דוח פרויקט

Image Duel

יבגני מורושקו

מרום סבג

מנחה: חביאר טורק

מהנדס מעבדה: ירון חונן

תוכן עניינים

תיאור הפרויקט 2
פירוט עיבודי התמונה 3
אלגוריתם 1 – Random Pixel Revealer 4
אלגוריתם 2 – Gaussian Blur 5
אלגוריתם 3 – Low Resolution Effect 6
אלגוריתם 4 – Spiral Twist Effect 7
אלגוריתם 5 – Polar Low Resolution 8
אלגוריתם 6 – Wave Effect 10
סכמת ממשק משתמש 12
השרת ופרוטוקול התקשורת 13
Single Duel Flow 13
מבנה המסד נתונים 14
פיצ'רים שמומשו באפליקציה 17
הצעות לשיפור 18

תיאור הפרויקט

במסגרת הפרויקט מימשנו משחק אינטראקטיבי בין שחקנים במהלכו כל שחקן יכול לשלוח לשחקן אחר תמונה, השחקן השני צריך לנחש מה יש בתמונה לאחר שהופעל אלגוריתם כלשהו לעיבוד תמונה המקשה על הזיהוי.
לאחר הניחוש, השחקן שולח בחזרה לשחקן השני עוד תמונה וכך המשחק ממשיך.

בפרויקט מומשו 6 אלגוריתמים לעיבוד תמונה כאשר האלגוריתם שמופעל בכל פעם נבחר באופן רנדומאלי.

השחקנים מקבלים ניקוד כתלות בזמן בו הצליחו לנחש, כלומר ניחוש מהיר יזכה בניקוד גבוה.

המשחק ממומש למכשירי אנדרואיד, ומשתמש בשרת מרוחק כדי לנהל את המסד נתונים ואת התקשורת בין משתמשים שמבוצעת בצורה א-סינכרונית.

פירוט עיבודי התמונה

במסגרת הפרויקט מימשנו 6 אלגוריתמים לעיבוד תמונה.
בכל אלגוריתם קיים פרמטר הניתן לשינוי והוא קובע עד כמה העיבוד "מוקצן" כלומר ע"י שינוי הפרמטר נוכל לגרום לכך שהפעלת העיבוד על התמונה יביא תמונה שתהיה דומה יותר או פחות לתמונה המקורית.

במהלך המשחק, כאשר על המשתמש לנחש את התמונה בפרק זמן קצוב, התמונה מוצגת לאחר שהופעל עליה אלגוריתם כלשהו לעיבוד תמונה, וככל שהזמן עובד מציגים תמונה שדומה יותר למקור ע"י שינוי הפרמטר הנ"ל.

אלגוריתם 1 – Random Pixel Revealer

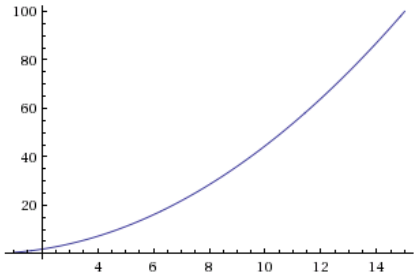
באלגוריתם זה אנחנו בוחרים להציג רק X% מפיקסלים (שנבחרים אקראית) מתוך התמונה המקורית.
התוצאה היא תמונה שבה חלק רנדומלי מהפיקסלים שייכים לתמונה המקורית והשאר לבנים, ולכן אנו מקבלים תמונה שקשה יותר לזהות.
ככל ש X יהיה קטן יותר, נקבל תמונה שקשה יותר לזהות.

