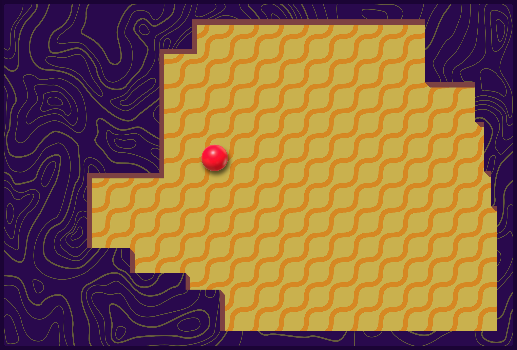
**Occupy game**

אלגוריתם 1 – לבדוק אם נקודה (השחקן) בתוך השטח או לא (שטח אויב)

הקלט – הנקודה, הגבול של השטח- רשימת קווים אנכיים, ורשימת קווים אופקיים

שלב 1 אם מעברים קו אופקי מהנקודה כלפי צד ימין (לכיוון X חיובי) , כמה קוים אנכים של השטח הקו יחצה.

נסמן את הנקודה בPx,Py)) לרשימת הקוים נקרא ver\_lines , ניצור משתנה crosses=0 עבור כמות הקווים שנחצו

עבור כל קו בver\_lines :

נסמן את שתי הנקודות של הקו ב A, B

Px < Ax

min(Ay,By) < Py < max(Ay,By)

אם ו-

תוסיף 1 ל crosses

אז

|  |  |  |
| --- | --- | --- |
|  | A |  |
|  | B |  |
|  |  |  |

(Px, Py)

**הקו האופקי**

**קודקוד חיצוני**

**הנקודה**

שלב 2 אם ערך הY של הנקודה שווה לאחד הקווים האופקיים, אז צריך לספור דרך כמה קודקודים חיצוניים הקו האופקי (שמתחתנו כביכול בשלב 1) עובר

קודקוד חיצוני, זה קודקוד שהשטח נמצא בתוך זווית ה90 של הקודקוד, ולא ה270.

|  |  |  |
| --- | --- | --- |
|  | **קודקוד חיצוני** |  |
|  |  |  |
|  | **לא קודקוד חיצוני** |  |

**הקו האופקי**

**הנקודה**

**קודקוד חיצוני**

*נמצא את הקו האנכי השמאלי ביותר (ערך* X *הקטן ביותר)*

נמצא מה מהנקודות ערך הY שלה הוא נמוך יותר וממנה נתחיל לכיוון הנקודה השנייה באותו קו

עבור כל נקודה נבדוק מה הכיוון ביחס לנקודה הבאה. (למעלה, ימינה, למטה, שמאלה) ונשמור במשתנה זמני.

בכל פעם ששני קוים יוצר רבע סיבוב **בכיוון השעון** אז נשמור את נקודת הקודקוד בין שני הקוים במשתנה נקודות חיצוניות

לבסוף נבדוק כמה מהנקודות החיצוניות בעלות אותו ערךY כמו של הנקודה. (כמה נקודות הקו האופקי חוצה)

שלב 3

נחבר את התוצאות של שלב1 ו2 (crosses)

אם יצא מספר זוגי – הנקודה מחוץ לשטח

אם יצא משפר אי זוגי – הנקודה בתוך השטח

פלט – if \_inside = True/False