



משחק הפוקימונים

במשחק הפוקימונים עליכם לנהל קבוצה של סוכנים שמטרתם לתפוס כמה שיותר פוקימונים לפני שייגמר הזמן. ככל שתתפסו יותר פוקימונים כך תצברו יותר נקודות. המשחק מתנהל על לוח משחק משתנה, בהתאם לשלב בו בחרתם לשחק (ישנם 24 שלבים שונים, בדרגת קושי עולה). התנועה בלוח היא על גבי גרף מכון ממושקל, עליו פזורים בצורה רנדומלית פוקימונים, אותם הסוכנים יכולים לתפוס ע"י מעבר בסמיכות אליהם על גבי הגרף.

ניהול המשחק יתנהל בשלבים הבאים:

1. בחירת שלב.
2. מיקום הסוכנים לקודקודי המוצא שלהם.
3. התחלת המשחק.
4. ניהול מתמשך של הכוונת הסוכנים עד שהמשחק יגמר.

את המשחק ינהל שרת, איתו ניתן לתקשר ע"י ה-API הבא:

Modifier and Type	Method	Description
boolean	<code>addAgent(int start_node)</code>	This method allows the user to add & locate the agents, all should be located in order to start a game.
long	<code>chooseNextEdge(int id, int next_node)</code>	This method is the main logical functionality, allows the client algorithm to direct each agent to the "next" edge.
java.lang.String	<code>getAgents()</code>	Returns a JSON string, representing all the Agents.
java.lang.String	<code>getGraph()</code>	Returns a JSON representation of graph as a JSON String.
default directed_weighted_graph	<code>getJava_Graph_Not_to_be_used()</code>	Returns an interface to the graph (should NOT be used) for final version - for testing only.
java.lang.String	<code>getPokemons()</code>	Returns a JSON string, representing all Pokemons (fixed bonus coin).
boolean	<code>isRunning()</code>	Returns the current status of the game (true: is running, false: NOT running).
boolean	<code>login(long id)</code>	Performs a login - so the results of the game will be stored in the data-base after the game, requires Internet connection.
java.lang.String	<code>move()</code>	moves all the agents along each edge, if the agent is on the node (nothing is done - requires to chooseNextEdge(int id, int next_node)
long	<code>startGame()</code>	Start a new game
long	<code>stopGame()</code>	Stops the game, after this method the <code>isRunning()</code> will return false
long	<code>timeToEnd()</code>	return the number of milli-seconds till the game is over

פירוט השלבים:

1. **בחירת שלב:** כדי לקבל רפרנס לשרת המשחק יש להפעיל את המתודה הבאה יחד עם שלב

המשחק בו רוצים לשחק:

```
game_service game = Game_Server_Ex2.getServer(level_number);
```

בעקבות פעולה זו השרת בונה בתוכו את לוח המשחק בהתאם לשלב שהתבקש. לאחר מכן אפשר להשיג מידע בנוגע ללוח, מידע כמו כמה פוקימונים וסוכנים יש באותו השלב. כמו כן אפשר לדעת איך הגרף של הלוח נראה (מידע על הצלעות והקודקודים), ולדעת איפה הפוקימונים ממוקמים, מאיזה סוג הם וכמה הם שווים. כדי לקבל מידע זה יש להשתמש במתודות ה-`get` השונות שמציע ה-API של השרת כמפורט בתמונה למעלה. כל המידע מהשרת מגיע בפורמט JSON. שימוש לב, כאשר ישנם שני קודקודים עם id 4 ו-9 למשל, המחוברים ביניהם בצורה דו כיוונית, כלומר קימות ביניהם שתי צלעות, מ-4 ל-9 ולהפך, ואם נמצא פוקימון בין קודקודים אלו, הפוקימון משתייך אך ורק לאחת מהן. כלומר שמעבר על הצלע השניה לא תאפשר תפיסה של הפוקימון, אלא אך ורק מעבר על הצלע הנכונה בין השתיים. כיצד ניתן לדעת לאיזו צלע הפוקימון משויך? ע"י הערך שב-type של אותו הפוקימון. type עם ערך של "1" אומר שהפוקימון נמצא על הצלע מהקודקוד הנמוך יותר (מבחינת ערך ה-id של הקודקוד) אל עבר הקודקוד הגבוה יותר. בדוגמה שלנו, הפוקימון

