Neuro-Evolutional "don't touch the Spikes" with Erlang

מגישים: טמיר כהן ונדב חדד

מרצה: יהודה בן-שימול

מתרגל: דוד לאון

קישור לסרטון הדגמה מהיוטיוב:  
[Don't Touch The Spikes - High Score (115) - iOS / Android Gameplay](https://www.youtube.com/watch?v=7t3qfK6mP64)

המאסטר:

מחלק עבודה בין 4 המחשבים. למשל אם בכל איטרציה של אוכלוסיה יש 1000 ציפורים, אז השרת שולח לכל אחד מהם 250 ציפורים (ויש 1000 רשתות נוירונים שונות סך הכל, כלומר משקולות שונות). כל מחשב נותן לציפור "לרוץ" ומגדיר לה פונקציית רווח (fitness function) שתהיה שווה לכמות הפריימים שהציפור בחיים.  
המחשבים אומרים לשרת מי הם 10% הציפורים הכי טובות שלהם והמאסטר משלב את המידע ומוליד את הדור הבא, שהוא השארה של הציפורים הכי טובות מהדור הקודם וגם מוטציות שלהן (ע"י רינדום קל במשקולות הרשת שלהם). וכך ממשיכים עוד איטרציות.

* נשלח הודעה im\_alive כל שניה ובמקרה והיא לא הגיעה נכריז על המחשב כנעדר ונחלק את העבודה בין שאר המחשבים.

**מכונת מצבים** של כל **ציפור** (לכל פרוסס):

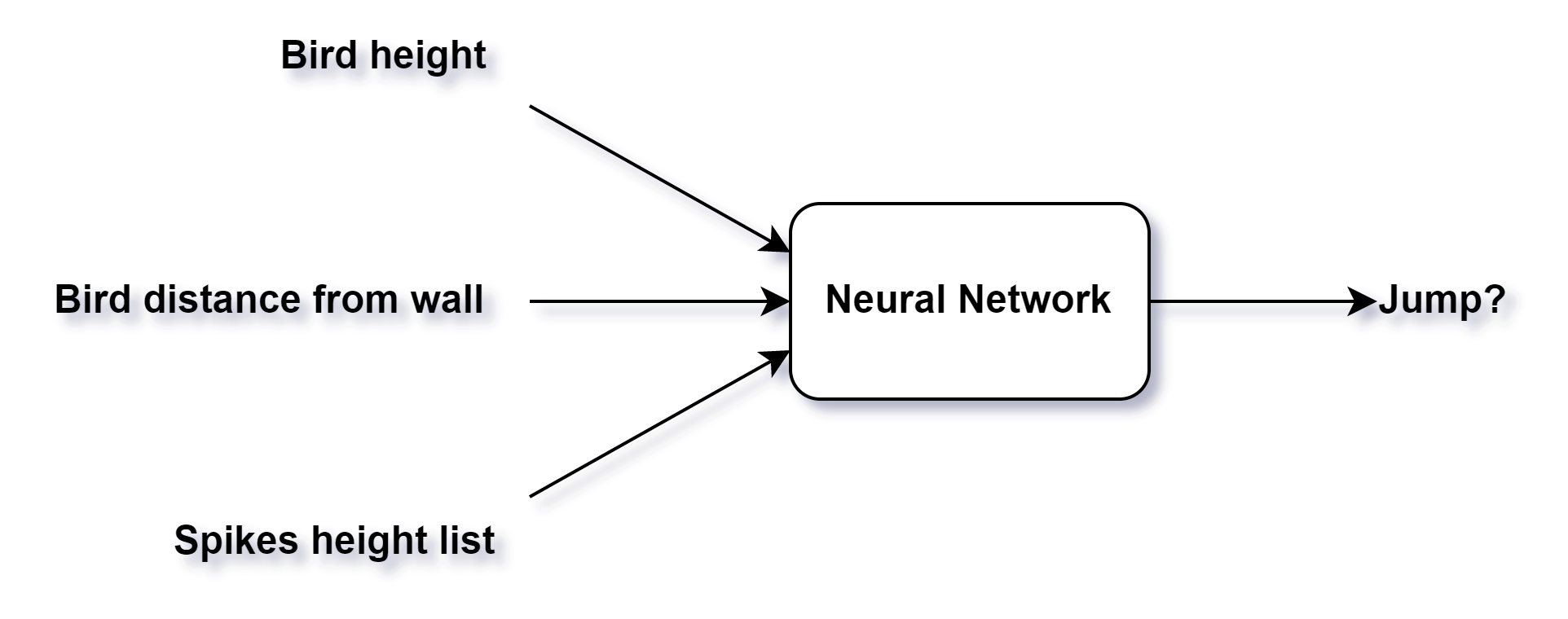
1. המתנה ללחיצה על הכפתור של המשחק או לקבלת רשת נוירונים לתפעול של הציפור. (**idle**)
2. התחלת ריצה של המשחק (בעזרת הרשת שקיבל) עד פסילה (נגיעה בקוץ). (**simulation**).

