 **Rubbery in the Depth - README**

**תיאור:**

המשחק מתרחש במצולות הים, והשחקן הוא צוללן ושודד אוצרות ערמומי  
המטרה – לשדוד כמה שיותר זהב ולהגיע לפתח היציאה על מנת לסיים את השלב.  
השחקן מנסה לחמוק ממלכדות מתוחכמות וקטלניות, להילחם באויבים אימתניים- שוטרים, כרישים ושאר מרעין בישין. תוך מעבר בין שלבים וזאת כדי להשיג שלל ותהילת עולם.  
השחקן שוחה מתחת המים, יש לו נשקים שהוא יכול להילחם איתם באויבים, תיבות שמכילות הפתעות כאשר המטרה היא לצבור נקודות ולעבור שלבים.

המשחק משלב פיסיקה תת מימית ברמה גבוהה (כגון זרמים, מערבולות וכדו'), חידות ופעולה.

**הסבר למשתמש:**

בתחילה מוצג המסך הראשי שכולל 3 אפשרויות:

1. **Start Game – פותח את מסך השלבים למשחק**
2. **Edit Level – פותח את מסך השלבים לעריכה**
3. **Exit – יציאה**

במסך השלבים ניתן לבחור שלב מסוים למשחק/עריכה.

המשחק מתנהל באופן הבא:  
השחקן נע עם החיצים (ארבעת הכיוונים). כמו-כן, השחקן יכול לקבל הפתעות שנותנות זהב, חיים ו/או נשקים.  
ניתן לירות ע"י מקש הרווח (במידה ויש נשק עם כדורים...), וע"י מקש ה-shift השמאלי ניתן להחליף נשק, במידה ויש מספר נשקים.  
ישנם מספר סוגי אויבים: כרישים אימתניים שנואשים לטרף קל, מפלצות עם נשקים שחושקות בבשר אדם טרי, מכונות ירייה, וסרטנים.

במידה והשחקן הגיע לפתח היציאה, יפתח מסך הניצחון שבו יהיה את כמות הזהב שצבר.  
במידה והשחקן נפסל, יפתח מסך ההפסד הלא נעים לצפייה, שמראה לו את כשלונו המר, עם אופציית restart כדי לשוב ולתקן את העוון.

מסך העריכה פועל באופן הבא:  
יש תפריט עליון הכולל כפתורי מחיקה (של דמות), שמירה ויציאה (ללא שמירה), וכן את רשימת הדמויות להוספה למפה.  
יתר-על-כן, יש את המפה עצמה שבה ניתן למקם את הדמויות השונות.  
ניתן לזוז במפה ע"י החיצים, ולהגדיל ולהקטין באמצעות הכפתורים 1,2.  
  
  
בכל המסכים ניתן לחזור למסך הקודם ע"י מקש Esc.

**הסבר למתכנת הסקרן:**

כל המשחק מבוסס על GUI שבנינו שמאד גנרי שכולל הגדרות לתצוגת אובייקטים ברמה המופשטת (מיקום, גודל, צבע/תמונת רקע וכו') עם אירועים (עכבר – כניסה, לחיצה ויציאה, מקלדת – לחיצה ועזיבה וכו'), לרמה הבסיסית (טקסט, כפתור, תמונה, אנימציה וכו'), וכולל אובייקטי "מקבץ" שלהם (HorizontalLayout, RelativeLayout וכו').  
כמו-כן, טעינת המשאבים (גופנים, סאונדים, מוסיקת רקע, תמונות ואנימציות) מתבצעת בצורה גנרית ע"י שליפת המידע מתוך קבצי JSON השונים.

פיסיקת המשחק מכילה את חוקי התנועה הבסיסיים של מהירות ותאוצה ובכך הגדרנו ציפה, חיכוך וכוח משיכה, ושילבנו בשחקנים בעלי יכולת תנועה (MovingObject)

המשחק כולל שימוש נרחב בחריגות (נפתח חלון שגיאה עם הסבר מתאים) כדי לדבג בצורה מהירה וקלה.

