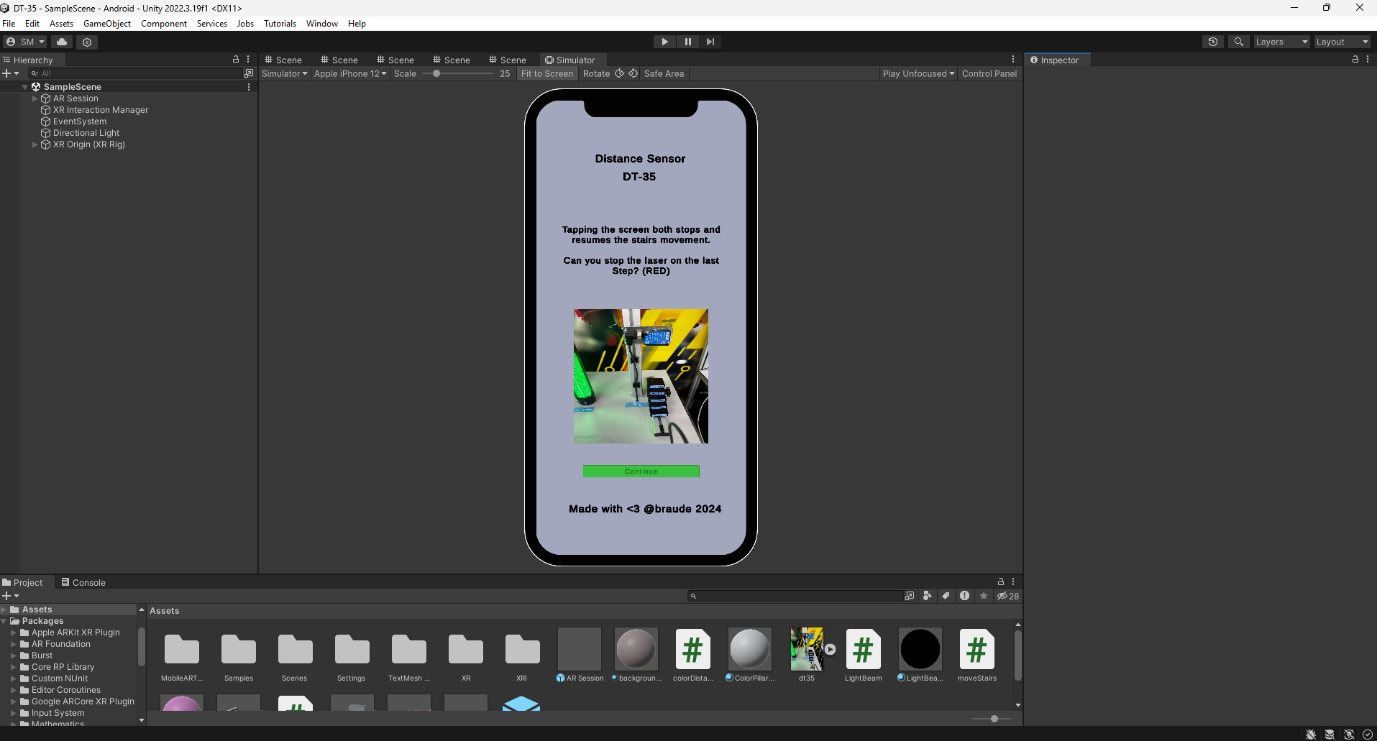
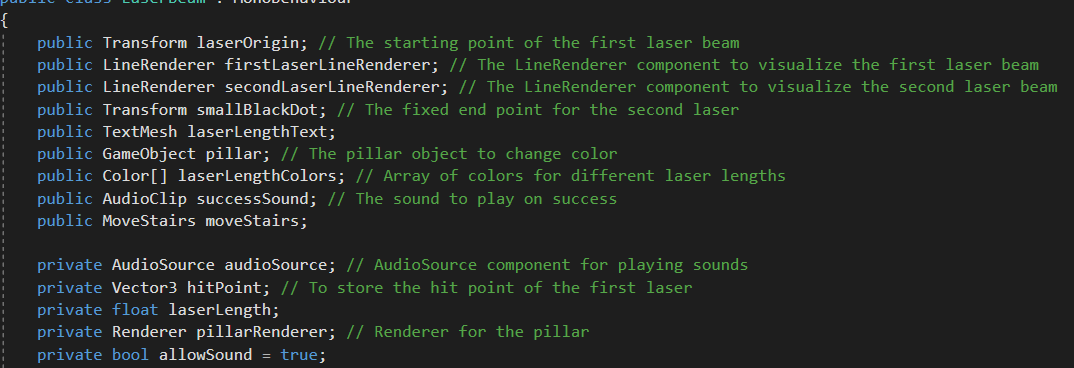
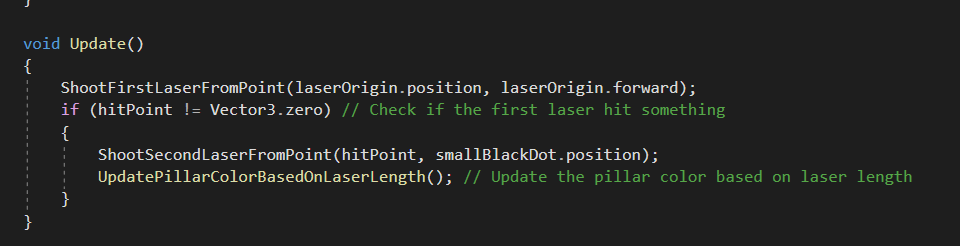
תיק מתכנת