כאשר אנו מציגים למשתמש את התמונות השונות שמשתנות ככל שהזמן עובר, אנו בוחרים את X בכל תמונה לפי הנוסחה הבאה:

$$X_i = \left(\frac{i}{NUM_OF_PICTURES} \right)^2 \cdot 100$$
$$i \in \{1..NUM_OF_PICTURES\}$$

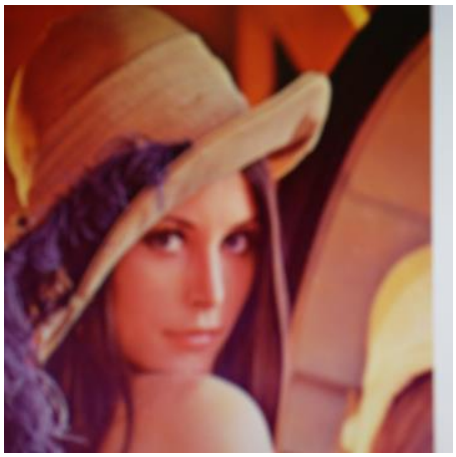
כאשר i הוא מספר התמונה שמציגים (כאשר $i=1$ זו התמונה הראשונה שתוצג).

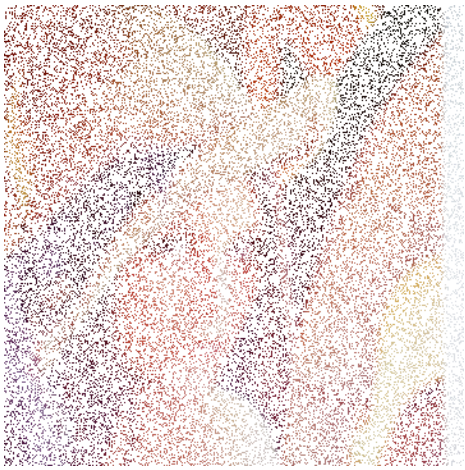
אנו מקבלים ש X משתנה באופן הבא:



השינוי הוא לא לינארי כדי ליצור רמת קושי מתאימה (שינוי לינארי מביא לכך שקל לנחש כבר בהתחלה).

דוגמאות שונות להפעלת האלגוריתם:





אלגוריתם 2 – Gaussian Blur

אלגוריתם זה, מייצר תמונה כשהיא נראית לא מפוקסת ביחס לתמונה המקורית (אפקט של blurring).

האלגוריתם מתבסס על פונקציית הגאוסיאן: $G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{\Delta x^2+\Delta y^2}{2\sigma^2}}$.

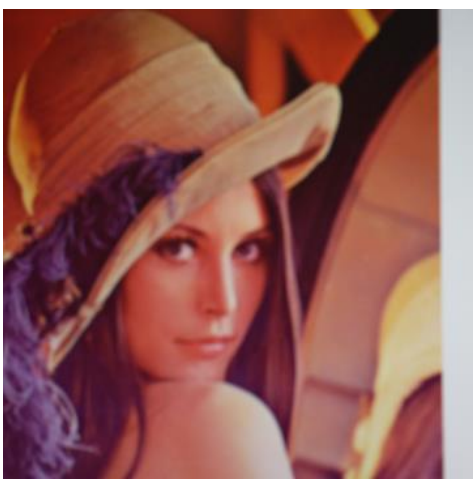
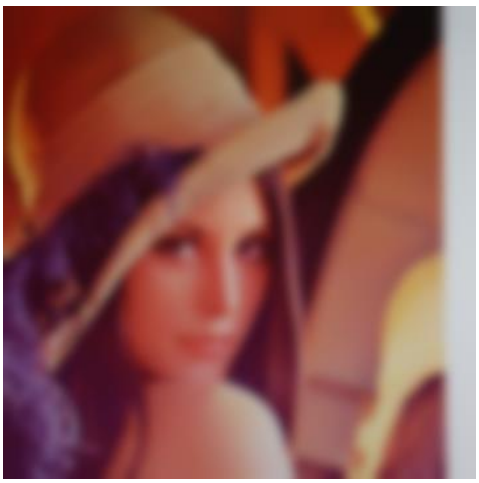
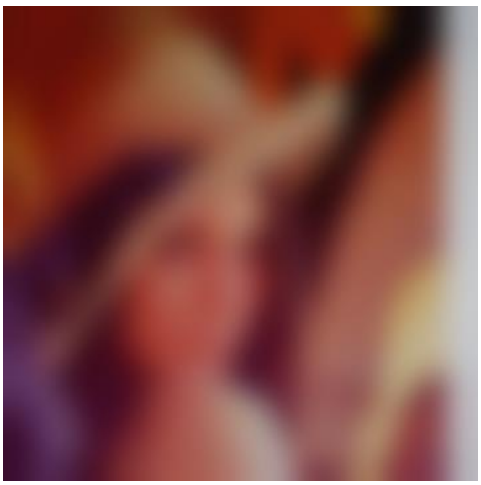
במהלך האלגוריתם אנו יוצרים מטריצת קונבולוציה בגודל $20x20$ פיקסלים שמחושבת בעזרת פונקציית הגאוסיאן הנ"ל.
ערכו של כל פיקסל מחושב על הפעלת מטריצת הקונבולוציה עליו, כלומר חישוב של סכום מכפלות הפיקסלים השכנים בערך המתאים במטריצת הקונבולוציה.

