



## המחלקה להנדסת תוכנה

שם הפרויקט: ביט אס משחק מציאות מדומה

Project Name: Beat Us VR Game

### דוח מסכם:

שם סטודנט 1: עמית פלרמן

שם סטודנט 2: גבריאל דבח

שם המרצה: ד"ר אלחנן גזית

תאריך ההגשה: 26.05.2020

### אישור המנחה:

עמית וגבריאלי שלום,  
אני מאשר את הגשת תיק פרוייקט לאחר שבוצעו בו כל התיקונים והשיפורים בהתאם  
למשוב ולהמלצותי.

בהצלחה,  
ד"ר אלחנן גזית

א.ג.  
ד"ר אלחנן גזית  
מייסד ג'ולות  
jult60t

## תודות:

ברצוננו להודות למנחה הפרויקט ד"ר חנן גזית על העזרה, ההיענות וההכוונה. בנוסף, תודה לחנן שחשב על הרעיון לפרויקט ודאג לזמינות הצידוד, תמך ודחף קדימה גם בתקופת הקורונה המורכבת.

כמו כן, ברצוננו לומר תודה מיוחדת לניר אלנברג ולמחלקת פרויקט גמר על האוזן הקשבת, הדינמיות, המענה המהיר והעזרה הרבה. במהלך הקורונה כאשר המכללה הייתה סגורה וכתוצאה מזאת לא יכולנו להגיע למעבדה להשתמש בצידוד האוקולוס אשר הכרחי לפיתוח הפרויקט, ניר נתן לנו את האפשרות להשאיל ציוד יקר ערך שבלעדיו לא היינו מצליחי להשלים את הפרויקט בזמן ובצורה הטובה ביותר.

## תוכן עניינים:

7	4. רשימות:
9	5. תקציר מנהלים:
10	6. Executive Summary:
11	7. מילון מונחים:
12	8. מבוא
13	9. מטרות ויעדים:
13	9.1. מטרת הפרויקט:
13	9.2. יעדי הפרויקט:
14	10. סקירת ספרות ביקורתית מורחבת / סקר שוק:
18	11. ניתוח חלופות מערכתי:
18	11.1.1 פיתוח משחק מחשב אישי
18	11.1.2 פיתוח משחק למכשיר מובייל
18	11.1.3 פיתוח משחק WEB
19	11.1.4 השוואה בין החלופות
21	12. דרישות המערכת:
21	12.1. דרישות מידע ופונקציונליות:
21	12.2. דרישות לא פונקציונליות:
21	12.3. דרישות הפעלה:
22	13. אפיון המערכת:
23	14. ניתוח חלופות טכנולוגיות:
23	14.1 חלופות טכנולוגיות:
23	14.1.1 פיתוח ל- Oculus Rift
23	14.1.2 פיתוח ל- Samsung Gear
23	14.1.3 פיתוח ל- HTC VIVE
24	טבלה 14.7 - פיתוח ל- HTC VIVE
24	14.1.4 השוואה בין החלופות
26	15. תכן המערכת:
28	16. התוצר:
30	17. תכנון הפרויקט:
32	18. בדיקות והערכה:
34	19. סיכום ומסקנות:

35	20. הצעה לעבודת המשך:
36	21. רשימת מקורות:
37	22. נספחים:
37	22.1 מסמך SRD
44	22.2 מסמך SDD
54	22.3 מסמך STD

## 4. רשימות:

### **איורים:**

- [איור 10.1 - משוואת ה-DWT](#)
- [איור 10.2 - דוגמה לפילטר ברמה 3.](#)
- [איור 10.3 - שלבי האלגוריתם שהוצג במאמר](#)
- [איור 10.4 - קטע קוד בשפת SCALA המבצע אלגוריתם זה ברמה הרצויה](#)
- [איור 22.2.1 - מסך ראשי של המשחק](#)
- [איור 22.2.2 - צילום מסך השיאים של המשחק](#)
- [איור 22.2.3 - צילום מסך הוראות המשחק](#)
- [איור 22.2.4 - צילום מסך האפשרויות](#)
- [איור 22.2.5 - צילום מהלך המשחק](#)
- [איור 22.4.1 - פוסטר הפרוייקט](#)

### **טבלאות:**

- [טבלה 11.1 - פיתוח משחק מחשב אישי](#)
- [טבלה 11.2 - פיתוח משחק למכשיר מובייל](#)
- [טבלה 11.3 - פיתוח משחק WEB](#)
- [טבלה 14.5 - פיתוח ל-Oculus Rift](#)
- [טבלה 14.6 - פיתוח ל-Samsung Gear](#)
- [טבלה 14.7 - פיתוח ל-HTC VIVE](#)
- [טבלה 14.8 - השוואה בין החלופות](#)
- [טבלה 22.1.1 - תיאור זרימה ראשית של התחלת המשחק](#)
- [טבלה 22.1.2 - תיאור זרימה ראשית של מהלך המשחק](#)
- [טבלה 22.1.3 - תיאור זרימה משנית של מהלך המשחק](#)
- [טבלה 22.1.4 - תיאור זרימה ראשית של צפייה בטבלת השיאים](#)
- [טבלה 22.1.5 - תיאור זרימה ראשית של צפייה בהוראות המשחק](#)
- [טבלה 22.1.6 - תיאור זרימה ראשית של בחירת שיר](#)
- [טבלה 22.1.7 - תיאור זרימה ראשית של בחירת רמת קושי](#)
- [טבלה 22.2.1 - פעולות משתמש במסך הראשי של המשחק](#)
- [טבלה 22.2.2 - פעולות משתמש במסך טבלת השיאים](#)

[טבלה 22.2.3 - פעולות משתמש במסך הוראות המשחק](#)

[טבלה 22.2.4 - פעולות משתמש במסך האפשרויות](#)

[טבלה 22.2.5 - פעולות משתמש במשחק](#)

[טבלה 22.3.1 - לוח זמנים לבדיקות המערכת](#)

## **גרפים:**

[גרף 15.1 - ארכיטקטורת MVC](#)

[גרף 22.1.1 - תרשים בלוקים של המערכת](#)

[גרף 22.1.2 - UseCase Diagram של המסך הראשי](#)

[גרף 22.2.1 - ארכיטקטורת המשחק](#)

[גרף 22.2.2 - Sequence Diagram של המשחק](#)

## 5. תקציר מנהלים:

במשך שנים רבות מציאות מדומה הייתה חלום בעולם הגיימינג. קשה להאמין שרק לפני עשור Oculus Rift הוציאו את אב הטיפוס הראשון שלהם וכל התחום הזה היה כל כך רחוק ולא בשל. אך כיום, עולם הגיימינג עובר שינוי משמעותי. בשילוב קונסולות חזקות ומתקדמות, מציאות מדומה היום מגדילה את ניתוחה בשוק וגם רכישת ציוד הולך ונהייה נגיש יותר וזול לעומת ימים עברו. עם כך, ברור לכל שמשחק במציאות מדומה נותנת לשחקן את ההרגשה הכי עמוקה שיכול לקבל. בדיוק זו הסיבה שבחרנו לשלב בפרויקט שלנו מציאות מדומה. תקציר זה מתאר את פיתוח משחק מציאות מדומה הנקרא "Beat Us". המטרה העיקרית של הפרויקט אותו נפתח הינו משחק מציאות מדומה המבוסס בינה מלאכותית, AI, שתדע לנתח את המוזיקה ולהציג אובייקטים לפי הקצב. יש לציין שפיתוח משחק כזה מורכב בשל האתגר ההנדסי שלו ומחדש מהמתחרים היום בשוק הקיימים בקונסולות שונות אותן סקרנו במהלך העבודה. השחקן יצטרך להכות באובייקטים בהתאם למוטות אותו יאחז, להתחמק מאובייקטים מסוימים וכך יצבור נקודות כשהמטרה כמובן היא לצבור כמה שיותר ולעמוד בראש הטבלה. המשחק מפותח ל-PC שכן נבחר מבין החלופות המערכתיות בשל היותו מספק גרפיקה ברמה גבוהה וחוויית משתמש בהתאם.



## 6. Executive Summary:

For many years incorporating virtual reality technologies in video games has been a dream.

It's hard to believe that just a decade ago the Oculus Rift released its first prototype and this whole area was so far away and undeveloped.

But today, the gaming world is undergoing significant change.

Combined with powerful and advanced consoles, the virtual reality ecosystem today is increasing market share, and also purchasing equipment is becoming more accessible and cheaper.

However, it is clear to everyone that playing in virtual reality gives the player the deepest feeling he can get.

This is exactly why we chose to combine virtual reality into our project.

This summary describes the development of a virtual reality game called "Beat Us".

The main goal of the project is a virtual reality game based on artificial intelligence, AI, that can analyze the music and display objects according to the beat.

It should be noted that such a game development is complex due to its engineering challenge and resumes from today's market competitors in various consoles that we reviewed during work.

The player will have to hit objects according to the handles he will hold, evading certain objects and thus accumulate points with the goal, of course, being to accumulate as much as possible and head the table.

The game is developed for PC as it is chosen from among the systemic alternatives due to providing high-level graphics and user experience accordingly.

## 7. מילון מונחים:

- מציאות מדומה - virtual reality** - מציאות מדומה הוא מדיום הכולל סימולציות אינטראקטיביות ממוחשבות שמסוגלות לחוש את מיקום המשתמש ופעולותיו ומסוגלות לייצג ולספק משוב שמחליף אחד או יותר מחושי הביולוגיים של המשתמש (ראיה, שמיעה, מגע, ריח) בזמן אמת, ליצירת הרגשה ותחושה של "היטמעות" (immersion) או "נוכחות" (presence) (בסימולציה (העולם הווירטואלי) [4].
- יוניטי - UNITY** - מנוע (מחולל) לפיתוח משחקים ואפליקציות למספר פלטפורמות מבוסס C#.
- אוקולוס - OCULUS** - החברה המייצרת את משקפי המציאות המדומה RIFT S האיכותיים בהן השתמשנו.
- הנדל - Handle** - ידיעות עזר לקסדת האוקולוס אותן אוזח המשתמש. במשחק מתפקדות כחרבות.
- ביט פר סקנד - BPM - Beat Per Second** - מספר הביטים בשניה, לפיו משוגרים האובייקטים במשחק.
- בוסט - BOOST** - מעין אובייקט בונס לשחקן שתדירותו נמוכה אשר משפיע על ניקוד השחקן לזמן קצוב או חד פעמי.
- בינה מלאכותית - AI** - בינה מלאכותית, היכולת של המחשב לזהות פעולות לחשוב ולפעול בהתאם.
- אסט - Asset** - משאב שניתן להורדה מהחנות של unity ובכל משאב כזה כלים ועזרים כמו אובייקטים וסקריפטים שניתן להשתמש בהם לפרויקט.
- דארק מוד - Dark Mode** - מצב בו המסך מוחשך ואין בו תאורה מרכזית.

## 8. מבוא

פרויקט זה הינו פרויקט אקדמאי הנעשה כפרויקט סיום ללימודי תואר ראשון בהנדסת תוכנה BSc במכללת אפקה והוא מיועד למשתמשים פרטיים. הפרויקט הינו פיתוח משחק מציאות מדומה מבוסס מוזיקה המשלב הנאה וקורדינציה.

