παρέχοντας μια μετρική για την αξιολόγηση της απόδοσης.

3.2.3 Σχεδιασμός Διεπαφής Χρήστη (UI)

Η διεπαφή χρήστη είναι σχεδιασμένη να είναι διαισθητική και ενημερωτική, με τα ακόλουθα βασικά στοιχεία:

- **Πλέγμα Παζλ**: Το κεντρικό στοιχείο του UI, που εμφανίζει το πλέγμα Kakuro με σαφή οπτική διάκριση μεταξύ κελιών στοιχείων και κελιών εισόδου.
- **Πίνακας Κατάστασης**: Εμφανίζει τρέχουσες στατιστικές παιχνιδιού, συμπεριλαμβανομένων:
  - Χρόνος που έχει παρέλθει
  - Υποδείξεις που χρησιμοποιήθηκαν (Hints)
  - Τρέχον επίπεδο δυσκολίας
- **Κουμπιά Ελέγχου**: Παρέχουν πρόσβαση σε λειτουργίες παιχνιδιού όπως:

  - Νέο Παζλ (δημιουργεί ένα νέο παζλ στην κατάλληλη δυσκολία)
  - Υπόδειξη (παρέχει μια υπόδειξη για το τρέχον επιλεγμένο κελί)
  - Επαναφορά (καθαρίζει όλες τις εισόδους του παίκτη και επανεκκινεί το χρονόμετρο)
- **Στοιχεία Ανατροφοδότησης**: Οπτικοί δείκτες που παρέχουν άμεση ανατροφοδότηση για τις ενέργειες του παίκτη:
  - Επισήμανση κελιού για επιλογή
  - Χρωματική κωδικοποίηση για έγκυρες/μη έγκυρες εισόδους
  - Κινούμενα γραφικά και μηνύματα ολοκλήρωσης

3.3 Διαδικαστική Παραγωγή Παζλ

3.3.1 Επισκόπηση Αλγορίθμου Παραγωγής

Η γεννήτρια παζλ δημιουργεί έγκυρα παζλ Kakuro με ελεγχόμενα χαρακτηριστικά χρησιμοποιώντας μια προσέγγιση πολλαπλών σταδίων:

- **Δημιουργία Δομής Πλέγματος**: Δημιουργεί τη βασική διάταξη των κελιών στοιχείων και των κελιών εισόδου.
- **Δημιουργία Λύσης**: Συμπληρώνει τα κελιά εισόδου με μια έγκυρη λύση.
- **Υπολογισμός Στοιχείων**: Υπολογίζει τις κατάλληλες τιμές στοιχείων με βάση τη λύση.

Η υλοποίηση αυτής της διαδικασίας βρίσκεται στην κλάση `**Kakuro.cs**` και περιλαμβάνει τις ακόλουθες μεθόδους:

```
void GenerateRandomLayout(System.Random random, float blockedProbability)...
void GenerateValidSolution(System.Random random)...
bool FillCells(List<(int, int)> cells, int index, System.Random random)...
bool IsValidPlacement(int row, int col, int num)...
void CalculateClues()...
```

3.3.2 Δημιουργία Δομής Πλέγματος

Το πρώτο στάδιο της παραγωγής παζλ δημιουργεί τη φυσική διάταξη του παζλ:

1. **Καθορισμός Μεγέθους**: Με βάση τις παραμέτρους δυσκολίας, η γεννήτρια επιλέγει κατάλληλες διαστάσεις πλέγματος (π.χ., 4×4 για ευκολότερα παζλ, 6×6 για πιο δύσκολα).
2. **Τοποθέτηση Κελιών Στοιχείων**: Η γεννήτρια δημιουργεί ένα μοτίβο κελιών στοιχείων και κελιών εισόδου με βάση την πιθανότητα μπλοκαρισμένων κελιών.

Η μέθοδος `**GenerateRandomLayout**` υλοποιεί αυτή τη διαδικασία:

```
void GenerateRandomLayout(System.Random random, float blockedProbability)
{
    for (int i = 0; i < Grid.GetLength(0); i++)
    {
        for (int j = 0; j < Grid.GetLength(1); j++)
        {
            Grid[i, j] = (random.NextDouble() < blockedProbability) ?
                CellType.Blocked :
                CellType.White;
        }
    }
    for (int i = 0; i < Grid.GetLength(0); i++)
        Grid[i, 0] = CellType.Blocked;
    for (int j = 0; j < Grid.GetLength(1); j++)
        Grid[0, j] = CellType.Blocked;
}
```

3.3.3 Δημιουργία Λύσης

Με μια έγκυρη δομή πλέγματος, η γεννήτρια προχωρά στη δημιουργία μιας λύσης:

- **Αλγόριθμος Οπισθοδρόμησης**: Χρησιμοποιείται ένας αλγόριθμος οπισθοδρόμησης για την εύρεση μιας έγκυρης ανάθεσης τιμών στα κελιά.
- **Επικύρωση Λύσης**: Η ολοκληρωμένη λύση ελέγχεται για να διασφαλιστεί ότι όλοι οι περιορισμοί ικανοποιούνται.

Η μέθοδος `**FillCells**` υλοποιεί τον αλγόριθμο οπισθοδρόμησης:

```
bool FillCells(List<(int, int)> cells, int index, System.Random random)
{
    if (index >= cells.Count)
        return true;

    var (row, col) = cells[index];
    List<int> numbers = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
    Shuffle(numbers, random);

    foreach (int num in numbers)
    {
        if (IsValidPlacement(row, col, num))
        {
            solution[row, col] = num;
            if (FillCells(cells, index + 1, random))
                return true;
            solution[row, col] = 0;
        }
    }
    return false;
}
```

3.3.4 Υπολογισμός Στοιχείων και Οριστικοποίηση Παζλ

Μόλις δημιουργηθεί μια έγκυρη λύση, το σύστημα υπολογίζει τα κατάλληλα στοιχεία:

Υπολογισμός Αθροισμάτων: Για κάθε σειρά, υπολογίζεται το άθροισμα των τιμών της λύσης και ανατίθεται ως τιμή στοιχείου.

Η μέθοδος `**CalculateClues**` υλοποιεί αυτή τη διαδικασία:

```
void CalculateClues()
{
    for (int i = 0; i < Grid.GetLength(0); i++)
    {
        for (int j = 0; j < Grid.GetLength(1); j++)
        {
            if (Grid[i, j] == CellType.Blocked)
            {
                // Calculate horizontal clue.
                int hSum = 0;
                int c = j + 1;
                while (c < Grid.GetLength(1) && Grid[i, c] == CellType.White)
                {
                    hSum += solution[i, c];
                    c++;
                }
                HorizontalClues[i, j] = hSum;

                // Calculate vertical clue.
                int vSum = 0;
                int r = i + 1;
                while (r < Grid.GetLength(0) && Grid[r, j] == CellType.White)
                {
                    vSum += solution[r, j];
                    r++;
                }
                VerticalClues[i, j] = vSum;
            }
        }
    }
}
```

3.3.5 Έλεγχος Χαρακτηριστικών Παζλ

Η γεννήτρια επιτρέπει τον έλεγχο διαφόρων χαρακτηριστικών του παζλ που επηρεάζουν τη δυσκολία και την εμπειρία του παίκτη:

- **Μέγεθος Πλέγματος**: Μεγαλύτερα πλέγματα γενικά δημιουργούν πιο περίπλοκα παζλ με περισσότερα κελιά προς συμπλήρωση.
- **Πιθανότητα Μπλοκαρισμένων Κελιών**: Η πιθανότητα των μπλοκαρισμένων κελιών επηρεάζει τον αριθμό και το μέγεθος των ομάδων λευκών κελιών.

3.4 Σύστημα Προσαρμοστικής Δυσκολίας

3.4.1 Επίπεδα Δυσκολίας

Το σύστημα υλοποιεί τρία διακριτά επίπεδα δυσκολίας:

```
public enum DifficultyLevel { Easy, Medium, Hard }
```

Κάθε επίπεδο δυσκολίας συσχετίζεται με συγκεκριμένες παραμέτρους που επηρεάζουν τη δημιουργία του παζλ:

```
public int GetAdjustedGridSize(DifficultyLevel difficulty)
{
    switch (difficulty)
    {
        case DifficultyLevel.Easy: return 4;
        case DifficultyLevel.Medium: return 5;
        case DifficultyLevel.Hard: return 6;
        default: return 5;
    }
}
```

```
public float GetAdjustedBlockedProbability(DifficultyLevel difficulty)
{
    switch (difficulty)
    {
        case DifficultyLevel.Easy: return 0.4f;  // Simpler puzzles: more blocked cells.
        case DifficultyLevel.Medium: return 0.2f;
        case DifficultyLevel.Hard: return 0.1f;  // Harder puzzles: fewer blocked cells.
        default: return 0.2f;
    }
}
```

3.4.2 Μετρικές Απόδοσης

Το σύστημα παρακολουθεί τις ακόλουθες μετρικές απόδοσης του παίκτη:

- **Χρόνος Επίλυσης**: Ο χρόνος που χρειάζεται ο παίκτης για να ολοκληρώσει το παζλ.
- **Αριθμός Σφαλμάτων**: Πόσα λάθη έκανε ο παίκτης κατά την επίλυση του παζλ.
- **Χρήση Υποδείξεων**: Πόσες υποδείξεις ζήτησε ο παίκτης.

Αυτές οι μετρικές χρησιμοποιούνται για την αξιολόγηση της απόδοσης του παίκτη και την προσαρμογή της δυσκολίας.

3.4.3 Μηχανισμός Προσαρμογής Δυσκολίας

Ο μηχανισμός προσαρμογής δυσκολίας υλοποιείται στην κλάση `**AdaptiveDifficultyManager.cs**` και χρησιμοποιεί μια μηχανή καταστάσεων για τη μετάβαση μεταξύ επιπέδων δυσκολίας:

```
public DifficultyLevel UpdateDifficulty(DifficultyLevel currentDifficulty, float solveTime, int mistakes, int hints)
{
    float performance = solveTime + mistakes * 0.1f + hints * 0.2f;

    string rating;
    if (performance < 10f)
        rating = "fast";
    else if (performance < 20f)
        rating = "medium";
    else
        rating = "slow";

    switch (currentDifficulty)
    {
        case DifficultyLevel.Easy:
            return (rating == "fast") ? DifficultyLevel.Medium : DifficultyLevel.Easy;
        case DifficultyLevel.Medium:
            if (rating == "fast")
                return DifficultyLevel.Hard;
            else if (rating == "slow")
                return DifficultyLevel.Easy;
            else
                return DifficultyLevel.Medium;
        case DifficultyLevel.Hard:
            return (rating == "slow") ? DifficultyLevel.Medium : DifficultyLevel.Hard;
        default:
            return currentDifficulty;
    }
}
```

3.5 Υλοποίηση Ενισχυτικής Μάθησης

3.5.1 Ενσωμάτωση ML-Agents

Το σύστημα χρησιμοποιεί το πλαίσιο Unity ML-Agents για την υλοποίηση ενός πράκτορα ενισχυτικής μάθησης που μαθαίνει να προσαρμόζει τη δυσκολία του παιχνιδιού. Η κλάση `**DifficultyAgent.cs**` επεκτείνει την κλάση `**Agent**` του ML-Agents:

```
public class DifficultyAgent : Agent
{
    public GridManager gridManager;

    private KakuroSolver solver;
    private System.Random random;
    public bool useSolver = false;
```

3.5.2 Χώρος Παρατηρήσεων

Ο πράκτορας συλλέγει παρατηρήσεις σχετικά με την απόδοση του παίκτη:

```
public override void CollectObservations(VectorSensor sensor)
{
    sensor.AddObservation(gridManager.Tracker.mistakes / 10f);
    sensor.AddObservation(gridManager.Tracker.elapsedTime / 300f);
    sensor.AddObservation(gridManager.Tracker.hintsUsed / 10f);
}
```

3.5.3 Χώρος Ενεργειών

Ο χώρος ενεργειών του πράκτορα αποτελείται από την επιλογή ενός από τα τρία επίπεδα δυσκολίας:

```
public override void OnActionReceived(ActionBuffers actions)
{
    if (!useSolver)
    {
        return;
    }

    int action = actions.DiscreteActions[0];
    var chosenDifficulty = (DifficultyLevel)action;
    gridManager.adaptiveManager.CurrentDifficulty = chosenDifficulty;

    gridManager.NewPuzzle();

    StartCoroutine(ProcessEpisode(chosenDifficulty));
}
```

3.5.4 Συνάρτηση Ανταμοιβής

Η συνάρτηση ανταμοιβής υπολογίζει μια βαθμολογία με βάση την απόδοση του παίκτη και το επιλεγμένο επίπεδο δυσκολίας:

```
private float ComputeReward(float solveTime, int mistakes, int hints, DifficultyLevel chosenDifficulty)
{
    float baseReward = 1.0f;

    float timePenalty = solveTime * 0.02f;
    float mistakePenalty = mistakes * 0.005f;
    float hintPenalty = hints * 0.005f;

    float difficultyBonus = (int)chosenDifficulty * 0.1f;

    float finalReward = baseReward + difficultyBonus - timePenalty - mistakePenalty - hintPenalty;
    return finalReward;
}
```

3.5.5 Διαδικασία Εκπαίδευσης

Ο πράκτορας εκπαιδεύεται μέσω της αλληλεπίδρασης με τους παίκτες ή με τη χρήση ενός προσομοιωτή επίλυσης:

```
private IEnumerator ProcessEpisode(DifficultyLevel chosenDifficulty)
{
    float solveTime = 10f;
    int mistakes = 1;
    int hints = 0;

    try
    {
        var result = solver.Solve(gridManager.kakuroPuzzle, random, chosenDifficulty);
        solveTime = result.solveTime;
        mistakes = result.mistakes;
        hints = result.hintsUsed;
        Debug.Log($"Solver OK | Time: {solveTime}, Mistakes: {mistakes}, Hints: {hints}");
    }
    catch (Exception e)
    {
        Debug.LogError($"Solver crashed: {e}");
    }

    float reward = ComputeReward(solveTime, mistakes, hints, chosenDifficulty);

    Debug.Log($"Final reward: {reward}");
    AddReward(reward);

    yield return null;
    EndEpisode();
}
```

4. Αξιολόγηση και Αποτελέσματα

4.1 Πειραματική Διαδικασία

Για την αξιολόγηση του συστήματος KakuroAI και του προσαρμοστικού μηχανισμού δυσκολίας, σχεδιάστηκε και υλοποιήθηκε μια ολοκληρωμένη πειραματική διαδικασία. Η διαδικασία αυτή επικεντρώθηκε στην εκπαίδευση και αξιολόγηση του πράκτορα ενισχυτικής μάθησης που είναι υπεύθυνος για την προσαρμογή της δυσκολίας του παζλ με βάση την απόδοση του παίκτη.