כדי לייעל את ביצוע האלגוריתם, אנו מבצעים בפועל שתי הפעלות חד ממדיות של האלגוריתם, פעם אחת על השורות, ואז עוד פעם על העמודות, הדבר מביא לחיסכון משמעותי בזמן ריצת האלגוריתם שכן בצורה זו נעשה יותר שימוש בערכים הנשמרים ב cache.

אלגוריתם זה שונה משאר האלגוריתמים, שכן אין כאן פרמטר כלשהו שמשתנה.
כאשר אנו רוצים להציג למשתמש את התמונות, השינוי בתמונות מתבצע ע"י כך שמפעילים את האלגוריתם מספר פעמים על אותה התמונה ולאחר מכן מציגים בסדר הפוך את תוצאות הפעלת האלגוריתם.

התמונה ה i שמוצגת היא התמונה המקורית לאחר שהופעל עליה האלגוריתם $NUM_OF_PICTURES-i$ פעמים.

דוגמאות להפעלת האלגוריתם:

אלגוריתם 3 – Low Resolution Effect

מטרת האלגוריתם הוא ליצור תמונה שנותנת הרגשה של רזולוציה נמוכה (תמונה מפוקסלת).

כדי להשיג את האפקט המבוקש אנו נעזרים באריתמטיקה של שלמים כדי לאחד מספר פיקסלים לפיקסל יחיד.

האלגוריתם מייצר תמונה חדשה בה כל פיקסל שנמצא במקום (x,y) שווה לפיקסל שהיה במקום $((x/C)\cdot C,(y/C)\cdot C)$ בתמונה המקורית.

האפקט מושג ע"י כך שכל ריבוע של פיקסלים בגודל CxC מאוחד לצבע אחיד.

לכן C הוא הפרמטר שקובע כד כמה התמונה שתיוצר תהיה קרובה לתמונה המקורית, ככל שנגדיל את C נקבל תמונה יותר "מפוקסלת" ולכן דומה פחות לתמונה המקורית.

כאשר התמונות מוצגות למשתמש לניחוש, אנו מייצרים תמונות עם ערכי C שונים לפי הנוסחה הבאה:

$$C_i = i\cdot 1.5$$
$$i \in \{1..NUM_OF_PICTURES\}$$

המכפלה ב 1.5 נועדה ליצור רמת קושי מתאימה.

דוגמאות להפעלת האלגוריתם:

אלגוריתם 4 – Spiral Twist Effect

באלגוריתם זה אנו יוצרים מראה של "ספירלה".
כדי לממש את האלגוריתם אנו עוברים לייצוג של פיקסלים בקואורדינטות פולריות, כלומר כל פיקסל מיוצג ע"י (r,θ), כאשר r הוא המרחק ממרכז התמונה ו θ זו הזווית.

כדי ליצור את התחושה של "ספירלה" אנו רוצים בתמונה החדשה לשנות את רכיב ה θ של הפיקסלים וע"י כך ליצור סיבוב.
השינוי שנביא לפיקסלים בערך ה θ יהיה ברדיוס שלהם (ערך ה r), פיקסלים שבערך ערכי r גבוהים יסובבו בצורה גדולה יותר מאשר פיקסלים בעלי ערכי r קטנים, וכך נקבל את התחושה הספירלית.

