



## מושגי יסוד ביוניטי

### אובייקט משחק

יוניטי היא סביבה מונחת-עצמים. המושג העיקרי ביוניטי הוא אובייקט-משחק – **GameObject**.

אובייקט-משחק יכול להיות דמות הפועלת במשחק, כגון מכונת, חללית, מפלצת וכדו'.

אבל יותר מזה, רוב האלמנטים של unity הם אובייקטים שבאים עם העורך הראשי, למשל מצלמה או תאורה.

יש אובייקטים בסיסיים שבאים עם יוניטי, למשל קובייה, צילינדר, קפסולה וכו', ואנחנו יכולים להרכיב אותם לאובייקטים מורכבים יותר, או ליצור חדשים בעצמנו.

לכל אובייקט יש אחד או יותר **רכיבים (Component)**: כל רכיב אחראי לחלק מההתנהגות של אותו אובייקט.

בסי שארפ, כל רכיב ממומש כמחלקה.

על מנת ליצור אובייקט חדש יש ללחוץ על GameObject שנמצא בתפריט הראשי, ולבחור בסוג האובייקט שאותו נרצה.

### יצירת אובייקט ריק

לצורך הדוגמה, ניצור קודם-כל אובייקט ריק – Create Empty. אנחנו רואים שגם לאובייקט ריק יש רכיב אחד והוא ה- **Transform**. זה רכיב שיש אוטומטית לכל אובייקט במשחק, והוא קובע את המיקום שלו ב-3 ממדים, הסיבוב שלו ב-3 ממדים, וההגדלה שלו (scaling) ב-3 ממדים.

כרגע, אנחנו בכלל לא רואים את האובייקט שלנו במשחק. מדוע? כי אין שום רכיב שמצייר אותו. הרכיב שמצייר אובייקט נקרא **Renderer** (המילה render משמעותה – לצייר אובייקט על המסך). אז כדי שנראה את הרכיב שייצרנו, צריך להוסיף לו **renderer**. איך נעשה את זה? בצד ימין, בחלון **Inspector**, נלחץ על "הוספת רכיב חדש", נתחיל לכתוב **renderer** ונראה כמה אפשרויות.

נבחר באפשרות **sprite renderer** – צייר ע"י "ספרייט" (תמונה). אנחנו רואים שלרכיב יש כמה שדות, ואחד מהם הוא התמונה עצמה. בשלב ראשון השדה הזה ריק – אין בו שום ערך – ולכן עדיין לא רואים את הרכיב. כדי שנוכל לראות אותו, נקח תמונה כלשהי מהמחשב שלנו (או נוריד מהאינטרנט – ניתן למצוא תמונות חנימיות במנוע החיפוש **Creative Commons Search**), ונשים אותה בתיקיית **assets**. עכשיו אפשר לגרור את קובץ התמונה מתיקיית **Assets** על הרכיב, או ללחוץ על העיגולון הקטן ליד השדה של ה **sprite** ולבחור את קובץ התמונה.

עכשיו סוף סוף רואים את האובייקט שלנו על המסך!

### יצירת קובייה

לצורך דוגמה נוספת, ניצור אובייקט מסוג קובייה (מהתפריט הראשי: **Cube -> 3D -> GameObject**). כדי שיהיה לנו קל לזהות אותה ניתן לה שם. שימו לב שלקובייה כבר יש **Renderer** (שנקרא **Mesh Renderer**), ולכן כבר רואים אותה על המסך.

הדבר הבא שוודאי נרצה לדעת הוא כיצד ניתן ליצור מניפולציות על האובייקט, כלומר לשנות את הגודל של האובייקט, מיקום סיבוב וכדו'. יש שתי דרכים עיקריות ליצירת מניפולציות על אובייקטים:




1) בחלון של האינספקטור (Inspector hierarchy) יש לנו אזור שמיועד למניפולציות על האובייקט עליו אנו עובדים- Transform.


ב- Transform יש לנו את האפשרות לשנות את המיקום של האובייקט יחסית לציר ה- $x, y, z$  של המסך הראשי. מתחת לזה יש את האפשרות לסובב את האובייקט, ומתחת לזה ניתן לשנות את גודל האובייקט בהתאם לצירים. כמובן ששימוש ב transform הוא הרבה פחות אינטואיטיבי, נשתמש בו בעיקר כדי להשוות גדלים/מיקום של אובייקטים, או, כפי שנראה בהמשך, לגרום לאובייקט לנוע בקו אחד.


2) דרך שניה ליצור מניפולציות על אובייקטים היא ע"י המניפולטורים המופיעים בצד שמאל למעלה:

Hand tool - מזיז את המצלמה בגרירה מהאובייקט שייצרנו.

Move tool - מאפשר להניע את האובייקט לפי צירי ה- $x, y, z$  

Rotate tool - מאפשר לסובב את האובייקט. 