המשחק נקרא Beat Us ושמו מרמז גם על תוכן המשחק. במהלך המשחק צריך להכות (TO BEAT) באובייקטים שמגיעים לכיוון השחקן לפי קצב ה-BEAT. לשחקן ישנן שתי מוטות המדמות חרבות, כל אחד בצבע שונה והוא צריך להכות באובייקטים המגיעים אליו באמצעות החרבות, בהתאם לצבען ובהתאם לכיוון החוץ שמופיע על האובייקט. בפרויקט זה ישנן 3 רמות קושי מותאמות ליכולות השחקן ומשתנות תוך כדי בהתאם לביצועיו. ככל שהרמה קשה יותר, כך האובייקטים מגיעים מהר יותר, יש מורכבות קשה יותר לפגוע בהם ואף מסיחים שיורידו ניקוד לו השחקן יפגע בהם. בנוסף, ישנם בוסטים שתדירותם נמוכה בהתאם לנדירות שלהם ולהשפעה שלהם על המשחק. כותבי הפרויקט: עמית פלרמן וגבריאל דבח. מנחה הפרויקט: ד"ר חנן גזית.

## 9. מטרת ויעדים:

### 9.1. מטרת הפרויקט:

מטרת הפרויקט היא לפתח משחק מציאות מדומה מהנה המתבסס על פי קצב המוזיקה אותה בחר השחקן.

### 9.2. יעדי הפרויקט:

- 9.2.1. שיטת פיתוח הפרויקט מבוסס על ארבע שלבים: בשלב הראשון בוצע מחקר ותיכון ראשוני של הטכנולוגיות ובמקביל פיתוח קונספט המשחק.
- 9.2.2. בשלב השני בוצע תכנון המערכת של המשחק והמודולים השונים המרכיבים אותו: UI, AI, צד שרת, תכנון בסיס נתונים, בחירת פלטפורמה לפיתוח מבין חלופות. בשלב השלישי פותחה אפליקציית המשחק לרמת מוקאפ עובד.
- 9.2.3. בשלב רביעי התבצע תהליך בדיקות ושיפורים של המשחק.

### 9.3. מדדים:

- 9.3.1. זמן התגובה הממוצע של המשחק לפעולת השחקן יפחת מ150ms.
- 9.3.2. לפחות 80% מהמשתמשים יעידו כי המשחק מהנה.
- 9.3.3. הצגת המידע אובייקטים לשחקן לא פחות מ80 ms.
- 9.3.4. המערכת תסכום את הנקודות במהלך המשחק ובסיומו תציג אותו לשחקן בלא פחות מ80 ms (מייד).
- 9.3.5. המשחק יצליח להציג ריבוי אובייקטים (עתידיים לפגיעה), לפחות 5 במסך.

לגבי תוצאות המדדים, מפורט בסעיף 18 ד.

## 10. סקירת ספרות ביקורתית מורחבת / סקר שוק:

משחקים מבוססי מוזיקה, דהיינו שהמוזיקה היא לא רק מוטיב אווירה אלא ממש חלק מהמשחק, איתנו עוד משנות השבעים. על פי מאמר [1] שסוקר משחקי וידאו מוזיקליים, ומחלק אותם לז'אנרים שונים, המשחק שאנו מציעים הוא בעיקר מהז'אנר של ריתם אקשן (Rhythm action) שהוא בעיקר ללחוץ על כפתור בטיימינג עם הקצב.

בסקירתנו הראשונית, בSOW, נעזרנו במאמר [2], שסמצאנו שיטה לניתוח מוזיקה שנקראת DFT. שיטת DFT שהיא שיטה מתמטית של התמרת פוריה המשמשת להעביר מפונקציה מחזורית, לפונקציה מחזורית אחרת. כיוון שקצב הוא אות שמגיע בצורה מחזורית שיטה זו מתאימה לניתוח הקצב של הקטע המוזיקלי. DFT זה אלגוריתם מהיר שניתן למקבל חלק ממנו. אולם ל DFT מגבלה משמעותית, היא לא מגלה תכיפות קצב נמוכות. לכן חיפשנו אלגוריתם אחר אשר יענה טוב יותר על דרישות המערכת.

בסקירה הבאה שלנו, במאמר [3] גילינו שיטה נוספת שאותה נעדיף במשחק שלנו: Discrete Wavelet, או בקיצור DWT.

שיטת WT מדמה את האוזן האנושית כאשר היא מספקת רזולוציית זמן-תדירות דומה. שיטת DWT היא מקרה מיוחד של WT, המספקת ייצוג קומפקטי של ניתוח האות בזמן ובתדירות, הניתן לחישוב יעיל.

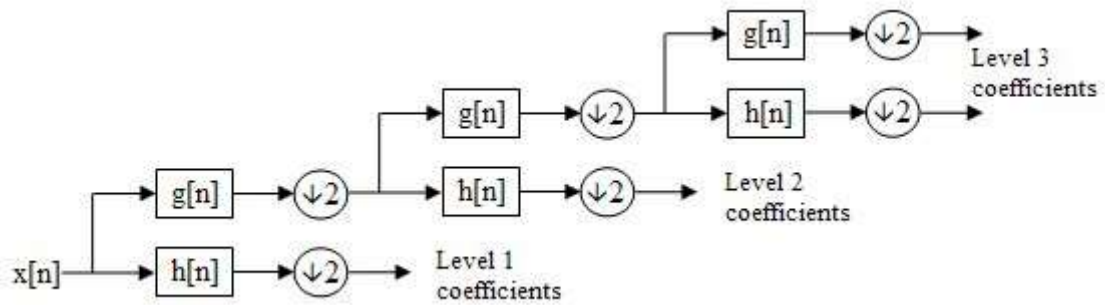
האלגוריתם של DWT לוקח דוגמיות משתי רצועות תדירות, גבוהה ונמוכה וכל אחת מהן מסוננת על ידי האלגוריתם במקביל. כך שוב ושוב מעבירים בשני הפילטרים לרמה הרצויה.

$$W(j, k) = \sum_j \sum_k x(k) 2^{-j/2} \psi(2^{-j}n - k) \quad (1)$$

$$y_{high}[k] = \sum_n x[n]g[2k - n] \quad (2)$$

$$y_{low}[k] = \sum_n x[n]h[2k - n] \quad (3)$$

איור 10.1 - משוואת ה-DWT



איור 10.2- דוגמה לפילטר ברמה 3 .

#### 1. LPF

Low pass filtering of the signal with a One Pole filter with alpha value 0.99 given by the equation:

$$y[n] = (1 - \alpha)x[n] - \alpha y[n] \quad (3)$$

#### 2. FWR

Full wave rectification given by the equation:

$$y[n] = \text{abs}(x[n]) \quad (4)$$

#### 3. DOWNSAMPLING

$$y[n] = x[kn] \quad (5)$$

#### 4. NORM

Normalization in each band (mean removal) :

$$y[n] = x[n] - E[x[n]] \quad (6)$$

#### 5. ACRL

Autocorrelation given by the equation:

$$y[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n]x[n+k] \quad (7)$$

איור 10.3 - שלבי האלגוריתם שהוצג במאמר

```
for (loop <- 0 until levels) {
  // Apply DWT
  val transform = new Transform(new FastWaveletTransform(wavelet));
  if (loop == 0) {
    val coefficients : Array[Array[Double]] = transform.decompose(data)
```

```

    val l = coefficients.length - 1
    aC = coefficients(1).slice(0, coefficients(1).length/2)
    dC = coefficients(1).slice(coefficients(1).length/2,
coefficients(1).length)
    dCMinLength = (dC.length/maxDecimation).toInt + 1
  } else {
    val coefficients : Array[Array[Double]] = transform.decompose(aC)
    val l = coefficients.length - 1
    aC = coefficients(1).slice(0, coefficients(1).length/2)
    dC = coefficients(1).slice(coefficients(1).length/2,
coefficients(1).length)
  }
  // Extract envelope from detail coefficients
  // 1) Undersample
  // 2) Absolute value
  // 3) Subtract mean
  val pace = pow(2, (levels-loop-1)).toInt
  dC = dC.undersample(pace).abs
  dC = dC - dC.mean
  // Recombine detail coefficients
  if (dCSum == null) {
    dCSum = dC.slice(0, dCMinLength)
  } else {
    dCSum = dC.slice(0, min(dCMinLength, dC.length)) |+| dCSum
  }
}
// Add the last approximated data
aC = aC.abs
aC = aC - aC.mean
dCSum = aC.slice(0, min(dCMinLength, dC.length)) |+| dCSum

// Autocorrelation
var correlated : Array[Double] = dCSum.correlate
val correlatedTmp = correlated.slice(minIndex, maxIndex)

// Detect peak in correlated data
val location = detectPeak(correlatedTmp)
// Compute window BPM given the peak

```

```
val realLocation = minIndex + location
val windowBpm : Double = 60.toDouble / realLocation *
(audioFile.sampleRate.toDouble/maxDecimation)
instantBpm += windowBpm;
```

איור 10.4 - קטע קוד בשפת SCALA המבצע אלגוריתם זה ברמה הרצויה



## 11. ניתוח חלופות מערכת:

### 11.1.1 פיתוח משחק מחשב אישי

יתרון	יותר רווח ופשוטות לפיתוח כמו כן למציאת מקורות.
יתרון	ישנו ציוד מגוון ואיכותי למשחק מציאות מדומה.
יתרון	מאפשר גרפיקה ברמה גבוהה.
חסרון	חוסר ניידות.
חסרון	מצריך התקנה.
חסרון	מצריך ציוד יקר יותר.

טבלה מס' 11.1 - פיתוח משחק מחשב אישי

### 11.1.2 פיתוח משחק למכשיר מובייל

יתרון	לרוב האנשים בעולם המערבי יש טלפון חכם שיוכל להריץ את המשחק.
יתרון	נייד, אפשר לשחק בכל מקום בכל זמן.
חסרון	גרפיקה ברמה נמוכה יותר.
חסרון	מחייב התקנה על המכשיר.
חסרון	מבזבז סוללה.
חסרון	מכביד על המכשיר הנייד, גורם להתחממותו ופליטת קרינה גבוהה יותר.

טבלה מס' 11.2 - פיתוח משחק למכשיר מובייל

### 11.1.3 פיתוח משחק WEB

יתרון	לא מצריך התקנה.
יתרון	יש אפשרות לשחק גם ממכשיר נייד וגם נייד(במידה ותהייה תאימות)
חסרון	מחייב רשת זמינה.
חסרון	מוגבל מבחינה גרפית.

חסרון	יכביד על המערכת ויכול להאט ביצועים.
-------	-------------------------------------

טבלה מס' 11.3 - פיתוח משחק WEB

#### 11.1.4 השוואה בין החלופות

##### קריטריונים למתן ציון:

התקנה: 5-לא מצריך התקנה, 1-מצריך התקנה

איכות גרפית- 5-הכי גבוהה שאפשר, 4- גבוהה אך לא מקסימלית, 3- גבוהה,

2-בינונית, 1- נמוכה.

חוויית משתמש- 5- הכי טובה, 4- טובה, 3- בינונית

ניידות- 5- נייד, 1- לא נייד

עלות - 5- חינם, 4- מאוד זול, 3- זול, 2- יקר

קרינה- 5- ללא קרינה כללי, 4- קרינה נמוכה, 3- קרינה בינונית, 1- קרינה גבוהה

משקלים: איכות גרפית-5, חוויית משתמש- 5, התקנה- 3, קרינה- 3, ניידות-1,עלות-1

קריטריון/חלופה	PC	MOBILE	WEB
התקנה	1	1	5
איכות גרפית	5	4	2
חוויית משתמש	5	4	3
ניידות	1	5	1
עלות	2	4	2
קרינה	3	1	4
ציון משוקלל	68	56	55

טבלה 11.4 - השוואת החלופות

### החלופה הנבחרת:

החלופה הנבחרת היא פיתוח המשחק ל-PC.