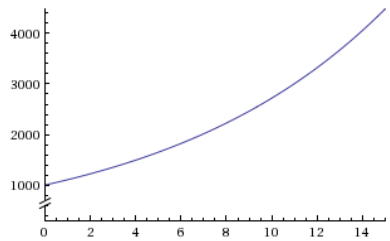
אם כן, ניצור תמונה חדשה, שבה כל פיקסל (r,θ) יילקח מהערך של הפיקסל $\left(r,\theta+\frac{r}{C}\right)$ בתמונה המקורית, הדבר מבטיח שכפי שרצינו, פיקסלים בערך ערכי r גדולים יותר יסתובבו יותר.

הפרמטר C מאשר לנו לשלוט על עוצמת הסיבוב. ככל שנגדיל את C נקבל תמונה קרובה יותר לתמונה המקורית.

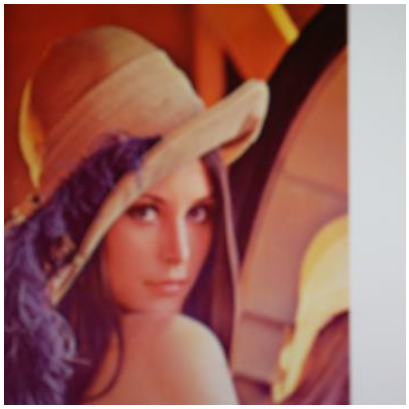
לכן, כאשר אנו מציגים למשתמש את התמונה לניחוש ומשתמשים באלגוריתם הזה, אנו משנים את ערכו של C לפי הנוסחה הבאה:

$$C_i = 1000 \cdot e^{\frac{i}{10}}$$
$$i \in \{1..NUM_OF_PICTURES\}$$

נקבל את השינוי הבא:

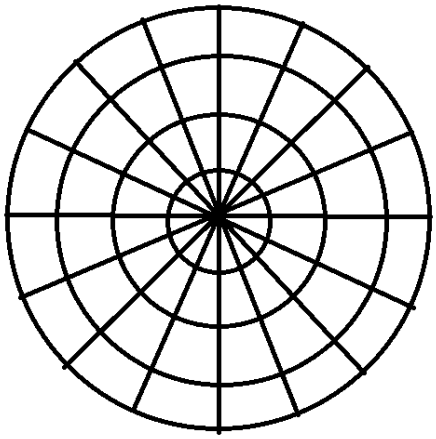


דוגמאות להרצת האלגוריתם:



אלגוריתם 5 – Polar Low Resolution

באלגוריתם זה נרצה ליצור מראית עין של "פקסול" אולם כאן נרצה שאפקט ה"פקסול" יראה מעגלי.
לצורך כך נרצה לחלק את התמונה לטבעות ולגזרות בגודל אחיד:



כעת, כל חלק נבצע בצבע אחיד לפי הערכים בתמונה המקורית וע"י כך נקבל אפקט של "פיקסול" בצורה מעגלית.

נממש את הרעיון הנ"ל בצורה הבאה:
נשתמש שוב בקואורדינטות פולריות כדי לייצג את התמונות.
נייצר תמונה חדשה, בה כל פיקסל (r,θ) יהיה הערך של הפיקסל $((r/C)\cdot C,(\theta/\mathrm{C})\cdot \mathrm{C})$ בתמונה המקורית.

אנו משתמשים באריתמטיקה של שלמים ולכן מקבל עיגול של הערך של החישוב גם בזווית וגם ברדיוס וכך מתקבל האפקט הרצוי.

הפרמטר C הוא הקבוע שמשפיע עד כמה התמונה שמתקבלת תהיה דומה לתמונה המקורית, הגדלה של C גורם לכך שיותר פיקסל יקבלו את אותו הערך ולכן גורם לתמונה הנוצרת להיות יותר "מפוקסלת" ודומה פחות למקורית.

