Εκπαίδευση ενός Πράκτορα Μηχανικής Μάθησης, να παίζει το παιχνίδι Pong, στο προγραμματιστικό περιβάλλον Scratch

Αναστάσιος Λαδιάς[1], Δημήτριος Λαδιάς[2]
ladiastas@gmail.com, ladimitr@gmail.com

[1]Ερευνητής STEM Education, [2]Game developer

Περίληψη. Η εργασία εστιάζει σε μια μέθοδο Τεχνητής Νοημοσύνης, την Επανενισχυόμενη Μάθηση και την εφαρμογή της σε ένα απλό παιχνίδι Pong στο οποίο ο παίχτης αντικαθίσταται από έναν αυτοεκπαιδευόμενο Πράκτορα που προγραμματίζεται σε Scratch. Το project απευθύνεται σε εκπαιδευτικούς πληροφορικής και σκοπός του είναι να γίνει κατανοητό από μικρούς μαθητές το πώς "μαθαίνει" από την εμπειρία του ένα πρόγραμμα Επανενισχυόμενης Μάθησης. Στο μοντέλο που υιοθετήθηκε ο Πράκτορας αλληλεπιδρά με το Περιβάλλον. Ο Πράκτορας αντιλαμβάνεται, αποφασίζει και δρα αποσκοπώντας να πετύχει τη βέλτιστη ανταμοιβή. Ο υπολογισμός της ανταμοιβής γίνεται με την εφαρμογή του αλγορίθμου Q-learning, λαμβάνοντας υπόψη την πολιτική που έχει επιλεγεί. Οι υπολογιζόμενες ανταμοιβές αποτίθενται σε ένα Πίνακα Λήψης Αποφάσεων. Μετά την εκπαίδευσή του όταν ο Πράκτορας κληθεί να παίξει μόνος του το παιχνίδι, συμβουλεύεται τον Πίνακα Λήψης Αποφάσεων και επιλέγει τη δράση με τη μεγαλύτερη αξία. Τα χαρακτηριστικά της εφαρμογής είναι η διαφάνεια του αλγορίθμου, ο κατανοητός κώδικας και η δυνατότητα παρέμβασης σε αυτόν, η αυτονομία του προγραμματιστικού μικρόκοσμου, τα νοηματοδοτημένα έργα για τους μαθητές και η προσέγγιση της εποικοδομητικής μάθησης για αυτούς.

Λέξεις κλειδιά: Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Πράκτορας Επανενισχυόμενης Μάθησης, Scratch, δευτεροβάθμια εκπαίδευση.

1. Εισαγωγή

Τον τελευταίο καιρό δεχόμαστε ένα καταιγισμό από εφαρμογές Τεχνητής Νοημοσύνης (ΤΝ) που προκαλούν το θαυμασμό για όσα επιτυγχάνουν. Η ΤΝ περιλαμβάνει διάφορες προσεγγίσεις, όπως τη Συμβολική ΤΝ / Symbolic AI, τα Έμπειρα Συστήματα, την Υπολογιστική Νοημοσύνη και τη Μηχανική Μάθηση (ΜΜ).

Η ΜΜ εστιάζει στην αυτόματη μάθηση από δεδομένα και στην αυτοβελτίωση του συστήματος. Η ΜΜ και χωρίζεται σε τρεις βασικές κατηγορίες:

- Την Εποπτευόμενη Μάθηση / Supervised learning,
- την Μη Εποπτευόμενη Μάθηση / Unsupervised learning και
- την Επανενισχυόμενη Μάθηση (ΕΜ) / Reinforcement learning.

Στην ΕΜ το πρόγραμμα ενός υπολογιστή αλληλεπιδρά με ένα δυναμικό περιβάλλον όπως για παράδειγμα ένα παιχνίδι, με στόχο να νικήσει. Σε επαναλαμβανόμενες παρτίδες του παιχνιδιού το πρόγραμμα συλλέγει δεδομένα από το περιβάλλον, τα αξιολογεί ως ανταμοιβές/rewards (που είναι ανάλογες με το βαθμό επιτυχίας του) και εφαρμόζοντας κάποιον αλγόριθμο προσπαθεί να τις μεγιστοποιήσει. Ενδεικτικοί αλγόριθμοι που επιτρέπουν σε ένα Πράκτορα -βασιζόμενοι σε δεδομένα της εκπαίδευσής του- να αυτοβελτιωθεί η απόδοσή του αναφέρονται οι Q-Learning, Deep Q-Learning, SARSA, Actor-

Critic, Monte Carlo Tree Search, Dyna-Q, Fitted Q-Iteration, Policy Gradients κ.λπ. (Marugán, 2023). Στην παρούσα εργασία χρησιμοποιείται ο αλγόριθμος Q-learning μέσω του οποίου αποτιμάται η αξία μιας επικείμενης δράσης που θα κάνει το αυτοεκπαιδευόμενο πρόγραμμα όταν βρίσκεται σε μια συγκεκριμένη κατάσταση (Russell & Norvig, 2010).

2. Ο σκοπός της εργασίας

Σε έναν εκπαιδευτικό πληροφορικής μετά από τον ειλικρινή θαυμασμό για τις καταπληκτικές δυνατότητες των εφαρμογών της ΤΝ, συνήθως δημιουργείται η ανάγκη για κατανόηση του πώς δουλεύει η ΤΝ.

Έτσι από αυτό το "πώς δουλεύει;" προκύπτει η επιθυμία του πληροφορικού να κατανοήσει που διαφέρει ο τρόπος που προγραμματίζεται μια ΤΝ από τον κλασικό τρόπο προγραμματισμού με τον οποίο είναι εξοικειωμένος. Ακολουθεί η επιθυμία του εκπαιδευτικού πληροφορικής να μπορέσει να απλοποιήσει αυτή τη γνώση για να τη μεταγγίσει με τον πιο αποτελεσματικό τρόπο σε μαθητές, όσο το δυνατόν μικρότερης ηλικίας (Ng et al., 2023).

Ο στόχος της εργασίας είναι να αποκαλύψει "πώς δουλεύει ένα πρόγραμμα ΜΜ" και πώς μπορεί να παρουσιαστεί αυτό σε μικρούς μαθητές ώστε αυτοί όχι μόνο να γίνουν κριτικοί καταναλωτές της Τ.Ν. αλλά και εν δυνάμει να αναδειχθούν σε παραγωγούς γνώσης σε αυτόν τον τομέα (Λαδιάς, 2024).

Για την εργασία οι προηγούμενες ανάγκες οδήγησαν στις εξής επιλογές:

- Να αναπτυχθεί ένα απλό ηλεκτρονικό παιχνίδι όπως το (πρωτόγονο) Pong, δεδομένου ότι η ενασχόληση με το παιχνίδι είναι ενδογενές κίνητρο για τα παιδιά.
- Να χρησιμοποιηθεί ένα προγραμματιστικό περιβάλλον όπως το Scratch με το οποίο είναι εξοικειωμένοι οι μαθητές και το οποίο μπορεί να ικανοποιήσει τις απαιτήσεις για προγραμματισμό ΜΜ, χωρίς τη χρήση βιβλιοθηκών. Επιπλέον πλεονέκτημα είναι το γεγονός ότι ο Scratch παρέχει τη δυνατότητα η υλοποίηση του παιχνιδιού και ο προγραμματισμός της εκπαίδευσης του αυτοεκπαιδευόμενου προγράμματος να συνυπάρχουν στο ίδιο προγραμματιστικό περιβάλλον.
- Να υιοθετηθεί η προσέγγιση της Επανενισχυόμενης Μάθησης, που δεν απαιτεί να προϋπάρχουν "μεγάλες ποσότητες δεδομένων" (big data), γιατί αυτά παράγονται κατά την εκπαίδευση του Πράκτορα μέσα από πολλές δοκιμές.
- Λαμβάνοντας υπόψη ότι ο μαθητές δεν έχουν υψηλές μαθηματικές γνώσεις, να υιοθετηθεί μια απλοποιημένη μορφή του αλγορίθμου Q-learning, εφαρμόζοντας έναν απλό αριθμητικό τύπο (Ρεπαντής, 2024).

Η μελέτη περίπτωσης της παρούσας εργασίας αφορά τη δημιουργία εκ του μηδενός ενός αυτοεκπαιδευόμενου προγράμματος με σκοπό να μάθει από την εμπειρία του να παίζει Pong. Το πρόγραμμα αυτό σχεδόν τριπλασίασε το ποσοστό επιτυχίας του μετά την εκπαίδευσή του.

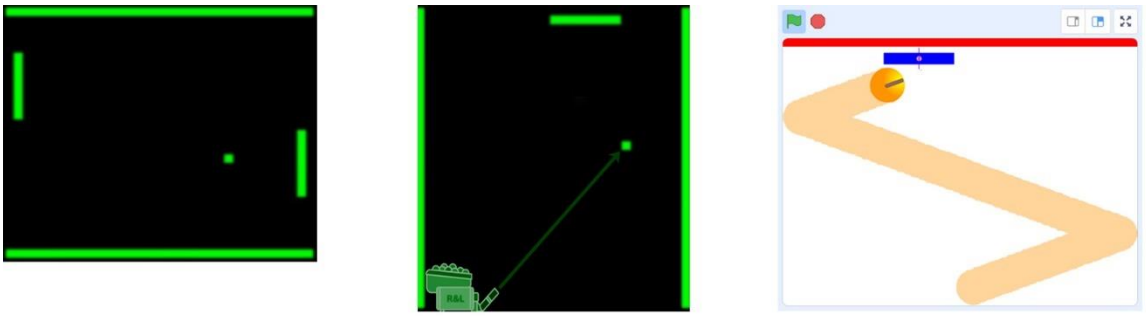
Η προσέγγιση για την παρουσίαση της εργασίας που ακολουθείται είναι "από κάτω προς τα πάνω" (bottom-up). Πρώτα γίνεται η ανάλυση βήμα προς βήμα της υλοποίησης του παραδείγματος της μελέτης περίπτωσης και στη συνέχεια ακολουθεί η σύνθεση με την ένταξη αυτών σε ένα ευρύτερο θεωρητικό πλαίσιο.

3. Το παιχνίδι Pong στο Scratch

Ο προγραμματισμός σε Scratch παρέχει δύο πλεονεκτήματα:

- επειδή το Scratch βασίζεται σε αντικείμενα (object based), βοηθά την ανάθεση των καθηκόντων σε αντικείμενα που παίζουν διακριτούς ρόλους με αντίστοιχη κατανομή του κώδικα.
- επειδή το Scratch είναι ένα προγραμματιστικό περιβάλλον καθοδηγούμενο από συμβάντα (event driven), είναι κατάλληλο για την ανάπτυξη παιχνιδιών.

Το Pong είναι ένα κλασικό βιντεοπαιχνίδι arcade που ως ιδέα βασίστηκε στο πινγκ-πονγκ. Το παιχνίδι παίζεται από δύο παίκτες σε ένα ορθογώνιο τερέν (Σχήμα 1). Στην εργασία έχει αντικατασταθεί ο ένας παίχτης με ένα αυτοεκπαιδευόμενο πρόγραμμα και ο άλλος παίχτης με έναν εκτοξευτή μπαλών. Το απλοποιημένο αυτό Pong προσομοιώνεται στο περιβάλλον του Scratch. Στο Scratch θα υπάρχει μια μπάλα που θα εκτοξεύεται από μια τυχαία θέση στο κάτω μέρος της σκηνής με τυχαία κατεύθυνση προς τα επάνω. Στο επάνω μέρος της σκηνής θα κινείται μια ρακέτα που θα ελέγχεται από το αυτοεκπαιδευόμενο πρόγραμμα που θα την κατευθύνει με στόχο να αποκρουστεί η μπάλα. Μια αντίστοιχη προσέγγιση για το Scratch-like περιβάλλον Snap! εντοπίστηκε στη βιβλιογραφία (Jatzlau, Michaeli, Seegerer, & Romeike, 2019).



Σχήμα 1. Αριστερά το κλασικό τερέν του Pong, στη μέση η χρησιμοποιούμενη διάταξη για την παρούσα εργασία και δεξιά η διάταξη αυτή όπως εμφανίζεται στη σκηνή του Scratch.

4. Το μοντέλο αλληλεπίδρασης του Πράκτορα με το Περιβάλλον του Pong

Στην Επανενισχυόμενη Μάθηση το αυτοεκπαιδευόμενο πρόγραμμα που αλληλεπιδρά με το δυναμικό περιβάλλον του Pong, προσωποποιείται ως Πράκτωρ / Agent (Σχήμα 2).