פיתוח המשחק ל-PC נותן את האיכות הגבוהה ביותר וכך לדעתנו חווית המשתמש הכי טובה שיכולים לספק במשחק VR.

החיסרון המשמעותי שלו הינו עלות הציוד היקרה יותר מאשר לטלפון נייד איננה פקטור בפרוייקט שלנו מפני שהציוד כבר קיים ולא צריך להירכש.

כמו כן גם ניידות אמנם פרמטר חשוב בתחומים רבים בעולם שלנו אך בעולם הגיימינג הוא לאו דווקא כה חשוב וזה מוכח לפי כך שרוב השחקנים המקצועיים משחקים במערכות ניידות.

## 12. דרישות המערכת:

### 12.1. דרישות מידע ופונקציונליות:

המערכת תציג תפריט שיאפשר לבחור את המוזיקה למשחק ואת רמות הקושי לבחירת השחקן.

המשחק יוצג בתלת מימד במשקפי תלת המימד אותם יענוד השחקן.

המערכת תציג לשחקן את המוטות אותן הוא אוזז כחרבות על גבי המרקע.

בעת פגיעה נכונה עם החרב או לא נכונה, יוצג חיווי על המסך לשחקן.

האובייקטים יתקרבו אל השחקן לפי קצב המוזיקה אשר ינותח מראש.

רמת הקושי תשתנה במהלך המשחק בהתאם לביצועי השחקן.

במהלך המשחק יופיע על המסך ניקודו של השחקן ובסיום המשחק תופיע הודעה עם ניקודו הסופי.

תוצאות השחקן ישמרו בטבלה אשר יתשמור את 10 התוצאות הטובות ביותר.

### 12.2. דרישות לא פונקציונליות:

ממשק משתמש פשוט ואינטואיטיבי אשר נשלט ע"י שלטי ה-Oculus Rift.

### 12.3. דרישות הפעלה:

משקפי מציאות מדומה : Oculus Rift

מחשב ניח או נייד ברמת קונפיגורציה המאפשרת הרצת Oculus Rift בהתאם לדרישות הפלטפורמה. ניתן לבדוק באתר החברה:

[/https://support.oculus.com/248749509016567](https://support.oculus.com/248749509016567)

### 13. אפיון המערכת:

#### מודל המערכת:

מודל המערכת בנוי ממספר חלקים.

משחק מציאות מדומה בעזרת קסדת ומוטות OCULUS RIFT מורץ בעזרת מנוע UNITY. בפיתוח המשחק ישנה הפרדה בין מודול ניתוח מוזיקה אנחנו בנינו ונותן לנו חיווי לשינוי BPM כל 3 שניות לבין המשחק עצמו שמפותח כיחידה עצמאית.

כל יחידה הינה עצמאית אך עם זאת הכרחית לקיום המשחק ואת הפירוט ניתן לראות בדוח ה SDD.

#### ביצועים עיקריים:

- המשחק פועל בביצועים טובים ביותר, ללא הפרעות ותקיעות תוך עבודה ממוקבלת בין הסקריפטים השונים.
- מודול ניתוח המוזיקה עובד מבעוד מועד ולא במהלך המשחק וכך איננו מהווה פקטור ל"כבדות" ההרצה או לזמן התגובה.
- ישנה התאמה בין ביצועי השחקן בזמן אמת לבין רמת הקושי.
- ישנן 3 רמות קושי. השוני ביניהן הוא במהירות תנועת האובייקטים, כמות ותדירות האובייקטים הכפולים והגעת מסיחים.

#### אילוצים:

- משקפי מציאות מדומה OCULUS RIFT + ידיות.
- מחשב העומד בכל התנאים להפעלת OCULUS RIFT בהתאם ללינק בדרישות ההפעלה.
- נקנו אפקטים ועיצובים מיוחדים במסגרת התקציב שניתן.

## 14. ניתוח חלופות טכנולוגיות:

### 14.1 חלופות טכנולוגיות:

#### 14.1.1 פיתוח ל- Oculus Rift

יתרון	איכות תמונה מעולה ברזולוציה גבוהה 1200×1080 לכל עין.
יתרון	איכות HDMI.
יתרון	זווית ראייה של 110 מעלות(ניתן לראות 360).
יתרון	אוזניות שמע מובנות ב-360 מעלות.
חסרון	לא נייד.
חסרון	צריך מחשב חזק ויקר בעל חומרה אולטימטיבית.

טבלה 14.5 - פיתוח ל- Oculus Rift

#### 14.1.2 פיתוח ל-Samsung Gear

יתרון	רענון 60Hz .
יתרון	נייד.
יתרון	זול.
חסרון	רזולוציה תלוייה במכשיר נייד.
חסרון	שמע באמצעות סמארטפון נלווה.
חסרון	זווית ראייה של 96 מעלות.
חסרון	מכביד על המכשיר הנייד, גורם להתחממותו ופליטת קרינה גבוהה יותר.

טבלה 14.6 - פיתוח ל-Samsung Gear

#### 14.1.3 פיתוח ל-HTC VIVE

יתרון	רענון 90Hz.
יתרון	זווית ראייה של 100 מעלות.

יתרון	איכות תמונה מעולה ברזולוציה גבוהה 1200×1080 לכל עין.
חסרון	מאוד יקר.
חסרון	לא נייד.
חסרון	זוג אוזניות שמע ברמה נמוכה.

טבלה 14.7 - פיתוח ל-HTC VIVE

#### 14.1.4 השוואה בין החלופות

##### קריטריונים למתן ציון:

דרישות מערכת חיצונית: 5- אין דרישות, 4-דרישות נמוכות, 3- דרישות בינוניות,

2- דרישות גבוהות, 1- דרישות הכי גבוהות.

רזולוציה- 5-הכי גבוהה שאפשר, 4- גבוהה אך לא מקסימלית, 3- גבוהה,

2-בינונית, 1- נמוכה.

זווית ראייה- 5- הכי טובה, 4- טובה, 3- בינונית

ניידות- 5- נייד, 1- לא נייד

עלות - 5- חינם, 4- מאוד זול, 3- זול, 2- יקר, 1- מאוד יקר

שמע- 5- הכי טוב, 4- טוב, 3- בינוני, 2- נמוך

קרינה- 5- ללא קרינה כללי, 4- קרינה נמוכה, 3- קרינה בינונית, 1- קרינה גבוהה

משקלים: רזולוציה 4-, זווית ראייה 4-, שמע 5-,עלות-2,קרינה-2, דרישות מערכת חיצונית-3,

ניידות-1.

קריטריון/חלופה	Oculus Rift	HTC Vive	Samsung Gear
דרישות מערכת חיצונית	1	2	3
רזולוציה	5	5	3
זווית ראייה	5	4	3
שמע	5	4	3
ניידות	1	1	5
עלות	2	1	3
קרינה	4	4	1
ציון משוקלל	81	73	61

טבלה 14.8 - השוואה בין החלופות

#### החלופה הנבחרת:

החלופה שנבחרה הינה Oculus Rift שבלטה מעל המתחרות בעיקר באיכות השמע ונוחות האוזניות.

אלו פרמטרים חשובים כאשר המשחק שלנו הוא משחק מוזיקלי וכמובן שאיכותו גם חשובה לנו.

בנוסף, קראנו בסקירות שונות שמוטות Oculus Rift הן הנוחות ביותר אך נוחות איננה דבר מדיד וזה סובייקטיבי ולכן לא נכנס לקריטריונים.

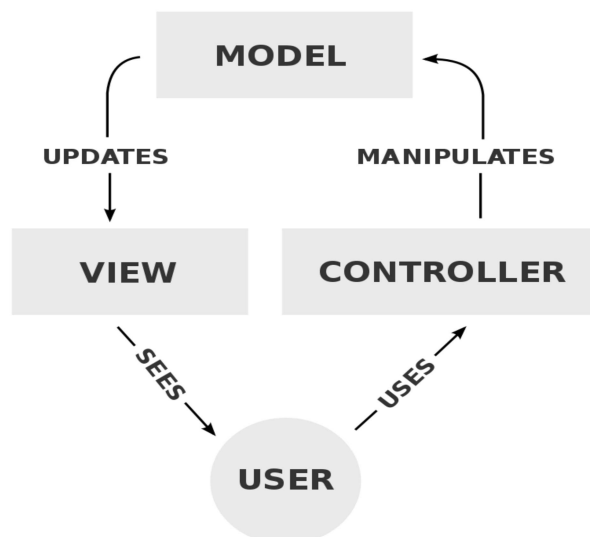
לגבי דרישות מערכת ועלות, הציוד הדרוש כבר קיים ונמצא ולכן קיבל משקל נמוך יותר.



## 15. תכן המערכת:

### ארכיטקטורת המערכת:

המערכת בנויה מתבנית ארכיטקטונית הנקראת MVC,



גרף 15.1 - ארכיטקטורת MVC

ניתן לקרוא בפירוט ולראות תרשים של הארכיטקטורה ודיאגרמת מצבים שמעמיקה ומסבירה את החלוקה בסעיף 2.2 של מסמך הSDD.

### תיכון:

Model או מודל - בתפקיד המודל ישנו GameManager שהוא סינגלטון. המודל מנהל את הנתונים, את הלוגיקה של המערכת ומתפעל את התצוגות השונות.  
View או תצוגה - בתפקיד התצוגה, ישנן מספר תצוגות אשר לא מכירות את המודל והבקר ומושפעות אך רק מהמודל.

Controller או בקר - בתפקיד הקונטרולר ישנן שלוש ישויות, 1-Spawner - תפקידו לייצר ולשגר אובייקטים 2- האובייקטים המשוגרים 3- החרבות. הקונטרולר מקבל חיווי מהמשתמש כאשר הוא מכה\מפספס וכו' ומעדכן את המודל בתוצאות.

## חלופות לתכן המערכת:

בבחירת תכן המערכת עמדו לפנינו שתי חלופות:

### 1. MVC - model view controller:

MVC משמשת כתבנית עיצוב אך נחשבת גם לתבנית ארכיטקטורה, בחלופה זאת ישנה חלוקה של היישום ל 3 חלקים: מודל, תצוגה אשר מחוברים בצימוד נמוך ובקר אשר מפעיל את החלקים האחרים לפי אירועים ביישום.

פירוט על ארכיטקטורת MVC ניתן למצוא במסמך ה SDD בסעיף 2.1.

### 2. Multitier architecture:

מודל שלוש השכבות היא ארכיטקטורת שרת-לקוח שבה ממשק המשתמש, הלוגיקה העסקית של היישום והגישה אל הנתונים מפותחים ומתוחזקים כמודולים נפרדים, בדרך כלל על גבי פלטפורמות שונות.

מודל שלוש השכבות היא ארכיטקטורת תוכנה ותבנית עיצוב.

## תרשים בלוקים/מודולים כללי של המערכת:

ניתן לראות תרשים בלוקים של המערכת במסמך ה SRD בלינק הבא:

[גרף 22.1.1 - תרשים בלוקים של המערכת](#)

## אלגוריתמים:

האלגוריתם בו מודול ניתוח המוזיקה משתמש בקרא DWT. זהו אלגוריתם ידוע אותו מימשנו בשפת #C על מנת שיהיה תואם לשפת פיתוח המשחק עצמו.

האלגוריתם מנתח את השיר וקובע את קצב ה BPM על כל 3 שניות של שיר.