כאשר מציגים למשתמש את התמונה לניחוש, אנו משנים את ערכו של C לפי הנוסחה הבאה:

$$C_i = i \cdot 0.95$$
$$i \in \{1..NUM_OF_PICTURES\}$$

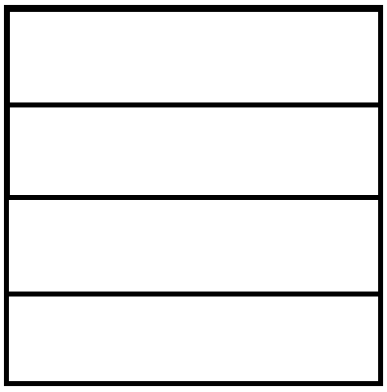
כאשר 0.95 הוא קבוע הנועד להבטיח רמת קושי נאותה.

דוגמאות להרצת האלגוריתם:

אלגוריתם 6 – Wave Effect

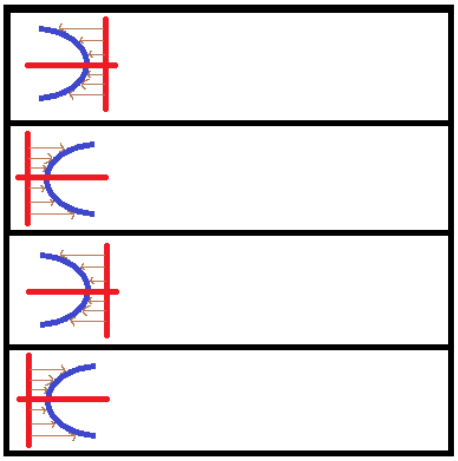
באלגוריתם זה אנו רוצים ליצור צורה של "גלים" לאורך התמונה.

כדי ליצור את האפקט הרצוי אנו מחלקים את התמונה לבלוקים רוחביים:



הבלוקים שנוצרו ממוספרים ומחולקים לקבוצות של בלוקים זוגיים ואי זוגיים.

בכל בלוק זוגי הפיקסלים מוזזים רוחבית לפי פרבולה חיובית, ובכל בלוק אי זוגי הפיקסלים מוזזים רוחבית לפי פרבולה שלילית:



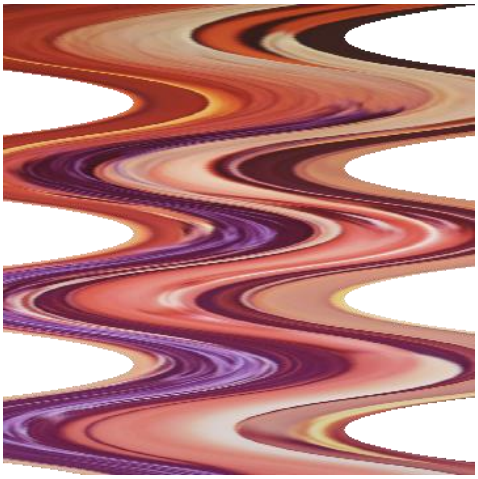
עוצמת האפקט נקבעת ע"י קביעת ערך המקדם החופשי במשוואת הפרבולה שבה משתמשים, ככל שנקבע ערך גדול יותר כך נקבל אפקט חזק יותר והתמונה המתקבלת תהיה שונה יותר מהתמונה המקורית.

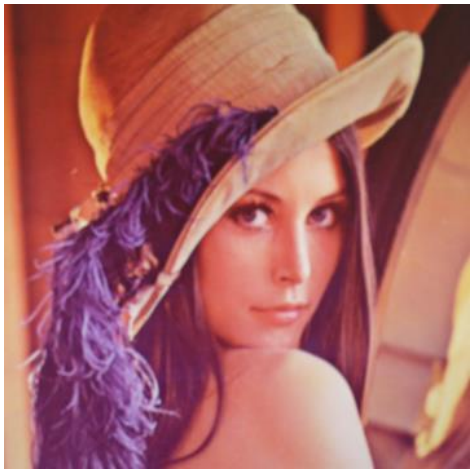
כאשר אנו מציגים את התמונה למשתמש אנו משנים את המקדם החופשי C לפי הנוסחה הבאה:

$$C_i = 100 \cdot \frac{i}{NUM_OF_IMAGES}$$

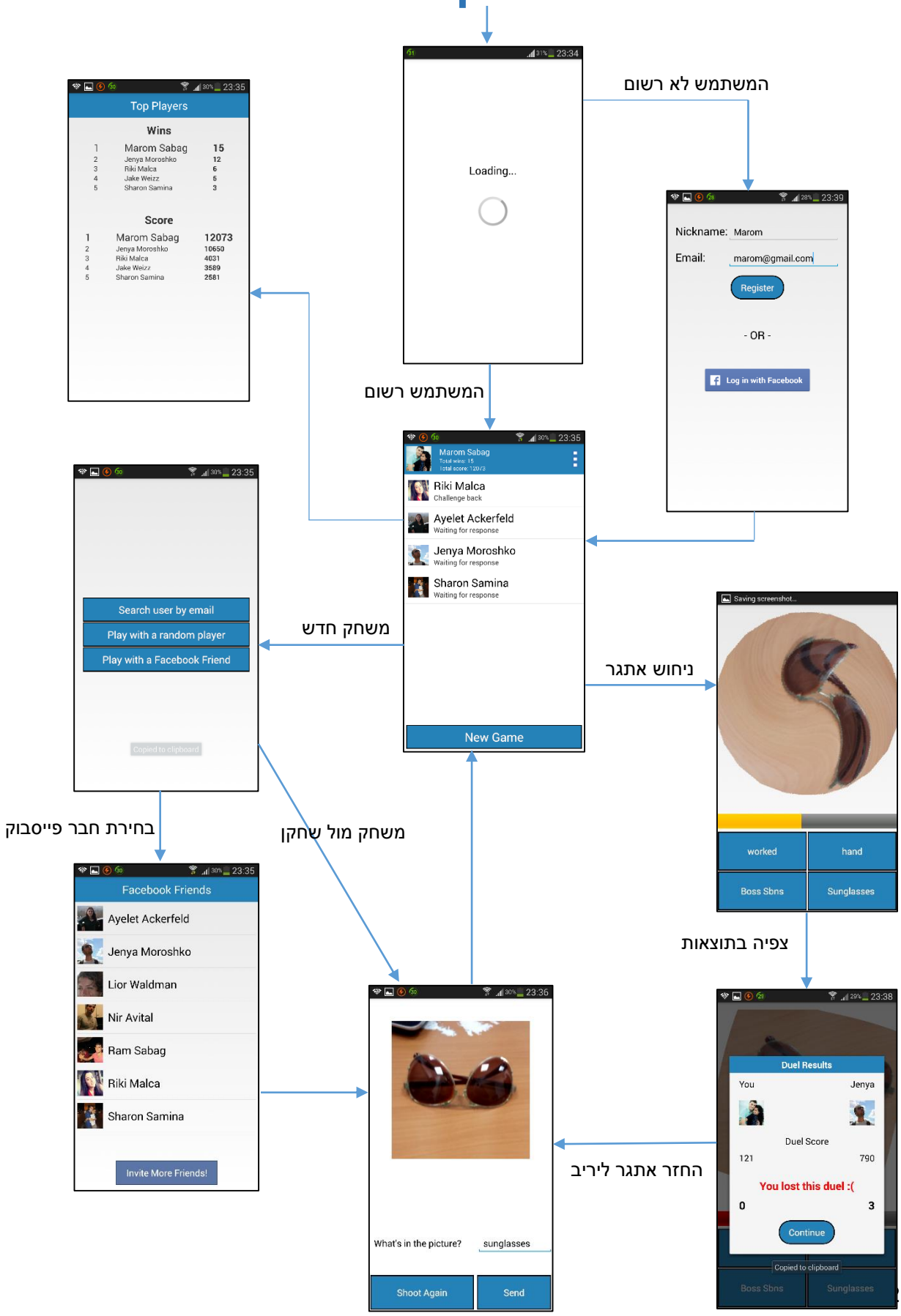
$$i \in \{1..NUM_OF_PICTURES\}$$

דוגמאות להרצת האלגוריתם:





סכמת ממשק משתמש



השרת ופרוטוקול התקשורת

האפליקציה מתקשרת עם שרת הממומש ב PHP ומשתמש במסד נתונים מסוג My SQL.