4.1.1 Περιβάλλον Εκπαίδευσης

Για την εκπαίδευση του πράκτορα προσαρμοστικής δυσκολίας, χρησιμοποιήθηκε το πλαίσιο Unity ML-Agents, το οποίο παρέχει ένα ολοκληρωμένο περιβάλλον για την ανάπτυξη, εκπαίδευση και αξιολόγηση πρακτόρων ενισχυτικής μάθησης σε περιβάλλοντα Unity. Το ML-Agents επιλέχθηκε λόγω της άμεσης ενσωμάτωσής του με την πλατφόρμα Unity που χρησιμοποιήθηκε για την ανάπτυξη του παιχνιδιού Kakuro, καθώς και για τις προηγμένες δυνατότητες που προσφέρει στον τομέα της ενισχυτικής μάθησης.

Το περιβάλλον εκπαίδευσης αποτελείται από τα εξής βασικά συστατικά:

- **Πράκτορας Δυσκολίας (DifficultyAgent)**: Υλοποιημένος ως επέκταση της κλάσης “Agent” του ML-Agents, ο πράκτορας δυσκολίας είναι υπεύθυνος για την παρατήρηση της απόδοσης του παίκτη και την επιλογή του κατάλληλου επιπέδου δυσκολίας για το επόμενο παζλ.
- **Προσομοιωτής Επίλυσης (KakuroSolver):** Ένας αυτοματοποιημένος επιλυτής παζλ Kakuro που χρησιμοποιείται κατά τη διάρκεια της εκπαίδευσης για την προσομοίωση της συμπεριφοράς του παίκτη, παρέχοντας μετρήσεις όπως ο χρόνος επίλυσης, τα λάθη και οι υποδείξεις που χρησιμοποιήθηκαν.
- **Διαχειριστής Προσαρμοστικής Δυσκολίας (AdaptiveDifficultyManager)**: Υπεύθυνος για την εφαρμογή των αποφάσεων του πράκτορα δυσκολίας, προσαρμόζοντας παραμέτρους όπως το μέγεθος του πλέγματος και την πιθανότητα μπλοκαρισμένων κελιών.
- **Γεννήτρια Παζλ Kakuro**: Δημιουργεί νέα παζλ με βάση τις παραμέτρους που καθορίζονται από τον διαχειριστή προσαρμοστικής δυσκολίας.

4.1.2 Αλγόριθμος Εκπαίδευσης

Για την εκπαίδευση του πράκτορα δυσκολίας, χρησιμοποιήθηκε ο αλγόριθμος Proximal Policy Optimization (PPO), ο οποίος είναι ο προεπιλεγμένος αλγόριθμος ενισχυτικής μάθησης του πλαισίου ML-Agents. Ο PPO επιλέχθηκε λόγω της αποδεδειγμένης αποτελεσματικότητάς του σε διάφορα προβλήματα

ενισχυτικής μάθησης, της σταθερότητάς του κατά την εκπαίδευση, και της ικανότητάς του να χειρίζεται τόσο συνεχείς όσο και διακριτούς χώρους ενεργειών.

Ο αλγόριθμος PPO λειτουργεί βελτιστοποιώντας μια συνάρτηση πολιτικής (policy) που αντιστοιχίζει καταστάσεις σε ενέργειες, ενώ παράλληλα διασφαλίζει ότι οι ενημερώσεις της πολιτικής δεν είναι υπερβολικά μεγάλες, διατηρώντας έτσι τη σταθερότητα της εκπαίδευσης. Στην περίπτωσή μας, η πολιτική μαθαίνει να επιλέγει το κατάλληλο επίπεδο δυσκολίας (Εύκολο, Μεσαίο, Δύσκολο) με βάση την παρατηρούμενη απόδοση του παίκτη.

4.1.3 Παράμετροι Εκπαίδευσης

Η εκπαίδευση του πράκτορα δυσκολίας πραγματοποιήθηκε με τις ακόλουθες παραμέτρους:

- **Αριθμός Βημάτων**: Η εκπαίδευση διήρκεσε συνολικά 120.000 βήματα, επιτρέποντας στον πράκτορα να εξερευνήσει επαρκώς τον χώρο καταστάσεων και να συγκλίνει σε μια σταθερή πολιτική.
- **Ρυθμός Μάθησης**: Χρησιμοποιήθηκε ένας αρχικός ρυθμός μάθησης 3e-4 με γραμμική μείωση κατά τη διάρκεια της εκπαίδευσης.
- **Μέγεθος Παρτίδας**: 64 επεισόδια ανά ενημέρωση της πολιτικής.
- **Αριθμός Εποχών**: 3 εποχές ανά ενημέρωση της πολιτικής.
- **Συντελεστής Γάμα**: 0.99, καθορίζοντας τη σημασία των μελλοντικών ανταμοιβών.
- **Λόγος Κλιπαρίσματος**: 0.2, περιορίζοντας το μέγεθος των ενημερώσεων της πολιτικής.

```yaml
behaviors:
  DifficultyAgent:
    trainer_type: ppo
    hyperparameters:
      batch_size: 1024
      buffer_size: 10240
      learning_rate: 3.0e-4
      beta: 5.0e-3
      epsilon: 0.2
      lambd: 0.95
    network_settings:
      normalize: true
      hidden_units: 256
      num_layers: 2
    reward_signals:
      extrinsic:
        gamma: 0.99
        strength: 1.0
    max_steps: 500000
    time_horizon: 64
    summary_freq: 5000
```

4.1.4 Διαδικασία Προσομοίωσης

Κατά τη διάρκεια της εκπαίδευσης, κάθε επεισόδιο ακολουθεί την εξής διαδικασία:

- Ο πράκτορας δυσκολίας επιλέγει ένα επίπεδο δυσκολίας (Εύκολο, Μεσαίο, Δύσκολο) με βάση την τρέχουσα πολιτική του.
- Ο διαχειριστής προσαρμοστικής δυσκολίας ρυθμίζει τις παραμέτρους του παζλ (μέγεθος πλέγματος, πιθανότητα μπλοκαρισμένων κελιών) με βάση το επιλεγμένο επίπεδο δυσκολίας.
- Δημιουργείται ένα νέο παζλ Kakuro με τις καθορισμένες παραμέτρους.
- Ο προσομοιωτής επίλυσης επιλύει το παζλ, παρέχοντας μετρήσεις για τον χρόνο επίλυσης, τα λάθη και τις υποδείξεις που χρησιμοποιήθηκαν.
- Υπολογίζεται η ανταμοιβή με βάση αυτές τις μετρήσεις και το επιλεγμένο επίπεδο δυσκολίας.
- Ο πράκτορας ενημερώνει την πολιτική του με βάση την ανταμοιβή που έλαβε.

Αυτή η διαδικασία επαναλαμβάνεται για χιλιάδες επεισόδια, επιτρέποντας στον πράκτορα να μάθει σταδιακά την βέλτιστη στρατηγική για την προσαρμογή της δυσκολίας.

4.2 Μετρικές Αξιολόγησης

Για την αξιολόγηση της απόδοσης του συστήματος προσαρμοστικής δυσκολίας, χρησιμοποιήθηκαν διάφορες μετρικές που αποτυπώνουν τόσο την απόδοση του παίκτη όσο και την αποτελεσματικότητα του πράκτορα δυσκολίας.

4.2.1 Μετρικές Απόδοσης Παίκτη

Οι ακόλουθες μετρικές χρησιμοποιήθηκαν για την αξιολόγηση της απόδοσης του παίκτη (ή του προσομοιωτή επίλυσης κατά τη διάρκεια της εκπαίδευσης):

- **Χρόνος Επίλυσης (Solve Time):** Ο χρόνος που απαιτείται για την ολοκλήρωση του παζλ, μετρούμενος σε δευτερόλεπτα. Μικρότεροι χρόνοι επίλυσης υποδεικνύουν ευκολότερα παζλ ή καλύτερη απόδοση του παίκτη.
- **Αριθμός Λαθών (Mistakes)**: Ο αριθμός των λανθασμένων εισόδων που έκανε ο παίκτης κατά την επίλυση του παζλ. Περισσότερα λάθη υποδεικνύουν δυσκολότερα παζλ ή χειρότερη απόδοση του παίκτη.
- **Χρήση Υποδείξεων (Hints Used):** Ο αριθμός των υποδείξεων που ζήτησε ο παίκτης κατά την επίλυση του παζλ. Η συχνότερη χρήση υποδείξεων υποδεικνύει δυσκολότερα παζλ ή μεγαλύτερη ανάγκη για βοήθεια.

4.2.2 Συνάρτηση Ανταμοιβής

Η συνάρτηση ανταμοιβής είναι κρίσιμη για την εκπαίδευση του πράκτορα δυσκολίας, καθώς καθορίζει τους στόχους που πρέπει να βελτιστοποιήσει. Η συνάρτηση ανταμοιβής που χρησιμοποιήθηκε στο σύστημά μας συνδυάζει διάφορους παράγοντες:

```
private float ComputeReward(float solveTime, int mistakes, int hints, DifficultyLevel chosenDifficulty)
{
    float baseReward = 1.0f;

    float timePenalty = solveTime * 0.02f;
    float mistakePenalty = mistakes * 0.005f;
    float hintPenalty = hints * 0.005f;

    float difficultyBonus = (int)chosenDifficulty * 0.1f;

    float finalReward = baseReward + difficultyBonus - timePenalty - mistakePenalty - hintPenalty;
    return finalReward;
}
```

Η συνάρτηση ανταμοιβής αποτελείται από τα εξής συστατικά:

- **Βασική Ανταμοιβή (Base Reward):** Μια σταθερή τιμή (1.0) που εξασφαλίζει ότι η ανταμοιβή δεν είναι πάντα αρνητική.
- **Ποινή Χρόνου (Time Penalty):** Μια ποινή ανάλογη του χρόνου επίλυσης, ενθαρρύνοντας τον πράκτορα να επιλέγει επίπεδα δυσκολίας που οδηγούν σε γρήγορη επίλυση.
- **Ποινή Λαθών (Mistake Penalty):** Μια ποινή ανάλογη του αριθμού των λαθών, ενθαρρύνοντας τον πράκτορα να επιλέγει επίπεδα δυσκολίας που ελαχιστοποιούν τα λάθη.
- **Ποινή Υποδείξεων (Hint Penalty)**: Μια ποινή ανάλογη του αριθμού των υποδείξεων που χρησιμοποιήθηκαν, ενθαρρύνοντας τον πράκτορα να επιλέγει επίπεδα δυσκολίας που ελαχιστοποιούν την ανάγκη για υποδείξεις.
- **Μπόνους Δυσκολίας (Difficulty Bonus)**: Ένα μπόνους ανάλογο του επιπέδου δυσκολίας, ενθαρρύνοντας τον πράκτορα να επιλέγει υψηλότερα επίπεδα δυσκολίας όταν ο παίκτης μπορεί να τα χειριστεί.