האלגוריתם פועל על שיר באופן חיצוני למערכת המשחק ונותן פלט וקטורי בקובץ טקסט אשר מתמקם בתיקייה ידועה למשחק.

בתחילת המשחק, וקטור ה BPM שנוצר ע"י האלגוריתם נטען והאובייקטים משוגרים לפי קצב זה.

ניתן לקרוא בהרחבה על האלגוריתם ודרך פעולתו, נכונותו וביצועיו [בפרק הסקירה ספרותית](#).

## 16. התוצר:

תוצר הפרויקט הינו משחק מציאות מדומה הפועל עם משקפי מציאות מדומה מדגם OCULUS RIFT בעזרת מנוע יוניט.

התוצר כתוב כולו בשפת C#, כך עובדים עם מנוע יוניטי וגם מודול ניתוח המוזיקה בשפה זו. המשחק מצריך מחשב בעל חומרה חזקה כפי שפורט בדרישות להפעלת האוקולוס ריפט. במהלך המשחק, השחקן אוחז בשתי חרבות, אחת בצבע אדום ואחת בצבע כחול. אל השחקן מגיעים אובייקטים לפי קצב הביט. ה BPM מחושב מראש ומשנה את קצב התזמון כל 3 שניות בהתאם לחישוב שעשה מלפני הרצת המשחק. רוב המשחק יגיעו אל השחקן שני אובייקטים, ריבוע כחול וריבוע אדום שלהם יהייה חץ. השחקן צריך להכות בהם בהתאם לצבע החרב ולכיוון החץ. פגיעה בכיוון החץ עם החרב הנכונה תזכה אותו במירב הנקודות ואילו אם יטעה ויכה עם החרב הלא נכונה או יפספס את הריבוע, יורד ניקוד. בנוסף, ישנם בוססים מיוחדים שיגיעו אל השחקן מדי פעם ויתנו לו אפשרות לקבל בונוס לניקוד, להכפיל את הניקוד לזמן מוגבל או לצאת משגרת המשחק ולהיות ב"דארק מוד" בו אפשר לפגוע בכל אובייקט.

ישנו לוח שמתעדכן בזמן אמת תוך כדי המשחק ומראה לשחקן את הניקוד, מספר הקומבו, זמן שנותר למשחק, רמה וכו'. בסיום השיר או לחילופין אם השחקן יפספס כמות אובייקטים מסויימת ברצף, משתנה בין רמה לרמה ייגמר המשחק ויופיע מסך סיכום.

במסך הסיכום יופיע גם כן נתוני המשחק והתוצאות ואם הצליח השחקן ונכנס לטבלת השיאים, הוא יוכל לרשום את שמו בעזרת מקלדת ולשמור את התוצאה. טבלת השיאים נשמרת בקבצי ג'ייסון, נותן לנו לשמור את הנתונים ולשלף אותם בזמן מהיר, מקומי ומהיר.

לגבי ניתוח המוזיקה, ישנו מודול חיצוני אותו פיתחנו בשפת C# שהייה תואם לשפת פיתוח המשחק והרצתו פשוטה. בסיום הריצה הוא שומר קובץ טקסט עם וקטור BPM לכל שיר ובזמן הרצת המשחק תהייה גישה לקובץ זה.

התוצר עומד בכל הדרישות שהוגדרו במסמך ה-SRD וכולל:

- המערכת מציגה תפריט שמאפשר לבחור את המוזיקה למשחק ואת רמת הקושי.
- המשחק מוצג בתלת מימד במשקפי תלת מימד אותם עונד השחקן.
- המערכת מציגה לשחקן את המוטות אותן הוא אוחז כחרבות על גבי המרקע.
- בעת פגיעה נכונה עם החרב או לא נכונה, מוצג חיווי על המסך לשחקן.
- האובייקטים מתקרבים אל השחקן לפי קצב המוזיקה.

- במהלך המשחק מופיע על המסך ניקודו של השחקן ובסיום המשחק מופיעה הודעה עם ניקודו הסופי.

- תוצאת השחקן נשמרת אם היא בין 10 התוצאות הטובות ביותר.

- ממשק משתמש פשוט ואינטואיטיבי אשר נשלט ע"י שלטי ה-Oculus Rift.

הצגה מפורטת של המסכים באפליקציה ניתן לראות בסעיף קודם או בנספח ה-SDD.

## 17. תכנון הפרויקט:

### סיכונים עיקריים ודרכי התמודדות:

#### 17.1.1 סיכון 1:

האטה והכבדה על זמן התגובה של המשחק משום האלגוריתם המורכב לפענוח קצב המוזיקה.

דרכי התמודדות:

17.1.1.1 לחפש כיצד אפשר לייעל את האלגוריתם ככל האפשר מבלי לפגוע

באיכות הביצוע.

17.1.1.2 פיתוח אלגוריתם שניגש לבעיה בצורה שונה.

התממשות:

הצלחה.

במהלך הפרויקט החלפנו את האלגוריתם לאלגוריתם "כבד" ומדויק יותר המודד שינוי בקצב המוזיקה כל 3 שניות ולכן לוקח גם יותר משאבים וזמן.

ע"מ לא להשפיע על המשחק, לא להאט את המשחק וגם לא לתת למשתמש לחכות בזמן הניתוח, הפרדנו את ניתוח המוזיקה למודול חיצוני אשר קורה מבעוד מועד ובסופו של דבר מערכת המשחק מקבלת רק קובץ עם וקטור השינויים של הBPM.

#### 17.1.2 סיכון 2:

תקינות האוקולוס ריפט שברשותנו. ישנו מכשיר אחד במעבדה, מכיוון שכל יכולות המשחק במציאות מדומה תלויות בתקינות המכשיר ברשותנו.

דרכי התמודדות:

17.1.2.1 לשמור את המכשיר בצורה המיטבית.

17.1.2.2 להתכונן להציג את המשחק גם בצורה שאינה מציאות מדומה.

17.1.2.3 לדאוג למכשיר חלופי.

התממשות:

הצלחה.

במהלך פיתוח המשחק נתקלנו גם בקושי חומרתי כאשר מכשיר האוקולוס התקלקל,

והמתנו תקופה קצרה עד שיירכש חדש וכך קרה.

כ-3 חודשים לפני הגשת הפרויקט, פרצה מגיפת הקורונה והרחיקה אותנו מהמעבדה וכמובן מהאוקולוס, סיכון שלא לקחנו בחשבון אך בעל אותה השפעה כמו הסיכון הזה, חוסר בצידוד. בזכות בדיקות וחתיירה לצידוד וכמובן היענות והסכמה מצד ניר ומחלקת פרויקטים, קיבלנו את הצידוד לבית וכך יכולנו להמשיך בפיתוח.

## 18. בדיקות והערכה:

א. כדי לוודא שהפרויקט עמד בציפיות ובייעדי התכנון כפי שהוגדרו בהגדרת הפרויקט תוכננו מספר בדיקות שיבדקו את החלקים השונים במערכת בחלוקה לשכבות. השכבות שנבדקו הן: שכבת ממשק המשתמש. שכבת הלוגיקה. שכבת הנתונים. כלל הבדיקות עברו בהצלחה והתוצר עומד ביעדיו. כלל הבדיקות שקבענו עברו בהצלחה והמוצר עמד ביעדים.

ב.1. דוגמת הפעלה 1:

1. The game is starting and Main-Menu shown.
2. The player click on "Hall Of Fame" button.
3. The Hall Of Fame scene opened.
4. The player can watch the highscores.
5. The High Scores table is sorted by the scores.
6. There is 0-10 scores in the table.
7. The player clicks the "MENU" button.
8. The "Main Menu" scene shown.

## ב.2. דוגמת הפעלה 2:

1. The game is starting and Main-Menu shown.
2. The player click on the Options button.
3. The Options screen opened.
4. The player selects a song from the list.
5. The player close the Options screen by click the Options button.

6. The Main Menu scene will shown.
7. The player will select to start a new game.
8. The game will start with the selected song playing in background.

ג. בדיקות שמישות:

לצערנו עקב אילוצים של מגפת הקורונה, לא ניתן היה לשתף אנשים ולתת להם לבחון את התוצר, לכן לא נעשו בדיקות שמישות מקיפות. עם זאת, היו בדיקות שמישות 'לא רשמיות' של בני הבית שעל פיהן שיפרנו את העיצוב, הוספנו עוד תכונות והגדלנו את הכתב במהלך המשחק.

ד. בדיקת מדדים:

תוצאות המדדים בסיום הפיתוח:

9.3.2.1 - זמן התגובה הממוצע קטן לפחות מ-50%.

9.3.2.2 - 100% מהשחקנים העידו שהמשחק מהנה אך עם זאת מספר השחקנים הינו

מועט בשל אילוצי מגפת הקורונה.

9.3.2.3 - האובייקטים מוצגים מיידית.

9.3.2.4 - במהלך המשחק המערכת סוכמת את הנקודות והנתונים ומציגה מיד בסיום.

9.3.2.5 - המשחק מצליח להציג עשרות אובייקטים במקביל.



## 19. סיכום ומסקנות:

תהליך העבודה על הפרויקט כלל מחקר מעמיק בכמה נושאים, ראשית נחקר הפיתוח למערכות מציאות מדומה. במהלך מחקר זה ניתן דגש לשימוש במחווות משתמש עם תפריטים מיוחדים שיפעלו עם שלטי קונסולת המציאות המדומה ביחד עם מעקב מבט המשתמש.

תוצאות מחקרים אלו הובילו את הפרויקט להיות ממומש באמצעות העורך של Unity לשימוש מיטבי באובייקטים בתלת מימד עם התוסף של Oculus למפתחים. השלב הבא היה תחילת עבודה על האב-טיפוס שהוא מימוש של שכבת ממשק המשתמש שכללה עבודה עם העורך של Unity וכתיבה ב C#

חלק חשוב מן הפרויקט היה למצוא דרך לנתח את המוזיקה ולתת תוצאה כמה שיותר מדויקת של BPM. בעזרת סקירה ספרותית מצאנו את האלגוריתם היעיל והטוב ביותר ופתחנו אותו לצרכים שלנו בהתאמה.

ניתוח המוזיקה הינו האתגר ההנדסי ואחת המטרות המרכזיות, אך לדעתנו לא פחות חשוב במשחק מחשב שהייה מהנה. כפי שצויין, עקב המגבלות היה קושי לסקור קהל רחב ולשאול אך אנו שבעי רצון מהתוצר וחושבים שהוא מאוד מהנה.

אחת המסקנות הנובעות מהפרויקט היא פתיחות מחשבתית, גם אם יש כבר משהו שעובד, למשל האלגוריתם, אולי כדאי להרהר ולחפש אחד שעובד טוב יותר.

בנוסף, תמיד כדאי ומומלץ לשפר, לעבוד להוסיף פיצ'רים למשחק וכך לאט לאט המשחק שלנו השתפר.

מסקנה חשובה ואחרונה היא ששמירה על הסדר תרמה לנו רבות.

מסמכים מסודרים, גרפים, יעדים וחלוקת עבודה חוסכת זמן רב ונותנת הכוונה, אך עם זאת כמו שצינו, יש תמיד מקום לחשוב ולפתוח את הראש אולי אפשר לשנות.