**קלט** לרשת נוירונים:

1. גובה הציפור.
2. מרחק הציפור מהקיר.
3. רשימת הקוצים שבקיר ממול.

**הפלט** מהרשת הוא האם לבצע קפיצה.

כמו באיור:



שלבי עבודה בסיסיים:

* תחילה נכין את המשחק שיעבוד אינטרקטיבית למשתמש:

מודולים/קבצים:

* קבועים.
* גרפיקה: מעדכנת את הגרפיקה של כל ציפור לפי המידע שיש לה ברקורד graphic\_state.
* ציפור: מכונת מצבים בדומה למה שיש למעלה:
* איידל: לחכות ללחיצה על סטארט
* סימולציה: ציפור קופצת עד פסילה (קפיצה בעזרת כפתור)
* אבלואציה: להדפיס ניקוד (כמות פריימים) וחזרה למצב איידל.
* לאחר מכן נשנה אותו כך שאת הבחירה האם לקפוץ תבצע רשת הנוירונים וגם נוסיף המון ציפורים.

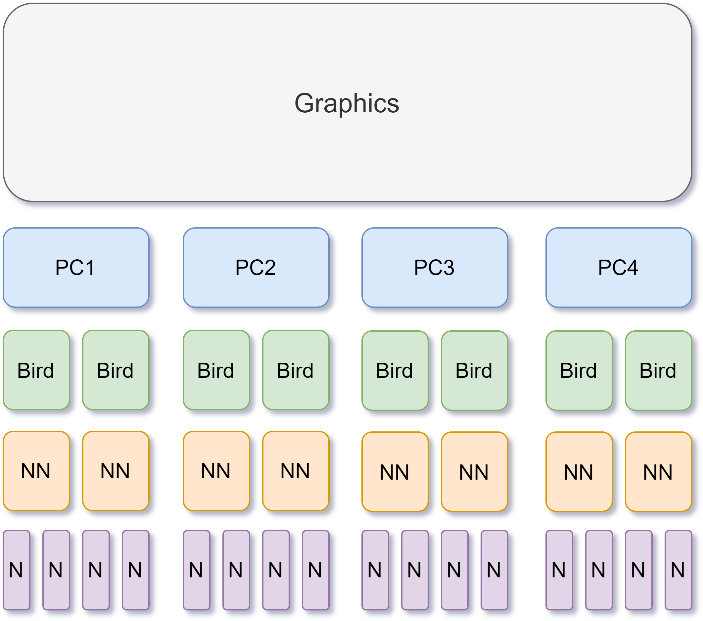
מודולים/קבצים:

* רשת נוירונים:
* התנהגות? אין
* אתחול של הרשת כולל אתחול של נוירונים
* לולאה מרכזית עם בלוק רסיב שיש בו, בהינתן מידע על ציפור, האם לקפוץ
* שליחה של המידע אל הציפור: האם לקפוץ, מבנה הרשת ורשימת משקולות
* לאחר מכן נפצל את העבודה בין טרמינלים שונים ולבסוף בין מחשבים שונים תוך תמיכה בנפילות.

שאלות ליהודה:

1. האם לעצור האלגוריתם מתישהו או שזה איטרטיבי לתמיד?  
   זה לא אמור להפסיק אלא אם כן אנחנו אומרים לו שהמשחק מפסיק כשמגיעים ל-score מסוים.
2. איך כדאי לייצג את רשימת הקוצים כקלט לרשת?  
   לפי מה שהבנתי ממנו, אפשר לייצג את זה כרשימת מספרים שמייצגת מיקום. בגלל שזה תמיד במקומות קבועים אז אפשר באמת לייצג את זה בצורה יותר נוחה של רשימת בוליאנים/ביטים. בהינתן מיקום של ביט ספציפי אפשר להתייחס אליו כאל מיקום בציר Y ולכן זה שקול.
3. מה קורה אם השרת עצמו נופל ולא אחד מ4 ה"עבדים"?  
   ניתן להניח שהשרת (שמריץ את הגרפיקה) לא נופל. רק 4 העבדים יכולים ליפול.

שכבות הפרויקט



הפרויקט שלנו מורכב מ-4 שכבות:

* + - 1. השכבה הראשונה היא Graphics. שכבה זו תנהל את הגרפיקה שהמשתמש מתנהל איתה (GUI). קיים רק מופע אחד של המודול הזה שירוץ אחד כזה זה ירוץ על המחשב הראשי והוא יהיה אחראי על יצירה ואחזקה של 4 תהליכים של PC.
      2. השכבה השנייה היא PC. כל מופע מהשכבה הזו מקבל הוראות מהגרפיקה שמעליו על מנת ליצור ולתחזק כמות גדולה של ציפורים.
      3. השכבה השלישית היא ציפור.
      4. השכבה הרביעית היא רשת הנוירונים של הציפור.
      5. השכבה החמישית היא נוירון בודד.