Στη συγκεκριμένη εργασία με το Pong το περιβάλλον συντίθεται:

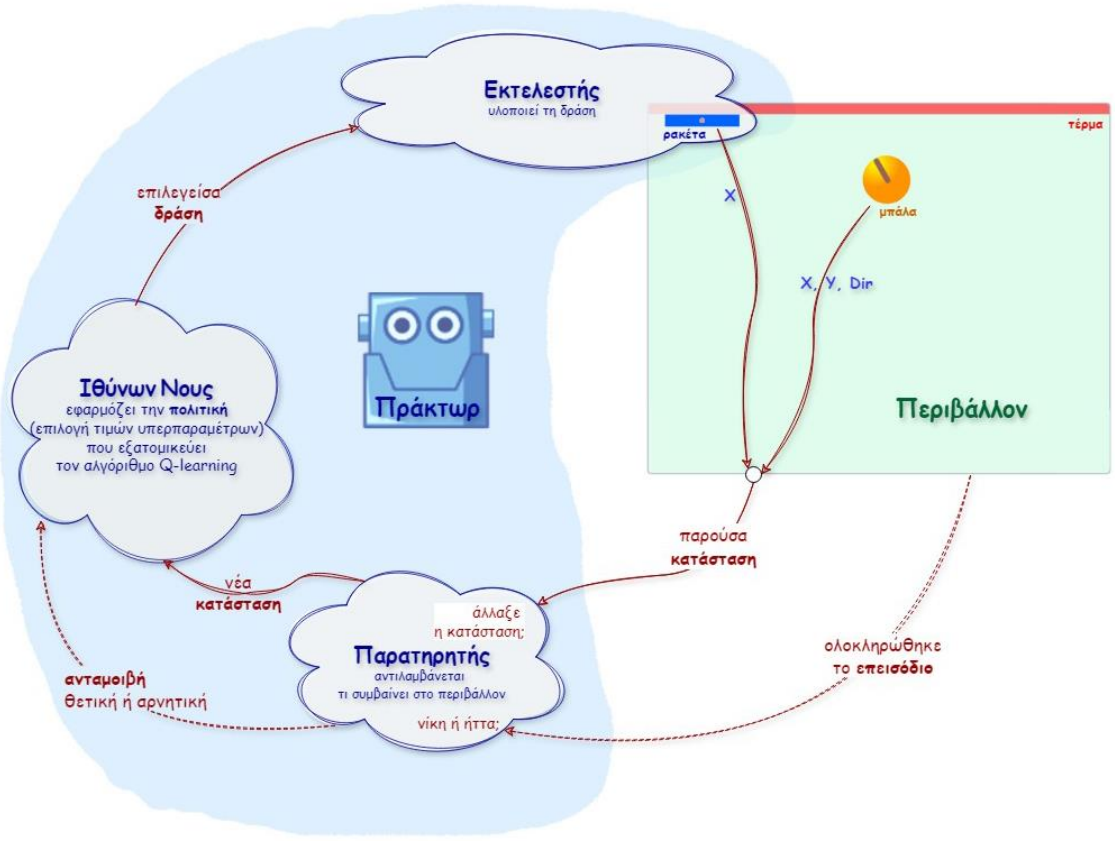
- από το τερέν που είναι η σκηνή του Scratch,
- από τη μπάλα που κινείται αυτόνομα και
- από το τέρμα στο οποίο όταν η μπάλα φτάνει και το αγγίζει, σημαίνει ότι η ρακέτα απέτυχε να την αποκρούσει (Omkar, 2019).

Στο πλαίσιο της ΤΝ ο Πράκτορας έχει την ικανότητα:

- να αντιλαμβάνεται τις αλλαγές που συμβαίνουν στο περιβάλλον του,
- να αποφασίζει για το ποια δράση θα κάνει και
- να δρα και με αυτό τον τρόπο να επηρεάζει το περιβάλλον.

Αυτά τα επιμέρους καθήκοντα αναλαμβάνουν να εξυπηρετήσουν αντίστοιχα τρία τμήματα του πράκτορα:

- ο Παρατηρητής που παρακολουθεί το περιβάλλον χρησιμοποιώντας αισθητήρες / sensors για να συλλέγει δεδομένα από αυτό.
- ο Ιθύνων Νους που ακολουθώντας μια πολιτική που καθορίζεται από τον προγραμματιστή, τροφοδοτεί τον αλγόριθμο Q-learning με τα δεδομένα του Παρατηρητή και αποφασίζει για τη δράση που θα κάνει.
- Ο Εκτελεστής που υλοποιεί με τις δράσεις που του υπαγορεύει ο Ιθύνων Νους, χρησιμοποιώντας ενεργοποιητές / activators. Ένας τέτοιος ενεργοποιητής είναι η ρακέτα που δρα ως ενεργούμενο του Εκτελεστή, η οποία βρίσκεται μέσα στο Περιβάλλον και το επηρεάζει, αλλά οργανικά είναι μέρος του Πράκτορα (Sutton & Barto, 2015).



Σχήμα 2. Το μοντέλο του βρόχου εκπαίδευσης του Πράκτορα στο Περιβάλλον της εργασίας.

Στη συνέχεια αφενός θα αναπαρασταθεί ψηφιακά το Περιβάλλον με τις καταστάσεις του και αφετέρου θα περιγραφούν οι δράσεις του πράκτορα σε συγκεκριμένη κατάσταση.

5. Αναπαράσταση του περιβάλλοντος

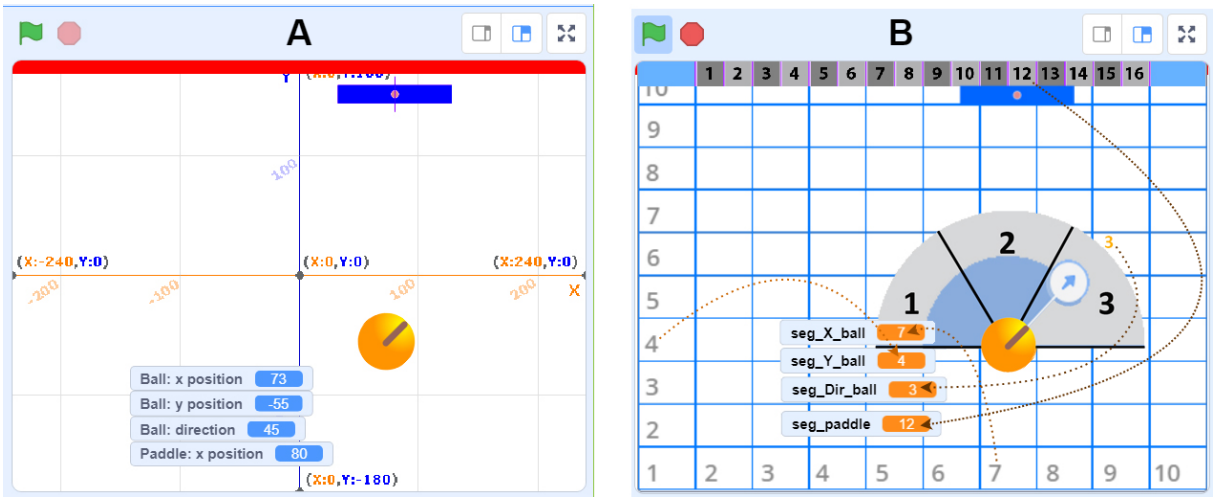
5.1 Περιγραφή μιας κατάστασης του περιβάλλοντος

Στο εξής θα αναφερόμαστε στο Περιβάλλον όπως αυτό οριοθετήθηκε στο Σχήμα 2. Η φωτογραφική αποτύπωση της σκηνής του Scratch αποδίδει ένα στιγμιότυπο.

Για έναν “εξωτερικό παρατηρητή” αυτό μπορεί να περιγραφεί από τη θέση (Ball: x position, Ball: y position) και την κατεύθυνση (Ball: direction) της μπάλας και από την οριζόντια θέση της ρακέτας (Paddle: x position) όπως φαίνονται στο Σχήμα 3Α. Με την παραδοχή ότι αυτά τα αντικείμενα κινούνται με ακέραια βήματα τότε κάθε διαδοχικό στιγμιότυπο (που

προσδιορίζεται από αυτές τις τέσσερες παραμέτρους) αντιστοιχεί σε μια κατάσταση του περιβάλλοντος.

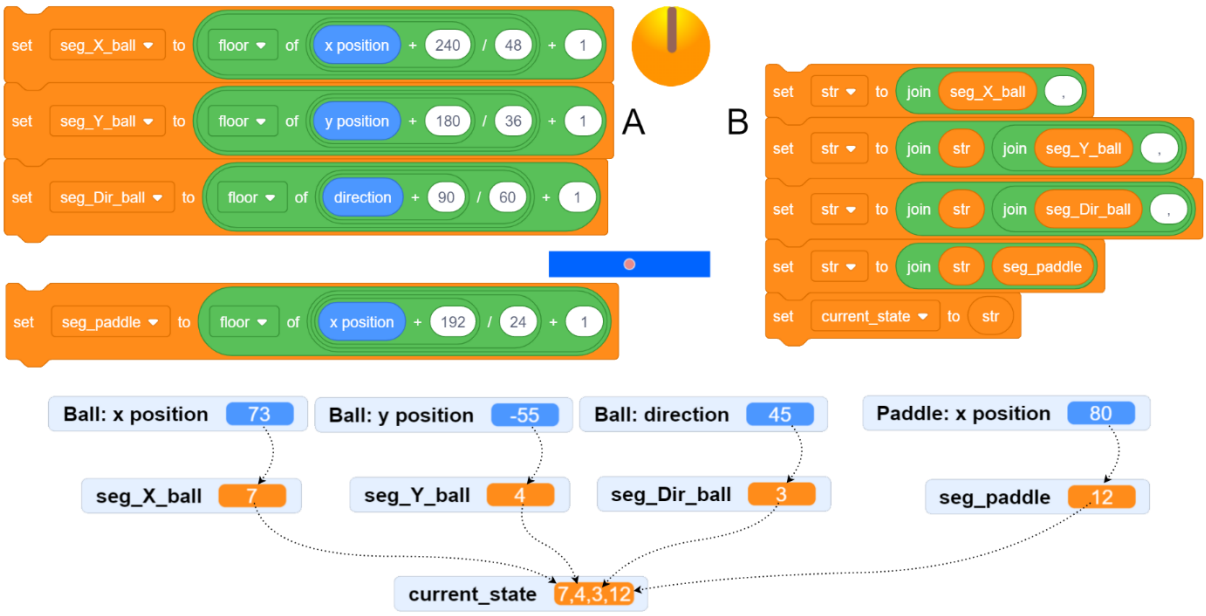
Το πλήθος των καταστάσεων στις οποίες μπορεί να βρεθεί το περιβάλλον προκύπτει από τον πολλαπλασιασμό των τιμών που μπορούν να πάρουν οι τέσσερες παράμετροι: Ball: x position, Ball: y position, Ball: direction και Paddle: x position. Το γινόμενο αντίστοιχα αυτών των τιμών είναι 480x360x180x480 που δίνει αποτέλεσμα 14.929.920.000 καταστάσεις. Οι σχεδόν 15 δισεκατομμύρια καταστάσεις είναι μη διαχειρίσιμο πλήθος καταστάσεων (Λαδιάς, 2024).



Σχήμα 3. Αριστερά οι τιμές των τεσσάρων παραμέτρων-συντεταγμένων της μπάλας και της ρακέτας και δεξιά οι αντίστοιχες τιμές των τομέων.

5.2 Διαχειρίσιμο σύνολο καταστάσεων

Για να γίνει διαχειρίσιμο τεχνικά το συνολικό πλήθος των καταστάσεων πρέπει να περιοριστεί σε μερικές χιλιάδες καταστάσεις. Ένας τρόπος για να επιτευχθεί αυτό είναι να ομαδοποιηθούν περιοχές τιμών των τεσσάρων παραμέτρων σε τομείς (Σχήμα 3Β).



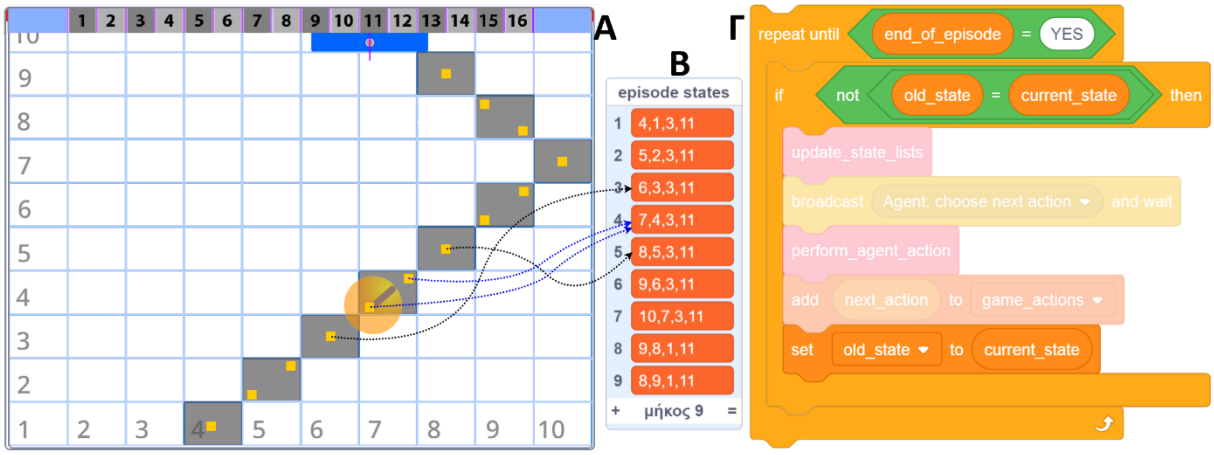
Σχήμα 4. Αριστερά ο κώδικας σε Scratch που μετασχηματίζει τις παραμέτρους της μπάλας και της ρακέτας σε τέσσερες τομείς και δεξιά ο κώδικας που ομαδοποιεί τους τέσσερες τομείς σε μια μεταβλητή-διάνυσμα.