## 20. הצעה לעבודת המשך:

לפריקט זה יכולים להיות מספר פיתוחים ויישומים עתידיים בתחומים שונים:

- הוספת יכולת להעלאת שיר ישירות למשחק וניתוח שלו בזמן אמת.
- שיפור אלגוריתם ניתוח השיר. במידה וייעשה בזמן אמת יש לשים על כך את מירב המאמצים והמשאבים כולל מיקבול ושימוש בכרטיס גרפי על מנת למקסם את הביצועים ולצמצם את זמן ההמתנה.
- הוספת רקעים וסביבות משחק שונות. למשל אפשרות לבחור שהמשחק ינוהל בחוף הים והאובייקטים בהתאם.
- הוספת מודולים של ML וחיזוק הAI בניתוח ביצועי השחקן והתאמת הקושי אליו, קיים היום אך ניתן לשיפור.
- הוספת רמות קושי.
- עיצוב! לדעתנו תמיד ניתן לשפר עיצוב למשחק ולהעלות את הרמה.

[1] Levels of Sound: On the Principles of Interactivity in Music Video Games.

M Pichlmair, F Kayali - DiGRA Conference, 2007

[2] Robert Kelly. Mathematics of Musical Rhythm. Departmental honors thesis, under the direction of Rachel

W. Hall. Published at <http://www.sju.edu/~rhall/Rhythms/bobby.pdf>. Saint Joseph's University, 2002.

[3] Tzanetakis, George & Essl, Georg & Cook, Perry. (2001). Audio Analysis using the Discrete Wavelet Transform. Proceedings of the Conference in Acoustics and Music Theory Applications. 318-323.

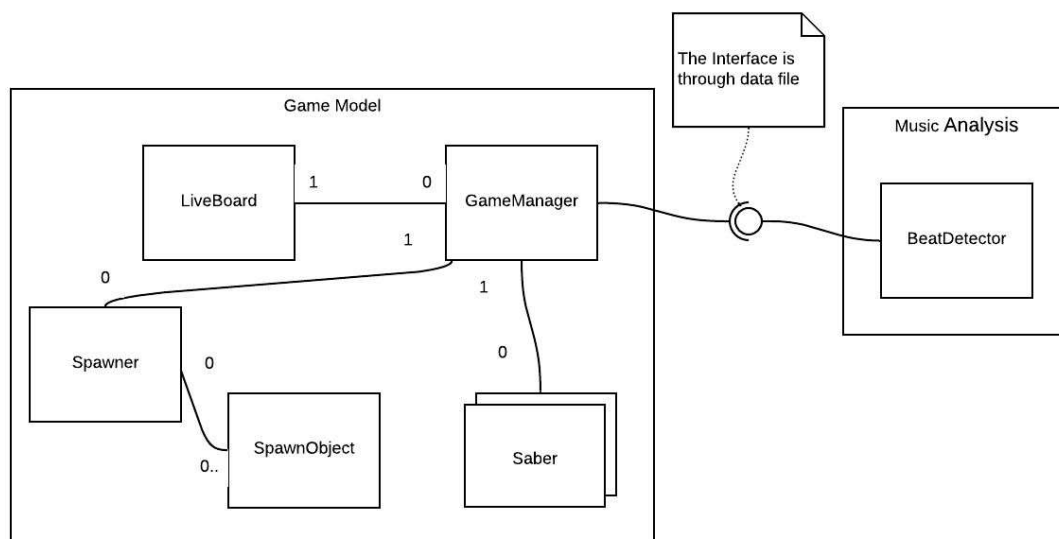
[4] Sherman, William R., and Alan B. Craig. Understanding virtual reality: Interface, application, and design. Elsevier, 2002.

**Software Requirements Document****1. Introduction**

Our game is a VR game, which requires a PC and Oculus VR system, the goal of the game is to hit objects with two swords (controlled by Oculus handles). The objects thrown by the tempo of the music. Each hit on the objects rewards with extra points.

**2. Model description**

Our model is a VR game using Oculus headset and handles. The backbone of the production of the game is with Unity game engine and development IDE, and our own music analysis module. The architecture is presented in the diagram below:



גרף 22.1.1 - תרשים בלוקים של המערכת

the reasons for this architecture are obvious:

1. Unity is a free and very popular game engine and development especially for beginners and small teams, everything is “out of the box” solution for development.
2. Oculus is available in our labs, and has very good connectivity with Unity.

### 3. Specific Requirements

#### 3.1 Functional Requirements

##### Actors & Goals

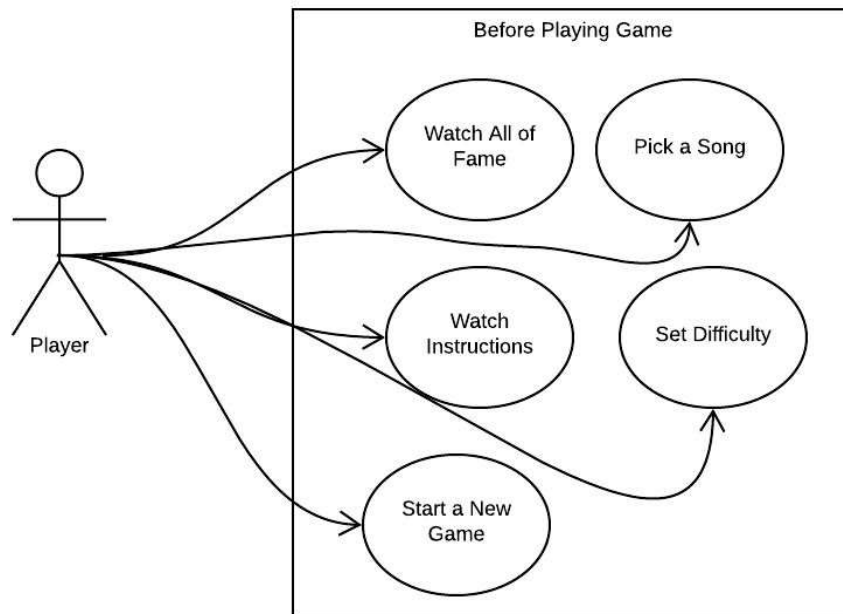
Actor name: Player

Type: primary

Goals:

- Play games
- Watch other players scores

## Use Case Diagram



גרף 22.1.2 UseCase Diagram של המסך הראשי

## Use Cases Details

### Start New Game

Main Actor: Player

Goal: the Player wishes to start a new game, System generate new game for Player.

Basic flow:

<ul style="list-style-type: none"><li>• Player enters game</li><li>• Player wish to start a new game</li></ul>	
	<ul style="list-style-type: none"><li>• System Generate a new game for Player in the level</li></ul>

	she chose
--	-----------

טבלה 22.1.1 - תיאור זרימה ראשית של התחלת משחק

### Play Game

Actor: Player

Goal: Player wants to play the game

Basic Flow:

<ul style="list-style-type: none"> <li>• Player is in the game screen</li> </ul>	
	<ul style="list-style-type: none"> <li>• Game start throwing objects at the player</li> </ul>
<ul style="list-style-type: none"> <li>• Player tries to hit objects and win points</li> <li>• Player tries to avoid hitting obstacles</li> </ul>	
	<ul style="list-style-type: none"> <li>• Game awards Player with points for every correct hit</li> <li>• Game ends</li> <li>• System show player his records</li> </ul>
<ul style="list-style-type: none"> <li>• Player leaves game</li> </ul>	

טבלה 22.1.2 - תיאור זרימה ראשית של מהלך המשחק

Alternate Flow:

<ul style="list-style-type: none"> <li>• Player is in the game screen</li> </ul>	
	<ul style="list-style-type: none"> <li>• Game start throwing objects at the player</li> </ul>

<ul style="list-style-type: none"> <li>• Player tries to hit objects and win points</li> <li>• Player tries to avoid hitting obstacles</li> </ul>	
	<ul style="list-style-type: none"> <li>• Game awards Player with points for every correct hit</li> <li>• Game ends</li> <li>• System show player his records</li> <li>• System notice that Player hit a record score</li> <li>• System ask Player's nickname</li> </ul>
<ul style="list-style-type: none"> <li>• Player gives a nickname</li> </ul>	
	<ul style="list-style-type: none"> <li>• System saves the record with the nickname given by Player</li> </ul>
<ul style="list-style-type: none"> <li>• Player leaves game</li> </ul>	

טבלה 22.1.3 - תיאור זרימה משנית של מהלך המשחק

### Watch Hall of Fame

Actor: Player

Goal: Player wants to see her and other players records

Basic Flow:

<ul style="list-style-type: none"> <li>• Player wishes to watch top game records</li> </ul>	
	<ul style="list-style-type: none"> <li>• System show Player top</li> </ul>



	records of the game
<ul style="list-style-type: none"> <li>• Player watches the top records</li> </ul>	

טבלה 22.1.4 - תיאור זרימה ראשית של צפייה בטבלת השיאים

### Watch Game Instructions

Actor: Player

Goal: the player wants to know how to play the game

Basic flow:

<ul style="list-style-type: none"> <li>• Player wants to know how to play the game</li> </ul>	
	<ul style="list-style-type: none"> <li>• System Show Player the game instructions</li> </ul>
<ul style="list-style-type: none"> <li>• Player learn the game rules</li> </ul>	

טבלה 22.1.5 - תיאור זרימה ראשית של צפייה בהוראות המשחק

### Pick Song

Actor: Player

Goal: the player set a song for the game

Basic flow:

	<ul style="list-style-type: none"> <li>• System ask Player to pick a song</li> </ul>
<ul style="list-style-type: none"> <li>• Player picks song</li> </ul>	
	<ul style="list-style-type: none"> <li>• System saves the song</li> </ul>

	preference
--	------------

טבלה 22.1.6 - תיאור זרימה ראשית של בחירת שיר

### Pick Difficulty

Actor: Player

Goal: the player set the level of difficulty for the game

Basic flow:

	<ul style="list-style-type: none"> <li>System ask Player to pick</li> </ul>
<ul style="list-style-type: none"> <li>Player picks level of difficulty</li> </ul>	
	<ul style="list-style-type: none"> <li>System saves the level of difficulty</li> </ul>

טבלה 22.1.7 - תיאור זרימה ראשית של בחירת רמת קושי

## **3.2 Non-Functional Requirements**

### **3.2.1 Performance requirements**

- Less than 150 ms average reaction time for the player action

### **3.2.2 Resource requirements**

- Windows 10 OS
- Oculus Rift VR System

### **3.2.3 Security requirements**

The whole game is installed locally and in addition there is no sensitive information stored in this application, therefore there is no security requirements.

## Software Design Description

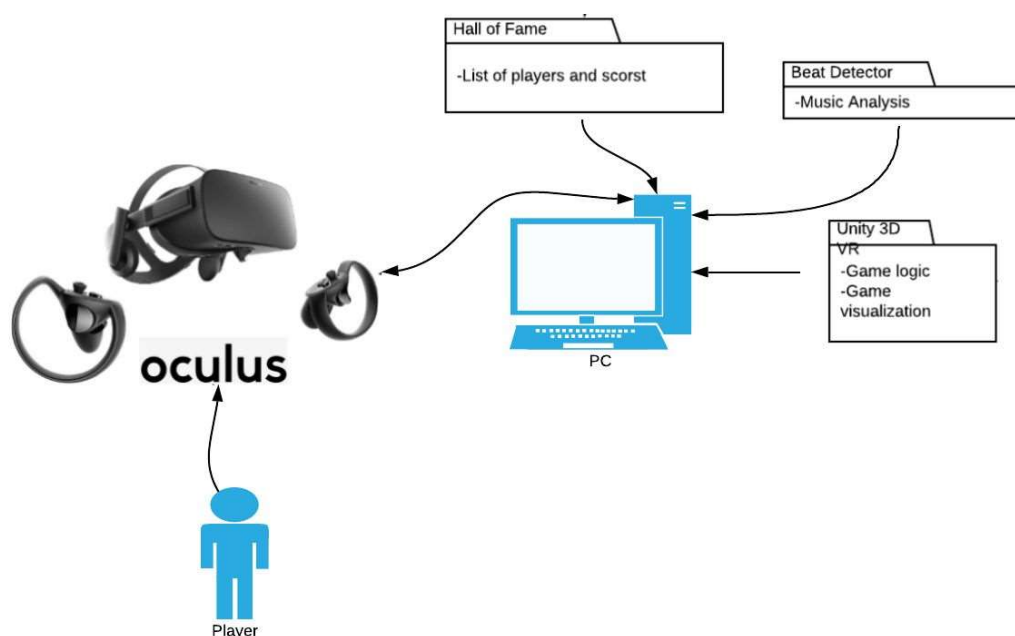
### 1. Introduction

Our game is a VR game, which requires a PC and Oculus VR system, the goal of the game is to hit objects with two swords (controlled by Oculus handles). The objects thrown by the tempo of the music. Each hit on the objects rewards with extra points.