1. **הקדמה**  
   הפרויקט הוא מידול תצוגה קיימת במכון ליצור מתקדם, המדמה שימוש בחיישן מרחק DT35.  
   הפרויקט רץ על גבי מכשירי Android בהתקנה פשוטה של קובץ apk.  
   הרצת הפרויקט ללא כניסה לקוד מתאפשרת על ידי הרצת הקובץ הנמצא פה: [קישור לדרייב](https://drive.google.com/drive/folders/10sv4zdnq4QWz_BZ9hNusyZWauBuU_312?usp=drive_link)  
   הפרויקט פותח באמצעות UNITY עם גרסה 2022.3.19f1.
2. **A screenshot of a computer

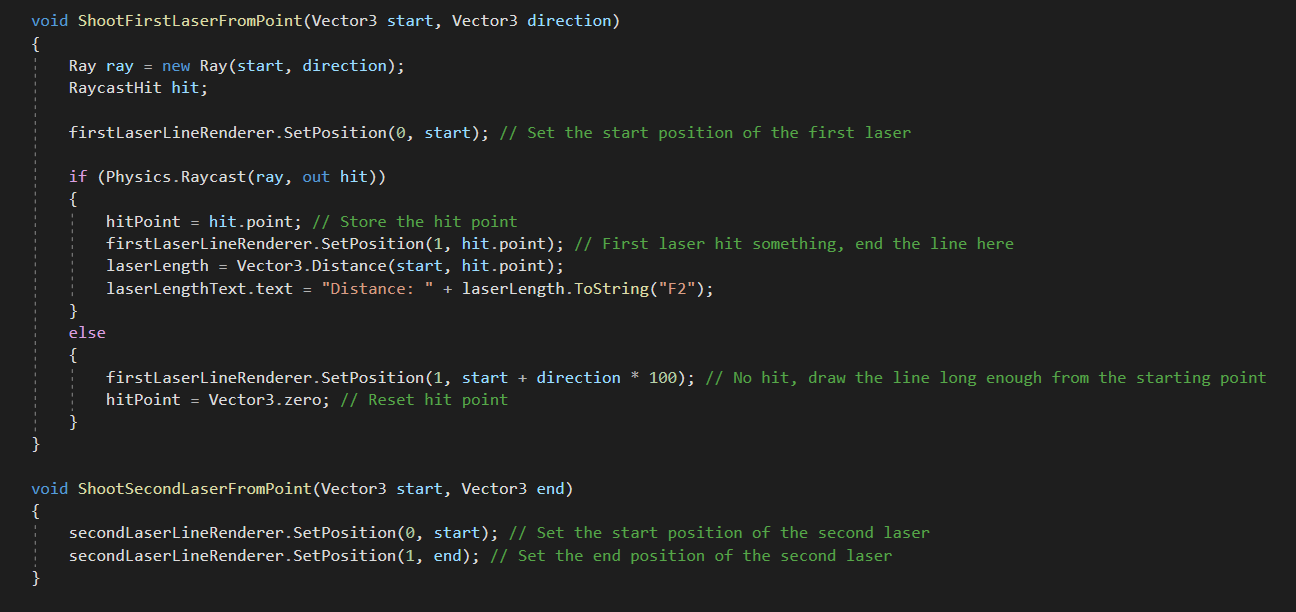
   Description automatically generatedהתקנת הפרויקט לסביבת פיתוח לוקאלית**  
   ניתן להתקין את הפרויקט באמצעות clone לפרויקט שנמצא בגיטהאב, [קישור לגיטהאב](https://github.com/SharonMor/AR-GAME)  
   הפקודה: git clone https://github.com/SharonMor/AR-GAME.git   
   לאחר מכן פתיחת unity hub ולחיצה על הכפתור Add, לאחר מכן Add project from disk ובחירת התיקייה שעודכנה, לאחר מכן לחיצה על התיקייה בunity hub תפתח את הפרויקט באופן לוקלאי.
3. **שימוש בסביבת העבודה**