Αποτέλεσμα της “τομεοποίησης” (κβάντισης) είναι να περιοριστεί το πλήθος των καταστάσεων σε 5.400 (= 10 x 10 x 3 x (16+2)) που είναι διαχειρίσιμο. Ο μετασχηματισμός των παραμέτρων σε τομείς γίνεται με τους κώδικες του Σχήματος 4Α που ανήκουν αντίστοιχα στη μπάλα και στη ρακέτα.

Οι τέσσερις τομείς ταυτοποιούν μια κατάσταση του περιβάλλοντος. Για να γίνει ευκολότερη η διαχείριση των τεσσάρων τομέων ομαδοποιούνται σε μια αλφαριθμητική μεταβλητή, την current_state που μέσω αυτής πλέον θα ορίζεται μια κατάσταση. Αυτό γίνεται με τον κώδικα του Σχήματος 4Β που ανήκει στον Παρατηρητή (Σχήμα 2).

5.3 Οι καταστάσεις ενός επεισοδίου

Ως επεισόδιο θεωρείται μια παρτίδα του παιχνιδιού κατά τη διάρκεια της οποίας η μπάλα ξεκινά από το κατώτερο σημείο και ακολουθώντας μια διαδρομή φτάνει στο ανώτερο σημείο που είτε αποκρούεται από τη ρακέτα είτε αγγίζει το τέρμα. Στο Σχήμα 5Α φαίνεται η εξέλιξη ενός επεισοδίου. Η μπάλα κάνει βήματα (τα κίτρινα ίχνη) που αντιστοιχούν σε διαδοχικά στιγμιότυπα, όμως με την “τομεοποίηση” των καταστάσεων παύει να ταυτίζεται το στιγμιότυπο με την κατάσταση. Η ροή με την οποία εξελίσσεται το παιχνίδι δεν βασίζεται σε χρονικά βήματα (όπως θα ήταν αν ακολουθούσε τα κίτρινα ίχνη της μπάλας) αλλά η ροή ακολουθεί τις μεταβάσεις από κατάσταση σε κατάσταση.

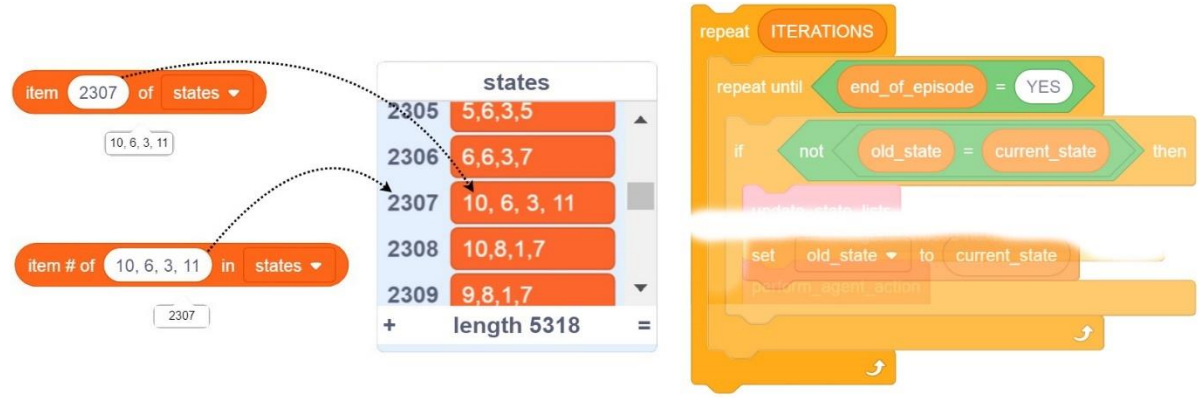


Σχήμα 5. Αναπαράσταση, κατά τη διάρκεια ενός επεισοδίου, των στιγμιοτύπων (κίτρινα ίχνη) και των καταστάσεων (γραμμοσκιασμένα ορθογώνια) που συμβαίνει με την προϋπόθεση ότι ο Πράκτορας επιλέγει να κρατά συνεχώς ακίνητη τη ρακέτα.

Αν για λόγους κατανόησης δεχτούμε ότι ο Πράκτορας δρα κρατώντας τη ρακέτα συνεχώς ακίνητη, οι καταστάσεις του επεισοδίου αντιστοιχούν στα γραμμοσκιασμένα ορθογώνια του Σχήματος 5Α. Παρατηρούμε ότι δύο στιγμιότυπα (κίτρινα ίχνη) μπορούν να αντιστοιχούν σε μια κατάσταση (γραμμοσκιασμένα ορθογώνια). Όλες οι καταστάσεις του επεισοδίου φυλάσσονται στη λίστα episode states (Σχήμα 5Β). Ο διαχωρισμός των καταστάσεων από τα στιγμιότυπα γίνεται σε ένα τμήμα κώδικα που ανήκει στον Ιθύνοντα Νου, το οποίο λειτουργεί ως φίλτρο διαχωρίζοντας τις καταστάσεις από τα διαδοχικά στιγμιότυπα (Σχήμα 5Γ). Το πλήθος των καταστάσεων ενός επεισοδίου εξαρτάται από τη διαδρομή της μπάλας και την “κινητικότητα” της ρακέτας. Έτσι μπορεί να κυμαίνεται από 9-10 καταστάσεις για διαδρομή με ακίνητη ρακέτα, μέχρι μερικές δεκάδες καταστάσεις. Μια μέση τιμή που παρατηρήθηκε σε μερικές εκπαιδεύσεις του Πράκτορα, ήταν περίπου 30 καταστάσεις ανά επεισόδιο.

5.4 Το σύνολο των καταστάσεων για την εκπαίδευση του παιχνιδιού

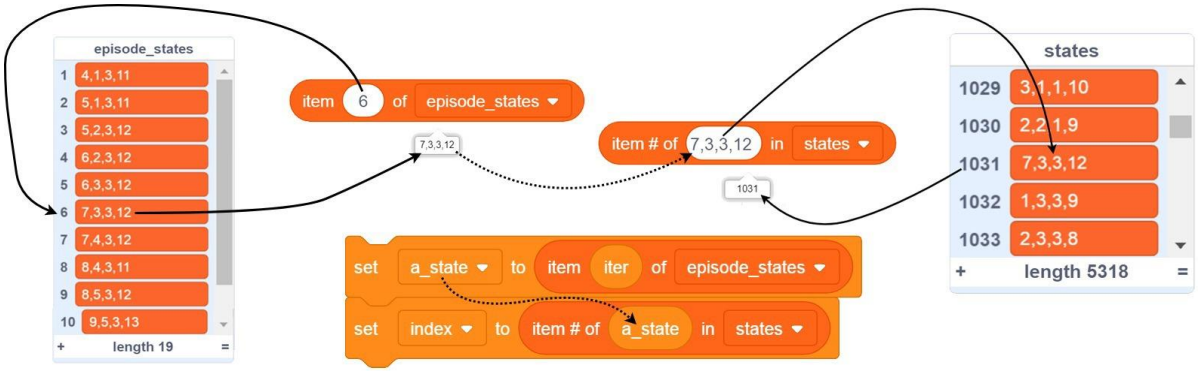
Ένα από τα ζητούμενα της εκπαίδευσης του Πράκτορα είναι να ανακαλύψει κατά την εκπαίδευσή του όσο το δυνατόν περισσότερες από τις προβλεπόμενες καταστάσεις (τις 5.400), γεγονός που απαιτεί να διεξαχθούν "πολλά" επεισόδια. Στην παρούσα εργασία για την εκπαίδευση διαπιστώθηκε ότι αρκεί να επαναληφθούν τα επεισόδια 20.000 φορές. Σε κάθε επεισόδιο η διαδρομή που ακολουθεί η μπάλα και η ρακέτα είναι τυχαία. Έτσι σε κάθε επεισόδιο (τουλάχιστον στην αρχή) ο πράκτορας θα ανακαλύπτει πολλές νέες καταστάσεις. Οι καταστάσεις που θα εμφανιστούν κατά τη διάρκεια της εκπαίδευσης υπολογίζονται σε περίπου 600.000 (=20.000 επεισόδια x 30 καταστάσεις ανά επεισόδιο). Υπό αυτές τις συνθήκες παρατηρήθηκε ότι σε διάφορες εκπαιδεύσεις ο Πράκτορας ανακαλύπτει κατά μέσο όρο περίπου 5.300 διαφορετικές καταστάσεις από τις 5.400 που έχουν οριστεί. Όλες αυτές οι καταστάσεις που θα ανακαλύψει ο Πράκτορας φυλάσσονται σε μια λίστα, τη states, η οποία θα λειτουργεί ως σύνολο, έχοντας καταγράψει ως (μοναδικά) στοιχεία του συνόλου την κάθε κατάσταση (Σχήμα 6). Η πρόσβαση σε στοιχείο του συνόλου μπορεί να γίνει είτε με την αναζήτηση του στοιχείου είτε με την αναζήτηση της θέσης του στοιχείου.



Σχήμα 6. Ό Πράκτορας εκπαιδεύεται παίζοντας 20.000 (η τιμή της σταθεράς ITERATION) παρτίδες-επεισόδια. Στο παράδειγμα του σχήματος μετά την εκπαίδευση ο Πράκτορας ανακάλυψε 5.318 διαφορετικές καταστάσεις (είναι το μήκος της λίστας states).

5.5 Η ταυτότητα μιας κατάστασης

Κατά τη διεξαγωγή ενός επεισοδίου ανακαλύπτονται διάφορες καταστάσεις που καταγράφονται με τη σειρά που εμφανίζονται στη λίστα episode_states.



Σχήμα 7. Ο τρόπος με τον οποίο μια κατάσταση που συναντιέται κατά τη διεξαγωγή ενός επεισοδίου βρίσκει τη θέση της στη λίστα states και αποκτά ταυτότητα.

Με την εμφάνιση κάθε διάδοχης κατάστασης ελέγχεται αν αυτή προϋπάρχει και αν όχι τότε προστίθεται στο τέλος λίστας states. Η θέση της κατάστασης στη λίστα states δηλ. ο δείκτης

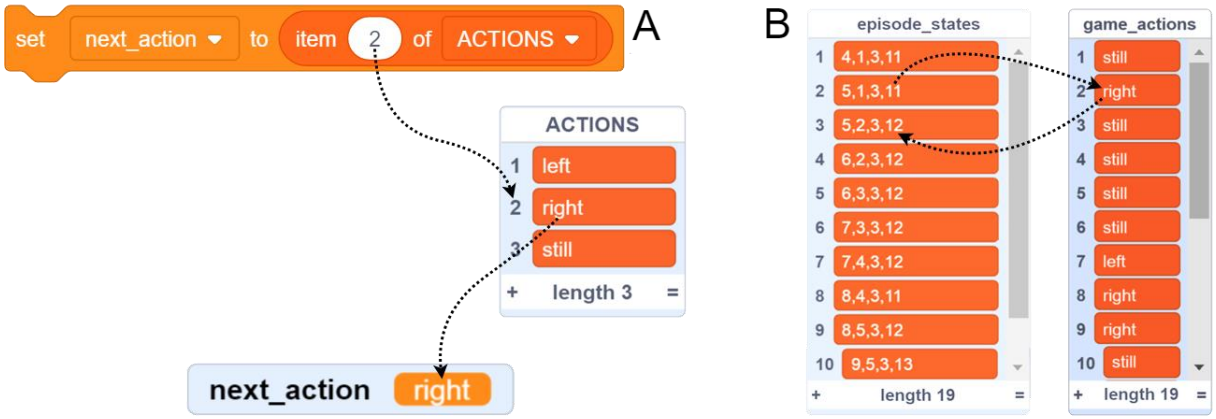
της είναι η (μοναδική) ταυτότητα της κατάστασης. Η σύνδεση της κατάστασης με την ταυτότητά της γίνεται με τον κώδικα του Σχήματος 7.

Μέχρι εδώ ασχοληθήκαμε με την αναπαράσταση του περιβάλλοντος (σε καταστάσεις). Στη συνέχεια θα ασχοληθούμε με τη δράση που επιλέγει να κάνει ο πράκτορας όταν βρεθεί σε μια κατάσταση.

