

## מטלה שבועית – בניית עולם ואלגוריתמים בשני ממדים

המשחק מהשיעור נמצא כאן: <https://github.com/gamedev-at-ariel/05-tilemap-pathfinding> ניתן להשתמש במשחק זה ולשנות אותו לפי ההוראות. לחלופין, אפשר ליצור משחק TileMap חדש לצורך המטלה. בצעו שני שינויים לבחירתכם, אחד מכל רשימה:

### 1. שינויים קלים

בחרו שינוי אחד מהרשימה הבאה:

- א. הוסיפו במפה סירה – שאם השחקן לוקח אותה, הוא יכול לשוט במים; עז – שאם השחקן לוקח אותה, הוא יכול לטפס על הרים; ומכוש – שאם השחקן לוקח אותו, הוא יכול לחצוב בהרים ולהפוך אותם לדשא.
- ב. שנו את הסריג ממלבני למשושה (hexagonal). שנו בהתאם את האלגוריתמים (מציאת מסלול + יצירה אוטומטית).
- ג. השתמשו במכונת המצבים כדי להוסיף לדמויות במשחק התנהגויות מעניינות יותר לפי בחירתכם.
- ד. באתר זה <https://www.myabandonware.com> ישנם משחקים רבים שהמפתחים נטשו אותם. בחרו משחק אחד שנראה לכם מעניין, ושיש בו עולם דו-ממדי במבט על, וציירו לפחות רמה אחת מתוכו במפת-האריחים של יוניטי.

### 2. אלגוריתמים

בחרו שינוי אחד מהרשימה הבאה:

- א. במקום אלגוריתם BFS, ממשו את אלגוריתם A\* (A-star) למציאת מסלול בגרף עם משקלים (אם אתם לא מכירים אותו, ממשו במקומו את דייקסטרה).
- ממשו את האלגוריתם קודם-כל בצורה מופשטת כמו שראינו בשיעור, ואז שלבו אותו במשחק ה-TileMap.
- שנו את המשחק כך שמהירות ההליכה של השחקן תהיה תלויה באריחים שהוא צועד עליהם, למשל: מהירות הליכה על גבעות או בתוך שיחים איטית יותר מאשר על דשא. השתמשו באלגוריתם שפיתחתם כדי למצוא את המסלול המהיר ביותר במקום המסלול הקצר ביותר.
- ב. כשיוצרים מפה אקראית ושמים בתוכה את השחקן, חשוב לוודא שהוא לא "תקוע" בתוך הר או מערה קטנה שאי-אפשר לצאת ממנה. כתבו קוד, המייצר מפה אקראית של מערה, מציב את השחקן בנקודה אקראית, ובודק שאפשר להגיע ממנה ל-100 אריחים שונים לפחות (רמז: היעזרו באלגוריתם BFS). אם אי-אפשר להגיע ל-100 אריחים שונים לפחות, המשחק שם את השחקן בנקודה אקראית אחרת ובודק שוב, עד שהשחקן מגיע לנקודה המקיימת את התנאי.
- ג. קראו על אלגוריתם "Binary Space Partitioning" ליצירה אוטומטית של חדרים (למשל כאן), והתאימו אותו למשחק TileMap, במקום ה-CaveGenerator שראינו בשיעור. איזה אלגוריתם נותן תוצאות מעניינות יותר?
- ד. שנו את המשחק כך שהשחקן יוכל לראות רק אריחים שאינם מוסתרים מאחרי הרים (כלומר, אם יש הר בין השחקן לבין אריח כלשהו על המפה, אז השחקן לא יראה את האריח הזה, אלא יראה ריבוע שחור).
- ה. בשיעור ראינו אלגוריתם המאפשר לאויבים לרדוף אחרי השחקן בצורה יעילה. המציאו (או מיצאו באינטרנט) אלגוריתם המאפשר לאויבים לברוח מהשחקן בצורה יעילה - כך שלשחקן יהיה קשה לתפוס אותם. בדקו את האלגוריתם שלכם על תרחישים שונים. וודאו שהאויבים לא "נתקעים" בפינות המאפשרות לשחקן לתפוס אותם בקלות.

הגשה כרגיל: המשחק באיץ' והקוד בגיטהאב, עם הסברים וקישורים הדדיים ביניהם.

