



אפקטורים – רכיבי הפעלת כוח

Effector2D הוא רכיב שעדיין לא יצא לנו לדבר עליו. מטרת הרכיב היא לכוון את הכוחות שמופעלים על עצם כלשהו המתנגש בעצם עם האפקטור.

משום-מה, האפקטורים מוגדרים רק במנוע הפיסיקלי הדו-ממדי – לכן השמות של כל האפקטורים נגמרים ב 2D. אם אתם רוצים אפקטורים בעולם פיסיקלי תלת-ממדי – תצטרכו לכתוב אותם בעצמכם.

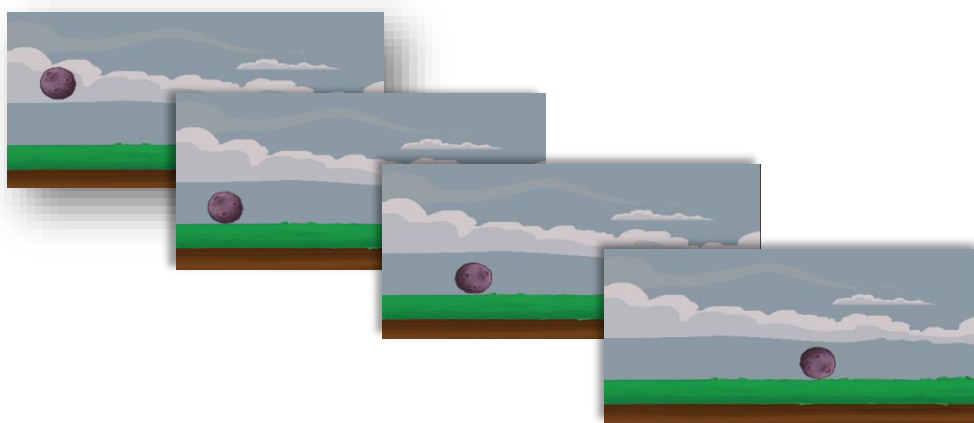
יש כמה סוגים של אפקטורים.

1. Surface Effector - אפקטור משטח

מדמה "סרט נע" – מביא כל גוף שמתנגש בקולידר שלו למהירות קבועה. לדוגמא: נפיל כדור דו-ממדי על משטח עם surface effector, הכדור ינוע לכיוון מסוים בהתאם לערך השדה "מהירות".

השדות של האובייקט:

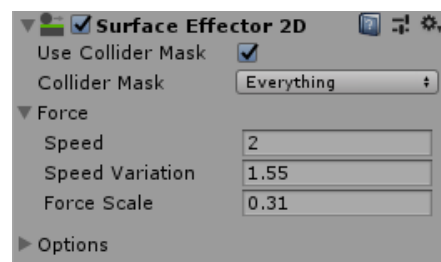
- **Speed** – המהירות הקבועה שאלה יגיע העצם (המהירות של ה"סרט נע"). עם מספר חיובי העצם המתנגש ינוע ימינה (בהנחה שהמשטח הוא אופקי). עם מספר שלילי העצם ינוע שמאלה, וככל שהערך יותר גדול (בערך מוחלט) הוא ינוע מהר יותר.
- **speed variation** – מספר רנדומלי בין 0 למספר שהוזן. הערך מוסף למהירות שהגדרנו ב-speed. עבור ערך שלילי נוספים כוחות נגדיים לכוח ששמונו ב-speed. המטרה היא ליצור מהירות אקראית.
- **force scale** – כמה זמן ייקח לאובייקט מרגע ההתנגשות להאיץ למהירות המרבית. ככל שהערך נמוך יותר ייקח לאובייקט יותר זמן להאיץ (ערכים בין 0 ל 1).
- **collider mask** – מגדיר באילו שכבות ישפיע האפקט.