### 2. System Architectural Design

#### 2.1 Chosen System Architecture

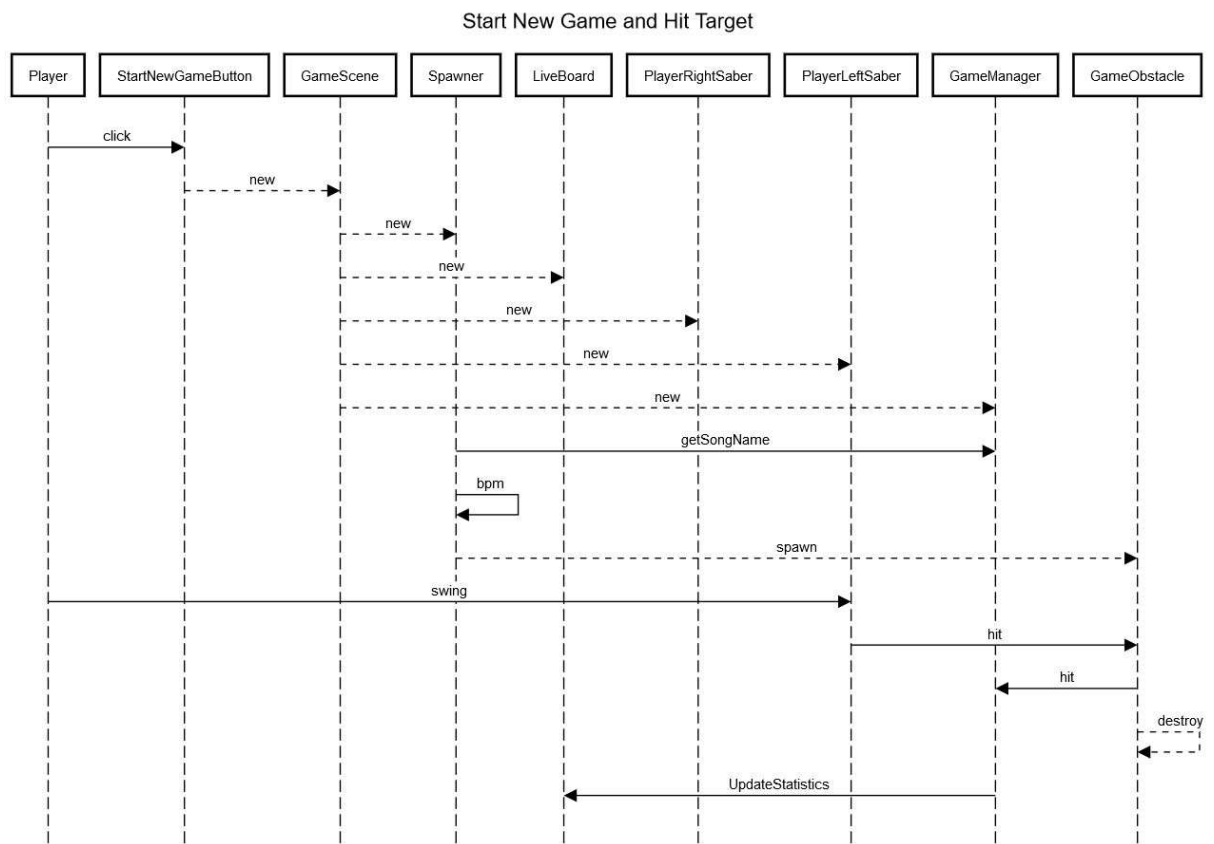
Our model is a VR game using Oculus headset and handles. The backbone of the production of the game is with Unity game engine and development IDE, and our own music analysis module. The architecture is presented in the diagram below:



גרף 22.2.1 - ארכיטקטורת המשחק

## 2.2 System Interface Description

### Start New Game



גרף 22.2.2 - Sequence Diagram של המשחק

## 3. Detailed Description of Main Components

### 3.1 Game Scene

**3.1.1 Type:** The main screen of the game, 3D environment

**3.1.2 Purpose:** Where the game is played

**3.1.3 Functional Inputs & Outputs:** Present to the player the game objects in 3D and music. The Player moves and hit with his handles.

**3.1.4 Interfaces:** Seen on the Oculus headset goggles, and react to the player movement and controls.

**3.1.5 Data:** there is data that the Unity modules manages.

## **3.2 Music Beat Detector**

**3.2.1 Type:** A separate application that analyze the song and create a file for the game to take the tempo from.

**3.2.2 Purpose:** gets the audio file of the music and analyze its tempo for the timed objects that thrown at the player direction.

**3.2.3 Functional Inputs & Outputs:** audio file as input, tempo as outputs.

**3.2.4 Interfaces:** The game has to each song the tempo analyzed in advance.

**3.2.5 Data:** The audio file is translated to an array, and the tempo that returns is an array of frames of tempo (every few seconds of the song).

## **3.3 Spawner**

**3.3.1 Type:** An object on the scene(not seen).

**3.3.2 Purpose:** Creates objects(cubes and obstacles) randomly, and throw them towards the player position.

**3.3.3 Functional Inputs & Outputs:** Input: the tempo of the music of the game level. Output: gameobject - cubes and obstacles.

**3.3.4 Interfaces:** Not visible to the player.

**3.3.5 Data:** Unity handles behavior.

### **3.4 LeftSaber and RightSaber**

**3.4.1 Type:** Two gameobjects 3D bats.

**3.4.2 Purpose:** Hit the objects thrown at the player.

**3.4.3 Functional Inputs & Outputs:** The player swings game handles supplied by Oculus. The game logic reacts to the player swing and direction. When intersect with other objects it destroys them if hit right.

**3.4.4 Interfaces:** Seen on the screen as first person, controlled by the player.

**3.4.5 Data:** Unity handles data.

### **3.5 GameManager**

**3.5.1 Type:** An object

**3.5.2 Purpose:** manages the points and strikes of the player in the game

**3.5.3 Functional Inputs & Outputs:** checks if an object hit or missed, then updates points or lives in the game.

**3.5.4 Interfaces:** Only the game objects interface with the Game Manager, not the player.

**3.5.5 Data:** GameManager saves the data on memory, during the game, if the player reaches high score it can save the points and name in the database.

## **4 User Interface Design:**

### **4.1 Main Menu Scene :**

#### **4.1.1 Description of the user interface**

Here the player can choose if to start new game, to watch Hall Of Fame, set options, read instructions or watch the credits.

#### **4.1.2 Objects and Actions**

<b><u>Object</u></b>	<b><u>Action</u></b>
Start New Game Button	Click
Hall of Fame Button	Click
Options Button	Click
How to Play Button	Click
Credits Button	Click

טבלה 22.2.1 - פעולות משתמש במסך הראשי של המשחק

#### **4.1.3 Screen Image**



איור 22.2.1 - מסך ראשי של המשחק

## 4.2 Hall Of Fame :

### 4.2.1 Description of the user interface

Here the player can watch the highscores of all players from all difficulty.

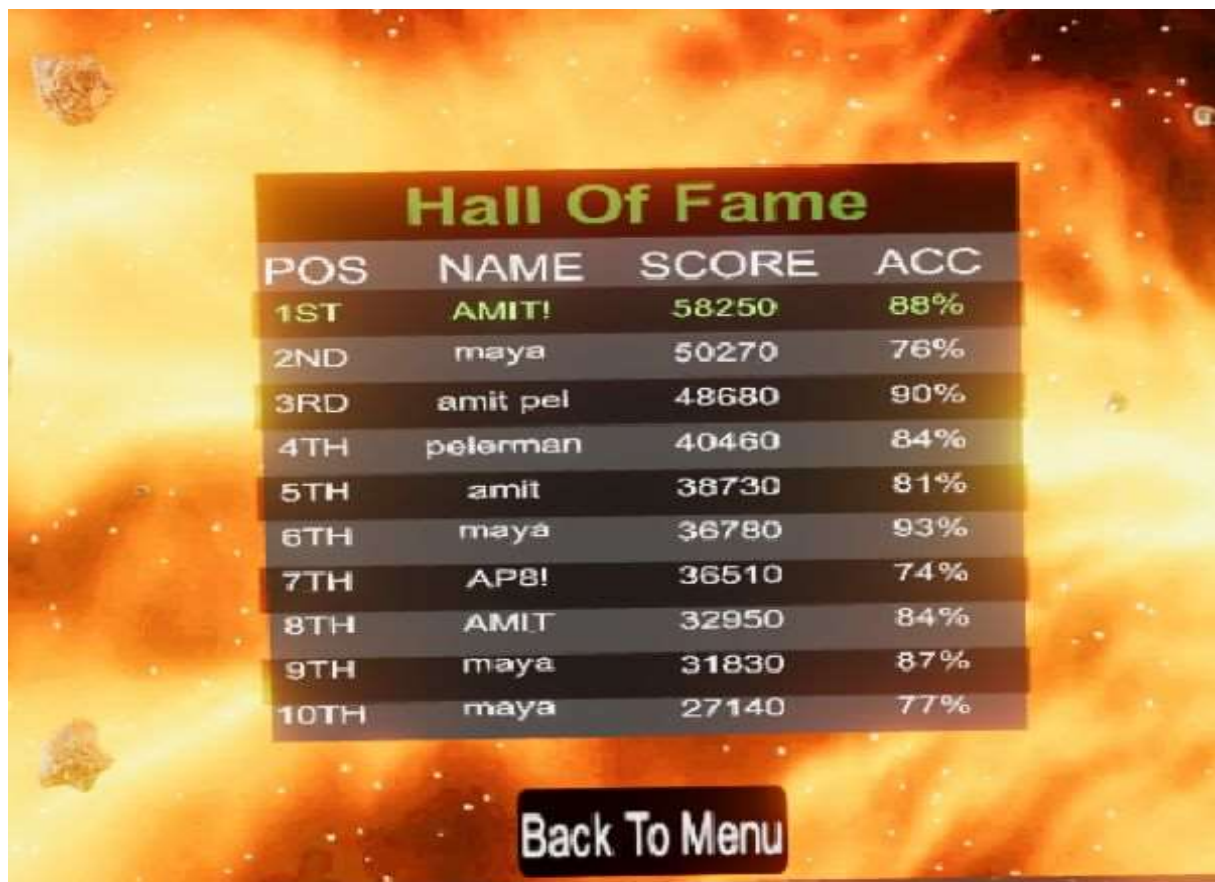
### 4.2.2 Objects and Actions

<u>Object</u>	<u>Action</u>
Hall of Fame Table	Watch only
Back to Menu Button	Click