6. Οι δράσεις του πράκτορα

6.1 Οι δράσεις του πράκτορα σε ένα επεισόδιο

Ο Πράκτορας όταν βρίσκεται σε μια συγκεκριμένη κατάσταση που επικρατεί στο περιβάλλον του παιχνιδιού, καλείται να δράσει χρησιμοποιώντας τη ρακέτα του, επιδιώκοντας να αποκρούσει την επερχόμενη μπάλα. Οι δράσεις που μπορεί να κάνει ο Πράκτορας είναι (α) να κινήσει τη μπάλα αριστερά ή (β) να κινήσει τη μπάλα δεξιά ή (γ) να αφήσει τη μπάλα ακίνητη. Αυτή η (προσεχής) δράση του πράκτορα φυλάσσεται στη μεταβλητή next_action, η οποία μπορεί να πάρει μια από τις τιμές "left" ή "right" ή "still" που βρίσκονται από την αρχικοποίηση ως σταθερά στοιχεία στη λίστα ACTIONS (Σχήμα 8Α). Κατά την εξέλιξη ενός επεισοδίου θα πρέπει ταυτόχρονα με την καταγραφή της τρέχουσας κατάστασης στη λίστα states να καταγράφεται και η μελλοντική δράση του πράκτορα σε μια “παράλληλη” λίστα, την game_actions. Οι δύο λίστες παρέχουν έτσι για ένα επεισόδιο ολόκληρη την ακολουθία ζευγών καταστάσεων-δράσεων (Σχήμα 8Β).



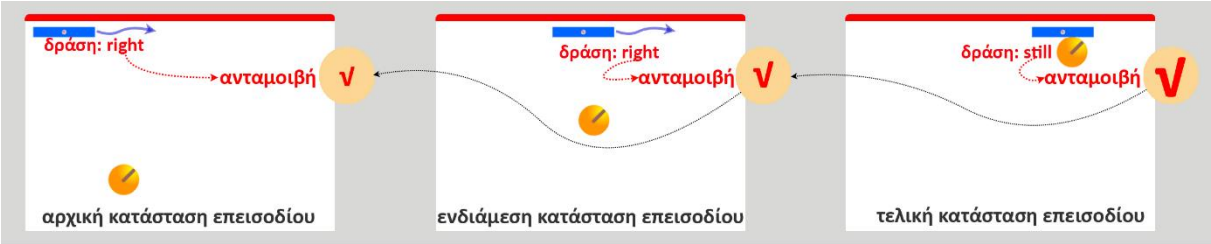
Σχήμα 8. (Α) με την εντολή set το 2ο στοιχείο της λίστας ACTION αντιγράφεται στη μεταβλητή next_action. (Β) Στη 2η κατάσταση η ρακέτα ήταν στον τομέα 11 (λίστα episode_states) και με την αντίστοιχη δράση (κίνηση right στη 2η θέση της λίστας game_actions), η ρακέτα κινείται δεξιά με αποτέλεσμα να βρεθεί στον τομέα 12 (3η κατάσταση στη λίστα episode_states).

6.2 Οι ανταμοιβές των δράσεων σε ένα επεισόδιο

Σε ένα επεισόδιο η διαδοχή των ζευγών καταστάσεων-δράσεων καταλήγει σε επιτυχία όταν αποκρούεται η μπάλα από τη ρακέτα ή διαφορετικά σε αποτυχία. Για κάθε μια από τις δράσεις του Πράκτορα -κατά τη διάρκεια ολόκληρης της ακολουθίας των καταστάσεων του επεισοδίου- πρέπει να υπάρξει ανταμοιβή.

Η ανταμοιβή θα είναι θετική σε περίπτωση επιτυχίας και αρνητική σε περίπτωση αποτυχίας. Οι ανταμοιβές θα αποδίδονται μετά την ολοκλήρωση του επεισοδίου γιατί τότε είναι γνωστή η επιτυχία ή η αποτυχία. Το ποσό της ανταμοιβής δεν θα είναι το ίδιο για όλες τις δράσεις της ακολουθίας (Σχήμα 9). Η ανταμοιβή (σε απόλυτη τιμή) πρέπει να είναι μέγιστη για τη

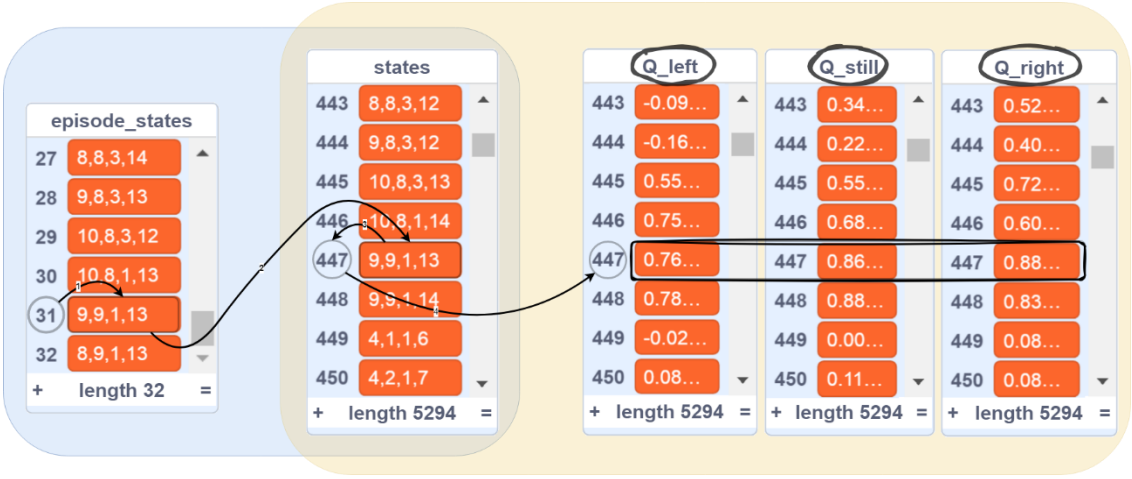
δράση της τελευταίας κατάστασης και να μειώνεται αναλογικά για τις προηγηθείσες δράσεις μέχρι την αρχική δράση που θα έχει τη μικρότερη ανταμοιβή (Laud Adam, 2004).



Σχήμα 9. Τρία ζεύγη καταστάσεων-δράσεων, σε αρχικό, ενδιάμεσο και τελικό στάδιο ενός επεισοδίου και τα ανάλογα μεγέθη των προσδοκώμενων ανταμοιβών.

6.3 Οι ανταμοιβές των δράσεων για ολόκληρη την εκπαίδευση

Για ολόκληρη την εκπαίδευση οι ανταμοιβές που παράγονται από τα διαδοχικά επεισόδια θα πρέπει να συσσωρεύονται αθροιστικά για κάθε μια ενδεχόμενη δράση μιας συγκεκριμένης κατάστασης. Έτσι για κάθε κατάσταση της λίστας states θα πρέπει να υπάρχουν οι αντίστοιχες ανταμοιβές για κάθε μια από τις (τρεις) δράσεις. Αυτό μπορεί να υλοποιηθεί συνδυάζοντας τη λίστα states με τρεις λίστες (τις Q_left, Q_right και Q_still) που λειτουργούν παράλληλα με αυτή (Σχήμα 10). Αυτό το σχήμα με λίστες θα χρησιμοποιηθεί για να αποδοθεί η ανταμοιβή σε όλες τις δράσεις που ακολουθούν τις καταστάσεις ενός επεισοδίου, χρησιμοποιώντας τον τρόπο διευθυνσιοδότησης που περιγράφηκε στον προαναφερθέντα ορισμό της κατάστασης.



Σχήμα 10. Αριστερά φαίνεται ο τρόπος που μια κατάσταση που εμφανίζεται σε κάποιο επεισόδιο αντιστοιχεί στη θέση της στη λίστα states και δεξιά ποιες είναι οι αντίστοιχες αξίες των τριών δράσεων για αυτή την κατάσταση.

7. Οι αποφάσεις του Πράκτορα

7.1 Το μοντέλο της Επανενισχυόμενης Μάθησης

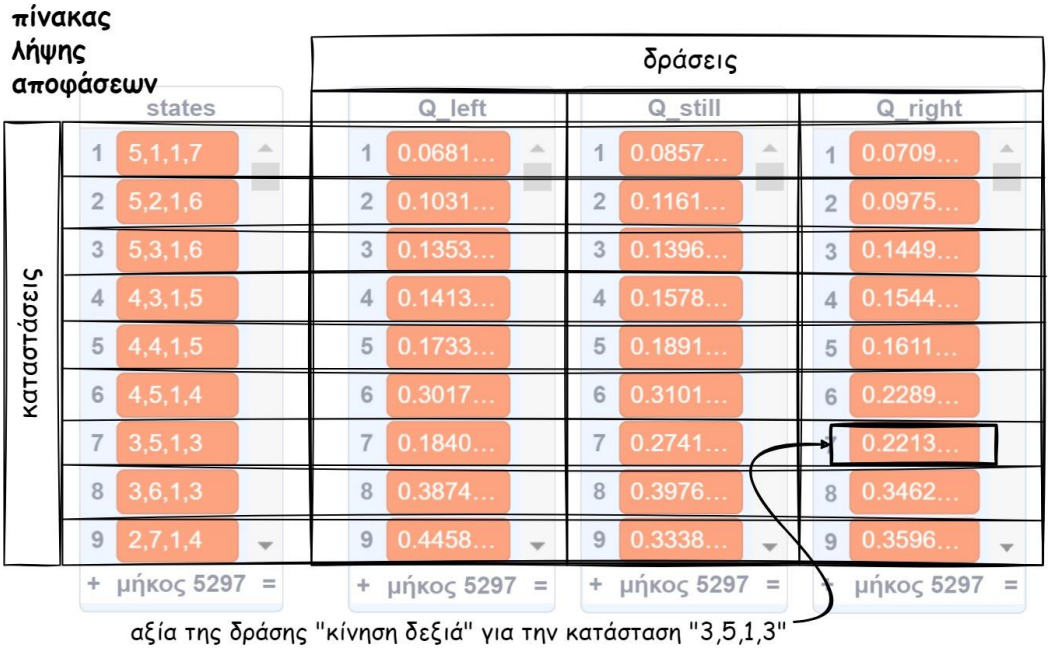
Σύμφωνα με τους Sutton & Barto (2015) το μοντέλο της Επανενισχυόμενης Μάθησης (ΕΜ) περιλαμβάνει:

- Έναν Πράκτορα που αλληλεπιδρά με το Περιβάλλον (βλ. ενότητα 4).

- Ένα σύνολο καταστάσεων του περιβάλλοντος που το κάνει κατανοήσιμο από τον Πράκτορα (βλ. ενότητες 5.3-5.4). Το πρόβλημα με την ύπαρξη άπειρων πιθανών καταστάσεων αντιμετωπίζεται με την quantization / τομεοποίηση (βλ. ενότητα 5.2) των πιθανών τιμών (Andre & Russell, 2002). Σύμφωνα με αυτή για να περιοριστεί το πλήθος των καταστάσεων, μπορούν να αντιστοιχηθούν πολλαπλές τιμές σε έναν "τομέα". Με αυτό τον τρόπο δεν γνωρίζουμε πχ την ακριβή γωνία σε μοίρες της κατεύθυνσης της μπάλας, αλλά κατά προσέγγιση αν αυτή κινείται προς τα αριστερά, προς τα πάνω ή προς τα δεξιά (Σχήμα 3Β).
- Μια δέσμη (τριών) δράσεων που μπορεί να κάνει ο Πράκτορας (βλ. ενότητα 6.1), οι οποίες μπορούν να έχουν ως αποτέλεσμα την εναλλαγή των καταστάσεων.
- Μια (αριθμητική) ανταμοιβή, θετική ή αρνητική, για κάθε ζεύγος κατάστασης-δράσης (βλ. ενότητες 6.2 και 6.3).

7.2 Η Διαδικασία Λήψης Αποφάσεων

Κατά την εκπαίδευση σε κάθε κατάσταση η ανταμοιβή κάθε δράσης του Πράκτορα φυλάσσεται σε ένα Πίνακα Λήψης Αποφάσεων (Sutton & Barto, 2015). Αυτός είναι ένας δισδιάστατος πίνακας, με κάθε γραμμή να αντιστοιχεί σε μια πιθανή τρέχουσα κατάσταση του συστήματος και κάθε στήλη να αντιστοιχεί σε μια πιθανή δράση που οδηγεί στην επόμενη κατάσταση. Ένα ισοδύναμο του Πίνακα Λήψης Αποφάσεων, όπως αυτός υλοποιείται στο Scratch, αναπαριστάνεται στο Σχήμα 11.