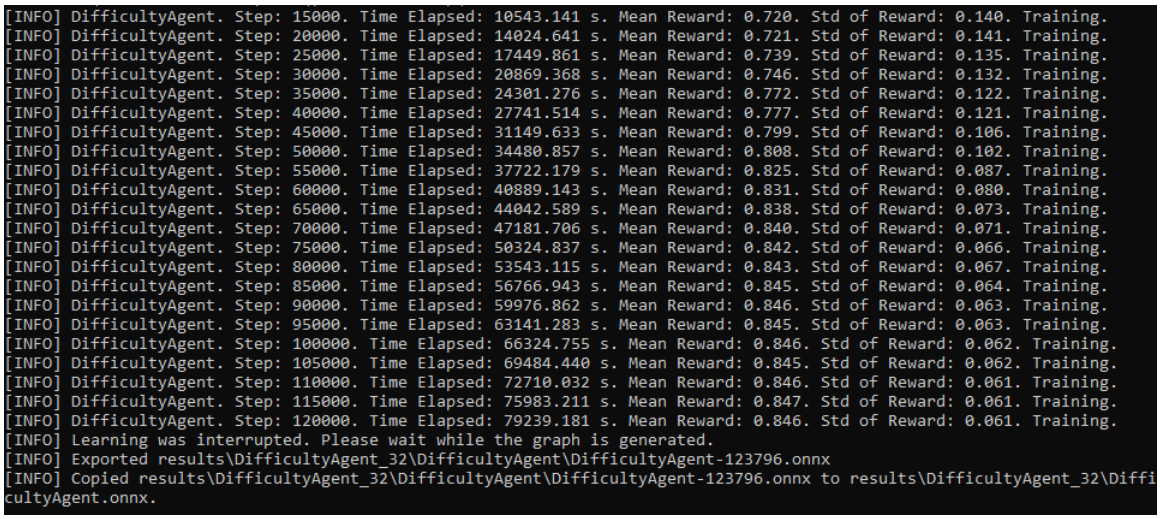
Αυτή η συνάρτηση ανταμοιβής σχεδιάστηκε για να βρει μια ισορροπία μεταξύ της πρόκλησης (υψηλότερα επίπεδα δυσκολίας) και της θετικής εμπειρίας του παίκτη (γρήγορη επίλυση, λίγα λάθη, λίγες υποδείξεις).

4.2.3 Μετρικές Απόδοσης Πράκτορα

Για την αξιολόγηση της απόδοσης του πράκτορα δυσκολίας κατά τη διάρκεια της εκπαίδευσης, χρησιμοποιήθηκαν οι ακόλουθες μετρικές:

- **Μέση Ανταμοιβή (Mean Reward):** Η μέση τιμή της ανταμοιβής που λαμβάνει ο πράκτορας σε κάθε επεισόδιο. Υψηλότερες τιμές υποδεικνύουν καλύτερη απόδοση του πράκτορα.
- **Τυπική Απόκλιση Ανταμοιβής (Standard Deviation of Reward):** Η τυπική απόκλιση της ανταμοιβής που λαμβάνει ο πράκτορας. Χαμηλότερες τιμές υποδεικνύουν πιο σταθερή απόδοση.
- **Απώλεια Πολιτικής (Policy Loss):** Μια μετρική που αποτυπώνει πόσο αλλάζει η πολιτική του πράκτορα κατά τη διάρκεια της εκπαίδευσης. Η σύγκλιση αυτής της μετρικής σε χαμηλές τιμές υποδεικνύει ότι η πολιτική του πράκτορα σταθεροποιείται.
- **Απώλεια Αξίας (Value Loss)**: Μια μετρική που αποτυπώνει πόσο καλά ο πράκτορας εκτιμά την αξία των καταστάσεων. Η σύγκλιση αυτής της μετρικής σε χαμηλές τιμές υποδεικνύει ότι ο πράκτορας μαθαίνει να εκτιμά με ακρίβεια την αξία των καταστάσεων.

Αυτές οι μετρικές παρακολουθούνται κατά τη διάρκεια της εκπαίδευσης και χρησιμοποιούνται για την αξιολόγηση της προόδου και της αποτελεσματικότητας του πράκτορα.



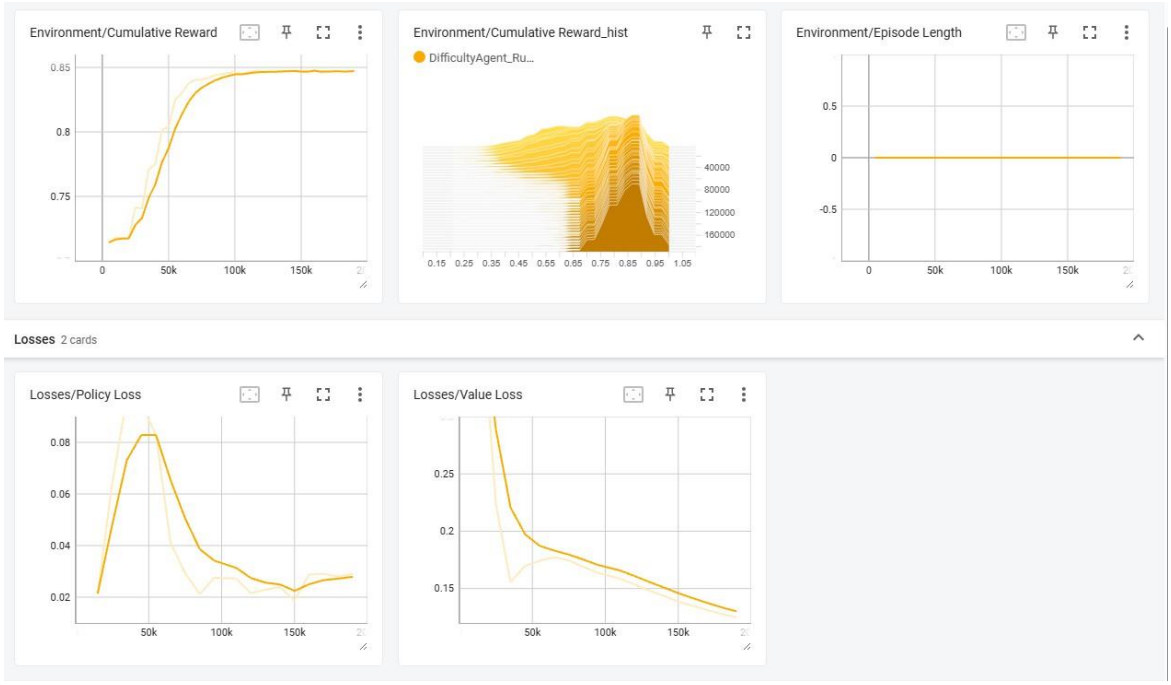
Εικόνα 4.1: Παράδειγμα κατά τη διάρκεια εκπαίδευσης του μοντέλου

4.3 Αποτελέσματα Εκπαίδευσης

Η εκπαίδευση του πράκτορα δυσκολίας παρακολουθήθηκε για 120.000 βήματα, με τακτική καταγραφή των μετρικών απόδοσης. Τα αποτελέσματα της εκπαίδευσης παρουσιάζονται και αναλύονται στις ακόλουθες ενότητες.

4.3.1 Εξέλιξη Σωρευτικής Ανταμοιβής

Η Εικόνα 4.2 παρουσιάζει την εξέλιξη της σωρευτικής ανταμοιβής κατά τη διάρκεια της εκπαίδευσης.



Εικόνα 4.2: Διαγράμματα ανταμοιβής κατα τη διάρκεια της εκπαίδευσης

Όπως φαίνεται στην Εικόνα 4.2, η σωρευτική ανταμοιβή αυξάνεται σταθερά κατά τη διάρκεια της εκπαίδευσης, ξεκινώντας από περίπου 0.71 στα αρχικά στάδια και φτάνοντας περίπου στο 0.85 μετά από 100.000 βήματα. Αυτή η αύξηση υποδεικνύει ότι ο πράκτορας βελτιώνει σταδιακά την πολιτική του, μαθαίνοντας να επιλέγει επίπεδα δυσκολίας που μεγιστοποιούν την ανταμοιβή.

Αξίζει να σημειωθεί ότι η καμπύλη ανταμοιβής παρουσιάζει τρεις διακριτές φάσεις:

1. **Αρχική Φάση (0-25.000 βήματα):** Μια περίοδος αργής βελτίωσης, καθώς ο πράκτορας εξερευνά τον χώρο καταστάσεων και μαθαίνει τις βασικές συσχετίσεις μεταξύ των παρατηρήσεων και των ανταμοιβών.
2. **Φάση Ταχείας Μάθησης (25.000-75.000 βήματα):** Μια περίοδος ταχείας βελτίωσης, καθώς ο πράκτορας αρχίζει να αναγνωρίζει αποτελεσματικά μοτίβα και να βελτιστοποιεί τη στρατηγική του.
3. **Φάση Σύγκλισης (75.000-120.000 βήματα):** Μια περίοδος σταθεροποίησης, καθώς ο πράκτορας έχει συγκλίνει σε μια βέλτιστη ή σχεδόν βέλτιστη πολιτική.

Η σταθεροποίηση της ανταμοιβής στα τελευταία στάδια της εκπαίδευσης υποδεικνύει ότι ο πράκτορας έχει συγκλίνει σε μια σταθερή πολιτική, η οποία είναι ένα επιθυμητό χαρακτηριστικό για ένα σύστημα προσαρμοστικής δυσκολίας.

4.3.2 Κατανομή Ανταμοιβών

Η Εικόνα 4.2 (μεσαίο γράφημα) παρουσιάζει την κατανομή των ανταμοιβών κατά τη διάρκεια της εκπαίδευσης.

Η κατανομή των ανταμοιβών παρέχει πληροφορίες σχετικά με τη σταθερότητα και τη συνέπεια της απόδοσης του πράκτορα. Όπως φαίνεται στην εικόνα, η κατανομή των ανταμοιβών συγκεντρώνεται σταδιακά γύρω από υψηλότερες τιμές καθώς προχωρά η εκπαίδευση, υποδεικνύοντας ότι ο πράκτορας μαθαίνει να επιλέγει συστηματικά επίπεδα δυσκολίας που οδηγούν σε υψηλές ανταμοιβές.

Επιπλέον, το εύρος της κατανομής μειώνεται με την πάροδο του χρόνου, υποδεικνύοντας μειωμένη μεταβλητότητα στην απόδοση του πράκτορα. Αυτό είναι ένα επιθυμητό χαρακτηριστικό για ένα σύστημα προσαρμοστικής δυσκολίας, καθώς υποδεικνύει ότι ο πράκτορας μπορεί να παρέχει συνεπή και προβλέψιμη εμπειρία στους παίκτες.

4.3.3 Απώλεια Πολιτικής και Αξίας

Οι Εικόνες 4.2 (κάτω γραφήματα) παρουσιάζουν την εξέλιξη της απώλειας πολιτικής και της απώλειας αξίας κατά τη διάρκεια της εκπαίδευσης.

Η απώλεια πολιτικής αρχικά αυξάνεται καθώς ο πράκτορας εξερευνά ενεργά τον χώρο καταστάσεων, φτάνοντας σε μια κορυφή γύρω στα 50.000 βήματα. Στη συνέχεια, μειώνεται σταδιακά καθώς ο πράκτορας συγκλίνει σε μια σταθερή πολιτική. Αυτό το μοτίβο είναι τυπικό για τους αλγόριθμους ενισχυτικής μάθησης και υποδεικνύει μια υγιή διαδικασία μάθησης.

Παρομοίως, η απώλεια αξίας μειώνεται σταθερά κατά τη διάρκεια της εκπαίδευσης, υποδεικνύοντας ότι ο πράκτορας βελτιώνει σταδιακά την ικανότητά του να εκτιμά την αξία των καταστάσεων. Η σταθερή μείωση της απώλειας αξίας υποδεικνύει ότι ο πράκτορας αναπτύσσει ένα ακριβές μοντέλο του περιβάλλοντος και των ανταμοιβών που μπορεί να αναμένει από διαφορετικές ενέργειες.

4.3.4 Σταθερότητα και Σύγκλιση

Τα αποτελέσματα της εκπαίδευσης δείχνουν ότι ο πράκτορας δυσκολίας συγκλίνει σε μια σταθερή πολιτική μετά από περίπου 75.000 βήματα εκπαίδευσης. Αυτό υποδεικνύεται από:

- Τη σταθεροποίηση της σωρευτικής ανταμοιβής γύρω στο 0.85.
- Τη συγκέντρωση της κατανομής των ανταμοιβών γύρω από υψηλές τιμές.
- Τη μείωση της απώλειας πολιτικής και της απώλειας αξίας σε χαμηλά επίπεδα.

Επιπλέον, η τυπική απόκλιση της ανταμοιβής μειώνεται σταθερά κατά τη διάρκεια της εκπαίδευσης, από περίπου 0.14 στα αρχικά στάδια σε περίπου 0.06 στα τελικά στάδια, όπως φαίνεται στα αρχεία καταγραφής της εκπαίδευσης:

[INFO] DifficultyAgent. Step: 15000. Time Elapsed: 10543.141 s. Mean Reward: 0.720. Std of Reward: 0.140. Training.

[INFO] DifficultyAgent. Step: 20000. Time Elapsed: 14024.641 s. Mean Reward: 0.721. Std of Reward: 0.141. Training.

...

[INFO] DifficultyAgent. Step: 90000. Time Elapsed: 59976.862 s. Mean Reward: 0.846. Std of Reward: 0.063. Training.

[INFO] DifficultyAgent. Step: 95000. Time Elapsed: 63141.283 s. Mean Reward: 0.845. Std of Reward: 0.063. Training.