האפליקציה מתקשרת עם השרת בעזרת בקשות HTTP GET/POST.

בכל פניה לשרת, האפליקציה מציינת בפרמטר service את סוג השירות שהיא מעוניינת לקבל מהשרת. על פי ערך הפרמטר service השרת יודע איזה שירות להעניק ולכן מפעיל את הקוד המתאים תוך שימוש בפרמטרים נוספים שהועברו.

השרת מחזיר מידע מקודד ב JSON, כאשר בכל תשובה קיים הפרמטר success המציין האם בפעולה המבוקשת הצליחה או לא, במידה וערך השדה success הוא 0, קיים גם שדה error_code המציין את סוג השגיאה שקרתה. במידה ולא קרתה שגיאה, שאר הפרמטרים מכילים את התשובה לשירות המבוקש.

השרת משתמש בשירות הענן של גוגל – Google Cloud Messaging על מנת לשלוח הודעות push למכשירים השונים בעת הצורך.

Single Duel Flow

1. משתמש 1 מצלם חפץ באמצעות מצלמת המכשיר ובוחר מילה המתארת את החפץ המופיע בתמונה (במידה ומדובר במשחק חדש הוא בוחר תחילה מי יהיה משתמש 2 שמולו ישחק).
2. האפליקציה פונה לשרת ומבקשת באמצעות service מתאים להוסיף duel חדש כאשר היא מצרפת כפרמטר את התמונה והמילה שהוזנה.
3. השרת מוודא שניתן להוסיף duel (למשל מוודא שהמשחק לא בוטל ע"י משתמש 2) ובמידה והפעולה חוקית הוא מוסיף את המידע למסד הנתונים ומשתמש בשירות הענן של google על מנת לשלוח למכשיר של משתמש 2 הודעת push המציינת שקיימת תמונה חדשה בשבילו בשרת.
4. המכשיר של משתמש 2 מקבל את הודעת ה push, פונה לשרת ומבקש ממנו את פרטי ה duel.
5. השרת מחזיר למשתמש 2 את התמונה שצולמה עבורו ואת 4 האפשרויות למילים, כאשר אחת מהאפשרויות היא המילה שמשתמש 1 הכניס ושלושת האפשרויות האחרות נבחרות אקראית מתוך המסד נתונים.

6. המכשיר של משתמש 2 מעבד את התמונה בעזרת האלגוריתם לעיבוד תמונה שנבחר בשרת, שומר את התוצאה במכשיר ומודיע למשתמש (בעזרת notification) שיש duel חדש שממתין עבורו.
7. כאשר משתמש 2 בוחר להגיב אל ה duel הוא מגיע למשחק הניחוש. כאשר המשתמש בוחר באחת האפשרויות (או שנגמר הזמן), המכשיר שלו פונה לשרת ומבקש ממנו לעדכן את תוצאת המשחק.
8. השרת מעדכן במסד נתונים את התוצאה ומחזיר למשתמש את תוצאת ה duel.
9. משתמש 2 רואה את תוצאת ה duel ומתבקש לשלוח בחזרה למשתמש 1 תמונה חדשה. בשלב זה ה flow מתחיל מהתחלה.

מבנה המסד נתונים

המסד נתונים מורכב מהטבלאות הבאות:

הטבלה Users:

הטבלה שומרת פרטים על כל המשתמשים באפליקציה.
הטבלה מכילה את השדות הבאים:

- Id – מספר סידורי (מזהה יחודי).
- Nickname – הכינוי של המשתמש (במידה ונרשם רגיל ולא התחבר דרך הפייסבוק).
- Os – מערכת ההפעלה של המכשיר עליו רצה האפליקציה (android/iphone).
- UDID – מספר סידורי של המכשיר.
- Mail – האימייל שהמשתמש הזין בעת הרישום.
- GCMID – מספר סידורי המזהה את המכשיר בשירות הענן של גוגל.