טבלה 22.2.2 - פעולות משתמש במסך טבלת השיאים



### 4.2.3 Screen Image



איור 22.2.2 צילום מסך השיאים של המשחק

### 4.3 Instructions screen:

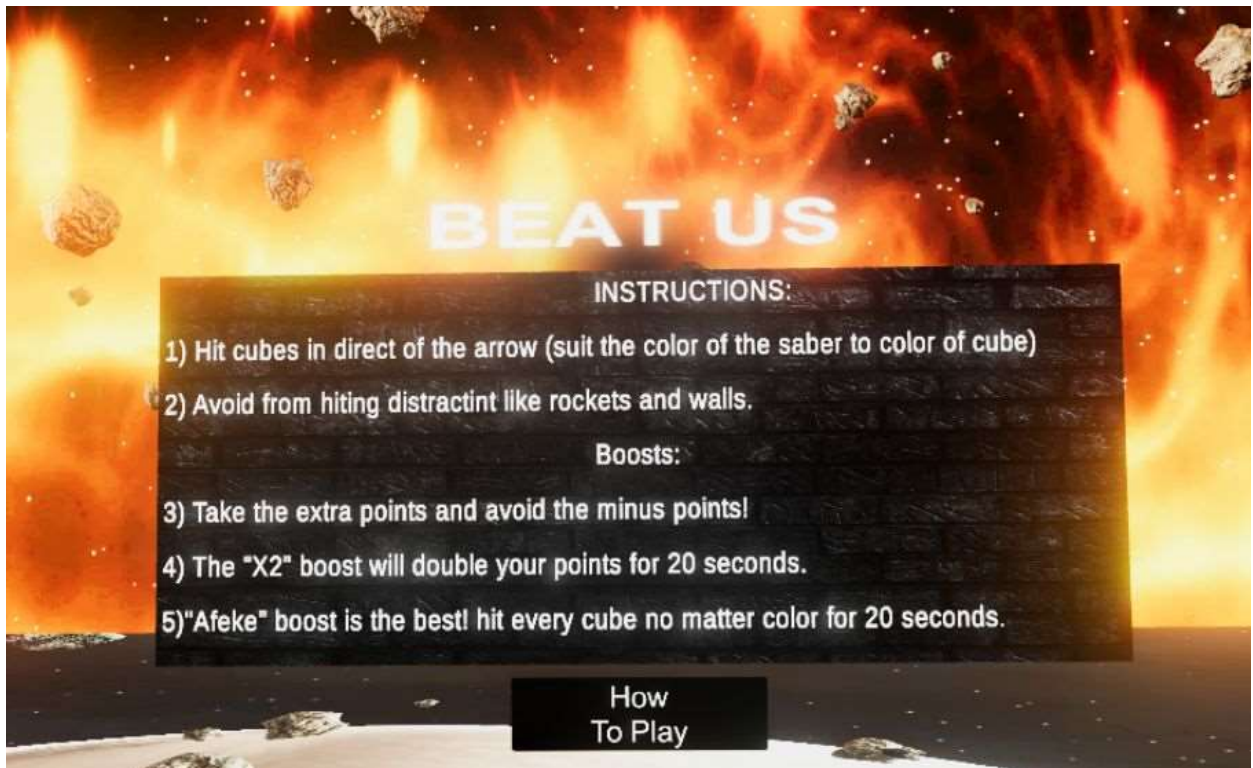
#### 4.3.1 Description of the user interface

Here the player can watch the instructions and learn the rules how to play the game.

#### 4.3.2 Objects and Actions

<u>Object</u>	<u>Action</u>
Instructions text	Watch only
Back to Menu Button	Click

### 4.3.3 Screen Image



איור 22.2.3 - צילום מסך הוראות המשחק

## 4.4 Options Screen:

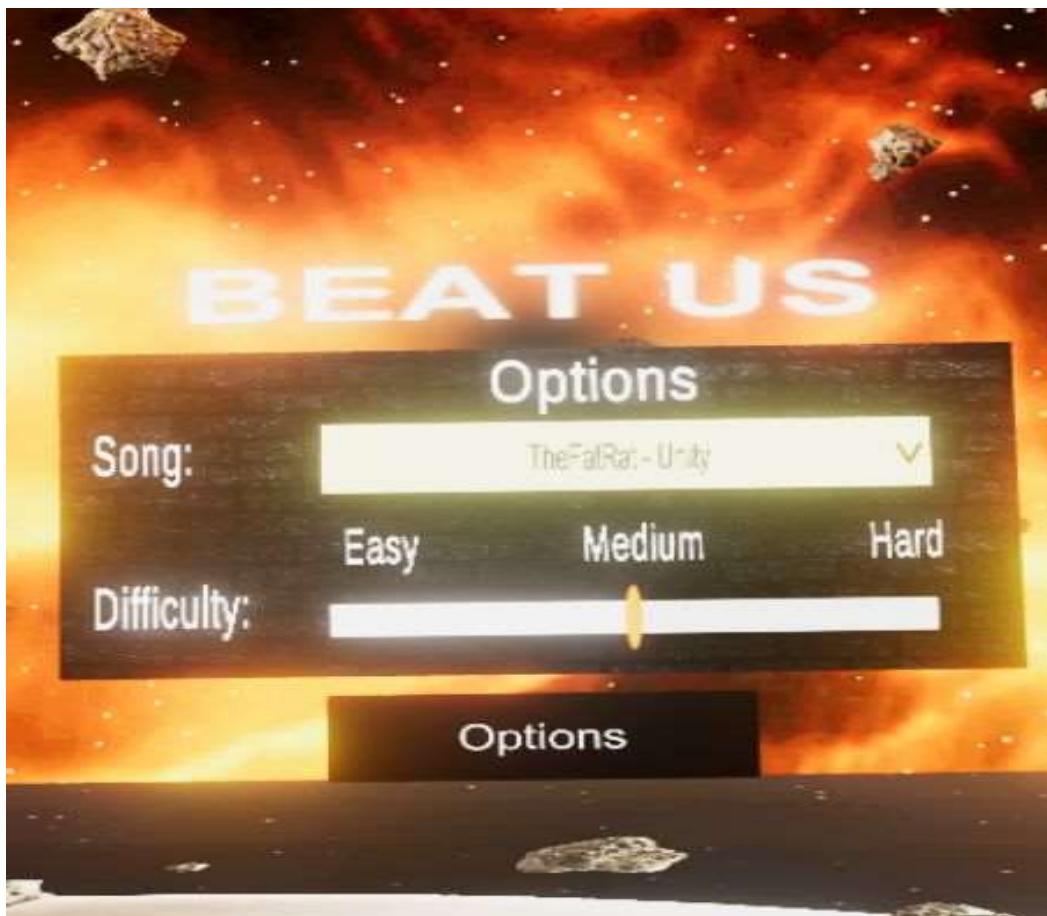
### 4.4.1 Description of the user interface

Here the player can set the difficulty and to pick a song to play with.

### 4.4.2 Objects and Actions

<u>Object</u>	<u>Action</u>
Songs Drop Table	Pick one
Difficulty track bar	Pick one
Back to Menu Button	Click

#### **4.4.3 Screen Image**



איור 22.2.4 - צילום מסך האפשרויות

#### **4.5 Game Screen :**

##### **4.5.1 Description of the user interface**

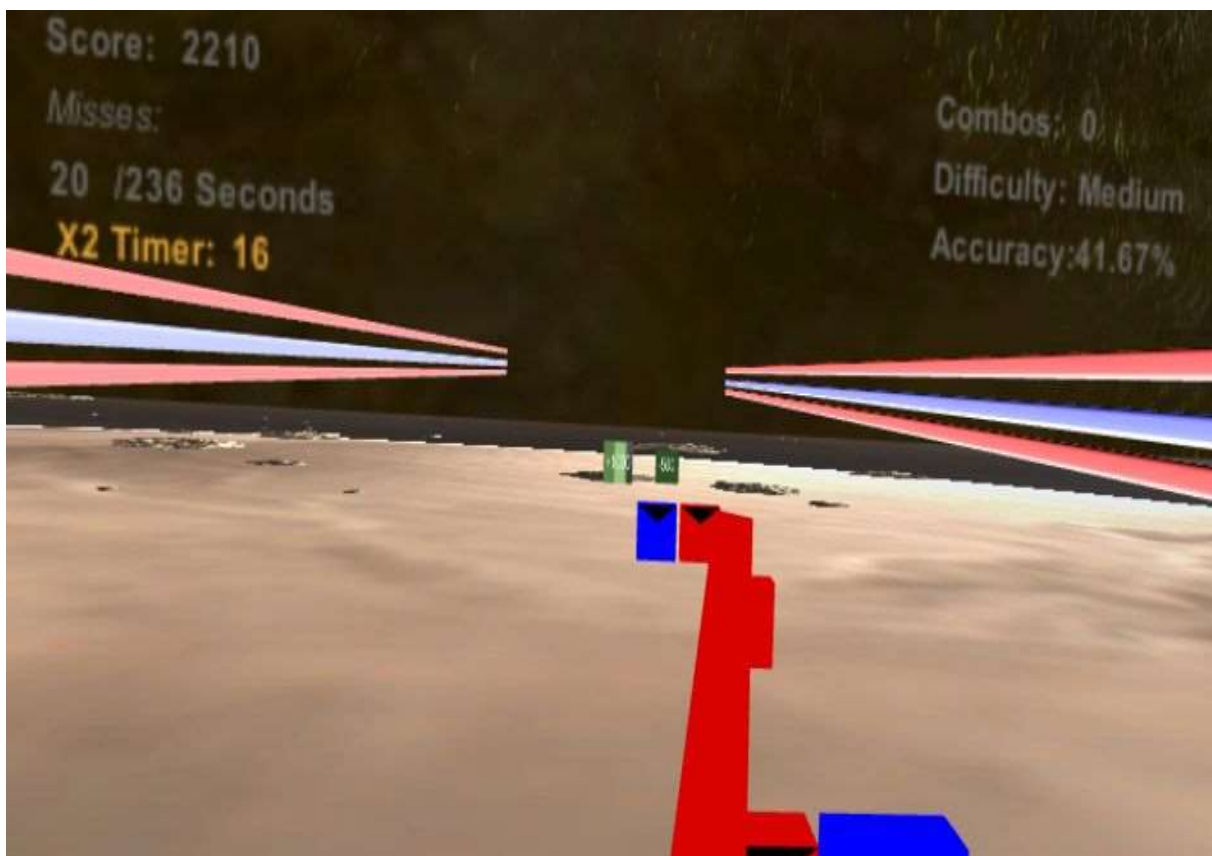
This is actually the game we developed. Here the player will play the game and his the red/blue cubes with his sabers.

#### 4.5.2 Objects and Actions

<u>Object</u>	<u>Action</u>
Right Bat handle	swing
Left Bat handle	swing
Strikes counter	accumulate
Cubes	thrown at the player
Points counter	accumulate

טבלה 22.2.5 - פעולות משתמש במשחק

#### 4.5.3 Screen Image



איור 22.2.5 - צילום מהלך המשחק

## **Software Test Documentation (STD):**

### **1. Introduction:**

In this part we will describe the tests that we will performed in the system to ensure compliance with the requirements and objectives we have set. We will set the test and how to do it, as well as the expected result.

### **2. QA and Sanity Test:**

#### **a. Features to be tested**

- Test Integration with Oculus headset and handles
- Test responsiveness of the handles with the screen bats.
- Test the objects running towards the player in a changing rhythm.
- Test the that the database is well interacted with the game.

