מטלה – דגמים וטריגרים

א. שינוי ושיפור המשחק מהשיעור

הורידו מגיטהאב את הקוד של משחק-החלליות שבנינו בהרצאה
(זה כאן https://github.com/gamedev-at-ariel פרוייקט מספר 02).

שימו לב: אין להוריד בעזרת zip אלא בעזרת git clone. כמו כן כדי לקצר את זמן ההורדה מומלץ להגביל את העומק:

```
git clone --depth=1 https://github.com/...git
```

ודאו שאתם מבינים את הקוד, ובצעו לפחות שינוי אחד מתוך הרשימה הבאה:

1. המגן לא נמצא על המסך בהתחלה, אלא מופיע מדי-פעם בנקודה אקראית. כשהשחקן מתנגש במגן, נוסף עיגול מסביב לחללית של השחקן. צבע העיגול נחלש משניה לשניה עד שהוא נעלם אחרי 5 שניות.
2. מדי-פעם מופיע על המסך תותח, בנקודה אקראית. כשהשחקן אוסף את התותח, הוא יכול לירות לייזר גדול וחזק יותר, למשך מספר שניות. התותח הוא חד-פעמי כמו המגן – נעלם אחרי שהשחקן אוסף אותו [אם אתם מסתבכים, תוכלו למצוא הסבר בגיטהאב של הקורס בתיקיה 6. אבל תנסו קודם לבד].
3. החללית של השחקן לא נהרסת מייד כשהוא מתנגש באויב, אלא יש לו בתחילת המשחק 3 "נקודות פגיעה" (hit points), כל פגיעה באויב מורידה לו נקודה אחת, ורק כשהוא מגיע לאפס הוא נהרס.
4. השחקן לא יכול לירות לייזרים בלי הפסקה, אלא חייב לחכות זמן מסויים (נניח חצי שניה) בין יריה ליריה הבאה.
5. יש שני סוגי אויבים: אחד איטי וכשפוגעים בו מקבלים נקודה אחת, והשני מהיר וכשפוגעים בו מקבלים שתי נקודות (כיתבו קוד כללי שיהיה קל להרחבה לשלושה סוגי אויבים או יותר).
6. שנו את גבולות המשחק – הפכו את העולם לעולם עגול – כשהשחקן מגיע לצד אחד של העולם, הוא מופיע בצד השני (השתמשו ברכיבי-התנגשות ולא במספרי-קסם).

בנוסף לשינוי מהרשימה, הוסיפו לפחות שינוי אחד מקורי.

ב. מימוש תהליכי-ליבה

כזכור, **תהליך-הליבה** הוא אוסף הפעולות המתבצעות שוב ושוב במהלך המשחק. הנושאים שלמדנו בשיעור הזה מאפשרים לכם כבר לממש תהליכי-ליבה של הרבה משחקים קלאסיים. ממשו את תהליך-הליבה במשחק כלשהו לבחירתכם. הנה כמה דוגמאות:

1. Jumper frog – צפרדע צריך לעבור את הכביש בלי להיפגע ממכוניות. *הרעיון*: צרו שלושה מסלולים שידמו את הכביש, וצרו spawner לכל נתיב שמתזמן את האובייקט שמייצג את המכוניות בזמן אקראי כלשהו. בנו קוד שמזיז את המכוניות בתנועה רציפה מימין לשמאל (או ההפך), ואת השחקן בהתאם לקלט ממקשי החיצים. השתמשו בטריגר כדי לבדוק אם השחקן נדרס בדרך, ובטריגר אחר כדי לבדוק אם השחקן הגיע לצד השני בהצלחה.

2. Guitar Hero - יש spawner שמתזמן רנדומלית קוביות שיורדות מהחלק העליון של המסך , הצבע של הקוביות נבחר בצורה אקראית מבין שלושה צבעים, וכאשר הקובייה מתנגשת באובייקט אחר שנמצא בתחתית המסך על השחקן ללחוץ על המקש המתאים בהתאם לצבע הקובייה. למשל מקשים z=אדום, x=כחול, ו-c =צהוב. לשינוי צבע אובייקט בזמן משחק מומלץ להיעזר בזה.

3. דו קרב: בנו משחק דו קרב יריות בין שני אובייקטי קובייה. השחקנים יזוזו משני קלטים שונים במקלדת (למשל קובייה אחת תזוז מהחצים והשנייה מהמקשים a,s,d,w) הקוביות צריכות לעמוד אחת מול השנייה כמו כן שימו לב שהכיוון של הלייזר של כל קובייה צריך להתאים למיקום של הדמות שיורה אותו. המנצח הוא מי שהצליח לפגוע שלוש פעמים בקובייה היריבה.

4. Tetris racing - השחקן הוא מכונית, והוא צריך להיזהר לא להתנגש במכוניות שמגיעות מהכיוון השני במסלול שלו. שימו לב שהשחקן זז רק על ציר ה-x. אם תרצו לשפר את המשחק הגדירו את התנועה של המכוניות האחרות מציר ה-z במקום מציר ה-y, והמצלמה קצת מעל לשחקן, כך המשחק ירגיש ייחודי.

5. Pac Man – צריך לעבור במבוך ולאכול את כל הנקודות בלי להתנגש ברוחות רפאים. יש נקודות מיוחדות שנותנות חסינות למשך מספר שניות.

אין צורך לממש את כל המשחק (עם כמה רמות, חיים, ניקוד, גרפיקה וכו'), אלא רק את תהליך-הליבה.

ההגשה אישית - כדי שכולכם תדעו לתכנת ותוכלו לתרום באופן שווה לעבודת הצוות בהמשך הקורס.

- העלו את שני המשחקים שכתבתם לחשבון שלכם ב-itch.io. ודאו שהמשחקים במצב public.
- העלו את הקוד לגיטהאב שלכם. והוסיפו קובץ README.md המסביר מה בדיוק עשיתם, כולל הפניות לקוד. ודאו שהתיקיה שאתם מעלים כוללת את הקבצים gitignore, gitattributes כמו בגיטהאב של הקורס, ושאתם לא מעלים את הקבצים הזמניים של יוניטי.
- הגישו במודל קישור למשחק באיץ' ובגיטהאב.