**רשימת קבצים:**

נחלק בין קבצי המשחק לבין קבצי ה-GUI:

**קבצי המשחק:**

|  |  |
| --- | --- |
| הסבר קצר | שם הקובץ |
| המסך הראשי. יורש BaseScreen<RelativeLayout>.  כולל 4 כפתורי תפריט. | MainScreen |
| יורש BaseScreen<RelativeLayout>  מסך המשחק. | GameScreen |
| יורש BaseScreen<RelativeLayout> מסך בחירת שלב | ChooseLevelScreen |
| יורש BaseScreen<RelativeLayout>  מסכי ניצחון/הפסד | WinScreen,  LoseScreen |
| יורש BaseScreen<RelativeLayout>  מסך עריכת השלבים. מציג את המפה ופאנל של דמויות להוספה על המפה | EditScreen |
| יורש AnimationView. אובייקט כללי של הדמויות במשחק  מחלקה אבסטרקטית בעלת פונקציות וירטואליות הנוגעות לכלל האובייקטים במשחק, כגון התנגשויות | BoardObject |
| יורש BoardObject. אובייקטים שאין להם אינטראקציה | UnInteractableObject |
| יורש UnInteractableObject. קיר, לא ניתן לעבור דרכו ולא ניתן להריסה | Wall |
| יורש BoardObject. אובייקטים שיש בינהם אינטראקציה | InteractableObject |
| יורש InteractableObject. דמויות שלא זזות. | StaticObject |
| יורש StaticObject | Container |
| יורש Conteiner. תיבה שנותנת מתנות | Chest |
| יורש StaticObject. ניתן להריסה ע"י יריות עליו | Box |
| יורש StaticObject. זרם. נותן תאוצה חיצונית לכיוון מסוים. (על המסך מוצג אותו דבר, כך שאין יכולת לדעת לאיזה כיוון. בעורך ניתן לקבוע כיוון.) | Flow |
| יורש StaticObject. דלת יציאה מהשלב | ExitLevel |
| יורש InteractableObject. דמויות זזות.  מכיל את פונקציות החישוב של התנועה של האובייקטים. (ע"י חיכוך, ציפה, מהירות ותאוצה) | MovingObject |
| יורש InteractableObject. פיצוץ | Explosion |
| יורש MovingObject | Character |
| יורש Character. השחקן. ניתן לשליטה ע"י מקשי המקלדת. | Player |
| יורש Non player character character. דמויות זזות שאינן השחקן (אויבים(  פונקציות כיוונים | NPC |
| יורש NPC. שודד  יש לו כלי נשק  רודף אחרי השחקן מרדיוס מסויים | Rubber |
| יורש NPC. כריש. רודף אחרי השחקן מרדיוס מסויים. | Shark |
| יורש NPC. סרטן הולך על הקרקעית | Crab |
| יורש NPC. מכונת ירייה. יש לו נשק ויורה צרורות | MachineGun |
| יורש MovingObject. קליע של כלי נשק | Projectile |
| יורש Projectile. כדור של רובה | Bullet |
| יורש Projectile. כדור של מטול רימונים | Granade |
| כלי עבודה (גם נשק) מכיל מידע על הכלים | Tool |
| כלי נשק. יורש Tool | Weapon |
| יורש weapon. כלי נשק חם | HotWeapon |
| יורש HotWeapon. רובה | AK47 |
| יורש HotWeapon. מטול רימונים | GrenadeLauncher |
| הפתעה | Surprise |
| יורש Surprise. מתנה | Gift |
| יורש Gift . מתנה מסוג כלי (נשק) | ToolGift |
| יורש Gift . מתנה של זהב | CoinGift |
| יורש Gift . מתנה של חיים | LifeGift |
| מציג את הכלי הנוכחי בתפריט המשחק | ToolView |
| Board Object Data Structure. מבנה הנתונים של הדמויות. | BODS |
| מחלקה ליצירת הדמויות מהמפה. משתמשת בתבנית העיצוב Factory | BOFactory |
| יורש View  אובייקט המציג את המפה של השלב | World |
| יורש HorizontalLayout<view>  תפריט המופיע במסך המשחק מעל המפה. מציג את המידע של השחקן | GameMenu |
| יורש HorizontalLayout<ImageView>  מציג את החיים של השחקן | LifeView |
| יורש ImageButton. מציג את הניקוד | CoinView |
| יורש ImageButton. מציג את הנשק הנוכחי של השחקן ואת הכמות. | ToolView |
| אחראי על תפעול המשחק.  מריץ את מסכי המשחק | GameController |
| יורש TextView. | GameAnimText |
| מנהל את הcamera של המשחק.  פונקציה שמעדכנת את המיקום של המצלמה | GameCamera |