Scale tool - מאפשר למתוח את האובייקט, המתיחה היא לפי הצירים אך דו כיוונית, כלומר גם לאזור החיוביים וגם השלילים של הצירים בו זמנית. 

Rect tool - מתיחה ריבועית. מאפשר למתוח את האובייקט לכיוון ספציפי. 

All in one - מאפשר לבצע את כל המניפולציות בו זמנית- מתיחה, סיבוב והזזה. 

טיפ: במקלדת הכפתורים Q-Y משמים כקיצורי דרך למניפולטורים השונים בהתאמה.

## יצירת חומרים לאובייקט

כל אובייקט חדש שיוצרים עשוי בד"כ מחומר דיפולטיבי. אם נרצה ליצור לאובייקט שלנו חומר או שנרצה שלפחות יהיה לצבע שונה מהאפור שמגיע כברירת מחדל, נצטרך ליצור אובייקט מסוג חומר (material) שיתאים לאובייקט עליו הוא יולבש.

נהוג ליצור תיקייה של 'חומרים' בחלון של הפרויקט (Project view) וכל חומר חדש לקרוא לו כשם האובייקט שלו, מקף תחתון והמילה mat (קיצור של material), למשל: player\_mat.

בשביל ליצור תיקייה יש ללחוץ מקש ימני בחלון ה-Project view -> create -> folder, לאחר שקראנו לה בשם נרצה להוסיף לה חומרים חדשים לרשימה, לשם כך יש ללחוץ בתיקה מקש ימני -> create -> material.

ובלחיצה על האובייקט חומר ניתן לערוך את החומר בחלון האינספקטור מצד ימין.

אחד היתרונות הגדולים של המנוע הגרפי unity הוא היכולת להזיז ולהתאים דברים באמצעות גרירה, כך שאם יצרנו את החומר בטעות מחוץ לתיקה ניתן לגרור אותו לתיקה הייעודית לחומרים, ואם נרצה להתאים בין האובייקט לחומר כל מה שנצטרך זה לגרור את החומר לתוך האובייקט- או לתוך האינספקטור של האובייקט, או לתוך האייקון של האובייקט בחלון ההיררכיה (Hierarchy view).

**שימו לב!** החומר (Material) הוא למעשה שדה של רכיב ה-Renderer (למרות שבחלון inspector הוא מוצג בנפרד). לכל אובייקט שיש לו רכיב Renderer, אפשר "להלביש" חומר.

## יצירת סקריפט

בכל משחק הדבר החשוב ביותר, לפני עיצוב השחקנים, התפאורה או מוזיקת רקע הוא יצירת מנגנון שינהיג את המשחק, כלומר מנגנון שיתן למשחק חוקיות: הדמות הזאת עושה ככה, כשמדברים איתו קורה ככה וככה, מתי נגמר המשחק, איך הוא מתחיל וכדו'. כאן בדיוק נכנס התפקיד של הקוד.



בעזרת תכונות בסי' שארפ, אנחנו יכולים ליצור סוגים חדשים של רכיבים (Component), ולהלביש אותם על העצמים שלנו, וכך לגרום להם להתנהג כמו שאנחנו רוצים.

למען הסדר הטוב נהוג ליצור תיקיה ייעודית לכל הסקריפטים של הסצנה, ולכל סקריפט לקרוא על שם האובייקט עליו הוא פועל. בשביל ליצור את הסקריפט יש ללחוץ על התיקיה הייעודית לסקריפטים מקש ימני- <create c# Script . בדיוק כמו ביצירת חומר לאובייקט גם כאן נלביש את הסקריפט לאובייקט ע"י גרירה. אם הגרירה עבדה כמו שצריך אמור להופיע הסקריפט בחלון האינספקטור של האובייקט, ובכדי לערוך את הקוד נצטרך ללחוץ מקש ימני על האייקון של הסקריפט בחלון האינספקטור- <Edit Script.

אם התקנתם את visual studio כמו שצריך זה אמור לפתוח לכם את התוכנה עם הקוד של הסקריפט.

שימו לב: כדי ליצור רכיב חדש, הקוד של הסקריפט אמור להיות בנוי כך:

- מחלקה עם שם הסקריפט שהיא יורשת ממחלקת MonoBehaviour (שהיא מחלקת האב לכל הרכיבים של unity)
- למחלקה שתי מתודות: void start() - מתודה שנקראת ראשונה עם אתחול האובייקט. המתודה נקראת פעם אחת בסקריפט;
- ו- void update() – בכל פריים (עדכון של המסך) נקראת הפונקציה והיא מייצגת את השינויים שמתחוללים באובייקט וכיצד הם נראים על המסך. למשל בשביל להזיז אובייקט במסך נצטרך לעדכן את המיקום שלו (ה- transform) בפונקציה, ואז ניתן יהיה לראות את השינוי על המסך.

בהמשך הקורס נלמד בהרחבה על בניית רכיבים ותכונותם באמצעות סקריפטים.