- יהיה על הצלע מ-4 ל-9. ואם ב-type של הפוקימון היה הערך "1-" אז הפוקימון ממוקם על הצלע היוצאת מהקודקוד הגבוה יותר, אל עבר הקודקוד הנמוך יותר. כלומר הצלע מ-9 ל-4.
2. **מיקום הסוכנים לקודקודי המוצא שלהם:** בזכות המידע שיש לכם מהשלב הקודם, אתם יכולים לקבל החלטה אסטרטגית היכן למקם את הסוכנים שלכם. המיקום מתבצע ע"י שימוש במתודה `addAgent(int startNode)`. שימו לב שיש לקרוא למתודה זו כמספר הסוכנים שצריכים להיות פעילים בשלב בו בחרתם לשחק. קריאה עבור הצבה של כל סוכן.
3. **התחלת המשחק:** התחלת המשחק מתבצעת ע"י שימוש במתודה `startGame()`. מרגע זה השעון מתקתק לאחור. ברגע שייגמר הזמן, המשחק ייגמר ולא ניתן לצבור עוד נקודות. ניתן לבדוק אם המשחק עוד פעיל או לבדוק כמה זמן נותר ע"י שימוש בפונקציות המתאימות שמציע ה-API.
4. **ניהול מתמשך של הכוונת הסוכנים עד שהמשחק יגמר:** לכל סוכן יש ערכי מיקום נוכחי ויעד אליו הסוכן מתכנן להגיע, המוגדרים כ- `src` ו- `dest`, שיכילו ערכי `id` של קודקודים על הגרף. שימו לב שה- `dest` יכול להיות אך ורק קודקוד שכן לקודקוד ה- `src`. ניתן לראות ערכים אלו, כמו הגדרות נוספות של הסוכנים (כגון מהירות וכד') ע"י קריאה למתודה `getAgents()` שתספק את המידע הרלוונטי בפורמט JSON.
- ניתן לומר לשרת המשחק להזיז את הסוכנים על גבי הגרף ע"י המתודה `move()`. ברגע שמתודה זו תופעל, השרת יזיז את כל אחד מהסוכנים במשחק, על גבי הגרף, מהיכן שהסוכן ממוקם כרגע, אל עבר קודקוד היעד שמוגדר לו בערך ה- `dest`, לפי המהירות המוגדרת לאותו סוכן ובהתאם לזמן שעבר מהפעם האחרונה בה הופעלה המתודה `move()`. לדוגמה, אם סוכן זז במהירות של 1 יחידת מרחק לשניה, ועברו 3 שניות מהפעם האחרונה שביקשתם שהשרת יזיז את הסוכן, אז כעת הבקשה תגרום לסוכן לזוז מרחק של 3 יחידות מרחק. עליכם לעדכן את `dest` לפי אסטרטגיית התנועה שתבחרו. התנועה של הסוכן תכונן לפי ה- `dest` שמוגדר לו במהלך המשחק. נשים לב שבזמן שסוכן נמצא על צלע, לא ניתן לעדכן את ה- `dest` שלו, כיוון שהוא כרגע בדרך אל עבר `dest` קודם כלשהו. ניתן לעדכן מחדש את `dest` של סוכן רק אם ה- `dest` הנוכחי הוא 1- (שזה אומר שהסוכן כרגע מוגדר כחסר יעד לתנועות הבאות), וזה יקרה כאשר הסוכן סיים להגיע לקודקוד יעד כלשהו. שימו לב לא לדלג בטעות מעל פוקימונים ובכך לפספס אותם (כדי ללכוד פוקימון עליכם לעבור בסמיכות קרובה אליו), פעולה שיכול לקרות אם ביצעתם קריאות `move` עם מרווחי זמן גדולים מידי, מה שיגרום לצעדים גדולים מידי בתנועות הסוכן על גבי הצלע. אחריי שפוקימון נתפס - אחר יופיע במקומו במיקום רנדומלי על גבי הגרף.

שימו לב! כל השימוש שתואר כאן, אינו כלל ממשק גרפי כלל. הבניה של הגרף בשרת והתנועה של הסוכנים על גביו היא לוגית בלבד. לכן, מלבד הניהול הלוגי של המשחק, עליכם לדאוג להציג את כל ההתרחשויות של מהלך המשחק בצורה ויזואלית. החל מתחילת המשחק ובמהלכו, כך שיהיה ניתן לצפות בהתקדמות של הסוכנים שלכם על הגרף ובתפיסת הפוקימונים שלהם. וכמובן לענות לכל הדרישות שפורטו במסמך המטלה הרשמי.

אחריי שהקוד והאלגוריתם שלכם פועלים – הוסיפו בקוד ביצוע של `login` עם ה- `ת.ז.` לאחר שלב מספר 1, על מנת שהתוצאות של ההרצות שלכם ישמרו. נסו להגיע לניקוד גבוהה ככל הניתן. שימו לב שהניקוד מושפע מכמות הפוקימונים שתפסתם (ככל שתפסתם יותר, כך הניקוד יהיה גבוה יותר) ומכמות הפעמים בהם הפעלתם את המתודה `move()` (ככל שקראתם למתודה יותר, כך הניקוד שלכם ירד). טיפ: אל תחסכו בקריאות למתודה `move()` בתור התחלה. אחריי שהמשחק שלכם פועל יוציב תתחילו לחשוב איך אפשר להפחית את הקריאות `move()` לשרת ובכך לשפר את הניקוד שלכם.

היעזרו (אך ממש לא חובה) במחלקות לדוגמה שקיבלתם עם המטלה. הן מדגימות מהלך פעיל של משחק וצורת תקשורת עם השרת. כמו כן, יש בהן דוגמה של שימוש בממשק גרפי והצגה של המשחק. כל הקוד הזה הוא כדי לשמש דוגמה ולתת עזרה והכוונה, אך אין חובה להשתמש בו או לראות בו הנחיה מחייבת, כזו או אחרת, לצורת מימוש המטלה.

בהצלחה!