בשלב זה ניתן לראות את התפריט משמאל המציין את מפרט הפרויקט ואת המסך הראשי editor בו ניתן לראות את הפרויקט בפועל. הכפתור הראשי הוא כפתור הplay שנמצא בצד העליון של המסך, הוא מדמה הרצה בפועל של האפליקציה, שם השינויים שנבצע לא ישמרו ויבוצעו רק בצורה לוקאלית. שם גם נוכל להבין כיצד הלוגיקה שביצענו מתנהגת בפועל (בסביבה לוקאלית).  
הפרויקט מורכב משני דברים עיקריים שמגיעים עם UNITY בפרויקט מסוג AR: AR Session, XR Origin, האחד מדמה את הסשן עצמו ואת הלוגיקה הכוללת, השני מספר טכנולוגיות לזיהוי המצלמה של הטלפון ועוד פרמטרים שונים הקשורים לתצוגת הסשן.

1. **קבצים עיקריים**ראשית נציין שאת השימוש בקבצי קוד, אנחנו מבצעים באמצעות גרירת הסקריפט אל האוביקט שמשתמש בו בunity, לדוגמא אוביקט של קרן לייזר יקבל את הקובץ LightBeam.  
   **קובץ ראשי - LightBeam.cs**  
   

המשתנים הpublic קשורים לחלקים אחרים באפליקציה בעוד שהחלקים הprivate הינם יחודיים רק לסקריפט הזה. ניתן לראות שלכל משתנה יש תאור קצר גם בתוך הסקריפט עצמו.  
המשתנים העיקריים הם laserOrigin שהוא המיקום ממנו יוצא הלייזר וmoveStairs שהוא אוביקט המדרגות הנעות.  
  


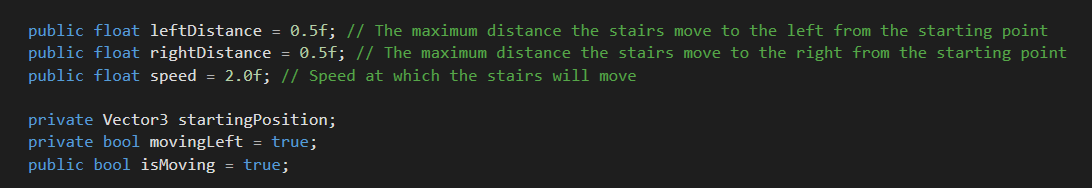
פונקציה מעניינת היא update, בעצם בכל עדכון הלייזר מתחיל מנק' הפתיחה ובודק מהיכן לשדר את הקרן חזרה, הקרן החוזרת יוצאת מנק' הפגיעה וחזרה אל נק' שנקראת smallBlackDot הנמצאת בסמוך לנק' היציאה.