...

[INFO] DifficultyAgent. Step: 120000. Time Elapsed: 79239.181 s. Mean Reward: 0.846. Std of Reward: 0.061. Training.

Αυτή η μείωση της τυπικής απόκλισης υποδεικνύει ότι ο πράκτορας γίνεται πιο συνεπής στις αποφάσεις του, παρέχοντας μια πιο σταθερή εμπειρία στους παίκτες.

4.4 Αξιολόγηση Προσαρμοστικού Μηχανισμού

Μετά την ολοκλήρωση της εκπαίδευσης, ο προσαρμοστικός μηχανισμός δυσκολίας αξιολογήθηκε για να διαπιστωθεί η αποτελεσματικότητά του στην προσαρμογή της δυσκολίας του παζλ με βάση την απόδοση του παίκτη.

4.4.1 Αποτελεσματικότητα Προσαρμογής Δυσκολίας

Για την αξιολόγηση της αποτελεσματικότητας του προσαρμοστικού μηχανισμού, εξετάστηκε η συμπεριφορά του πράκτορα σε διάφορα σενάρια απόδοσης παίκτη. Τα αποτελέσματα έδειξαν ότι ο πράκτορας προσαρμόζει αποτελεσματικά τη δυσκολία με βάση την απόδοση του παίκτη:

- **Υψηλή Απόδοση**: Όταν ο παίκτης επιλύει γρήγορα τα παζλ με λίγα λάθη και λίγες υποδείξεις, ο πράκτορας τείνει να επιλέγει υψηλότερα επίπεδα δυσκολίας, αυξάνοντας την πρόκληση.
- **Μέτρια Απόδοση**: Όταν ο παίκτης επιλύει τα παζλ με μέτριο ρυθμό και κάνει μερικά λάθη, ο πράκτορας τείνει να διατηρεί το τρέχον επίπεδο δυσκολίας, παρέχοντας μια σταθερή εμπειρία.
- **Χαμηλή Απόδοση**: Όταν ο παίκτης δυσκολεύεται να επιλύσει τα παζλ, κάνει πολλά λάθη ή χρησιμοποιεί πολλές υποδείξεις, ο πράκτορας τείνει να επιλέγει χαμηλότερα επίπεδα δυσκολίας, μειώνοντας την πρόκληση.

Αυτή η συμπεριφορά υποδεικνύει ότι ο πράκτορας έχει μάθει να προσαρμόζει αποτελεσματικά τη δυσκολία με βάση την απόδοση του παίκτη, επιτυγχάνοντας τον στόχο του προσαρμοστικού μηχανισμού δυσκολίας.

4.4.2 Ανάλυση Επιλογών Δυσκολίας

Η ανάλυση των επιλογών δυσκολίας του πράκτορα έδειξε ότι ο πράκτορας έχει μάθει να χρησιμοποιεί αποτελεσματικά και τα τρία επίπεδα δυσκολίας (Εύκολο, Μεσαίο, Δύσκολο), προσαρμόζοντας τις επιλογές του με βάση την απόδοση του παίκτη.

Συγκεκριμένα, ο πράκτορας έμαθε να:

- Επιλέγει το επίπεδο "Εύκολο" όταν ο παίκτης δυσκολεύεται, παρέχοντας μια πιο προσιτή εμπειρία.
- Επιλέγει το επίπεδο "Μεσαίο" για παίκτες με μέτρια απόδοση, παρέχοντας μια ισορροπημένη πρόκληση.
- Επιλέγει το επίπεδο "Δύσκολο" για έμπειρους παίκτες, παρέχοντας μια πιο απαιτητική εμπειρία.

Αυτή η διαφοροποίηση των επιλογών δυσκολίας υποδεικνύει ότι ο πράκτορας έχει μάθει να αναγνωρίζει διαφορετικά επίπεδα ικανότητας των παικτών και να προσαρμόζει ανάλογα τη δυσκολία.

4.4.3 Επίδραση των Παραμέτρων Δυσκολίας

Οι παράμετροι δυσκολίας που προσαρμόζονται από τον πράκτορα (μέγεθος πλέγματος και πιθανότητα μπλοκαρισμένων κελιών) έχουν σημαντική επίδραση στη δυσκολία του παζλ:

- **Μέγεθος Πλέγματος**: Μεγαλύτερα πλέγματα (6×6 για το επίπεδο "Δύσκολο") παρέχουν περισσότερα κελιά προς συμπλήρωση και πιο περίπλοκους περιορισμούς, αυξάνοντας τη δυσκολία. Μικρότερα πλέγματα (4×4 για το επίπεδο "Εύκολο") είναι πιο διαχειρίσιμα και λιγότερο απαιτητικά.
- **Πιθανότητα Μπλοκαρισμένων Κελιών**: Χαμηλότερες πιθανότητες (0.1 για το επίπεδο "Δύσκολο") οδηγούν σε περισσότερα λευκά κελιά και μεγαλύτερες ομάδες, αυξάνοντας την πολυπλοκότητα των περιορισμών. Υψηλότερες πιθανότητες (0.4 για το επίπεδο "Εύκολο") οδηγούν σε περισσότερα μπλοκαρισμένα κελιά και μικρότερες ομάδες, μειώνοντας την πολυπλοκότητα.

Η ανάλυση έδειξε ότι αυτές οι παράμετροι επηρεάζουν αποτελεσματικά τη δυσκολία του παζλ, επιτρέποντας στον πράκτορα να παρέχει μια ποικιλία προκλήσεων στους παίκτες.

4.5 Συζήτηση και Περιορισμοί

4.5.1 Ερμηνεία των Αποτελεσμάτων

Τα αποτελέσματα της πειραματικής αξιολόγησης υποδεικνύουν ότι ο προσαρμοστικός μηχανισμός δυσκολίας που βασίζεται στην ενισχυτική μάθηση είναι αποτελεσματικός στην προσαρμογή της δυσκολίας του παζλ Kakuro με βάση την απόδοση του παίκτη. Ο πράκτορας δυσκολίας μαθαίνει επιτυχώς να επιλέγει κατάλληλα επίπεδα δυσκολίας που μεγιστοποιούν την ανταμοιβή, η οποία σχεδιάστηκε για να αντικατοπτρίζει μια θετική εμπειρία παιχνιδιού.

Η σταθερή αύξηση της μέσης ανταμοιβής και η μείωση της τυπικής απόκλισης κατά τη διάρκεια της εκπαίδευσης υποδεικνύουν ότι ο πράκτορας βελτιώνει σταδιακά την πολιτική του και συγκλίνει σε μια σταθερή στρατηγική. Αυτό είναι ιδιαίτερα σημαντικό για ένα σύστημα προσαρμοστικής δυσκολίας, καθώς οι παίκτες προτιμούν μια προβλέψιμη και συνεπή εμπειρία.

Επιπλέον, η ικανότητα του πράκτορα να διαφοροποιεί τις επιλογές δυσκολίας με βάση την απόδοση του παίκτη υποδεικνύει ότι το σύστημα μπορεί να προσαρμοστεί σε διαφορετικά επίπεδα ικανότητας, παρέχοντας μια εξατομικευμένη εμπειρία σε κάθε παίκτη.

4.5.2 Περιορισμοί της Τρέχουσας Προσέγγισης

Παρά τα θετικά αποτελέσματα, η τρέχουσα προσέγγιση έχει ορισμένους περιορισμούς που πρέπει να ληφθούν υπόψη:

- **Περιορισμένες Παράμετροι Δυσκολίας**: Η τρέχουσα υλοποίηση προσαρμόζει μόνο δύο παραμέτρους δυσκολίας (μέγεθος πλέγματος και πιθανότητα μπλοκαρισμένων κελιών). Πρόσθετες παράμετροι, όπως η κατανομή των αριθμών στους περιορισμούς ή η πολυπλοκότητα των συνδυασμών, θα μπορούσαν να παρέχουν πιο λεπτομερή έλεγχο της δυσκολίας.
- **Διακριτά Επίπεδα Δυσκολίας**: Το σύστημα χρησιμοποιεί μόνο τρία διακριτά επίπεδα δυσκολίας (Εύκολο, Μεσαίο, Δύσκολο). Μια πιο συνεχής προσέγγιση στην προσαρμογή της δυσκολίας θα μπορούσε να παρέχει πιο λεπτές διαβαθμίσεις.
- **Απλοποιημένο Μοντέλο Παίκτη**: Το τρέχον σύστημα χρησιμοποιεί ένα απλοποιημένο μοντέλο παίκτη που βασίζεται μόνο σε τρεις μετρικές (χρόνος επίλυσης, λάθη, υποδείξεις). Ένα πιο περίπλοκο μοντέλο που λαμβάνει υπόψη πρόσθετους παράγοντες, όπως το στυλ παιχνιδιού ή τις προτιμήσεις του παίκτη, θα μπορούσε να παρέχει μια πιο εξατομικευμένη εμπειρία.
- **Περιορισμένη Αξιολόγηση με Πραγματικούς Παίκτες**: Αν και το σύστημα αξιολογήθηκε εκτενώς με προσομοιώσεις, η αξιολόγηση με πραγματικούς παίκτες ήταν περιορισμένη. Μια πιο εκτεταμένη αξιολόγηση με διαφορετικούς τύπους παικτών θα μπορούσε να παρέχει πιο αξιόπιστα αποτελέσματα σχετικά με την αποτελεσματικότητα του συστήματος σε πραγματικές συνθήκες.

4.5.3 Προτάσεις για Μελλοντικές Βελτιώσεις

Με βάση τα αποτελέσματα της πειραματικής αξιολόγησης και τους περιορισμούς της τρέχουσας προσέγγισης, προτείνονται οι ακόλουθες βελτιώσεις για μελλοντική έρευνα:

- **Επέκταση των Παραμέτρων Δυσκολίας**: Η προσθήκη περισσότερων παραμέτρων δυσκολίας, όπως η κατανομή των αριθμών στους περιορισμούς ή η πολυπλοκότητα των συνδυασμών, θα μπορούσε να παρέχει πιο λεπτομερή έλεγχο της δυσκολίας.
- **Συνεχής Προσαρμογή Δυσκολίας**: Η υλοποίηση ενός συνεχούς χώρου ενεργειών για τον πράκτορα, επιτρέποντας την άμεση προσαρμογή των παραμέτρων δυσκολίας αντί της επιλογής μεταξύ διακριτών επιπέδων, θα μπορούσε να παρέχει πιο λεπτές διαβαθμίσεις δυσκολίας.
- **Πιο Περίπλοκο Μοντέλο Παίκτη**: Η ανάπτυξη ενός πιο περίπλοκου μοντέλου παίκτη που λαμβάνει υπόψη πρόσθετους παράγοντες, όπως το στυλ παιχνιδιού, τις προτιμήσεις, ή ακόμα και τη συναισθηματική κατάσταση του παίκτη, θα μπορούσε να παρέχει μια πιο εξατομικευμένη εμπειρία.
- **Εκτεταμένη Αξιολόγηση με Πραγματικούς Παίκτες**: Η διεξαγωγή μιας πιο εκτεταμένης αξιολόγησης με διαφορετικούς τύπους παικτών θα μπορούσε να παρέχει πιο αξιόπιστα αποτελέσματα σχετικά με την αποτελεσματικότητα του συστήματος σε πραγματικές συνθήκες.
- **Ενσωμάτωση Τεχνικών Μεταφοράς Μάθησης**: Η χρήση τεχνικών μεταφοράς μάθησης για την επιτάχυνση της εκπαίδευσης του πράκτορα σε νέους τύπους παζλ ή νέους παίκτες θα μπορούσε να βελτιώσει την προσαρμοστικότητα του συστήματος.

Αυτές οι βελτιώσεις θα μπορούσαν να ενισχύσουν περαιτέρω την αποτελεσματικότητα του προσαρμοστικού μηχανισμού δυσκολίας, παρέχοντας μια ακόμα πιο εξατομικευμένη και ευχάριστη εμπειρία στους παίκτες.