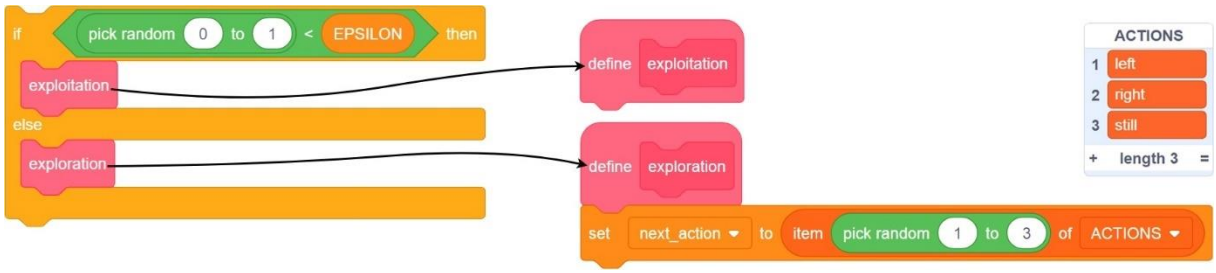


Σχήμα 11. Ο Πίνακας Λήψης Αποφάσεων όπως αυτός υλοποιείται στο Scratch.

Η συμπλήρωση του Πίνακα Λήψης Αποφάσεων με δεδομένα είναι ο τρόπος που κατά την εκπαίδευσή του ο Πράκτορας συσσωρεύει εμπειρία, είναι ο τρόπος που ο Πράκτορας μαθαίνει. Σε κάθε στοιχείο του πίνακα προστίθενται οι ανταμοιβές, αντιπροσωπεύοντας την αξία/πιθανότητα που θα έχει η συγκεκριμένη δράση στη δεδομένη κατάσταση για να κερδηθεί η παρτίδα. Έτσι με βάση τα δεδομένα του πίνακα που έχουν συσσωρευτεί, ο Πράκτορας μπορεί να δρα “ορθολογιστικά” λαμβάνοντας τις “σωστές” αποφάσεις για το ποια δράση θα επιλέξει. Αυτό γίνεται με τη Διαδικασία Λήψης Αποφάσεων. Σε αυτή ο Πράκτορας για δεδομένη κατάσταση είτε επιλέγει τη δράση εκείνη που έχει τη μεγαλύτερη

αξία (διαδικασία exploitation / εκμετάλλευσης δεδομένων) είτε επιλέγει τυχαία μια από τις τρεις δράσεις (διαδικασία exploration / εξερεύνησης εναλλακτικών). Με τη διαδικασία "εκμετάλλευσης δεδομένων" ο Πράκτορας επιλέγει εκείνη τη δράση που προσδοκά να αποδώσει άμεσα όφελος ενώ με τη διαδικασία "εξερεύνησης εναλλακτικών" ο Πράκτορας διερευνά τη δυνατότητα να υπάρχουν και άλλες (καλύτερες) επιλογές.

Η ποσόστωση της επιλογής μεταξύ των δύο διαδικασιών καθορίζεται από τη σταθερά Έψιλον (Coggan, 2004). Μια μεγάλη τιμή του Έψιλον κάνει τον Πράκτορα να συμπεριφέρεται πιο συντηρητικά και κοντόφθαλμα ενώ μια μικρή τιμή του Έψιλον τον κάνει πιο ριψοκίνδυνο και τυχοδιώκτη. Ο κώδικας σε Scratch που υλοποιεί τη Διαδικασία Λήψης Αποφάσεων είναι αυτός του Σχήματος 12.



Σχήμα 12. Ο κώδικας σε Scratch βάσει του οποίου αποφασίζεται αν η συμπεριφορά του Πράκτορα κατά την εκπαίδευση θα είναι συντηρητική αποσκοπώντας σε μικρά αλλά σίγουρα κέρδη ή τυχοδιωκτική παίρνοντας ρίσκα επιδιώκοντας μεγάλα αλλά επισφαλή κέρδη.

7.3 Ο υπολογισμός της ανταμοιβής

Όπως αναφέρεται στην εισαγωγή, υπάρχουν διάφοροι αλγόριθμοι που μπορούν να εφαρμοστούν για να μεγιστοποιηθεί η ανταμοιβή/αξία των δράσεων στην Επανενισχυόμενη Μάθηση. Μια από αυτές τις μεθόδους που θα χρησιμοποιηθεί στη παρούσα εργασία βασίζεται στον αλγόριθμο/τεχνική Q-learning. Σύμφωνα με αυτό ο Πράκτορας «σκέφτεται»: "Ημουν στην κατάσταση Κ, έκανα τη δράση Δ και είχα το αποτέλεσμα Α. Αυτή την εμπειρία μου αποτυπώνω άμεσα σε ένα πίνακα Μ". Αυτή η εκάστοτε ανταμοιβή μιας δράσης μιας κατάστασης (πχ το στοιχείο 447 της λίστας Q_left με τιμή 0.76 στο Σχήμα 10) προστίθεται στην αξία που είχε συσσωρεύσει αυτή η δράση από τα προηγούμενα επεισόδια. Ο υπολογισμός της ανταμοιβής γίνεται χρησιμοποιώντας τον απλοποιημένο τύπο του Σχήματος 13. Μεγαλύτερη εμβάθυνση στο πως καταλήγουμε σε αυτό τον τύπο ξεφεύγει από τους στόχους της εργασίας που απευθύνεται σε μικρούς μαθητές.

Σχήμα 13. Ο τύπος που κατά την εκπαίδευση υπολογίζει πόσο θα αλλάξει η αξία μιας δράσης του Πράκτορα όταν βρεθεί σε μια συγκεκριμένη κατάσταση κατά τη διεξαγωγή ενός επεισοδίου.

Τι εκφράζουν οι διάφορες μεταβλητές στον τύπο;

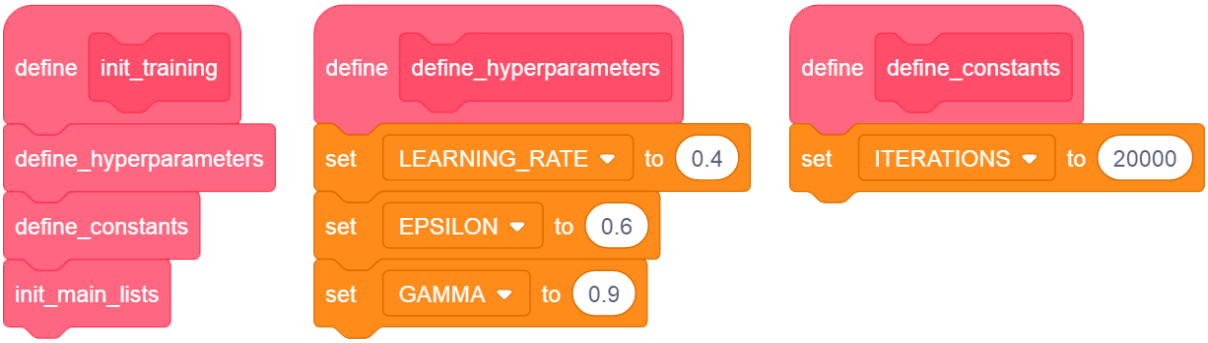
- Η μεταβλητή game_reward είναι το ποσοστό της ανταμοιβής που αντιστοιχεί για το συγκεκριμένο βήμα του επεισοδίου (§6.2).
- Ο Ρυθμός Μάθησης (LEARNING_RATE) καθορίζει σε ποιον βαθμό οι νεοαποκτηθείσες πληροφορίες αντικαθιστούν τις παλιές πληροφορίες. Μια τιμή 0 του Ρυθμού Μάθησης κάνει τον Πράκτορα να μην μαθαίνει τίποτα (και να βασίζεται αποκλειστικά

σε προηγούμενες γνώσεις), ενώ μια τιμή 1 κάνει τον Πράκτορα να λαμβάνει υπόψη μόνο τις πιο πρόσφατες πληροφορίες (αγνοώντας τις προηγούμενες γνώσεις).

- Ο Συντελεστής Έκπτωσης γ (GAMMA)εκφράζει την προσδοκία για μακροπρόθεσμα μεγαλύτερες ανταμοιβές. Μια τιμή 0 του γ θα κάνει τον Πράκτορα "μυωπικό", βλέποντας μόνο στις άμεσες / προφανείς ανταμοιβές, ενώ μια μεγαλύτερη τιμή του γ (που δεν ξεπερνά το 1) θα κάνει τον Πράκτορα να αποβλέπει μακροπρόθεσμα σε υψηλές ανταμοιβές, αγνοώντας τις άμεσες.
- Η μεταβλητή new_q_val είναι η ζητούμενη Μέγιστη Ανταμοιβή για την επόμενη κατάσταση που μπορεί να υπάρξει για δεδομένους τον Ρυθμό Μάθησης και τον Συντελεστή Έκπτωσης.

7.4 Υπερπαράμετροι και πολιτική της εκπαίδευσης του Πράκτορα

Η σταθερά Έψιλον μαζί με το Ρυθμό Μάθησης και τον Συντελεστή Έκπτωσης γ αποτελούν τις υπερπαραμέτρους του προγράμματος. Για κάθε εκπαίδευση του Πράκτορα οι υπερπαράμετροι έχουν συγκεκριμένες τιμές. Στο Σχήμα 14 φαίνονται οι τιμές των υπερπαραμέτρων που ορίζονται στην αρχικοποίηση του προγράμματος-Πράκτορα.



Σχήμα 14. Τμήματα του κώδικα στα οποία αποδίδονται οι αρχικές τιμές των υπερπαραμέτρων οι οποίες καθορίζουν την πολιτική της εκπαίδευσης του Πράκτορα στη μελέτη περίπτωσης της παρούσας εργασίας.

Η επιλογή των τιμών των υπερπαραμέτρων εκφράζει την “πολιτική” στην οποία βασίζεται η εκπαίδευση του Πράκτορα (Luca, 2024). Σε μια Διαδικασία Λήψης Αποφάσεων, ο στόχος είναι να βρεθεί η κατάλληλη "πολιτική" που να βελτιστοποιεί τον τρόπο που αποφασίζει ο Ιθύνων Νους, δηλαδή να βρεθεί εκείνος ο συνδυασμός των τιμών των υπερπαραμέτρων που θα έχει ως αποτέλεσμα ο Πράκτορας να πετύχει την καλύτερη απόδοση (Parker-Holder et al. 2022].

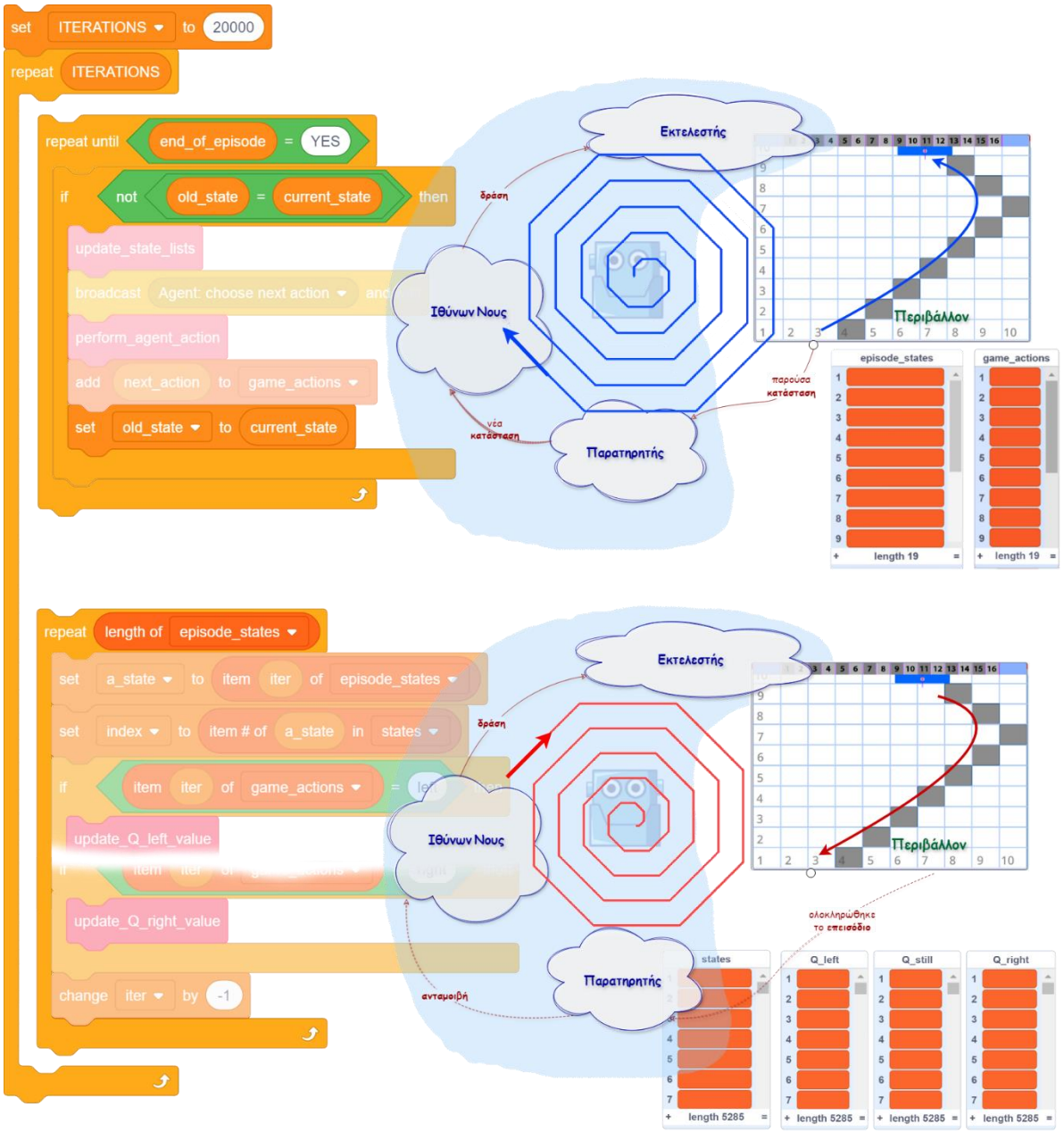
Όπως είδαμε οι δράσεις για τις αντίστοιχες καταστάσεις δημιουργούν μια ακολουθία ζευγών καταστάσεων-δράσεων στη διάρκεια ενός επεισοδίου.

