



K.L.Y.R

strategic variation on the classic rock-paper-scissors game

++++

++++



מבוא

K.L.Y.R הוא משחק אסטרטגי המבוסס על אבן-נייר-מספריים המיושם בפייתון. במצגת זו נספק סקירה כללית של יישום המשחק, הכללים, האלגוריתמים, פונקציות ההערכה, שפת התכנות והאתגרים שעמדו בפנינו במהלך הפיתוח.



++++

מימוש

מימשנו את המשחק K.L.Y.R בשפת Python, יצרנו מספר מודולים שונים המיישמים את הלוגיקה של המשחק, תנועת השחקנים, ניהול מצבי משחק וכמובן שחקן מחשב המבצע את צעדיו לאחר ניתוח המצב הקיים ושימוש באלגוריתם IDA* הכולל שתי פונקציות הערכה: מרחק מנהטן ופונקציה נוספת שבודקת לפני כל תור מספר מצבי משחק ספציפיים המרמזים על איום נוכחי ו/או חשש להפסד או ניצחון במשחק.

כללי המשחק

++++

מטרה:

ללכוד את דגל היריב על ידי הזזת חיילים בזירת הקרב.

מהלך המשחק:

- כל שחקן בתורו בוחר להזיז חייל, השחקן הנבחר יסומן בצבע ירוק.
- תוצאת הבחירה (אבן מנצחת מספריים, מספריים מנצחים נייר, נייר מנצח אבן) קובעת את תנועת החיילים. אם בתוצאת הבחירה יש את אותו הכלי, המערכת תיצור כלי רנדומלי לכל חייל שאיתו ילחמו עד שמוכרעת התוצאה.
- באופן רנדומלי בתחילת המשחק, המערכת תבחר משבצת ממרכז הלוח (לא באיזור שבו נמצאים החיילים של השחקנים) שתהיה בה חומה גלוייה.
- הפסד בעימות בין שני חיילים – מוציא את החייל המפסיד מהלוח והמנצח עובר למשבצתו.
- חייל לא יכול לדרוך על חומה ולעבור דרכה.

כללי המשחק

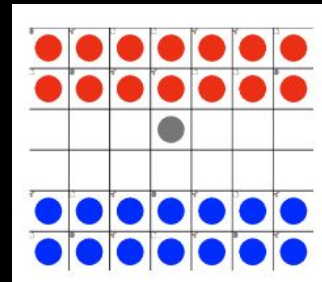
++++

ניצחון: המשחק ממשיך עד שאחד השחקנים מצליח להזיז את חייליו וללכוד את דגל היריב.

כוח מיוחד: לכל שחקן בכל סיטואציה יכול לרנדר את הכלים של החיילים שלו על ידי בחירת מקש "החלף כלים" – החלפת הכלים מתבצעת באופן רנדומלי לחלוטין-כולל הדגל.

זירת הקרב:

- זירת הקרב מורכבת מ 6 שורות ו 7 עמודות.
- לכל שחקן יש 2 שורות עם 14 חיילים – 7 חיילים בכל שורה.
- החיילים ממוקמים בתחילת המשחק בשורות בשתי השורות הראשונות, המחשב בשורות העליונות והשחקן בשורות התחתונות
- לדוגמא:



שפת תכנות - פייתון

בחרנו בשפת Python עבור מימוש המשחק K.L.Y.R בשל הפשטות, הקריאות שלה והספריות הנרחבות שמסייעות בפיתוח משחקים (pygame) ובמימוש בינה מלאכותית.

אלגוריתמים

המשחק משתמש במספר אלגוריתמים לקביעת המהלכים הטובים ביותר.

1. אלגוריתם IDA*

2. Random Choice

אלגוריתמים - IDA*

מטרה: חיפוש מהלך אופטימלי במשחק מבוסס גריד כמו שלנו.

תיאור: IDA* משתמש בהיוריסטיקה מבוססת מרחק מנהטן (Manhattan Distance) לחישוב המרחק מהשחקן לדגל של היריב.

יתרונות:

- שיטה אפקטיבית לחישוב מרחקים בגריד.
- שליטה בזיכרון על ידי אי שמירת כל הצמתים.
- מותאם במיוחד למשחקים מבוססי גריד.

אלגוריתמים - Random Choice

מטרה: בחירת כלי חדש במקרה של שוויון בקרב.

תיאור: כאשר שני השחקנים משתמשים באותו כלי, מתבצעת בחירה אקראית של כלים חדשים.

