

פרויקט בבינה מלאכותית 236502 - Tower Defense

תומר לוי stlevy@t2.technion.ac.il

19 באפריל 2014

1 תיאור הבעיה - משחק Tower Defense

Tower Defense (או TD) הוא ז'אנר משחקי מחשב מסוג אסטרטגיה בזמן אמת. מטרת המשחק היא לנסות לעצור את מעברן של דמויות עוינות מצד אחד של מפת המשחק למשנהו על ידי בניית מגדלים שירו בדמויות. למבנים שונים יש יכולות שונות, גם בהתאם לאויבים השונים ויש להם עלויות שונות. הריגת דמות אויב מביאה להרווחת כסף או נקודות, אשר בתורם משמשים את המשתמש בקניית מגדלים נוספים, או בשדרוג הקיימים. **הבחירה בסוג המגדלים ומיקומם במפה היא חלק מהאסטרטגיה החיונית במשחק.** משחקים רבים, מכילים דמויות עוינות העוברות במבוך מוגדר מראש ואילו משחקים אחרים, מכילים מפה ריקה ומאפשרים לאויבים לנוע עליה בצורה חופשית, ובכך מאלצים את השחקן להשתמש במגדלים כדי לבנות את המבוך. ישנן גם גרסאות אשר הן הכלאה של שני סוגים האלה. [[מתוך ויקיפדיה]]
הפרויקט יעסוק בכתיבת "שחקן" למשחק TD , שהמטרה שלו היא לנצח - בעצם לגרום לכך שכמה שפחות אויבים יגיעו לצד השני של המפה. לעתים נתייחס במסמך למפת המשחק כ"מבוך", אם כי זה לא מבוך במובן הקלאסי אלא מסלול (יחיד) מתחילת המסך לסיום.

2 תיאור דרך הפתרון

כיוון שאין לנו במשחק שני שחקנים (כי ה"אויב" הוא בעצם מדיניות שחרור אויבים) הבעיה היא לא בעיית משחק קלאסית ($minmax$), ואנו נפתור אותה בדרך אחרת. כמו שהודגש בתיאור משחקי TD , מה שנעשה בפרויקט יהיה

- בחירת מיקומי המגדלים בהינתן כמות כסף, ומחיר כל מגדל.
- ניתן זמן קבוע בתחילת המשחק - לפני שמתחיל הגל הראשון
- בזמן המשחק - למשך זמן קבוע (או משתנה - בהתאם לאסטרטגיה) באינטרוולים קבועים.
- (אולי בשלב מתקדם) בחירת סוגי מגדלים שונים בהינתן המחיר שלהם והכסף הנתון.
- (אולי בשלב מתקדם) בחירת אסטרטגיה לכל מגדל - איזה אויבים (מפלצות) שנמצאת ברדיוס שלו כדאי לו לתקוף.

3 תיאור המערכת לפתרון הבעיה

המערכת תכלול שני אלמנטים עיקריים:

1. ה- $Server$ יהיה אחראי על:

- דיסקרטיזציה של הזמן
- זימון המפלצות מצד אחד של המפה
- התקדמות המפלצות במבוך
- פגיעה של המגדלים במפלצות
- הורדת חיים למפלצות
- מוות של מפלצות
- תגמול כספי לשחקן כשמפלצת מתה

2. השחקן יהיה אחראי על:

- בהינתן מחיר של מגדל, וכמות כסף, בנה קבוצת מגדלים בצורה אידאלית על מנת למנוע מהמפלצות להגיע לצד השני של המפה.
- (אולי בשלב מתקדם) החלט לכל בניין על איזה מפלצת הכי כדאי לו לירות

4 תיאור הניסויים

4.1 פרמטרים ויוריסטיקות לבדיקה

- נבדוק את השפעת בחירת יוריסטיקות שונות על תוצאות המשחק. יוריסטיקות אפשריות:
 - חפיפה מקסימלית בין המגדלים השונים
 - חפיפה מינימלית בין המגדלים
 - פיזור אקראי למחצה (אקראי אבל צמוד למסלול מבוך)
- אלגוריתמים שונים. אנו נעשה חיפוש לוקלי וננסה להתגבר על מקדם הסיעוף הגדול. דרכים אפשריות:
 - חיפוש אלומה
 - first choice
 - סידור האופרטורים (למשל, מפרצים קודם)

4.2 מזדים להערכת ביצועים

- מספר המפלצות שהצליחו להגיע לצד השני של המבוך
- זמן ממוצע להריגת מפלצת (כאשר אם היא לא מתה נספור את זה כזמן שלקח לה להגיע לצד השני)

5 שונות

5.1 לוחות זמנים משוערכים

- 15 למאי - משחק בסיסי עם סוג מגדלים אחד, סוג מפלצות אחד גרפיקה בסיסית, ויכולת ניתוח תוצאות.
- 1 יוני - subset selection ע"י חיפוש מקומי, כולל השוואה עם אלגוריתמים שונים
- 30 לספטמבר - עבודה על פיצ'רים מתקדמים אלגוריתמים שונים, והרחבת המשחק (למשל: מפלצות שונות, סוגי מגדלים שונים, שדרוג מגדלים)

5.2 שפת תכנות ספריות וקוד קיים

שפת התכנות תהיה Java, אני מתכוון להשתמש במדריך שמצאתי ב-youtube על מנת לכתוב משחק בסיסי, ולהשתמש בספריות של Aima לבינה מלאכותית ב-Java.