8. Εκπαίδευση του Πράκτορα

8.1 Πώς ο Πράκτορας εκπαιδεύεται

Θα εστιάσουμε στη συνολική εικόνα της εκπαίδευσης του Πράκτορα. Η όλη εκπαίδευση βασίζεται σε 20.000 επαναλήψεις μιας παρτίδας του παιχνιδιού (ενός επεισοδίου). Κάθε μια από αυτές τις παρτίδες έχει δύο φάσεις: Στην πρώτη φάση (Σχήμα 15 επάνω) εξελίσσεται η αλληλουχία καταστάσεων-δράσεων από την αρχή του επεισοδίου μέχρι να φτάσει στο τέλος

(τερματική κατάσταση) οπότε είναι γνωστή η επιτυχής ή μη έκβαση του παιχνιδιού. Από αυτή τη φάση έχουν καταγραφεί ως δεδομένα στις λίστες episode_states και game_actions οι καταστάσεις στις οποίες βρέθηκε το Περιβάλλον και οι αντίστοιχες δράσεις του Πράκτορα.



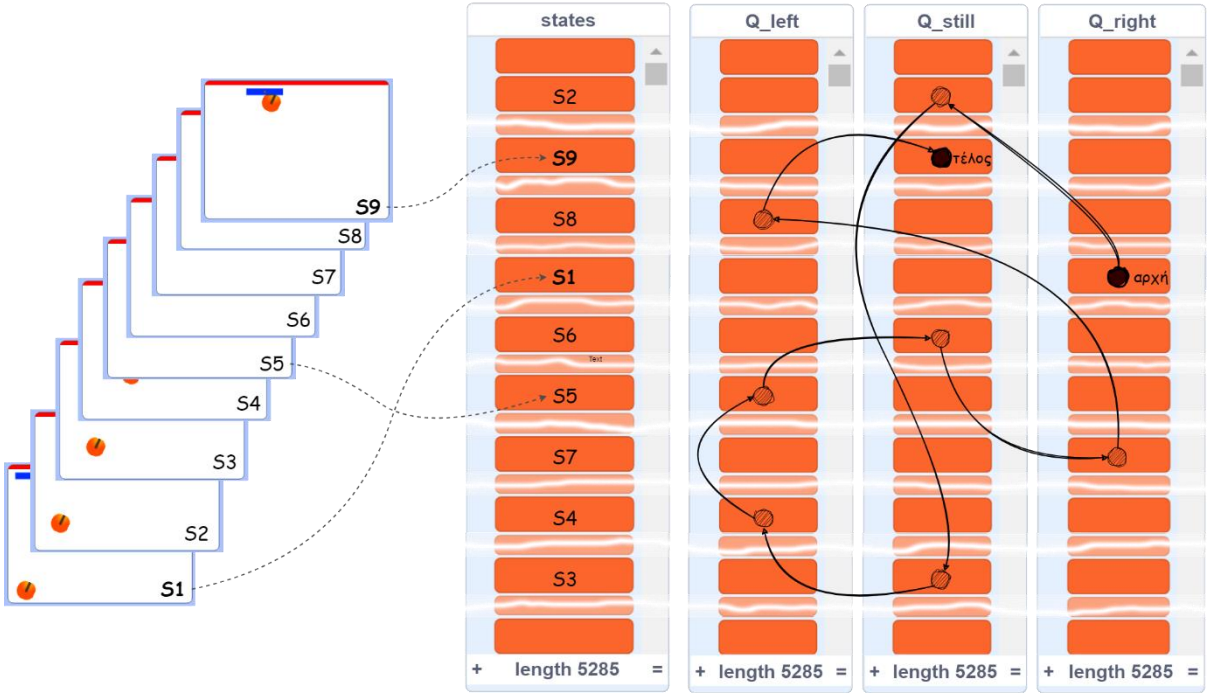
Σχήμα 15. Διάγραμμα που συνδέει τις επαναλήψεις του κώδικα με το μοντέλο του βρόχου εκπαίδευσης του Πράκτορα στο Περιβάλλον και τα προκύπτοντα αποτελέσματα που αποτυπώνονται στις λίστες των δεδομένων.

Στη δεύτερη φάση (Σχήμα 15 κάτω) και αφού πλέον είναι γνωστή η έκβαση του παιχνιδιού, ακολουθεί μια αντίστροφη πορεία από την τελευταία δράση προς την πρώτη δράση του Πράκτορα, σε κάθε βήμα της οποίας αποδίδεται η αντίστοιχη ανταμοιβή σε κάθε δράση. Από αυτή τη φάση ενημερώνονται οι τιμές των αντίστοιχων ανταμοιβών/αξιών στις λίστες Q_left, Q_still και Q_right του Πίνακα Λήψης Αποφάσεων. Ο υπολογισμός γίνεται με τον τύπο της ενότητας 7.3.

8.2 Πώς ο Πράκτορας παίζει

Όταν ολοκληρωθεί η εκπαίδευση του Πράκτορα (για 20.000 επεισόδια-παρτίδες), αυτός έχει στη διάθεσή του έναν Πίνακα Λήψης Αποφάσεων με ικανοποιητικό πλήθος καταστάσεων (περίπου 5.300 από τις υπάρχουσες 5.400) και με μια τιμή για κάθε μια από τις τρεις δράσεις για όλες τις καταστάσεις που βρήκε.

Όταν ο Πράκτορας παίζει, πρέπει σε κάθε κατάσταση να επιλέξει την κατάλληλη δράση η οποία θα κινήσει τη ρακέτα για να αποκρούσει τη μπάλα στην τερματική κατάσταση. Αυτό επιτυγχάνεται με την ενημέρωση του Πράκτορα σε ποια κατάσταση βρίσκεται το Περιβάλλον, την εύρεση αυτής της κατάστασης στον Πίνακα Λήψης Αποφάσεων και την επιλογή εκείνης της δράση που έχει τη μεγαλύτερη αξία. Σε ένα επεισόδιο η ακολουθία όλων των ζευγών καταστάσεων-δράσεων που επιλέγει ο Πράκτορας δημιουργεί μια Αλυσίδα Λήψης Αποφάσεων (Σχήμα 16).

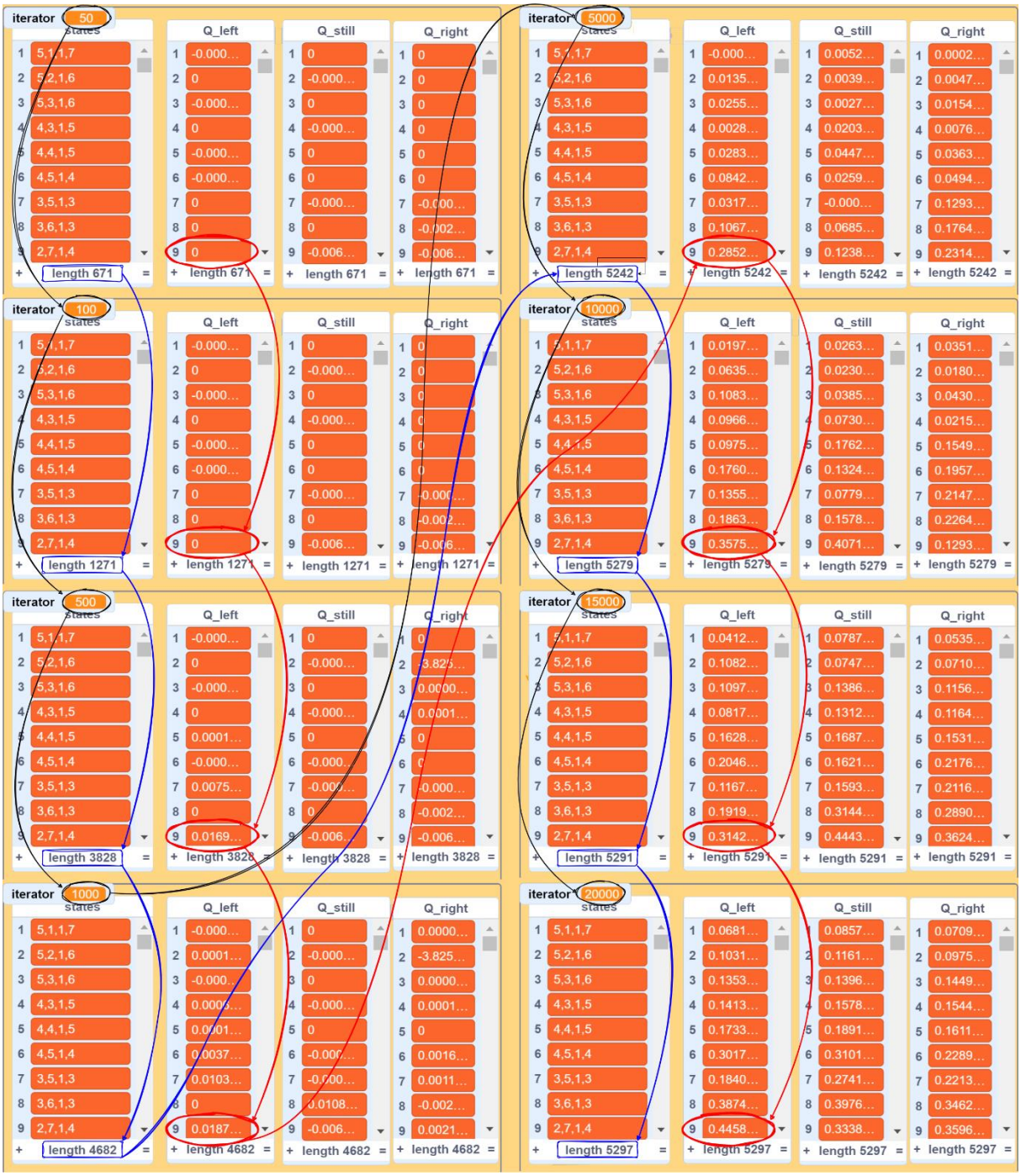


Σχήμα 16. Απεικόνιση ενός παραδείγματος μιας Αλυσίδας Λήψης Αποφάσεων του Πράκτορα στη διάρκεια ενός επεισοδίου. Ξεκινά από την κατάσταση S1 και καταλήγει στην κατάσταση S9.

Όπως αναφέρθηκε στην αρχή της εργασίας ο ανεκπαίδευτος Πράκτορας που παίζει στην τύχη καθώς ο Πίνακας Λήψης Αποφάσεων είναι άδειος, έχει ποσοστά επιτυχίας 30%-35% ενώ ο εκπαιδευμένος Πράκτορας (με τη συγκεκριμένη πολιτική) έχει ποσοστά επιτυχίας 90%-95%.

9. Παραγωγή μεγάλων ποσοτήτων δεδομένων (big data)

Στην Επανενισχυόμενη Μάθηση δεν προαπαιτούνται δεδομένα για την εκπαίδευση του Πράκτορα. Οι απαραίτητες μεγάλες ποσότητες δεδομένων στις οποίες θα βασιστεί η εκπαίδευση του Πράκτορα παράγονται κατά τη διάρκεια της εκπαίδευσης.
Κατά την εκπαίδευση ο Πίνακας Λήψης Αποφάσεων είναι αρχικά άδειος και βαθμιαία γεμίζει με δεδομένα. Οκτώ στιγμιότυπα από μια εκπαίδευση του Πράκτορα παρουσιάζονται στο Σχήμα 17. Τα στιγμιότυπα έχουν ληφθεί στις υπ’ αριθμόν 50, 100, 500, 1000, 5000, 10.000, 15.000 και 20.000 επαναλήψεις (μαύρες γραμμές στο Σχήμα 17).