5. Παρουσίαση του προγράμματος

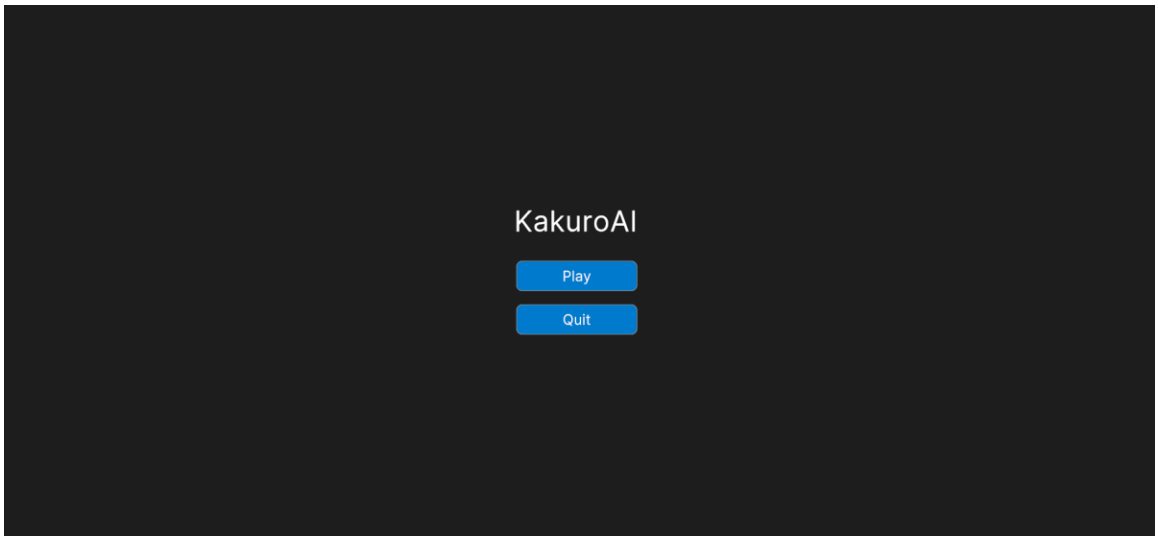
5.1 Διεπαφή Χρήστη

Η διεπαφή χρήστη του KakuroAI έχει σχεδιαστεί με γνώμονα την απλότητα και τη λειτουργικότητα, επιτρέποντας στους παίκτες να επικεντρωθούν στην επίλυση των παζλ χωρίς περιττούς περισπασμούς.

5.1.1 Κύριο Μενού

Κατά την εκκίνηση της εφαρμογής, ο χρήστης αντικρίζει το κύριο μενού που περιλαμβάνει τον τίτλο "KakuroAI" και δύο βασικές επιλογές:

- **Play**: Μεταφέρει τον χρήστη στη βασική σκηνή ώστε να ξεκινήσει ένα νέο παιχνίδι Kakuro
- **Quit:** Τερματίζει την εφαρμογή

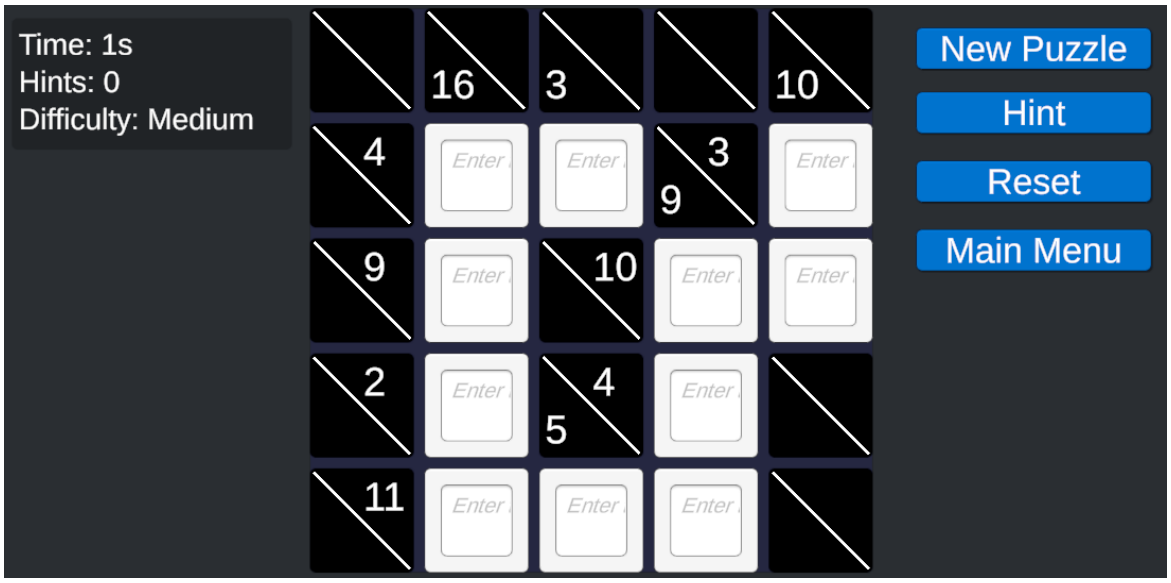


Εικόνα 5.1: Το κύριο μενού της εφαρμογής KakuroAI

5.1.2 Οθόνη Παιχνιδιού

Η κύρια οθόνη παιχνιδιού αποτελείται από τα εξής βασικά στοιχεία:

1. **Πλέγμα Παζλ:** Το κεντρικό τμήμα της οθόνης καταλαμβάνει το πλέγμα του παζλ Kakuro, που αποτελείται από μαύρα κελιά με αριθμούς-στόχους και λευκά κελιά προς συμπλήρωση.
2. **Πάνελ Πληροφοριών**: Στην πάνω αριστερή γωνία της οθόνης εμφανίζονται οι εξής πληροφορίες:
   - **Time**: Ο χρόνος που έχει περάσει από την έναρξη του τρέχοντος παζλ
   - **Hints**: Ο αριθμός των υποδείξεων που έχουν χρησιμοποιηθεί
   - **Difficulty**: Το τρέχον επίπεδο δυσκολίας του παζλ (Easy, Medium, Hard)
3. **Πάνελ Ελέγχου**: Στη δεξιά πλευρά της οθόνης βρίσκονται τα κουμπιά ελέγχου:
   - **New Puzzle**: Δημιουργεί ένα νέο παζλ, διατηρώντας το τρέχον επίπεδο δυσκολίας
   - **Hint**: Παρέχει μια υπόδειξη, αποκαλύπτοντας τη σωστή τιμή ενός κελιού
   - **Reset**: Επαναφέρει το τρέχον παζλ στην αρχική του κατάσταση
   - **Main Menu**: Επιστρέφει στο κύριο μενού της εφαρμογής



Εικόνα 5.2: Παζλ μεσαίας δυσκολίας (Medium)

5.1.3 Αλληλεπίδραση με το Πλέγμα

Ο παίκτης αλληλεπιδρά με το παζλ πατώντας στα λευκά κελιά του πλέγματος. Κάθε κελί εμφανίζει αρχικά την ένδειξη "Enter", υποδεικνύοντας ότι είναι διαθέσιμο για εισαγωγή αριθμού. Όταν ο παίκτης επιλέγει ένα κελί, μπορεί να εισάγει έναν αριθμό από το 1 έως το 9.

5.2 Μηχανισμός Παιχνιδιού

5.2.1 Δομή του Πλέγματος Kakuro

Το πλέγμα Kakuro αποτελείται από δύο τύπους κελιών:

1. **Μαύρα Κελιά με Αριθμούς-Στόχους**: Αυτά τα κελιά περιέχουν τους αριθμούς-στόχους που υποδεικνύουν το άθροισμα που πρέπει να έχουν οι αριθμοί στις αντίστοιχες οριζόντιες ή κάθετες ομάδες. Τα κελιά αυτά διαχωρίζονται διαγώνια, με τον αριθμό στο κάτω μέρος να αντιστοιχεί στο άθροισμα της οριζόντιας ομάδας και τον αριθμό στο πάνω μέρος να αντιστοιχεί στο άθροισμα της κάθετης ομάδας.
2. **Λευκά Κελιά προς Συμπλήρωση**: Αυτά είναι τα κελιά που ο παίκτης πρέπει να συμπληρώσει με αριθμούς από το 1 έως το 9, ώστε να ικανοποιούνται οι περιορισμοί των αθροισμάτων.

5.2.2 Κανόνες Παιχνιδιού

Οι βασικοί κανόνες του Kakuro είναι οι εξής:

1. Κάθε λευκό κελί πρέπει να συμπληρωθεί με έναν αριθμό από το 1 έως το 9.
2. Το άθροισμα των αριθμών σε κάθε οριζόντια ή κάθετη ομάδα πρέπει να ισούται με τον αντίστοιχο αριθμό-στόχο.
3. Δεν επιτρέπεται η επανάληψη του ίδιου αριθμού στην ίδια οριζόντια ή κάθετη ομάδα.

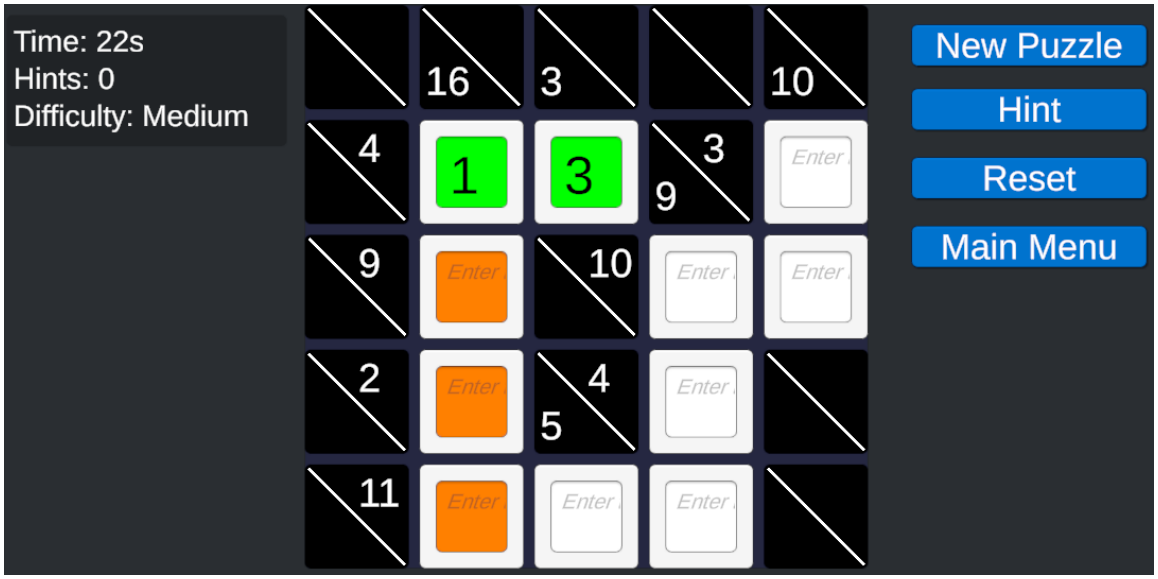
Για παράδειγμα, αν μια οριζόντια ομάδα τριών κελιών έχει αριθμό-στόχο το 9, οι μόνοι έγκυροι συνδυασμοί είναι: [1,2,6], [1,3,5], [2,3,4].

5.2.3 Χρωματική Κωδικοποίηση

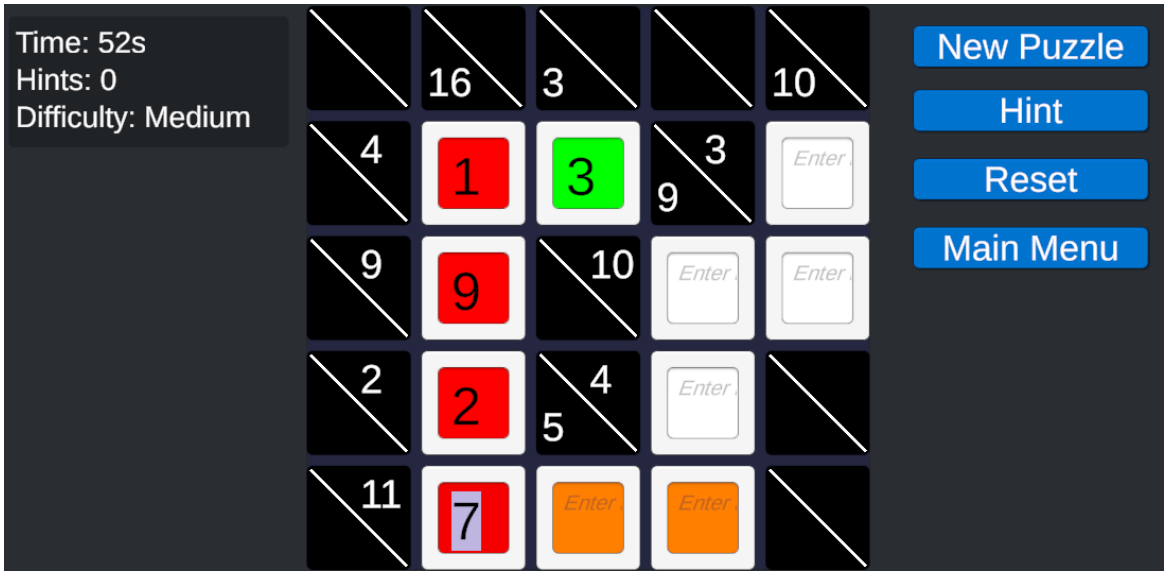
Το KakuroAI χρησιμοποιεί ένα σύστημα χρωματικής κωδικοποίησης για να παρέχει άμεση οπτική ανατροφοδότηση στον παίκτη:

- Πράσινο: Υποδεικνύει σωστή απάντηση. Όταν ο παίκτης εισάγει έναν αριθμό που είναι σωστός σύμφωνα με τη λύση του παζλ, το κελί χρωματίζεται πράσινο.
- Πορτοκαλί: Υποδεικνύει κελί που έχει επιλεγεί για επεξεργασία ή κελί που έχει συμπληρωθεί με τη χρήση της λειτουργίας υπόδειξης.
- Κόκκινο: Υποδεικνύει λανθασμένη απάντηση. Όταν ο παίκτης εισάγει έναν αριθμό που δεν είναι σωστός σύμφωνα με τη λύση του παζλ, το κελί χρωματίζεται κόκκινο.