התמונות לעיל ממחישות שימוש של האפקטור. הוספנו למשטח את האפקטור 2D surface effector -I collider2D, בקולידר של המשטח גם הגדרנו **used by effector**, כדי שנוכל להשתמש אפקטור, ולאסטרואיד הוספנו גוף קשיח דינמי + קולידר. כאשר הכדור נוחת על המשטח בהרצת המשחק, הוא "מתגלגל" ימינה במהירות.

באפקטור לצורך הדוגמא הגדרנו כך :





שימו לב – הקולידר של העצם צריך להיות לא טריגר, כדי שהעצם המתנגש יוכל להתגלגל על המשטח שלו ולא יחדור לתוכו.

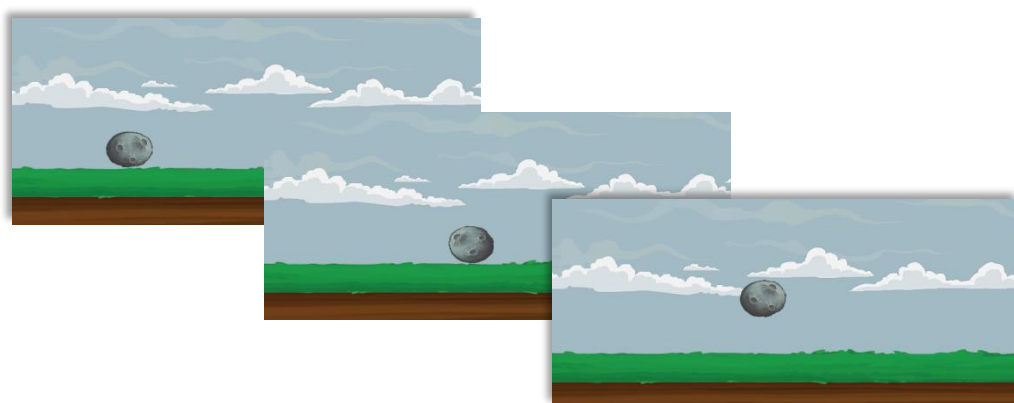
2. Area Effector - אפקטור אזורי

מפעיל כוח בזווית מסויימת על כל עצם שנכנס לתחום הקולידר שלו. למשל נניח הכדור מהדוגמא הקודמת ממשיך לנוע על המשטח עם ה-surface effector, עד שהוא נכנס לאזור שמפעיל עליו כוח וגורם לו לקפוץ. נמשיך מהדוגמא שעשינו למעלה- ניצור אובייקט ריק חדש ונוסיף לו colliderBox2D ובקולידר נגדיר use by effector כדי שנוכל להשתמש באפקטור area. שימו לב – בניגוד לקודם, הקולידר של העצם צריך להיות טריגר, כדי שיוכל לחפוף לקולידר של העצם המתנגש.

נוסיף את area effector 2D ונסמן את use collider mask. נשנה את הגודל של הקולידר והמיקום שלו להיכן שנרצה שבו יופעל האפקטור. בעזרת השדה force magnitude נוכל להגדיר את עוצמת הכוח שיופעל על האובייקט שיכנס למתחם של האפקטור, ובעזרת force angle את הזווית של הכוח.

דוגמאת הרצה:

האסטרואיד ממשיך לנוע לאותו כיוון עד שהוא נכנס לקולידר של האובייקט הריק שעשינו, ואז בגלל AreaEffector מופעל על האסטרואיד כוח שגורם לו לקפוץ בזווית.