Σχήμα 17. Οκτώ στιγμιότυπα του Πίνακα Λήψης Αποφάσεων κατά την εκπαίδευση του Πράκτορα

Κατά τη διάρκεια της εκπαίδευσης υπάρχει η δυνατότητα να μελετηθούν από τους μαθητές και να αναπαρασταθούν σε γραφικές παραστάσεις:

- Το πως μεταβάλλεται το πλήθος των καταστάσεων που ανακαλύπτονται σε σχέση με τον αύξοντα αριθμό των επαναλήψεων των επεισοδίων. Ενδεικτικά στο Σχήμα 17 παρουσιάζεται το πώς αυξάνεται το πλήθος των καταστάσεων που ανακαλύπτει ο Πράκτορας που είναι 671, 1271, 3828, 4682, 5242, 5279, 5291 και 5297 (μπλε γραμμές στο Σχήμα 17). Φαίνεται ο ραγδαίος ρυθμός ανακάλυψης νέων καταστάσεων στις πρώτες χίλιες επαναλήψεις, ενώ ο ρυθμός σταθεροποιείται μετά τις 5.000 επαναλήψεις.

- Ο τρόπος που εξελίσσεται / αλλάζει η αξία ενός συγκεκριμένου ζεύγους κατάστασης-δράσης κατά τη διάρκεια της εκπαίδευσης δηλαδή πώς αλλάζει η τιμή ενός κελιού του Πίνακα Λήψης Αποφάσεων. Στο Σχήμα 17 μπορεί να παρατηρηθεί το πώς αλλάζει κατά τη διάρκεια της εκπαίδευσης η τιμή ενός συγκεκριμένου ζεύγους κατάστασης-δράσης π.χ. της υπ. αρ. 9 κατάστασης (2,7,1,4) με τη δράση "κίνηση αριστερά" (κόκκινες γραμμές στο Σχήμα 17).
- Η συχνότητα με την οποία συναντάται μια κατάσταση σε συνάρτηση με τον αύξοντα αριθμό των επαναλήψεων των επεισοδίων.
- Ο χρόνος της εκπαίδευσης του Πράκτορα για 20.000 επαναλήψεις, με τη συγκεκριμένη πολιτική και "τρέχοντας" σε ένα απλό οικιακό υπολογιστή είναι περίπου 45 λεπτά.

Τέλος ένα επιπλέον στοιχείο που μπορούν να παρατηρήσουν οι μαθητές πειραματιζόμενοι με την εκπαίδευση του Πράκτορα είναι η απόδοσή του όταν παίζει (όχι κατά την εκπαίδευση) σε σχέση με το πλήθος των επαναλήψεων της εκπαίδευσή του. Ενδεικτικές τιμές φαίνονται στον Πίνακα 1.

Πίνακας 1. Εκπαίδευση και απόδοση Πράκτορα

Επαναλήψεις εκπαίδευσης Πράκτορα	**Απόδοση Πράκτορα (%)**
0	35
50	35
100	35
500	48
1.000	66
5.000	88
10.000	92
15.000	94
20.000	95

Οι γραφικές παραστάσεις μπορούν είτε να υλοποιηθούν με κώδικα στο Scratch είτε να γίνουν σε ένα λογιστικό φύλλο, αφού υπάρχει η δυνατότητα το περιεχόμενο μιας λίστας στο Scratch να εξαχθεί σε ένα αρχείο ".txt".

10. Εναλλακτικές επιλογές σχεδίασης και πολιτικές εκπαίδευσης

Το λογισμικό που προσομοιώνει το παιχνίδι Pong μαζί με το λογισμικό του Πράκτορα αποτελεί ένα μικρόκοσμο. Ο μικρόκοσμος αυτός προσφέρεται για δραστηριότητες από τους μαθητές με τις οποίες θα κατανοήσουν τον τρόπο λειτουργίας της Μηχανικής Μάθησης. Τέτοιες δραστηριότητες που πρέπει να συσχετιστούν με την απόδοση του Πράκτορα (μετά την επανεκπαίδευσή του) κατά το παιχνίδι μπορεί να είναι:

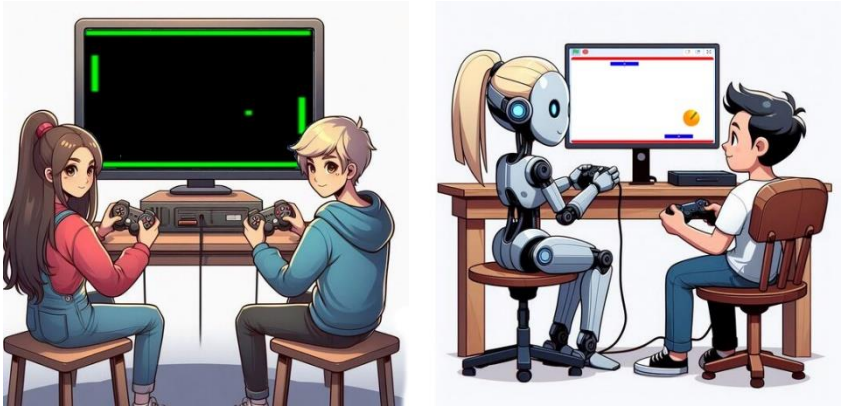
- Να γίνουν αλλαγές στο μέγεθος ή/και την ταχύτητα της μπάλας ή/και της ρακέτας.
- Να τροποποιηθεί η "τομεοποίηση" στον ορισμό των καταστάσεων ώστε να έχουμε περισσότερο ή λιγότερο λεπτομερείς καταστάσεις.

- Να αναζητηθούν άλλες εναλλακτικές παράμετροι που να ορίζουν μια κατάσταση π.χ. να θεωρηθεί ότι ο παρατηρητής είναι κινούμενος (και κάθεται πάνω στη ρακέτα) και να χρησιμοποιηθούν ως παράμετροι η απόσταση ρακέτας-μπάλας και η σύγκλιση ή μη των διανυσμάτων ταχυτήτων της ρακέτας με τη μπάλα. Η προσέγγιση αυτή ίσως είναι πιο κατανοητή από τους μαθητές καθώς μοιάζει περισσότερο με τον τρόπο που θα χειρίζονταν αυτοί τη ρακέτα αν έπαιζαν Pong.
- Να αλλάξουν οι τιμές των υπερπαραμέτρων, δηλαδή να αλλάξει η πολιτική εκπαίδευσης του Πράκτορα.

11. Προοπτικές εξέλιξης του Πράκτορα

Ο συγκεκριμένος Πράκτορας αναπτύχθηκε για να αποκρούσει τις “μπαλιές” από ένα αυτόματο εκτοξευτή της μπάλας. Όμως με τις σχετικές μικρές και εύκολες τροποποιήσεις στο περιβάλλον του Scratch μπορεί να χρησιμοποιηθεί:

- Για να φτιαχτεί ένα παιχνίδι που ο Πράκτορας να παίζει με έναν άνθρωπο (Σχήμα 18).
- Για να φτιαχτεί ένα παιχνίδι που ο Πράκτορας να παίζει εναντίον ενός άλλου Πράκτορα που έχει εκπαιδευτεί με διαφορετική πολιτική. Αυτό θα μπορούσε να εξελιχθεί σε ένα διαγωνισμό μεταξύ Πρακτόρων τους οποίους αναπτύσσουν διαφορετικές ομάδες μαθητών.
- Η εκπαίδευση του Πράκτορα να επαναλαμβάνεται μέσα σε ένα βρόχο πολλές φορές με διαφορετικές επιλεγμένες πολιτικές κάθε φορά και στη συνέχεια να γίνεται η σύγκριση της αποτελεσματικότητας για κάθε πολιτική.
- Να χρησιμοποιηθεί ένας εκπαιδευμένος Πράκτορας Α ως αντίπαλος κατά την εκπαίδευση ενός νέου ανεκπαίδευτου Πράκτορα Β. Στη συνέχεια ο Πράκτορας Β μπορεί να χρησιμοποιηθεί στην επανεκπαίδευση του Πράκτορα Α. Αυτό επαναλαμβανόμενο δημιουργεί μια σπείρα αυτοβελτίωσης των δυνατοτήτων των Πρακτόρων.
- Να χρησιμοποιηθούν περισσότεροι του ενός Πράκτορες, που έχουν εκπαιδευτεί με την ίδια ή διαφορετικές πολιτικές, οι οποίοι να συναποφασίζουν μετά από ψηφοφορία (με βαρύτητες) για το ποια θα είναι η επόμενη δράση σε μια συγκεκριμένη κατάσταση.



Σχήμα 18. Από το κλασικό Pong στο παιχνίδι πράκτορα με άνθρωπο. Η βασική εικόνα είναι προϊόν Τεχνητής Νοημοσύνης με τροποποιήσεις στο Photoshop.

Οι πολλαπλοί Πράκτορες στο Scratch μπορεί να είναι είτε αντίγραφα του αρχικού Πράκτορα είτε κλώνοι του. Σημειώνεται ότι τα αντίγραφα των Πρακτόρων δημιουργούνται από τον

προγραμματιστή στη φάση της δημιουργίας του προγράμματος και η φύση τους δεν μπορεί να αλλάξει κατά την εκτέλεση του προγράμματος. Αντίθετα η χρήση κλώνων προγραμματίζεται μεν από τον προγραμματιστή αλλά η δημιουργία τους γίνεται κατά την εκτέλεση του προγράμματος, παρέχοντας ευελιξία στη χρήση του προγράμματος.

Να σημειωθεί ότι όλα τα προηγούμενα μπορεί να γίνουν πειραματικά από μαθητές της δευτεροβάθμιας εκπαίδευσης και έτσι αυτοί να αποκτήσουν εμπειρίες χρήσιμες στο να κατανοούν πώς εκπαιδεύεται η Τεχνητή Νοημοσύνη.

12. Τα χαρακτηριστικά της εφαρμογής της Επανενισχυόμενης Μάθησης

Η εργασία περιγράφει μια ολοκληρωμένη διδακτική προσέγγιση για την ενσωμάτωση της Επανενισχυόμενης Μάθησης στην τάξη για μαθητές γυμνασίου. Καθοδηγεί τους εκπαιδευτικούς και τους μαθητές σε διάφορες φάσεις, από την εισαγωγή των εννοιών της Επανενισχυόμενης Μάθησης έως την εφαρμογή τους σε παιχνίδια, επιτρέποντας μια βαθύτερη κατανόηση των διαδικασιών Μηχανικής Μάθησης. Για τους μαθητές τα χαρακτηριστικά αυτής της δομημένης προσέγγισης είναι:

- **Η Τεχνητή Νοημοσύνη δεν αντιμετωπίζεται ως Μαύρο Κουτί.** Ο εφαρμοζόμενος αλγόριθμος της Τεχνητής Νοημοσύνης δεν αντιμετωπίζεται ως ένα Μαύρο Κουτί αλλά παρέχεται η δυνατότητα στους μαθητές να δουν τι συμβαίνει στο εσωτερικό του, να τον αναλύσουν σύμφωνα με το μοντέλο του βρόχου εκπαίδευσης του Πράκτορα στο Περιβάλλον της εργασίας και να επεξεργαστούν τα διάφορα τμήματά του, κατανοώντας έτσι τον τρόπο λειτουργίας του (Λαδιάς, 2024).
- **Ο κώδικας του Πράκτορα είναι διαφανής**. Το γεγονός ότι η υλοποίησή του Πράκτορα γίνεται στο Scratch, που είναι ένα ισχυρό προγραμματιστικό περιβάλλον με το οποίο οι μαθητές του γυμνασίου είναι εξοικειωμένοι, έχει ως αποτέλεσμα ο κώδικας να είναι κατανοητός και διαφανής για τους μαθητές. Σε αυτό βοηθά ότι το Scratch χρησιμοποιεί αντικείμενα και ο κώδικας κατανέμεται σε διαφορετικά αντικείμενα ανάλογα με το σκοπό που εξυπηρετεί. Αποτέλεσμα είναι να γίνεται εύκολα κατανοητός ο κώδικας του αντικειμένων “Παρατηρητής”, “Ιθύνων Νους” και “Εκτελεστής” που είναι τα επιμέρους τμήματα του Πράκτορα. Όλα τα τμήματα και οι μεταβλητές του κώδικα που υλοποιεί τον αλγόριθμο μπορούν να ελεγχθούν, επιτρέποντας στους μαθητές να παρατηρήσουν πώς αλλάζουν οι τιμές με την πάροδο του χρόνου και να κατανοήσουν τη λειτουργία του αλγορίθμου (Jatzlau, et al. 2019).
- **Η εφαρμογή ολοκληρώνεται μέσα σε ένα αυτόνομο μικρόκοσμο**. Στο ίδιο προγραμματιστικό περιβάλλον υλοποιείται και το παιχνίδι Pong και ο αυτοεκπαιδευόμενος Πράκτορας, μια ολοκλήρωση που προσφέρεται για να κατανοήσουν οι μαθητές την ολότητα του μικρόκοσμου στον οποίο δουλεύουν. Η εφαρμογή είναι αυτόνομη και όλα τα έργα των μαθητών δημιουργούνται στο ίδιο πλαίσιο. Δεν απαιτούνται υπηρεσίες από εξωτερικούς πόρους για την εκπαίδευση μοντέλων ή υπολογισμών. Επιπλέον η επιλογή της Επανενισχυόμενης Μάθησης παράγει μόνη της τα απαιτούμενα σύνολα από δεδομένα και δεν απαιτεί την τροφοδοσία του Πράκτορα με εξωτερικά δεδομένα (Jatzlau, et al., 2019). Όμως ο μικρόκοσμος δεν είναι απομονωμένος από τον υπόλοιπο κόσμο γιατί στο Scratch παρέχεται η δυνατότητα εξαγωγής/εισαγωγής των δεδομένων μιας λίστας.