**קבצי ה-Editor:**

|  |  |
| --- | --- |
| הסבר | שם הקובץ |
| יורש HorizontalLayout<view>  מציג את רשימת הדמויות שניתן להוסיף למפה | EditMenu |
| יורש ImageView. מציג דמות בתוך ה-Editmenu | GameObjectView |
| יורש HorizontalLayout<GameObjectView>. רשימה של דמויות להצבה בעורך | GameObjectList |
| מפענח רשימת אובייקטים להצבה | GOIFileParser |
| מידע על אובייקט להצבה | GameObjectInfo |
| יורש std::exception. שגיאה בעת פענוח של אובייקט להצבה | ParseGOIException |
| יורש TableLayout<MapCellView>  מפת העריכה של השלב. ניתן להוסיף דמות לכל תא במפה | EditMapView |
| יורש ImageView. תא במפת העריכה | MapCellView |
| מנהל את השלבים מהקובץ levels.json (עריכה, הוספה וכו') | LevelFileManager |
| מכיל מידע על השלב. (מטריצה, שם וכו') | LevelInfo |

**קבצי ה-GUI:**

|  |  |
| --- | --- |
| תיאור | שם הקובץ |
| מחלקה מופשטת המייצגת אובייקט שמוצג על המסך. כוללת מיקום, רוחב, גובה, רקע (תמונה או צבע), גבול (צבע וגודל) כוללת אירועים: לחיצה, כניסה ויציאה עם העכבר. לחיצה ועזיבת מקש במקלדת. פונקציה לציור, קבלת מלבן חוסם, מצב תצוגה/הסתרה, מצב אפשור/ביטול אירועים. אפשרות למצלמה.  לא ניתן ליצור מופע של המחלקה באופן ישיר אלא ע"י ירושה בלבד. | View |
| מחלקה המייצגת טקסט. ניתן לשנות את הפונט, הטקסט וצבעו, ואת כיוון היישור שלו (ימין, מרכז או שמאל) | TextView |
| כפתור. יורש TextView | Button |
| אנימציה. יורש View. מכיל TextureInfo מקושר וטיימר, ובכך מצייר אנימציה. כולל אופציה לשינוי תדירות עדכון התמונות. | AnimationView |
| מחלקה מופשטת תבניתית המכילה וקטור של אובייקטים ViewType. ViewType חייב להיות צאצא של View  מטרת המחלקה לגרום לכך שכל אירוע שמתקבל יגיע לאובייקטים המוכלים ב- ViewGroup  כמו כן, גם מצב התצוגה והאפשור מועבר לילדים. כלומר, לאובייקטים המוכלים בו.  לא ניתן ליצור מופע של המחלקה באופן ישיר אלא ע"י ירושה בלבד. כולל הנחה שאין אובייקטים מצטלבים (הסיבה היא בגלל חישוב יעיל של האירועים) | ViewGroup<ViewType> |
| שכלול של ViewGroup שמסדר את האובייקטים בצורה אנכית אחד אחרי השני. גודל האובייקטים מחושבים באופן יחסי בהתאם לממדי ה- VerticalLayout | VerticalLayout<ViewType> |
| שכלול של ViewGroup שמסדר את האובייקטים בצורה אופקית אחד אחרי השני. גודל האובייקטים מחושבים באופן יחסי בהתאם לממדי ה- HorizonalLayout | HorizonalLayout<ViewType> |
| שכלול של ViewGroup שמסדר את האובייקטים באופן מותאם אישית. גודל האובייקטים מחושבים באופן יחסי בהתאם לממדי ה- RelativeLayout | RelativeLayout<ViewType> |
| אובייקט המייצג תמונה. יורש View | ImageView |
| שכלול של VerticalLayout המכיל ImageView ו-TextView | ImageButton |
| מכיל מידע על תמונה או צבע ומצייר אותה בהתאם למלבן חוסם שניתן לו | ImageTexture |
| גבול – מכיל צבע ועובי הגבול ומצייר אותו בהתאם למלבן חוסם שניתן לו. | Border |
| עורך טקסט. יורש TextView. | EditText |
| פונקציות עזר עבור מלבנים | RectUtils |
| מנגנון אירועים גנרי תבניתי. מכיל וקטור של פונקציות שמאזינות לאירוע מסוים. כולל פונקציות הוספת האזנה וקריאה לאירוע. | IHandler<T> |
| תא – מכיל מספר שורה ומספר עמודה | Cell |
| מטריצה תבניתית. כוללת תמיכה ב-iterator | Matrix<T> |
| תמונה/אנימציה. מכיל Texture שמייצג תמונה או סדרת תמונות ומטאדטה נלווה – שם, מספר שורות, עמודות ומספר תמונות. | TextureInfo |
| מחלקה מופשטת סינגלטונית לטעינת משאבים | ResourceLoader |
| טוען תמונות/אנימציות מקובץ JSON. יורש ResourceLoader | TextureManager |
| טוען צלילים של המשחק ומוסיקת רקע מקובצי JSON. יורש ResourceLoader | SoundManager |
| טוען גופנים מקובץ JSON. יורש ResourceLoader | FontManager |
| חריגה בעת טעינת משאב מסוים. יורש std::exception | LoadResourceException |
| טיימר. מחזיק שעון עם פונקציה שמופעלת כל מרווח זמן.  פונקציות התחלת שעון וסיום שעון. ובדיקה של השעון. משתנה בוליאני האם השעון פעיל. | Timer |
| שעון עצר. מחזיק שעון עם פונקציה שמופעלת כאשר הטיימר יסיים. כולל אירוע בדיקה כל מרווח זמן. | StopWatch |
| מחלקה אבסטרקטית של מסך. יורש ViewType. | BaseScreen<ViewType> |