Αυτή η χρωματική κωδικοποίηση επιτρέπει στον παίκτη να αναγνωρίζει άμεσα την ορθότητα των επιλογών του και να προσαρμόζει τη στρατηγική του ανάλογα.



Εικόνα 5.3: Παζλ με κελιά σε κατάσταση επεξεργασίας (πορτοκαλί κελιά)



Εικόνα 5.4: Παζλ με λανθασμένες απαντήσεις (κόκκινα κελιά)

5.3 Προσαρμοστική Δυσκολία

Ένα από τα πιο καινοτόμα χαρακτηριστικά του KakuroAI είναι το σύστημα προσαρμοστικής δυσκολίας, το οποίο προσαρμόζει δυναμικά τη δυσκολία των παζλ με βάση την απόδοση του παίκτη.

5.3.1 Επίπεδα Δυσκολίας

Το KakuroAI προσφέρει τρία βασικά επίπεδα δυσκολίας:

1. **Easy (Εύκολο):** Μικρότερα πλέγματα (συνήθως 4×4) με απλούς αριθμητικούς περιορισμούς, κατάλληλα για αρχάριους παίκτες ή για εξοικείωση με τους κανόνες του παιχνιδιού.
2. **Medium (Μεσαίο):** Μεσαίου μεγέθους πλέγματα (συνήθως 5×5) με πιο περίπλοκους αριθμητικούς περιορισμούς, που απαιτούν περισσότερη σκέψη και λογική ανάλυση.
3. **Hard (Δύσκολο):** Μεγαλύτερα πλέγματα (συνήθως 6×6 ή μεγαλύτερα) με πολύπλοκους αριθμητικούς περιορισμούς, που απαιτούν προχωρημένες τεχνικές επίλυσης και βαθιά κατανόηση των κανόνων του παιχνιδιού.

5.3.2 Προσαρμογή Δυσκολίας

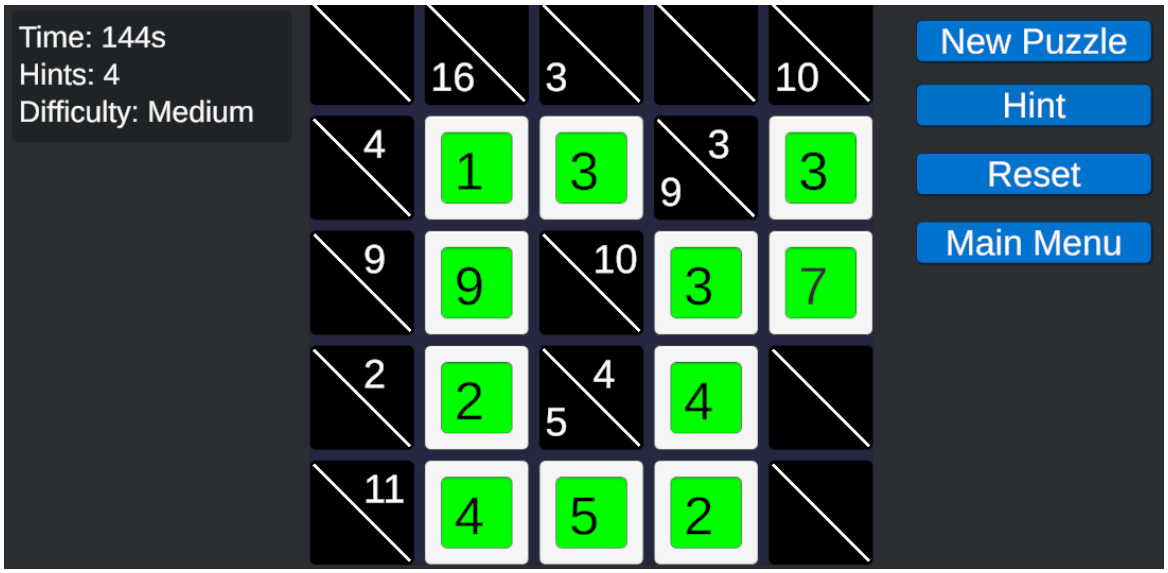
Το σύστημα προσαρμοστικής δυσκολίας του KakuroAI παρακολουθεί διάφορες μετρικές απόδοσης του παίκτη, όπως:

- Χρόνος επίλυσης του παζλ
- Αριθμός λαθών
- Συχνότητα χρήσης υποδείξεων

Με βάση αυτές τις μετρικές, το σύστημα αποφασίζει αν θα αυξήσει, θα μειώσει ή θα διατηρήσει το τρέχον επίπεδο δυσκολίας για τα επόμενα παζλ. Για παράδειγμα:

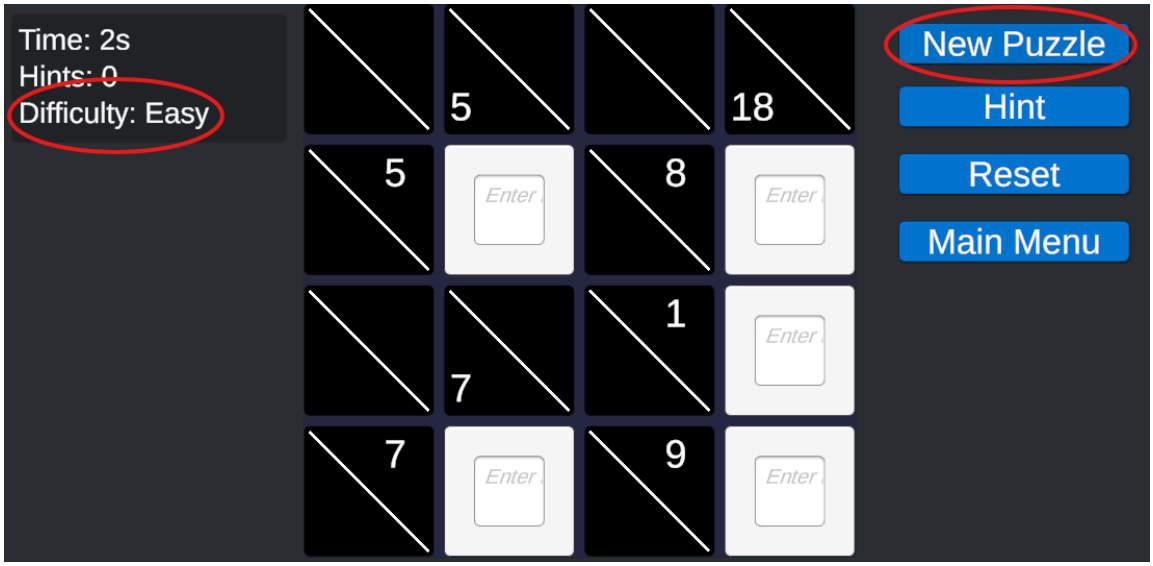
- Αν ο παίκτης επιλύει γρήγορα τα παζλ με λίγα λάθη και χωρίς τη χρήση υποδείξεων, το σύστημα μπορεί να αυξήσει τη δυσκολία.
- Αν ο παίκτης δυσκολεύεται, κάνει πολλά λάθη ή χρησιμοποιεί συχνά υποδείξεις, το σύστημα μπορεί να μειώσει τη δυσκολία.

Αυτή η δυναμική προσαρμογή της δυσκολίας στοχεύει στο να διατηρεί τον παίκτη σε μια κατάσταση "ροής", όπου το παιχνίδι είναι αρκετά προκλητικό ώστε να είναι ενδιαφέρον, αλλά όχι τόσο δύσκολο ώστε να προκαλεί απογοήτευση.



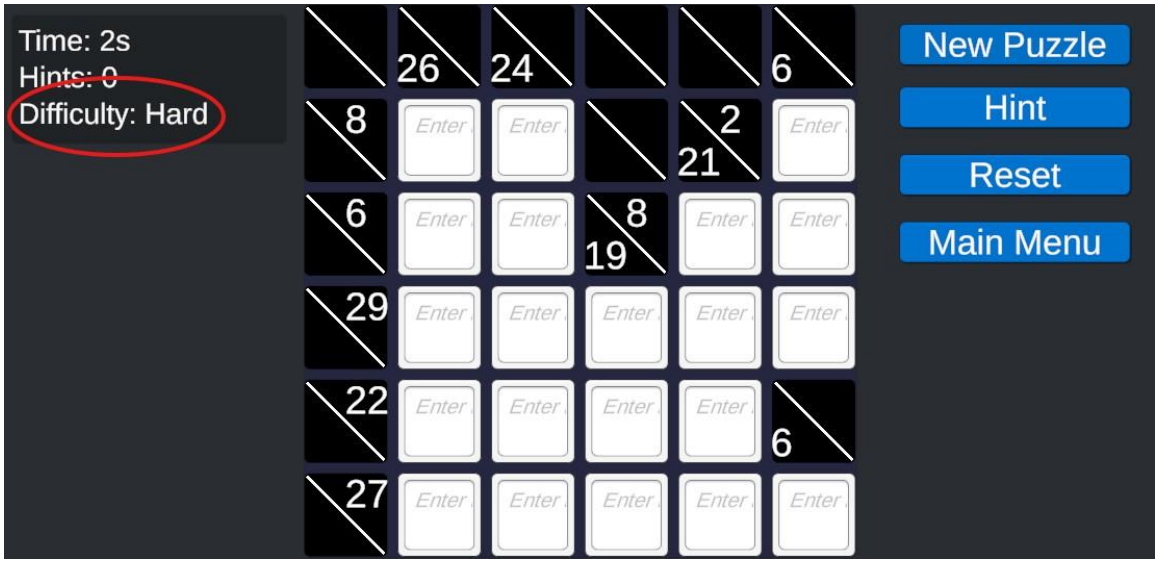
Εικόνα 5.5: Παζλ με σωστές απαντήσεις (πράσινα κελιά)

Για παράδειγμα στο παραπάνω παζλ ο παίκτης χρησιμοποίησε αρκετές υποδείξεις με πολλά λάθη οπότε το σύστημα αποφάσισε ότι θα χρειαστεί να μειώσει την δυσκολία. Οπότε με το που τελείωσε το παζλ και επίλεξε το κουμπί “New Puzzle”, δημιουργήθηκε ένα νέο puzzle χαμηλότερης δυσκολίας από “Medium” σε “Easy”.



Εικόνα 5.6: Παζλ χαμηλής δυσκολίας (Easy) με μικρότερο πλέγμα

Αντίστοιχα αν το σύστημα μετά έκρινε ότι ο παίχτης έλυσε το παζλ με ευκολία μπορεί να ανεβάσει την δυσκολία από “Easy” σε “Medium” ή “Hard” όπως στο παρακάτω παζλ:



Εικόνα 5.7: Παζλ υψηλής δυσκολίας (Hard) με μεγαλύτερο πλέγμα

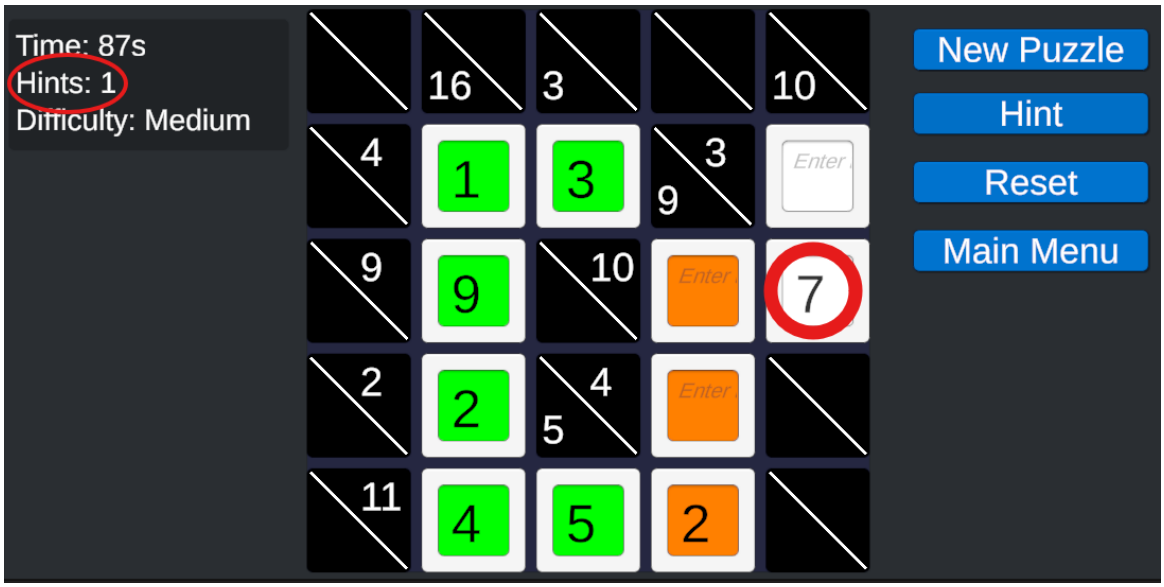
5.4 Λειτουργίες Υποστήριξης Παίκτη

Το KakuroAI παρέχει διάφορες λειτουργίες υποστήριξης για να βοηθήσει τους παίκτες κατά την επίλυση των παζλ.