- **Δημιουργούνται νοηματοδοτούμενα έργα με εποικοδομιστική προσέγγιση**. Το έργο με το οποίο ασχολούνται οι μαθητές είναι ένα ηλεκτρονικό παιχνίδι και ως τέτοιο είναι νοηματοδοτούμενο για τους μικρούς μαθητές. Αυτοί μπορούν να χρησιμοποιούν το αυτοεκπαιδευόμενο πρόγραμμα και τροποποιώντας το να δημιουργούν δικά τους αυθεντικά έργα, με τα οποία μπορούν να παίξουν έχοντας έτσι ενδογενή κίνητρα για να ασχοληθούν με αυτά. Τα έργα αυτά μπορούν να τα "πειράξουν", να πειραματιστούν και με αυτό τον τρόπο να κατανοήσουν ως βίωμα τις έννοιες της Μηχανικής Μάθησης με έναν απτό και ουσιαστικό τρόπο. Όλη αυτή η μαθησιακή προσέγγιση εντάσσεται στον εποικοδομιστικό τρόπο μάθησης (Papert, 1991).
- **Οι μαθητές αναπτύσσουν μεταγνωστικές δεξιότητες**. Οι μαθητές εκπαιδεύοντας τον Πράκτορα να συσσωρεύει εμπειρία και να μαθαίνει με συμπεριφοριστικό τρόπο αλλά και ταυτόχρονα πειραματιζόμενοι με τις υπερπαραμέτρους συνειδητοποιούν τη διαφορά και την αξία του δικού τους εποικοδομητιστικού τρόπου σκέψης με αυτόν του συμπεριφορισμού. Παρέχεται στους μαθητές η ευκαιρία να κατανοήσουν πώς μαθαίνει ένας Πράκτορας αλληλεπιδρώντας με το περιβάλλον χρησιμοποιώντας μια διαδικασία ανταμοιβής των δράσεών του.
- **Το κέντρο βάρος του προγράμματος μετατοπίζεται από τον αλγόριθμο προς τα δεδομένα**. Σύμφωνα με τον Wirth (1990), "Πρόγραμμα = Αλγόριθμος + Δεδομένα". Στον κλασικό τρόπο προγραμματισμού ο προγραμματιστής πρέπει να αναπτύξει έναν αλγόριθμο που να επιλύει βήμα προς βήμα το πρόβλημα, με τα δεδομένα να παίζουν δευτερεύοντα ρόλο. Στην προσέγγιση της Επανενισχυόμενης Μάθησης το κέντρο βάρους του προγράμματος μετατοπίζεται από τον αλγόριθμο προς τα δεδομένα. Τα δεδομένα αναπαριστούν το περιβάλλον του παιχνιδιού στο πρόγραμμα και σε αυτά αποθηκεύεται η εμπειρία που συσσωρεύει το πρόγραμμα κατά την εκπαίδευσή του (Ρεπαντής, 2024, Λαδιάς, 2024).
- **Ο ίδιος αλγόριθμος της Μηχανικής Μάθησης μπορεί να λύσει διαφορετικά προβλήματα**. Αν η μέθοδος της επανενισχυόμενης μάθησης εφαρμοστεί σε διαφορετικά προβλήματα / παιχνίδια, τότε θα μπορέσει ο μαθητής να διαπιστώσει ότι δεν απαιτείται διαφορετικός αλγόριθμος για κάθε πρόβλημα (όπως στον κλασικό προγραμματισμό) αλλά ένας αλγόριθμος (με μικρές τροποποιήσεις κάθε φορά) που μαθαίνει από διαφορετικά περιβάλλοντα/παιχνίδια κάθε φορά.

Συμπεράσματα

Η παρούσα εργασία είχε ως σκοπό αφενός να γίνει κατανοητό μέσα από ένα παράδειγμα του πώς δουλεύει η Τεχνητή Νοημοσύνη στο πλαίσιο της μηχανικής μάθησης και αφετέρου πώς μπορεί αυτό να παρουσιαστεί σε μικρούς μαθητές με τους υπάρχοντες περιορισμούς στις μαθηματικές και προγραμματιστικές τους γνώσεις. Για να το πετύχει περιέγραψε τον τρόπο που εκπαιδεύεται ένας Πράκτορας (το αυτοεκπαιδευόμενο πρόγραμμα) μαθαίνοντας από την εμπειρία του όταν παίζει το παιχνίδι Pong. Η εκπαίδευσή του βασίστηκε στην Επανενισχυόμενη Μάθηση, με τον αλγόριθμο Q-learning. Ο Πράκτορας υλοποιήθηκε στο προγραμματιστικό περιβάλλον Scratch που απευθύνεται σε παιδιά και στην εργασία παρέχεται μια γενική περιγραφή του κώδικα. Με την πολιτική που επιλέχθηκε ο Πράκτορας αύξησε το ποσοστό επιτυχίας 30%-35% που είχε πριν την εκπαίδευση σε 90%-95% μετά την

εκπαίδευση. Η προσέγγιση για την παρουσίαση της εργασίας ακολούθησε μια προσέγγιση "από κάτω προς τα πάνω" και στη συνέχεια έγινε η ένταξη αυτού του παραδείγματος σε ένα ευρύτερο θεωρητικό πλαίσιο. Κατά τη θεωρητικοποίηση έγινε προσπάθεια να χρησιμοποιηθεί η δόκιμη ορολογία που χρησιμοποιείται στην Επανενισχυόμενη Μάθηση. Επίσης έγινε προσπάθεια να παρέχονται οπτικοποιήσεις βήμα προς βήμα της όλης διαδικασίας. Γενικά η εργασία παρουσιάζει μια όσο το δυνατόν πιο απλή και κατανοητή υλοποίηση της Επανενισχυόμενης Μάθησης για το Pong, κατάλληλη για εκπαιδευτικούς σκοπούς, ευελπιστώντας να οδηγήσει σε ένα βαθμό στην απομυθοποίηση της Τεχνητής Νοημοσύνης. Η επιτυχής εκπαίδευση του Πράκτορα αποδεικνύει τη δυνατότητα της Επανενισχυόμενης Μάθησης να λύνει προβλήματα λήψης αποφάσεων σε δυναμικά περιβάλλοντα.

Με αφορμή την εργασία οι μαθητές μπορούν να καταλήξουν σε διαπιστώσεις όπως ότι ο Πράκτορας έχει μάθει μόνος του τι να κάνει για να πετύχει το σκοπό που του έχει οριστεί από τον προγραμματιστή, αλλά (ο Πράκτορας) δεν έχει κατανοήσει γιατί το κάνει. Αξίζει να σημειωθεί ότι η μέθοδος της εκπαίδευσης του πράκτορα έχει πολλά κοινά χαρακτηριστικά με τη συμπεριφοριστική μέθοδο μάθησης.

Μια συνέχεια του παρόντος άρθρου μπορεί να περιλαμβάνει μια εκτενή αναφορά στον τρόπο που δομείται το πρόγραμμα του Πράκτορα και στον τρόπο που επικοινωνεί αυτό με το Περιβάλλον αναδεικνύοντας την προσανατολισμένη σε αντικείμενα δομή του Scratch σε συνδυασμό με τον καθοδηγούμενο από συμβάντα χαρακτήρα του. Ένα χρήσιμο εργαλείο για την παρουσίαση αυτών των χαρακτηριστικών του προγράμματος μπορεί να είναι η αναπαράσταση του προγράμματος με το ΚωδικΌραμα (Ladias, Mikropoulos, Ladias, & Bellou, 2021).

Συμπληρωματικά πεδία που θα μπορούσαν μελλοντικά να διερευνηθούν είναι να γίνει μια πιο λεπτομερής ανάλυση του αλγορίθμου και των περιορισμών του, να αναλυθεί ο τρόπος που συσσωρεύονται τα δεδομένα κατά την εκπαίδευση και να εστιάσει στην επιλογή μιας πιο αποτελεσματικής (ίσως βέλτιστης) πολιτικής που να πετυχαίνει υψηλότερη απόδοση. Επίσης να χρησιμοποιηθεί η ίδια μεθοδολογία και σε άλλα παιχνίδια έτσι ώστε οι μαθητές να διαπιστώσουν την κοινή λογική που έχει στον προγραμματισμό η εφαρμογή της Επανενισχυόμενης Μάθησης.

Αναφορές

Andre, D. and Russell, S. (2002). *State Abstraction for Programmable Reinforcement Learning Agents*. American Association for Artificial Intelligence (www.aaai.org) https://people.eecs.berkeley.edu/~russell/papers/aaai02-alisp.pdf

Coggan, M. (2004). *Exploration and Exploitation in Reinforcement Learning.* Ανακτήθηκε στις 12 Φεβρουαρίου 2024 από file:///C:/Users/USER/Desktop/FinalReport.pdf

Jatzlau, S., Michaeli, T., Seegerer, S., & Romeike, R. (2019). It's not Magic After All - Machine Learning in Snap! using Reinforcement Learning. *IEEE Blocks and Beyond Workshop.* *https://computingeducation.de/pub/2019_Jatzlau-Michaeli-Seegerer-Romeike_BLOCKSANDBEYOND19.pdf*

Ladias, A., Mikropoulos, A., Ladias, D., & Bellou, I. (2021). CodeOrama: A two-dimensional visualization tool for Scratch code to assist young learners' understanding of computer programming. *Themes in eLearning*, Vol 14 (2021).

Laud Adam, (2004). *Theory and application of reward shaping in Reinforcement Learning*. Urbana, Illinois. Ανακτήθηκε στις 3 Ιανουαρίου 2024 από https://core.ac.uk/download/pdf/4820036.pdf

Luca, Gabriele, De. (2024). *What Is a Policy in Reinforcement Learning?* Ανακτήθηκε στις 2 Μαΐου 2024, από https://www.baeldung.com/cs/ml-policy-reinforcement-learning

Marugán, A. P. (2023). Applications of Reinforcement Learning for maintenance of engineering systems: A review. *Advances in Engineering Software*, 183, https://doi.org/10.1016/j.advengsoft.2023.103487.

Ng, D.T.K., Lee, M., Tan, R.J.Y., Hu, X., J. Downie, S., Chu, S.K.W. (2023). A review of AI teaching and learning from 2000 to 2020. *Education and Information Technologies*, 28, 8445–8501.

Omkar, V. (2019). *Episodic Reinforcement Learning for Pong*. Ανακτήθηκε στις 10 Νοεμβρίου 2023, από https://towardsdatascience.com/intro-to-reinforcement-learning-pong-92a94aa0f84d

Papert, S., (1991). *Νοητικές θύελλες. Παιδιά, Ηλεκτρονικοί Υπολογιστές και Δυναμικές Ιδέες. Τα πάντα γύρω από τη Logo*. Αθήνα: Οδυσσέας.

Parker-Holder, et al. (2022). Automated Reinforcement Learning (AutoRL): A Survey and Open Problems. *Journal of Artificial Intelligence Research,* 74 (2022) 517-568. https://arxiv.org/pdf/2201.03916

Russell, S. and Norvig, P. (2010). *Artificial Intelligence. A Modern Approach*. New Jersey: Prentice Hall. https://people.engr.tamu.edu/guni/csce421/files/AI_Russell_Norvig.pdf

Sutton, P., and Barto, A. (2015). *Reinforcement Learning: An Introduction*. London: The MIT Press. A Bradford Book. https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

Wirth, N., (1990). *Αλγόριθμοι και Δομές Δεδομένων.* Αθήνα: Κλειδάριθμος.

Λαδιάς, Α. (2024). *Ας κοιτάξουμε μέσα σε ένα μαύρο κουτί Μηχανικής Μάθησης.* 3ο Πανελλήνιο Επιστημονικό Συμπόσιο "Το Δέντρο της Logo εν Ελλάδι". Ανακτήθηκε στις 15 Μαρτίου 2024, από https://www.logotreegr.net/presentations

Ρεπαντής, Β. (2024). *Reinforcement Learning στο Scratch. Από τον προγραμματισμό βάσει κανόνων, στη σκέψη που βασίζεται σε δεδομένα. Προσκεκλημένη ομιλία στο* 3ο Πανελλήνιο Επιστημονικό Συμπόσιο "Το Δέντρο της Logo εν Ελλάδι". Ανακτήθηκε στις 2 Μαΐου 2024, από https://www.logotreegr.net/presentations