#### **b. Features not to be tested**

- Test the game on VR Headset that are not Oculus Rift's brand.
- Test the game with "weak computers" which has the minimum requirements to play the game but not properly.
- Test the Oculus Rift's performances and limitations.

#### **c. Environmental needs**

- Oculus Rift - VR headset and handles.
- Unity engine must be installed on the computer with minimum version 2019.1.1.

- PC or laptop that has at least the minimum requirements and system specifications for Oculus Rift S - can be found in the following link:  
<https://support.oculus.com/248749509016567/>

#### **d. Main Test Cases**

##### **2.4.1 Case – 1:**

Options button - pick a song.

##### **2.4.1.1 Purpose:**

Check if song that had been selected in the Options will be played in the game.

##### **2.4.1.2 Inputs**

2.4.1.2.1 Selected song from list.

##### **2.4.1.3 Expected Outputs & Pass / Fail Criteria:**

2.4.1.3.1 The song will be played in the game.

##### **2.4.1.4 Test Procedure:**

2.4.1.4.1 The game is starting and Main-Menu shown.

2.4.1.4.2 The player click on the Options button.

2.4.1.4.3 The Options screen opened.

2.4.1.4.4 The player selects a song from the list.

2.4.1.4.5 The player close the Options screen by click the Options button.

2.4.1.4.6 The Main Menu scene will shown.

2.4.1.4.7 The player will select to start a new game.

2.4.1.4.8 The game will start with the selected song playing in background.

## **2.4.2 Case – 2:**

Watch Instructions.

### **2.4.2.1 Purpose:**

Check the Instructions screen.

### **2.4.2.2 Inputs - קלטים:**

2.4.2.2.1 Click on “How To Play” button.

### **2.4.2.3 Expected Outputs & Pass / Fail Criteria:**

2.4.2.3.1 The Instructions screen opened.

### **2.4.2.4 Test Procedure:**

2.4.2.4.1 The game is starting and Main-Menu shown.

2.4.2.4.2 The player click on “How To Play” button.

2.4.2.4.3 The Instructions screen opened.

2.4.2.4.4 The Instructions screen will include the instructions to the game and the player can read it.

2.4.2.4.5 The player clicks the Instruction button.

2.4.2.4.6 The Instructions screen closed and main menu shown.

### **2.4.3 Case – 3:**

Watch Hall Of Fame.

#### **2.4.3.1 Purpose:**

Check the Hall Of Fame scene.

#### **2.4.3.2 Inputs - קלטים:**

2.4.3.2.1 Click on “Hall Of Fame” button.

#### **2.4.3.3 Expected Outputs & Pass / Fail Criteria:**

2.4.3.3.1 The Highscore table will be shown.

#### **2.4.2.4 Test Procedure:**

2.4.3.4.1 The game is starting and Main-Menu shown.

2.4.3.4.2 The player click on “Hall Of Fame” button.

2.4.3.4.3 The Hall Of Fame scene opened.

2.4.3.4.4 The player can watch the highscores.

2.4.3.4.5 The High Scores table is sorted by the scores.

2.4.3.4.6 There is 0-10 scores in the table.

2.4.3.4.7 The player clicks the “MENU” button.

2.4.3.4.8 The “Main Menu” scene shown.

### **2.4.3 Case – 4:**

Bats hit target objects

#### **2.4.4.1 Purpose:**



Check if the bats hit an object and the game reacts to it.

#### **2.4.4.2 Inputs - קליטים:**

2.4.4.2.1 Swinging bats at object (A cube).

#### **2.4.4.3 Expected Outputs & Pass / Fail Criteria:**

2.4.4.3.1 The live score will be raise by 100 points.

#### **2.4.2.4 Test Procedure:**

2.4.4.4.1 The game is starting and Main-Menu shown.

2.4.4.4.2 The player click on "New Game" button.

2.4.4.4.3 The game will start.

2.4.4.4.4 The system spawn cubes that coming forward to player.

2.4.4.4.5 The player will hit the red cube with his red saber by the arrow.

2.4.4.4.6 The cube will be crushed and disappear.

2.4.4.4.7 The live score will be raise by 100 points.

#### **2.4.5 Case – 5:**

Finish game.

#### **2.4.5.1 Purpose:**

Check the finish game screen.

#### **2.4.5.2 Inputs - קלטים:**

2.4.5.2.1 Finish to play a game.

#### **2.4.5.3 Expected Outputs & Pass / Fail Criteria:**

2.4.3.3.1 The Finish screen will show with the following details: score, hits, combos, button of Menu and restart. In case of achieving new highscore: Name field and "Save Highscore" button.

#### **2.4.5.4 Test Procedure:**

2.4.5.4.1 The game is starting and Main-Menu shown.

2.4.5.4.2 The player click on "New Game" button.

2.4.5.4.3 The game starting.

2.4.5.4.4 The system spawn cubes that coming forward to player.

2.4.5.4.5 The player plays the game.

2.4.5.4.6 The song was over.

2.4.5.4.7 There is no more cubes in the screen.

2.4.5.4.8 The Live score in the top left screen disappeared.

2.4.5.4.9 The Finish screen shown.

2.4.5.4.10 The Finish screen will show the score,hits and combos of the last game.

2.4.5.4.11 The player clicks the Restart button.

2.4.5.4.12 The game will restart and play the same song.

#### **2.4.6 Case – 6:**

Game Over

##### **2.4.6.1 Purpose:**

Check the Game Over working well.

##### **2.4.6.2 Inputs - קלט:**

2.4.6.2.1 Play the game without hit the objects.

##### **2.4.6.3 Expected Outputs & Pass / Fail Criteria:**

2.4.6.3.1 The Finish screen will show after 5 missed objects with “Game Over” title and will play a sound of “GAME OVER”.

##### **2.4.6.4 Test Procedure:**

2.4.6.4.1 The game is starting and Main-Menu shown.

2.4.6.4.2 The player click on “New Game” button.

2.4.6.4.3 The game starting.

2.4.6.4.4 The system spawn cubes that coming forward to player.

2.4.6.4.5 The player ignore the cubes and doesn't do anything.

2.4.6.4.6 The system will add “X” in the top left, under the score when each cube hit the player body.

2.4.6.4.7 The system will update the score top left with 80 points less.

2.4.6.4.8 The Finish screen will show when the 5 "X" will appear.

2.4.6.4.9 The system will stop the spawn of objects and all the objects disappear.

2.4.6.4.11 The system will stop the song.

2.4.6.4.12 The system will play a sound of "Game Over".

2.4.6.4.13 The Finish title will be "Game Over".

2.4.6.4.14 The Score will be -400.

2.4.6.4.15 The Combos will be 0.

2.4.6.4.16 The Hits will be 0.

2.4.6.4.17 The player will select the Menu Button.

2.4.6.4.18 The Main Menu scene will be shown.

#### **2.4.7 Case – 7:**

Save High Score

##### **2.4.7.1 Purpose:**

Check the add score to High Scores.

##### **2.4.7.2 Inputs - קלטים:**

2.4.7.2.1 Play the game.

##### **2.4.7.3 Expected Outputs & Pass / Fail Criteria:**

2.4.7.3.1 The High Score table will update with the player score.

##### **2.4.7.4 Test Procedure:**

2.4.7.4.1 The game is starting and Main-Menu shown.

2.4.7.4.2 The player clicks on "New Game" button.

2.4.7.4.3 The game starting.

- 2.4.7.4.4 The system spawn cubes that coming forward to player.
- 2.4.7.4.5 The player plays according to the rules to gain maximum points.
- 2.4.7.4.6 The system will raise the score according to the player hits.
- 2.4.7.4.7 The song was over.
- 2.4.7.4.8 There is no more cubes in the screen.
- 2.4.7.4.9 The Live score not shown anymore.
- 2.4.7.4.10 The Finish Screen shown with title: "HIGH SCORE"
- 2.4.7.4.11 There is fireworks on the screen and player can hear it.
- 2.4.7.4.12 The Score is positive number.
- 2.4.7.4.13 The rest of the numeric fields is positive.
- 2.4.7.4.14 There is "Save High Score" button exist in the screen.
- 2.4.7.4.15 The player clicks the "Save High Score" button without fill name.
- 2.4.7.4.16 There is no change in screen and "wrong" sound play.
- 2.4.7.4.17 The player type his name in the Name field.
- 2.4.7.4.18 The player clicks the "Save High Score" button.
- 2.4.7.4.19 Confirmation sound played.
- 2.4.7.4.20 The "Save High Score" button is disable.
- 2.4.7.4.21 The player clicks the "Menu" button.
- 2.4.7.4.22 The "Main Menu" scene shown.
- 2.4.7.4.23 The player clicks the "Hall Of Fame" button.
- 2.4.7.4.24 The player clicks the "Save High Score" button.
- 2.4.7.4.25 The player's High Score shown once only with his name and score.
- 2.4.7.4.26 The High Score table is sorted by scores..

### **3. Functional & Usability Tests**

We will form test users group and let them play the game.

We will ask them if they understand and enjoy to play the game.

#### **3.2:**

We will check the indices that we set in the sow.

### **4. Schedule**

Test	Start Date	End Date	Duration
Options button - pick a song	7.5.20	7.5.20	1 day
Watch Instructions	7.5.20	7.5.20	1 day
Watch Hall of Fame	7.5.20	7.5.20	1 day
Bats hit target objects	7.5.20	10.5.20	3 days
Check the finish game screen	8.5.20	10.5.20	2 days
Check the Game Over working	10.5.20	10.5.20	1 day
Save High Score	11.5.20	12.5.20	2 days
Usability Testing	12.5.20	15.5.20	4 days

טבלה 22.3.1 - לוח זמנים לבדיקות המערכת

## 22.4 נספח פוסטר הפרויקט:

המכללה המקומית  
להנדסה בתל-אביב

**אפקה**

### הנדסת תוכנה

עמית פלרמן וגרי דבנ  
מנחה: ד"ר חנן גזית

ארכיטקטורת ממשקי המערכת:

בתמונה: על"ית רמה בהתאם לביצועי השחקן ולוח אשר נותן נתונים על המשחק בזמן אמת.

בתמונה: מצב מיוחד שקורה בעקבות פגיעה ב"בוסט" ייחודי במהלך המשחק.

### BEAT US

#### משחק ספורטיבי מבוסס מוזיקה בטכנולוגיית מציאות מדומה

**הפרויקט הינו משחק מציאות מדומה מבוסס על אלגוריתם ניתוח מוזיקה.**

**במהלך המשחק, השחקן משתמש בחרבות וירטואליות כדי לפגוע בקוביות צבעוניות שונות שמוצגות לו בהתאם לקצב המוזיקה.**

**המשחק כולל שלוש רמות קושי ומודול בינה מלאכותית שמאתגר את המתשמש בהתאם לרמת הביצועים שלו.**

**דרישות המערכת:**

- האובייקטים מוצגים ומעים לכיוון השחקן על פי קצב המוזיקה שמנותח על ידי אלגוריתם בינה מלאכותית.
- בעת פגיעה נכונה עם החרב או לא נכונה, מוצג לשחקן חיזוי ומשוב בזמן אמת.
- רמת הקושי של המשחק משתנה ומסתגלת בהתאם לביצועי השחקן ומימונתו.

ארכיטקטורת המערכת:

איור 22.4.1 -פוסטר הפרוייקט

### נספח מאמר:

בחרנו לצרף את מאמר [3] שלדעתנו היה המאמר המשפיע ביותר על הפרויקט.

המאמר מופיע בדיסק תחת תיקיית "ARTICLE".