|  |  |
| --- | --- |
| שם המפה | קונספט |
| Silence in the Depths | מפה גדולה, בסיסית שמטרתה להציג את הפונקציונליות של המשחק |
| Lost in the Depths | משחק הרפתקאות קלאסי |
| The Shark Territory | כשמה כן היא, הרבה אקשן |
| Into the Storm | מדגישה את הפן החשבתי במשחק |
| Tsunami on the Way | מדגישה את הצד של הזרמים במשחק |

**מבני נתונים עיקריים:**

* את הדמויות החזקנו באופן הבא:

1. תור בקשות להוספת דמויות.
2. תור בקשות להסרת דמויות.
3. מפת הדמויות (עץ אדום שחור) שכל מפתח בה הוא int המייצג את עדיפות ציור, וכל ערך הוא קבוצה לא סדורה (טבלת גיבוב) של דמויות עם עדיפות הציור הזה.

הסיבה לבנייה הזאת היא כי לכל דמות יש עדיפות ציור כלשהי ונרצה שהן יצוירו על פיה,  
התורים נחוצים על מנת שהדמויות עצמן יוכלו ליצור או להרוג דמויות אחרות (למשל, כדור שהורג כריש) בלי לפגוע במבנה הנתונים, לכן אנו מוסיפים את הבקשה הזאת לתור המתאים ומטפלים בה ב"זמן נח".  
כמו-כן, זמן הוספת/הסרת דמות לוקחת log(n).

* מפות לא סדורות (טבלאות גיבוב) של פונטים, אנימציות וסאונדים עם מפתח של שמם (מחרוזת).
* עץ AABBTree לבדיקת התנגשויות יעילה.

**אלגוריתמים הראויים לציון:**

* המשחק מדמה פיסיקה מתחת למים הכוללת- חיכוך, תאוצה ומהירות. כל דמות דינאמית מושפעת הן מהתנועה שלה (תאוצה עצמית), והן ממקורות חיצוניים (זרמים), שהם בתורם גורמים למהירות לעלות/לרדת. כל זאת תחת השפעת החיכוך עם המים, זאת אומרת שככל שהמהירות עולה כך גם החיכוך דבר הגורם להאטת ובסופו של דבר עצירת הדמות.
* כיוון שהמפות מאד גדולות במשחק עם המון דמויות, אז בהתחלה בדיקת ההתנגשויות ארכה זמן רב, לכן החלטנו פה אחד להשתמש באלגוריתם חכם של עץ דינאמי מאוזן AABBTree, הנמצא בספרייה Box2D כדי לבדוק התנגשויות בזמן יעיל ביותר של log(n)

**ספריות צד ג':**

* [**nlohmann - json**](https://github.com/nlohmann/json)
* [**Box2D**](https://github.com/erincatto/Box2D)

**הערות:**

* בעורך ניתן רק לערוך שלבים ולא ליצור אותם או למחוק אותם. ניתן לעשות את הפעולות הללו ישירות על קובץ השלבים levels.json.
* במחלקות השונות יש פונקציה toString שמיועדת לצורך דיבאגינג בלבד.

**מגישים:**  
מתנאל אביוף  
נחום נגה  
עקיבא גולדשטיין