יתרונות:

- הוספת אלמנט של אקראיות ושונות למשחק.
- מגביר את האתגר והעניין.

פונקציות הערכה

פונקציות ההערכה שבחרנו למשחק מעריכות את המהלכים הפוטנציאליים ואת תוצאותיהם. פונקציות אלו מתייחסות למצב הנוכחי של המשחק, למצבים אפשריים בעתיד ולסיכוי ההצלחה של כל מהלך.

הערכת מהלך:

- **פונקציה:** `evaluate_move(old_pos, new_pos, game_logic_)`
- **מטרה:** הערכת המהלך פוטנציאלי של השחקן הממוחשב.
- **מרכיבים:**
 - בדיקת אם המהלך מוביל לניצחון.
 - חישוב המרחק לדגל של היריב באמצעות IDA*.
- **תיאור:** הפונקציה מחשבת את הניקוד של מהלך לפי המרחק לדגל ולפי הסיכוי לנצח בקרב.

פונקציות הערכה

פונקציית מרחק לדגל:

- **פונקציה:** `distance_flag(start_pos, flag_pos, game_logic)`
- **מטרה:** חישוב המרחק מהשחקן לדגל של היריב באמצעות IDA*.
- **מרכיבים:**
 - היוריסטיקה מבוססת מרחק מנהטן.
 - חיפוש בעומק חוזר עם סף (bound).
- **תיאור:** הפונקציה מחשבת את המרחק לדגל ומחזירה את המרחק השלילי כפונקציית הערכה למהלך.

פונקציות הערכה

פונקציית ניצחון בקרב:


































- **פונקציה:** `is_winner(old_pos, new_pos, game_logic)`
- **מטרה:** לבדוק אם מהלך מסוים מוביל לניצחון בקרב.
- **מרכיבים:**
 - השוואת הכלי של השחקן הממוחשב לכלי של היריב.
 - קביעת ניצחון או הפסד בהתאם לכללים של Rock-Paper-Scissors.
- **תיאור:** הפונקציה מחזירה ניקוד חיובי אם המחשב מנצח בקרב וניקוד שלילי אם הוא מפסיד.

אתגרים

התמודדנו עם מספר אתגרים במהלך פיתוח המשחק:

- מימוש אלגוריתמים יעילים לבינה המלאכותית.
- ניהול מצב המשחק והבטחת עקביות.
- אופטימיזציה של הלוגיקה של המשחק לשיפור הביצועים.
- טיפול בתרחישי משחק בלתי צפויים ובמקרי קצה.