3. Point Effector - אפקטור נקודתי

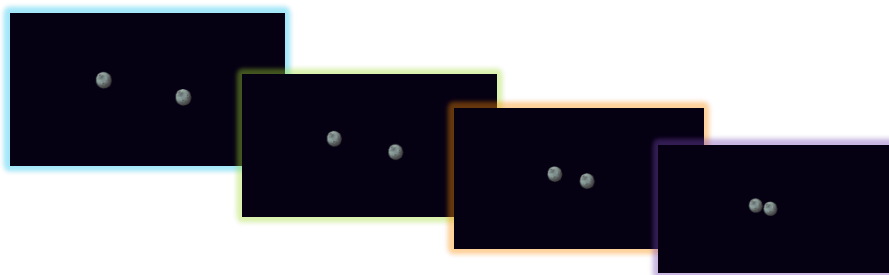
מפעיל משיכה או דחיה על כל קולידר שנכנס לקולידר שלו, לכיוון מרכז המסה של הגוף הקשיח שלו. מדמה מגנט הבנוי על העצם שאליו הוא מוצמד. Force Magnitude מגדיר בכמה כוח האובייקט מושך אליו אובייקטים אחרים, או דוחה אותם. עבור ערך חיובי יש דחייה של האובייקט לגופים אחרים, ועבור ערך שלילי הוא מושך אותם אליו.

כברירת מחדל האובייקט מוגדר ב-Force Mode constant. השדה force mode מגדיר את הדרך בה תוצג ההתנגשות. Constant למשל מגדיר שהמשיכה, או הדחייה בין האובייקטים תעשה כאשר האובייקטים יכנסו למתחם



הקולידר אחד של השני. לפעמים נרצה שההתנגשות תראה יותר ריאליסטית, כלומר ככל שהאובייקטים קרובים אחד לשני יותר ככה הם ינוע אחד לשני במהירות גבוהה יותר, או במילים קצת יותר פורמליות נרצה שהגופים יקיימו את חוק הכבידה האוניברסלי: $F = G * \frac{m_1 * m_2}{r^2}$ כאשר F הוא הכוח המופעל, והוא שווה למכפלת המסות של הגופים ביחד עם G- כוח הכבידה האוניברסלי, חלקי המרחק בין הגופים בריבוע.

למזלנו החברה של יוניטי חשבו גם האסטרו-פיזיקאים חובבי המשחקים, ונתנו את האפשרות להגדיר את ההתנגשות בהתאם לנוסחה לעיל, כל מה שצריך לעשות הוא לשנות ה-force mode ל-Inverse squared.



אז איך משתמשים באפקטור? נסיף לעצם שלנו rigidbody2D ושני קולידרים - אחד לא טריגר ובו נגדיר את גבולות האובייקט, והשני טריגר ובו נגדיר את מסגרת הכוח של המשיכה/דחיה, כלומר השדה שאם אובייקט אחר נכנס אליו הוא נמשך לאובייקט עם ה-pointEffector. בקולידר של שדה הכוח נגדיר Used by effector.

שימו לב – אם יש שני גופים עם PointEffector2D, אז שניהם ימשכו אחד את השני ברגע שהקולידרים שלהם נחתכים (כאן "חוק הפעולה והתגובה" של ניוטון מתקיים). אבל אם רק לאחד מהם יש PointEffector2D, אז הוא ימשוך את הגוף השני ברגע שהוא נכנס לקולידר שלו, אבל לא יימשך אליו (כאן נראה ש"חוק הפעולה והתגובה" לא מתקיים).