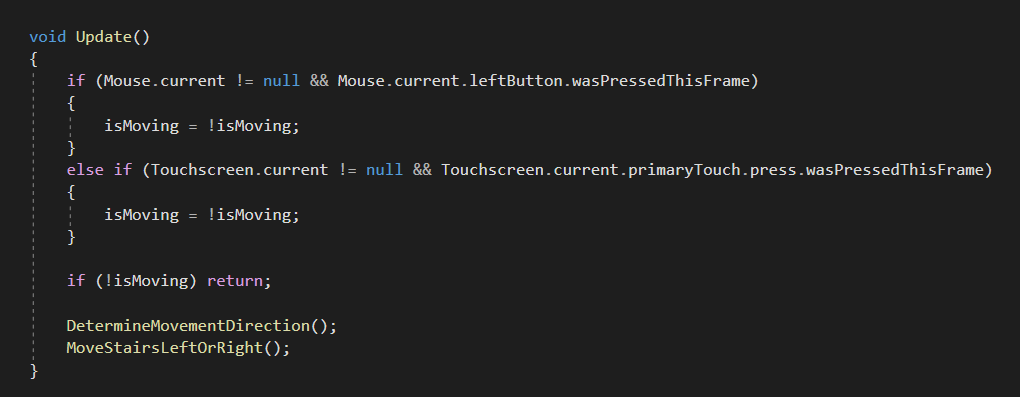
אלו הפונקציות המבצעות בפועל את השיגור של הקרן הראשונה והשניה. ניתן לראות כי השתמשנו בlineRenderer, פונקציה מובנית של unity המייצרת קו line במרחב. עידכנו את גודל הקו בהתאם.

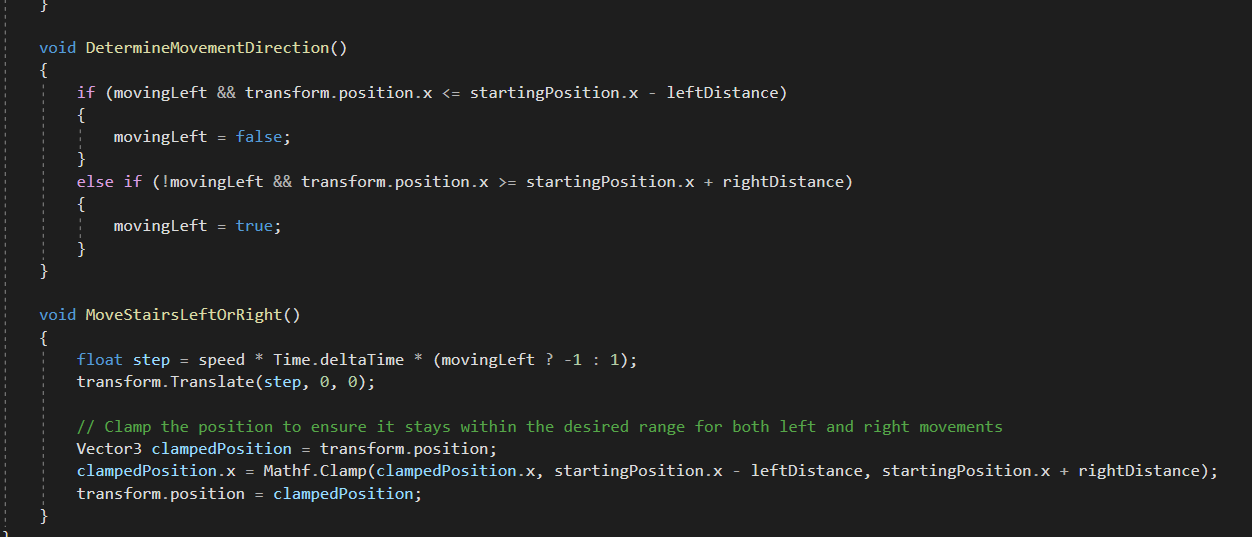


ישנה הפונקציה הזאת, היא מעדכנת את צבע העמוד לפי המרחק שמד הלייזר מצא (אורך הלייזר מנק' היציאה לנק' הפגיעה).