מצב פתיחה: כחול-שחקן, אדום-מחשב, אפור-חומה

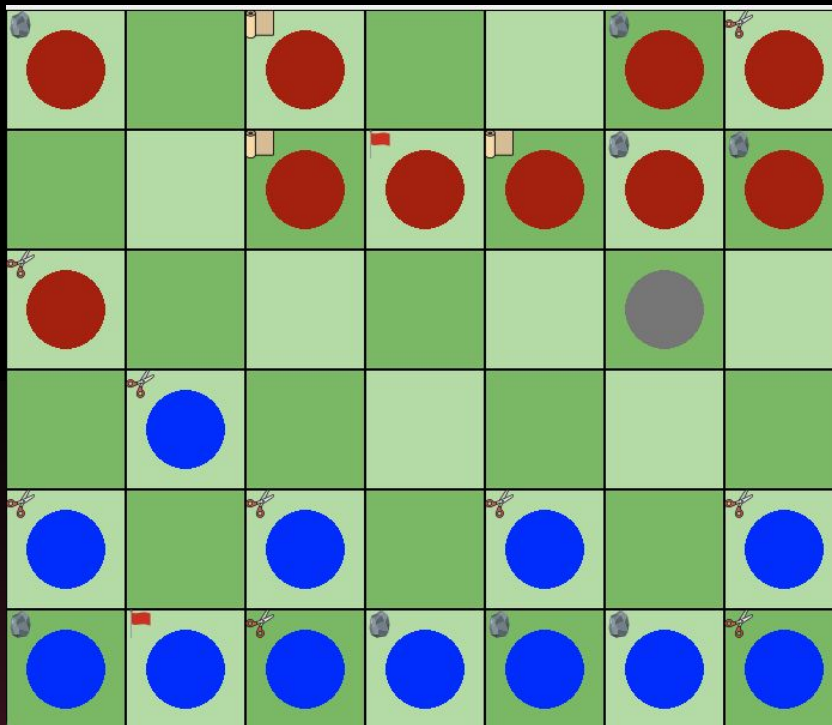
Player's Turn

Player Soldiers: 14

Computer Soldiers: 14

Shuffle

מהלך המשחק




























Player's Turn

Player Soldiers: 12

Computer Soldiers: 10

Shuffle

תור השחקן, החייל שנבחר מסומן במסגרת לבן

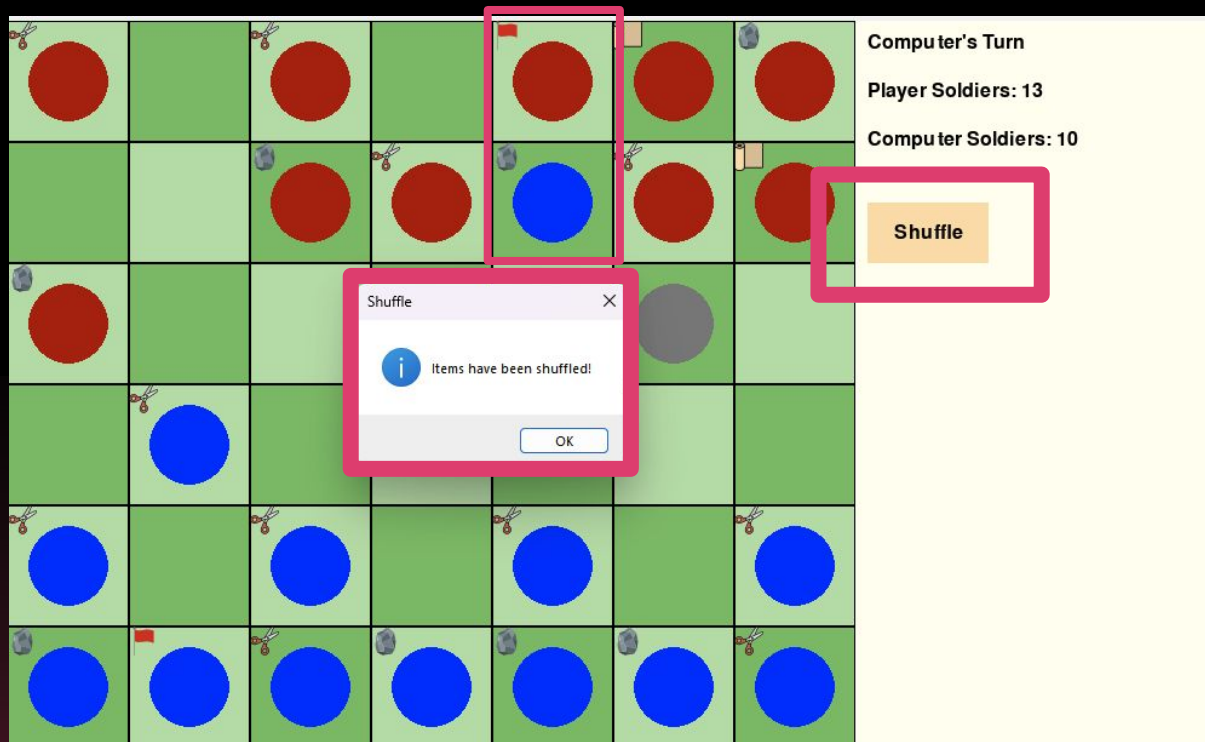
Player's Turn

Player Soldiers: 13

Computer Soldiers: 11

Shuffle

המחשב מבצע ערבוב לאחר איום על הדגל

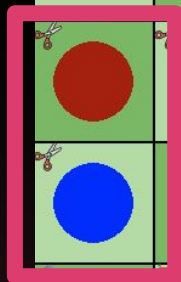


צילומי מסך של המשחק - המשך

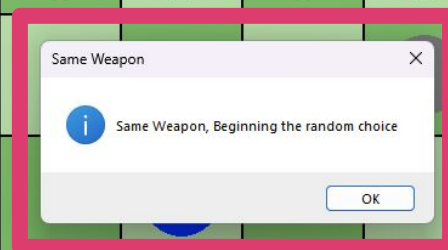


3

1



2



Computer's Turn

Player Soldiers: 12

Computer Soldiers: 10

Shuffle

צילומי מסך של המשחק - ניצחון השחקן