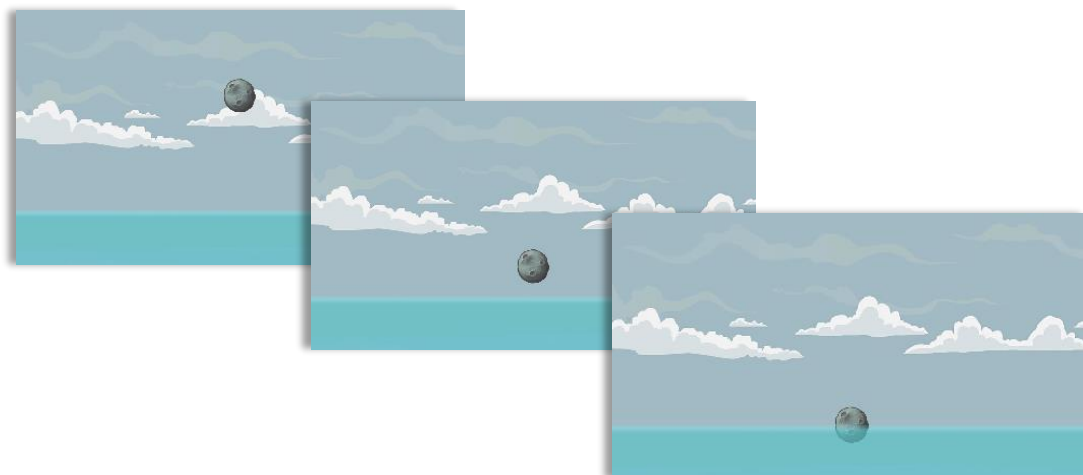
4. Buoyancy Effector - אפקטור ציפה

מפעיל כוחות של שקיעה וציפה, כמו בתוך מים או ג'לי, על כל גוף שנכנס לקולידר שלו.

בעזרת ה-density נוכל להגדיר את רמת הצפיפות של הנוזל. בהתאם לחוקי הפיסיקה, גוף עם צפיפות נמוכה יותר מהנוזל – יצוף (כמו עץ על-פני מים, או אדם על-פני ים המלח), גוף עם צפיפות גבוהה יותר – ישקע (כמו פלדה במים). ככל שהצפיפות של העצם נמוכה יותר הוא יצוף ויהיה פחות בתוך המים מאשר אובייקט ששוקל יותר. ניקח לדוגמא כדור חוף לעומת כדורסל, כדור החוף יצוף יותר מהכדורסל וזה יתבטא בין היתר שפחות ממנו יהיה בתוך המים, לעומת הכדורסל שחלק יותר גדול ממנו יהיה שקוע במים. הצפיפות של גוף היא המסה (שדה של RigidBody) חלקי הנפח (גודל הקולידר).

ה-surface level מגדיר את גובה פני המים. כוחות הציפה מתחילים לפעול על הגוף רק מרגע שהקולידר שלו חותך את פני המים. בנוסף יש אפשרות לשלוט בזרימה של הנוזל. בדיוק כמו באפקטורים הקודמים, flow magnitude יכול לשלוט במהירות וכיוון הזרימה.

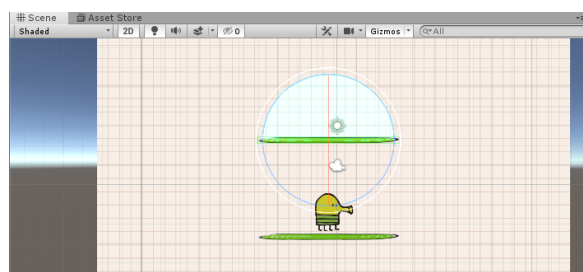




גם כאן ניצור את האובייקט בצורה דומה לקודמים. נוסיף לו קוליידר נסמן את Used by effector ו- is Trigger ונוסיף Buoyancy effector 2D.

5. Platform Effector – אפקטור פלטפורמה

מי שמכיר את המשחק Doodle jump או את המשחק icy tower הרעיון דומה: מצמידים אפקטור זה לפלטפורמות או משטחים למיניהם כדי שהדמות תוכל להתנגש בקוליידר של הפלטפורמה מכיוון מסוים ולא מכל מקום. למשל ב-icy tower השחקן מגיע מלמטה אבל לא נתקע מלמטה הפלטפורמה עליה הוא קופץ, אלא רק כשהוא כבר מעל לפלטפורמה הוא יכול "להתנגש" בה. האפקטור מגיע בצורה של קשת בקוליידר אליו הוא נצמד וניתן לשנות את הזווית שלו ב-surface arc או את הכיוון שממנו תורגש ההתנגשות ב-Rotational offset.



מקורות

- סרטונים ביוטיוב

סיכום: מעוז גרוסמן