**קובץ moveStairs.cs**  


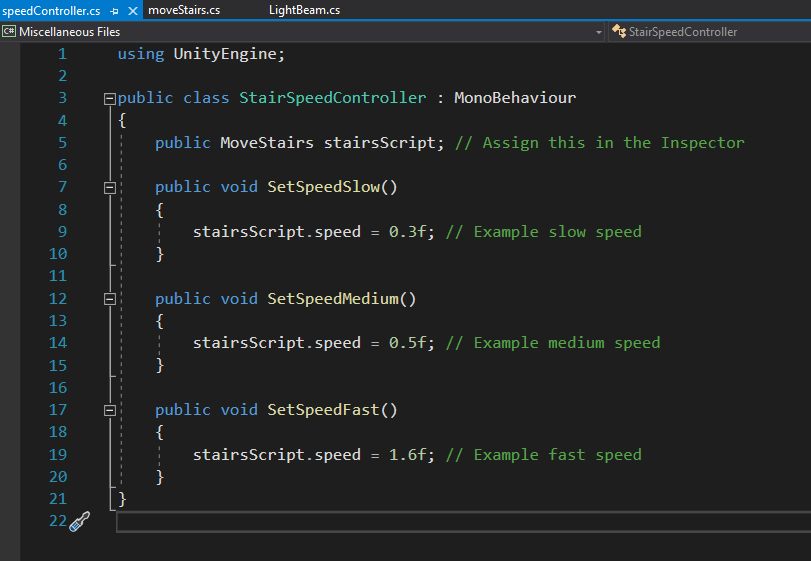
בדומה לקובץ lightbeam, גם פה המשתנים הpublic מגיעים מהui והמשתנים הprivate הם בשביל הלוגיקה הפנימית בעיקר.

המשתנים העיקריים הם המהירות של תנועת המדרגות speed, והמרחקים שהמדרגות יכולות לזוז בהם 0.5 יחידות לכל צד.  
בנוסף ישנו משתנה בוליאני שנקרא isMoving שמציין האם המדרגות בתנועה או לא. 

הפונקציה הראשית בקובץ זה היא update הרצה בכל עדכון, פה מתבצעת הלוגיקה שמאפשרת שליטה על תנועת המדרגות, כל לחיצה על המסך עוצרת/ממשיכה את תנועה המדרגות. כמו toggle.  


פה ניתן לראות כיצד חישבנו מתי לסיים את תנועת המדרגות ולהחליף כיוון, וכיצד בחרנו לאיזה כיוון כאן נכנס השימוש העיקרי במשתנים שלנו.

**קובץ speedController.cs**

זהו קובץ util פשוט שתקפידו לשלוט במהירות תזוזת המדרגות  


בקובץ זה נעשה שימוש בפונקציונאליות של הכפתורים של המהירות, כך נשלוט במהירות.

אלו קבצי הלוגיקה העיקריים אותם כתבנו, בנוסף להם יש קבצים שמגיעים עם הunity.  
הערה אחרונה: ישנה לוגיקה מסוימת שנקבעה ללא סקריפטים, למשל פונקציונאליות מסוימת של כפתורים שהגדרנו בUI, לא הוספנו פירוט שלהם שם כי הם מאוד ישירים.

לכל שאלה ניתן לפנות לאחד מחברי הצוות ונשמח להרחיב ולעזור!  
שרון מור, יובל בדיחי, יובל זוהר וליאל פסמן