

## מטלה שבועית – מבוא ליוניטי

השאלות במטלה זו נועדו לעזור לכם להרחיב את הידע שלכם בנושאים שנלמדו בהרצאה.

- יש להעלות את קוד-המקור לגיטהאב של ה"אירגון" (organization) שפתחתם עבור הצוות. היעזרו בקובץ ([4-github-and-unity](#)) שנמצא בתיקיה של השבוע הנוכחי בקורס, כדי לעלות את הפרויקט שבניתם בצורה יעילה ובלי קבצים מיותרים.
- יש להעלות את המשחקים עצמם לחשבון של הצוות, שתפתחו באתר [itch.io](#).
- יש לשים קישור מקובץ ה-Readme.md בגיטהאב למשחק ב [itch.io](#), וכן קישור מהערות המשחק ב [itch.io](#) לגיטהאב של המשחק, כך שמי שמוצא את המשחק שלכם באחד האתרים יוכל להגיע לשני.
- יש להגיש בתיבת ההגשה במודל שני קישורים: קישור לגיטהאב וקישור ל-[itch.io](#).
- יש לכתוב קוד קריא, מתועד, בלי מספרי קסם וכו' – כמו שלמדתם בקורסים הקודמים.

### א. רכיבים

1. בהרצאה בנינו רכיב בשם **Mover** המזיז עצם-משחק בכיוון קבוע ובמהירות קבועה. בנו רכיב חדש בשם **Oscillator**, המזיז עצם-משחק בתנועת מטוטלת, למשל: ימינה עד גבול מסויים, ואז שמאלה עד גבול מסויים, וחוזר חלילה.

- מרכז התנועה צריך להיות המקום הנוכחי של העצם בסצינה (למשל אם אני שם אותו במיקום 2,3, אז העצם יזוז ימינה ושמאלה סביב נקודה זו).
  - מהירות העצם צריכה להשתנות בהדרגה, כמו במציאות (העצם נע ימינה, מאט בהדרגה, נעצר, נע שמאלה, מאיץ בהדרגה, מאט בהדרגה, נעצר וכו'). רמז: השתמשו בפונקציה טריגונומטרית.
  - הגדירו את הרכיב באופן כמה שיותר כללי – עם כמה שיותר משתנים המאפשרים לשלוט בכיוון התנועה וגודלה מתוך יוניטי.
  - הדגימו את הרכיב שלכם על כמה עצמים, עם ערכים שונים לפרמטרים.
2. בנו רכיב בשם **Rotator**, המסובב עצם-משחק במהירות נתונה סביב עצמו (באיזה ציר-סיבוב צריך להשתמש?).
3. בנו עצם-משחק בצורת כדור, הגדל בהדרגה וקטן בהדרגה, כמו לב פועם.
4. בנו רכיב המאפשר לשחקן להסתיר את העצם שלו (העצם שהרכיב שייך אליו) בלחיצת כפתור, ולהציג אותו מחדש בלחיצת כפתור. רמז: השתמשו ברכיב האחראי על הצגה של העצם.
- הגשה:** כדי להדגים את כל הרכיבים בסעיף זה, אפשר ליצור פרויקט אחד עם סצינה אחת, ולשים בה עצמים שונים המדגימים את הרכיבים השונים.
- מומלץ לכתוב ליד כל עצם מה בדיוק הוא מדגים. לשם כך אפשר להוסיף טקסט לסצינה באופן הבא:
- הוסיפו עצם-משחק חדש ריק (**Create Empty**)
  - הוסיפו לו רכיב חדש (**Add Component**) מסוג **Text Mesh Pro – Text** (\*).
  - שנו את מאפייני הרכיב כדי לקבוע את הגודל, הצבע, הכיוון והשפה. אפשר לכתוב בעברית או אנגלית.

(\*) אם אתם לא מוצאים את *Text Mesh Pro*, ייתכן שאתם צריכים להתקין אותו באופן חד-פעמי. לשם כך היכנסו ל *Window* ואז ל *Package Manager*, חפשו את החבילה *Text Mesh Pro* והתקינו אותה.

## ב. מצלמות ונקודות-מבט

1. קיראו כאן: <https://docs.unity3d.com/Manual/class-Camera.html> על מצלמות ביוניטי.
2. בנו "משחק" שבו השחקן יכול לזוז בעולם דו-ממדי בעזרת מקשי החצים, ובמקביל הוא רואה את המיקום שלו על מפה קטנה (mini-map) בצד ימין למעלה.
3. בנו "משחק" לשני שחקנים, כל שחקן יכול להזיז דמות אחת, וכל שחקן רואה את העולם מנקודת-המבט של הדמות שלו, על חצי מסך, כמו במשחק העתיק "מרגל נגד מרגל": <https://youtu.be/Yu2e866bEcM> (למידע נוסף על שימוש בשתי מצלמות ראו כאן <https://forum.unity.com/threads/2-cameras-running-the-same-time.44911>).

### הגשה:

- בכל השאלות אין צורך לבנות משחק מלא, אלא רק "משחק" מינימלי המדגים את פעולת המצלמה.
- בסעיף זה יהיה לכם נוח יותר לכתוב פרויקט נפרד עבור כל שאלה.
- העלו לאתר [itch.io](https://itch.io) את אחד מהפרויקטים לפי בחירתכם (2 או 3).

## ג. סי' שארפ - לימוד עצמי - הגשה אישית

קיראו בגיטהאב, בתיקיה של שבוע זה, את המדריך על הבדלים בין C# לבין Java ([5-java-vs-csharp](https://www.hackerrank.com/domains/algorithms))

כדי לתרגל את מה שלמדתם, בחרו אחת מהאפשרויות הבאות:

1. האפשרות הקלה – אתר האקראנק שאתם כבר מכירים. לכו למסלול אלגוריתמים: <https://www.hackerrank.com/domains/algorithms>
  - באפשרות זו יש לפתור 5 שאלות לבחירתכם, ברמת קושי בינונית ומעלה. בכל פתרון אפשר לבחור שפה – תבחרו C#.
  2. האפשרות המהנה יותר – אתר קודינגייס - <https://www.codingame.com> - אתר דומה להאקראנק רק עם משחקים ורכיבים דרמטיים.
  - באפשרות זו יש לפתור שלוש שאלות או יותר לבחירתכם.
  - כמוכן, אם בחרתם באפשרות זו, תארו את ההבדלים בין אתר זה לבין האקראנק: איזה רכיבים משחקיים נוספו כאן? האם הם תורמים ליעילות הלימוד? האם הם תורמים להנאה מהלימוד? האם הייתם ממליצים להשתמש באתר זה במקום האקראנק בקורס "תכנות מערכות ב", ומדוע?
- הגשה:** בסעיף זה ההגשה אישית – יש להגיש במודל קישור לפרופיל שלכם בהאקראנק / קודינגייס.
- למה?** – כדי שכולכם תדעו לתכנת ותוכלו לתרום באופן שווה לעבודת הצוות בהמשך הקורס.