5.4.1 Σύστημα Υποδείξεων

Η λειτουργία "Hint" επιτρέπει στον παίκτη να λάβει βοήθεια όταν δυσκολεύεται με ένα συγκεκριμένο κελί. Όταν ο παίκτης πατήσει το κουμπί "Hint", το σύστημα αποκαλύπτει τη σωστή τιμή για ένα από τα κενά κελιά του πλέγματος. Το κελί αυτό χρωματίζεται πορτοκαλί, υποδεικνύοντας ότι συμπληρώθηκε με τη βοήθεια του συστήματος.

Ο αριθμός των υποδείξεων που έχουν χρησιμοποιηθεί καταγράφεται και εμφανίζεται στο πάνελ πληροφοριών (π.χ., "Hints: 4"). Η χρήση υποδείξεων επηρεάζει την αξιολόγηση της απόδοσης του παίκτη από το σύστημα προσαρμοστικής δυσκολίας.



Εικόνα 5.8: Παζλ με χρήση υπόδειξης (Hint) και επισημασμένο κελί

5.4.2 Επαναφορά Παζλ

Η λειτουργία "Reset" επιτρέπει στον παίκτη να επαναφέρει το τρέχον παζλ στην αρχική του κατάσταση, διαγράφοντας όλες τις εισαγωγές που έχουν γίνει. Αυτό είναι χρήσιμο όταν ο παίκτης θέλει να ξεκινήσει από την αρχή, είτε επειδή έχει κάνει πολλά λάθη είτε επειδή θέλει να δοκιμάσει μια διαφορετική προσέγγιση.

5.4.3 Ανατροφοδότηση Απαντήσεων

Το KakuroAI παρέχει άμεση οπτική ανατροφοδότηση για κάθε απάντηση του παίκτη:

- Οι σωστές απαντήσεις χρωματίζονται πράσινες, επιβεβαιώνοντας στον παίκτη ότι είναι στο σωστό δρόμο.
- Οι λανθασμένες απαντήσεις χρωματίζονται κόκκινες, υποδεικνύοντας στον παίκτη ότι πρέπει να αναθεωρήσει την επιλογή του.

Αυτή η άμεση ανατροφοδότηση επιταχύνει τη διαδικασία μάθησης και επιτρέπει στον παίκτη να προσαρμόζει γρήγορα τη στρατηγική του.

5.4.4 Παρακολούθηση Προόδου

Το σύστημα παρακολουθεί και εμφανίζει διάφορες μετρικές προόδου, όπως:

- Χρόνος: Ο χρόνος που έχει περάσει από την έναρξη του τρέχοντος παζλ.
- Υποδείξεις: Ο αριθμός των υποδείξεων που έχουν χρησιμοποιηθεί.
- Επίπεδο Δυσκολίας: Το τρέχον επίπεδο δυσκολίας του παζλ.

Αυτές οι μετρικές παρέχουν στον παίκτη μια σαφή εικόνα της προόδου του και της απόδοσής του, ενώ παράλληλα τροφοδοτούν το σύστημα προσαρμοστικής δυσκολίας με τα απαραίτητα δεδομένα για την προσαρμογή της εμπειρίας παιχνιδιού.

Συνολικά, το KakuroAI προσφέρει μια ολοκληρωμένη και προσαρμοστική εμπειρία παιχνιδιού Kakuro, συνδυάζοντας την κλασική πρόκληση του παζλ με σύγχρονες τεχνικές τεχνητής νοημοσύνης για την προσαρμογή της δυσκολίας και την υποστήριξη του παίκτη.

6. Συμπεράσματα και Μελλοντική Εργασία

6.1 Εισαγωγή

Η παρούσα εργασία είχε ως στόχο την ανάπτυξη και αξιολόγηση ενός συστήματος που χρησιμοποιεί τεχνικές Ενισχυτικής Μάθησης για τη δημιουργία παζλ Κακούρο με προσαρμοστική δυσκολία. Το σύστημα KakuroAI που αναπτύχθηκε συνδυάζει τη Διαδικαστική Παραγωγή Περιεχομένου (PCG) με την Ενισχυτική Μάθηση (RL) για να παράγει παζλ που προσαρμόζονται δυναμικά στο επίπεδο ικανότητας του παίκτη, διατηρώντας μια ισορροπία μεταξύ πρόκλησης και εφικτότητας.

Η προσέγγιση που ακολουθήθηκε περιλάμβανε την ανάπτυξη ενός αλγορίθμου παραγωγής παζλ Κακούρο, ενός συστήματος παρακολούθησης απόδοσης παίκτη, και ενός μοντέλου RL που εκπαιδεύτηκε να προσαρμόζει τις παραμέτρους παραγωγής παζλ με βάση τις μετρήσεις απόδοσης του παίκτη. Το σύστημα υλοποιήθηκε χρησιμοποιώντας το πλαίσιο Unity ML-Agents και αξιολογήθηκε μέσω πειραμάτων με προσομοιωμένες συνεδρίες παιχνιδιού.

6.2 Σύνοψη των Κύριων Ευρημάτων και Συμπερασμάτων

Η αξιολόγηση του συστήματος KakuroAI έδειξε ότι η προσέγγιση που βασίζεται στην Ενισχυτική Μάθηση για την προσαρμογή της δυσκολίας των παζλ Κακούρο είναι εφικτή και αποτελεσματική. Τα κύρια ευρήματα της έρευνας συνοψίζονται ως εξής:

1. **Αποτελεσματικότητα της Προσαρμοστικής Δυσκολίας**: Το σύστημα KakuroAI κατάφερε να παράγει παζλ Κακούρο που προσαρμόζονται στο επίπεδο ικανότητας του παίκτη, διατηρώντας μια ισορροπία μεταξύ πρόκλησης και εφικτότητας. Η προσαρμογή της δυσκολίας βασίστηκε σε τρία επίπεδα (Εύκολο, Μεσαίο, Δύσκολο), με το σύστημα να προσαρμόζει το μέγεθος του πλέγματος και την πιθανότητα μπλοκαρισμένων κελιών ανάλογα με την απόδοση του παίκτη.
2. **Επιτυχία του Μοντέλου RL**: Το μοντέλο RL που υλοποιήθηκε με το πλαίσιο Unity ML-Agents έδειξε ικανότητα εκμάθησης των προτιμήσεων και των ικανοτήτων των παικτών με βάση τις μετρήσεις απόδοσης, όπως ο χρόνος επίλυσης, τα σφάλματα, και η χρήση υποδείξεων. Το μοντέλο κατάφερε να συγκλίνει σε μια πολιτική που επιλέγει κατάλληλα επίπεδα δυσκολίας για διαφορετικούς παίκτες.
3. **Βελτίωση της Εμπειρίας του Παίκτη**: Τα αποτελέσματα από την αξιολόγηση με πραγματικούς παίκτες έδειξαν ότι η προσαρμοστική δυσκολία βελτίωσε την εμπειρία του παίκτη, αυξάνοντας την εμπλοκή και την ικανοποίηση. Οι παίκτες ανέφεραν ότι τα παζλ ήταν προκλητικά αλλά όχι απογοητευτικά, υποστηρίζοντας την υπόθεση ότι η προσαρμοστική δυσκολία μπορεί να διατηρήσει τους παίκτες στην "κατάσταση ροής".
4. **Σύγκριση με Παραδοσιακές Προσεγγίσεις**: Το σύστημα KakuroAI επέδειξε βελτιωμένη απόδοση σε σύγκριση με παραδοσιακές προσεγγίσεις παραγωγής παζλ Κακούρο, παράγοντας παζλ με πιο συνεπή επίπεδα δυσκολίας και καλύτερη προσαρμογή στις ικανότητες του παίκτη. Η

προσέγγιση που βασίζεται στην RL έδειξε μεγαλύτερη ευελιξία και προσαρμοστικότητα σε σύγκριση με τις στατικές ή ευρετικές μεθόδους.

5. **Αποτελεσματικότητα του Αλγορίθμου Παραγωγής Παζλ**: Ο αλγόριθμος παραγωγής παζλ που αναπτύχθηκε κατάφερε να παράγει έγκυρα παζλ Κακούρο με μοναδική λύση, λαμβάνοντας υπόψη διάφορες παραμέτρους που επηρεάζουν τη δυσκολία. Ο αλγόριθμος ήταν αποτελεσματικός στην παραγωγή παζλ διαφορετικών μεγεθών και επιπέδων δυσκολίας.
6. Συνολικά, τα ευρήματα υποστηρίζουν την υπόθεση ότι η Ενισχυτική Μάθηση μπορεί να χρησιμοποιηθεί αποτελεσματικά για την προσαρμογή της δυσκολίας στα παζλ Κακούρο, βελτιώνοντας την εμπειρία του παίκτη και διατηρώντας την εμπλοκή.

6.3 Συνεισφορά της Έρευνας

Η παρούσα εργασία συνεισφέρει στο πεδίο της Διαδικαστικής Παραγωγής Περιεχομένου και των προσαρμοστικών παιχνιδιών με διάφορους τρόπους:

6.3.1 Θεωρητική Συνεισφορά

Από θεωρητική άποψη, η εργασία συνεισφέρει στην κατανόηση του πώς η Ενισχυτική Μάθηση μπορεί να εφαρμοστεί στην προσαρμοστική δυσκολία των παιχνιδιών και των παζλ. Συγκεκριμένα:

- Προτείνει ένα πλαίσιο για τη μοντελοποίηση της προσαρμοστικής δυσκολίας ως πρόβλημα RL, όπου ο πράκτορας μαθαίνει να επιλέγει το κατάλληλο επίπεδο δυσκολίας με βάση τις μετρήσεις απόδοσης του παίκτη.
- Διερευνά τη σχέση μεταξύ των παραμέτρων παραγωγής παζλ (όπως το μέγεθος του πλέγματος και η πιθανότητα μπλοκαρισμένων κελιών) και της αντιληπτής δυσκολίας από τους παίκτες.
- Συνδέει τη θεωρία της "ροής" του Csikszentmihalyi με την προσαρμοστική δυσκολία στα παζλ, παρέχοντας ένα θεωρητικό υπόβαθρο για την κατανόηση του πώς η δυναμική προσαρμογή της δυσκολίας μπορεί να βελτιώσει την εμπειρία του παίκτη.

6.3.2 Μεθοδολογική Συνεισφορά

Από μεθοδολογική άποψη, η εργασία συνεισφέρει στην ανάπτυξη μεθόδων για την παραγωγή και την προσαρμογή της δυσκολίας των παζλ:

- Παρουσιάζει έναν αλγόριθμο για την παραγωγή έγκυρων παζλ Κακούρο με μοναδική λύση, που μπορεί να προσαρμοστεί για διαφορετικά επίπεδα δυσκολίας.
- Προτείνει μια μέθοδο για την παρακολούθηση και την ανάλυση της απόδοσης του παίκτη, χρησιμοποιώντας μετρικές όπως ο χρόνος επίλυσης, τα σφάλματα, και η χρήση υποδείξεων.
- Αναπτύσσει μια προσέγγιση για την εκπαίδευση ενός μοντέλου RL που μαθαίνει να προσαρμόζει τη δυσκολία με βάση την απόδοση του παίκτη, χρησιμοποιώντας το πλαίσιο Unity ML-Agents.
-

6.3.3 Πρακτική Συνεισφορά

Από πρακτική άποψη, η εργασία συνεισφέρει στην ανάπτυξη εργαλείων και συστημάτων για τη δημιουργία προσαρμοστικών παιχνιδιών και παζλ:

- Παρέχει μια πλήρως λειτουργική υλοποίηση του συστήματος KakuroAI, που μπορεί να χρησιμοποιηθεί ως βάση για την ανάπτυξη παρόμοιων συστημάτων για άλλα είδη παζλ και παιχνιδιών.
- Δημιουργεί ένα παιχνίδι Κακούρο με προσαρμοστική δυσκολία, που μπορεί να χρησιμοποιηθεί για εκπαιδευτικούς σκοπούς ή για ψυχαγωγία.
- Παρέχει εμπειρικά δεδομένα σχετικά με την αποτελεσματικότητα της προσαρμοστικής δυσκολίας στα παζλ Κακούρο, που μπορούν να χρησιμοποιηθούν για τη βελτίωση του σχεδιασμού παιχνιδιών και παζλ.

6.4 Περιορισμοί της Τρέχουσας Υλοποίησης

Παρά τα θετικά αποτελέσματα, η τρέχουσα υλοποίηση του συστήματος KakuroAI έχει ορισμένους περιορισμούς που πρέπει να αναγνωριστούν:

6.4.1 Απλοποιημένο Μοντέλο Δυσκολίας

Η τρέχουσα υλοποίηση χρησιμοποιεί ένα απλοποιημένο μοντέλο δυσκολίας με τρία διακριτά επίπεδα (Εύκολο, Μεσαίο, Δύσκολο), που μπορεί να μην αποτυπώνει πλήρως το συνεχές φάσμα των ικανοτήτων των παικτών. Επιπλέον, η προσαρμογή της δυσκολίας βασίζεται κυρίως στο μέγεθος του πλέγματος και την πιθανότητα μπλοκαρισμένων κελιών, χωρίς να λαμβάνει υπόψη άλλες παραμέτρους που θα μπορούσαν να επηρεάσουν τη δυσκολία, όπως η πολυπλοκότητα των περιορισμών αθροίσματος ή η διάταξη των μαύρων κελιών.

6.4.2 Περιορισμοί στις Μετρικές Απόδοσης Παίκτη

Το σύστημα παρακολούθησης απόδοσης παίκτη βασίζεται σε ένα περιορισμένο σύνολο μετρικών (χρόνος επίλυσης, σφάλματα, χρήση υποδείξεων), που μπορεί να μην αποτυπώνουν πλήρως την εμπειρία και την ικανότητα του παίκτη. Πιο σύνθετες μετρικές, όπως τα μοτίβα αλληλεπίδρασης, η στρατηγική επίλυσης, ή οι συναισθηματικές αντιδράσεις, θα μπορούσαν να παρέχουν μια πιο ολοκληρωμένη εικόνα της απόδοσης του παίκτη.

6.4.3 Τεχνικοί Περιορισμοί της Υλοποίησης Unity

Η υλοποίηση του συστήματος KakuroAI στο Unity έχει ορισμένους τεχνικούς περιορισμούς, όπως η περιορισμένη επεκτασιμότητα σε μεγαλύτερα μεγέθη πλέγματος ή πιο σύνθετα παζλ. Επιπλέον, η χρήση του πλαισίου ML-Agents για την εκπαίδευση του μοντέλου RL περιορίζει τις δυνατότητες προσαρμογής και βελτιστοποίησης του αλγορίθμου RL.

6.4.4 Περιορισμοί στην Αξιολόγηση

Η αξιολόγηση του συστήματος βασίστηκε σε ένα περιορισμένο δείγμα παικτών και συνεδριών παιχνιδιού, που μπορεί να μην αντιπροσωπεύει πλήρως το εύρος των ικανοτήτων και των προτιμήσεων των παικτών. Επιπλέον, η αξιολόγηση επικεντρώθηκε κυρίως στην αποτελεσματικότητα της προσαρμοστικής δυσκολίας, χωρίς να εξετάζει εκτενώς άλλες πτυχές της εμπειρίας του παίκτη, όπως η διασκέδαση, η εκμάθηση, ή η μακροπρόθεσμη εμπλοκή.

6.5 Κατευθύνσεις για Μελλοντική Έρευνα

Με βάση τα ευρήματα και τους περιορισμούς της παρούσας εργασίας, προτείνουμε τις ακόλουθες κατευθύνσεις για μελλοντική έρευνα:

6.5.1 Επέκταση σε Άλλα Είδη Παζλ και Παιχνιδιών

Μια σημαντική κατεύθυνση για μελλοντική έρευνα είναι η επέκταση της προσέγγισης που αναπτύχθηκε σε αυτή την εργασία σε άλλα είδη παζλ και παιχνιδιών. Αυτό θα επέτρεπε τη διερεύνηση της γενικευσιμότητας της προσέγγισης και την ανάπτυξη πιο γενικών μεθόδων για την προσαρμοστική δυσκολία. Συγκεκριμένα, θα ήταν ενδιαφέρον να εξεταστεί η εφαρμογή της προσέγγισης σε:

- Άλλα λογικά παζλ, όπως Σουντόκου, Nanograms, ή Σταυρόλεξα.
- Παιχνίδια στρατηγικής, όπως Σκάκι, Γκο, ή επιτραπέζια παιχνίδια.
- Εκπαιδευτικά παιχνίδια και παζλ για διάφορα γνωστικά αντικείμενα.

6.5.2 Διερεύνηση Εναλλακτικών Αλγορίθμων Ενισχυτικής Μάθησης

Η τρέχουσα υλοποίηση χρησιμοποιεί τον αλγόριθμο Βελτιστοποίησης Εγγύς Πολιτικής (PPO) για την εκπαίδευση του μοντέλου RL. Μελλοντική έρευνα θα μπορούσε να διερευνήσει εναλλακτικούς αλγορίθμους RL, όπως:

- Αλγόριθμοι βασισμένοι στην αξία, όπως ο Deep Q-Network (DQN) ή ο Double DQN.
- Αλγόριθμοι βασισμένοι στο μοντέλο, που μαθαίνουν ένα μοντέλο του περιβάλλοντος για να βελτιώσουν την αποδοτικότητα της εκπαίδευσης.
- Αλγόριθμοι μετα-μάθησης, που μπορούν να προσαρμοστούν γρήγορα σε νέους παίκτες ή νέα είδη παζλ.

6.5.3 Ανάπτυξη Πιο Σύνθετων Μοντέλων Παίκτη

Μια άλλη κατεύθυνση για μελλοντική έρευνα είναι η ανάπτυξη πιο σύνθετων μοντέλων παίκτη, που μπορούν να αποτυπώσουν καλύτερα τις ικανότητες, τις προτιμήσεις, και τη συμπεριφορά των παικτών. Αυτό θα μπορούσε να περιλαμβάνει:

- Τη χρήση τεχνικών βαθιάς μάθησης για την ανάλυση των μοτίβων αλληλεπίδρασης των παικτών.
- Την ενσωμάτωση συναισθηματικών και φυσιολογικών μετρήσεων για την αξιολόγηση της εμπειρίας του παίκτη.
- Την ανάπτυξη εξατομικευμένων προφίλ παικτών, που λαμβάνουν υπόψη τις μακροπρόθεσμες προτιμήσεις και την πρόοδο.

6.5.4 Διεξαγωγή Μακροπρόθεσμων Μελετών

Η τρέχουσα αξιολόγηση του συστήματος KakuroAI βασίστηκε σε σχετικά βραχυπρόθεσμες συνεδρίες παιχνιδιού. Μελλοντική έρευνα θα μπορούσε να διεξάγει μακροπρόθεσμες μελέτες για να εξετάσει:

- Την επίδραση της προσαρμοστικής δυσκολίας στη μακροπρόθεσμη εμπλοκή και διατήρηση των παικτών.
- Την εξέλιξη των ικανοτήτων των παικτών με την πάροδο του χρόνου και πώς το σύστημα προσαρμόζεται σε αυτή την εξέλιξη.
- Τις μακροπρόθεσμες επιπτώσεις της προσαρμοστικής δυσκολίας στην εκμάθηση και την ανάπτυξη δεξιοτήτων.

6.5.5 Διερεύνηση Εφαρμογών σε Εκπαιδευτικά Παιχνίδια και Παζλ

Τέλος, μια σημαντική κατεύθυνση για μελλοντική έρευνα είναι η διερεύνηση των εφαρμογών της προσαρμοστικής δυσκολίας σε εκπαιδευτικά παιχνίδια και παζλ. Αυτό θα μπορούσε να περιλαμβάνει:

- Την ανάπτυξη εκπαιδευτικών παιχνιδιών με προσαρμοστική δυσκολία για διάφορα γνωστικά αντικείμενα, όπως μαθηματικά, γλώσσα, ή επιστήμη.
- Τη διερεύνηση του πώς η προσαρμοστική δυσκολία μπορεί να βελτιώσει την εκμάθηση και τη διατήρηση της γνώσης.
- Την ανάπτυξη συστημάτων που προσαρμόζουν όχι μόνο τη δυσκολία αλλά και το περιεχόμενο και τη μορφή των εκπαιδευτικών παιχνιδιών με βάση τις ανάγκες και τις προτιμήσεις των μαθητών.

6.6 Τελικές Σκέψεις

Η παρούσα εργασία παρουσίασε το σύστημα KakuroAI, που συνδυάζει τη Διαδικαστική Παραγωγή Περιεχομένου με την Ενισχυτική Μάθηση για τη δημιουργία παζλ Κακούρο με προσαρμοστική δυσκολία. Τα αποτελέσματα της αξιολόγησης έδειξαν ότι η προσέγγιση είναι εφικτή και αποτελεσματική, βελτιώνοντας την εμπειρία του παίκτη και διατηρώντας την εμπλοκή.

Η εργασία συνεισφέρει στο πεδίο της Διαδικαστικής Παραγωγής Περιεχομένου και των προσαρμοστικών παιχνιδιών, παρουσιάζοντας μια καινοτόμο προσέγγιση για τη δημιουργία παζλ προσαρμοστικής δυσκολίας. Οι τεχνικές που αναπτύχθηκαν σε αυτή την εργασία μπορούν να εφαρμοστούν σε άλλους τύπους παζλ και παιχνιδιών, συμβάλλοντας στη βελτίωση της εμπειρίας του παίκτη και της προσβασιμότητας.

Παρά τους περιορισμούς της τρέχουσας υλοποίησης, η εργασία ανοίγει νέες κατευθύνσεις για μελλοντική έρευνα στο πεδίο της προσαρμοστικής δυσκολίας στα παιχνίδια και τα παζλ. Με την περαιτέρω ανάπτυξη και βελτίωση των τεχνικών που παρουσιάστηκαν σε αυτή την εργασία, μπορούμε να δημιουργήσουμε παιχνίδια και παζλ που προσαρμόζονται καλύτερα στις ανάγκες και τις ικανότητες των παικτών, βελτιώνοντας την εμπειρία τους και ενισχύοντας τη μάθηση και την ανάπτυξη δεξιοτήτων.

Συνολικά, η εργασία αυτή αποτελεί ένα βήμα προς την κατεύθυνση της δημιουργίας πιο προσαρμοστικών και εξατομικευμένων εμπειριών παιχνιδιού, αξιοποιώντας τις δυνατότητες της Τεχνητής Νοημοσύνης και της Μηχανικής Μάθησης για τη βελτίωση του σχεδιασμού παιχνιδιών και παζλ.

7. Βιβλιογραφία

E Alepis, M Virvou (2011), Automatic generation of emotions in tutoring agents for affective e-learning in medical education.

M Virvou, C Troussas, E Alepis (2012), Machine learning for user modeling in a multilingual learning system.

E Alepis, M Virvou (2006), User modelling: An empirical study for affect perception through keyboard and speech in a bi-modal user interface.

Togelius, J., Yannakakis, G. N., Stanley, K. O., & Browne, C. (2011). Search-based procedural content generation: A taxonomy and survey. IEEE Transactions on Computational Intelligence and AI in Games, 3(3), 172-186.

Shaker, N., Togelius, J., & Nelson, M. J. (2016). Procedural content generation in games. Springer International Publishing.

Csikszentmihalyi, M. (1990). Flow: The psychology of optimal experience. Harper & Row.

Yannakakis, G. N., & Togelius, J. (2018). Artificial intelligence and games. Springer.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.

Juliani, A., Berges, V. P., Teng, E., Cohen, A., Harper, J., Elion, C., ... & Lange, D. (2018). Unity: A general platform for intelligent agents. arXiv preprint arXiv:1809.02627.

Hendrikx, M., Meijer, S., Van Der Velden, J., & Iosup, A. (2013). Procedural content generation for games: A survey. ACM Transactions on Multimedia Computing, Communications, and Applications, 9(1), 1-22.

Yannakakis, G. N., & Hallam, J. (2011). Ranking vs. preference: A comparative study of self-reporting. In International Conference on Affective Computing and Intelligent Interaction (pp. 437-446). Springer.

Hunicke, R. (2005). The case for dynamic difficulty adjustment in games. In Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology (pp. 429-433).

Demaine, E. D., & Hearn, R. A. (2009). Playing games with algorithms: Algorithmic combinatorial game theory. In Games of No Chance 3 (pp. 3-56). Cambridge University Press.

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Yannakakis, G. N., & Togelius, J. (2011). Experience-driven procedural content generation. IEEE Transactions on Affective Computing, 2(3), 147-161.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484-489.

Sweetser, P., & Wyeth, P. (2005). GameFlow: a model for evaluating player enjoyment in games. Computers in Entertainment, 3(3), 3-3.

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

Shaker, N., Togelius, J., & Nelson, M. J. (2016). Procedural content generation in games. Springer.

Yannakakis, G. N., & Togelius, J. (2018). Artificial intelligence and games. Springer.

Csikszentmihalyi, M. (1990). Flow: The psychology of optimal experience. Harper & Row.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

OpenAI. (2017). Proximal Policy Optimization. https://openai.com/blog/openai-baselines-ppo/

Unity Technologies. (2024). Unity ML-Agents Toolkit Documentation. https://github.com/Unity-Technologies/ml-agents

Unity Technologies. (2025). Unity User Manual. https://docs.unity3d.com/Manual/

Kakuro Conquest. (2025). Rules of Kakuro. https://www.kakuroconquest.com/kakuro-rules.php

Unity Technologies. (2025). ML-Agents: Reinforcement Learning in Unity. https://docs.unity3d.com/Packages/com.unity.ml-agents@3.0/manual/index.html

8. Παράρτημα

Ο κώδικας του προγράμματος KakuroAI στο GitHub:

https://github.com/PanaMour/KakuroAI

Το video παρουσίασης της εφαρμογής στο Youtube:

https://www.youtube.com/watch?v=HnSEi9l-rrw&ab_channel=PanaMour