- Fbid – ה id של פרופיל הפייסבוק של המשתמש (במידה והתחבר עם הפייסבוק).
- Fbname – השם של המשתמש כפי שמופיע בפרופיל הפייסבוק שלו (במידה והתחבר עם הפייסבוק).
- Score – הניקוד המצטבר של המשתמש.
- gamesWon – מספר הניצחונות של המשתמש.
- gamesLost – מספר ההפסדים של המשתמש.

הטבלה Game:

הטבלה מכילה מידע על כל המשחקים הפתוחים בין זוגות משתמשים.
הטבלה מכילה את השדות הבאים:

- Id – מספר מזהה של המשחק (מזהה ייחודי).
- sourceUser – המספר מזהה של המשתמש שהתחיל את המשחק.
- Opponent – המספר מזהה של המשתמש שמולו נפתח לראשונה המשחק.
- scoreSourceUser – מספר הניצחונות של ה sourceUser במשחק.
- scoreOponent – מספר הניצחונות של ה opponent במשחק.
- isActive – שדה גדל המציין האם המשחק פעיל או לא (המשחק נהיה לא פעיל כאשר אחד המשתמשים בוחר לבטל את המשחק).

הטבלה Duel:

הטבלה מכילה מידע לגבי כל ה duels שהתקיימו.
הטבלה מכילה את השדות הבאים:

- Id – מספר מזהה של ה duel (מזהה ייחודי).
- GameId – המספר מזהה של המשחק שבמסגרתו מתבצע ה duel הזה.
- RightWordId – המזהה (בטבלת ה Words) של המילה שאותה הכניס המשתמש.
- Word1ID – המזהה של אחת המילים שנבחרו באופן אקראי להיות אחת האפשרויות לניחוש ב duel הזה.
- Word2ID – המזהה של אחת המילים שנבחרו באופן אקראי להיות אחת האפשרויות לניחוש ב duel הזה.
- Word3ID – המזהה של אחת המילים שנבחרו באופן אקראי להיות אחת האפשרויות לניחוש ב duel הזה.
- Score – הציון של המשתמש ב duel (אחרי שהוא ניחש).
- ImageMixingMethod – מספר מזהה של האלגוריתם לעיבוד תמונה שייעשה בו שימוש ב duel זה.

- sourceUser – המספר המזהה של המשתמש ששלח את ה duel.
- Processed – גדל שמציין האם משתמש היעד של ה duel כבר הפעיל את האלגוריתם לעיבוד תמונה על התמונה שנשלחה או לא.
- isActive – דגל המציין האם ה duel פעיל (duel מפסיק להיות פעיל ברגע שמשתמש יעד סיים לנחש ועודכן ה score).

הטבלה Words

הטבלה מכילה את כל המאגר מילים מהם יילקחו 3 האפשרויות הנוספות לניחוש של המילים.
הטבלה מכילה את השדות הבאים:

- Id – מספר מזהה (מזהה ייחודי).
- Word – המחרוזת שמייצגת את המילה.

פיצ'רים שמומשו באפליקציה

כדי לשפר את חווית השימוש באפליקציה, בנוסף למימוש הבסיסי הוספו את הפיצ'רים הבאים:

1. **התחברות באמצעות הפייסבוק** – במקום להרשם לאפליקציה, ניתן להתחבר עם חשבון הפייסבוק.
התחברות עם חשבון הפייסבוק מאפשר לשחק עם חברי פייסבוק שגם התקינו את האפליקציה.
כמו כן, ניתן **לשלוח הזמנה** למשחק לחברים בפייסבוק שעדיין אין להם את האפליקציה.
כאשר המשתמש מתחבר עם חשבון הפייסבוק שלו הוא רואה את תמונת הפרופיל שלו ושל חבריו לפייסבוק שמולם הוא משחק.
2. **ביטול משחק** – ניתן לבטל משחק ע"י ביצוע swipe על המשחק במסך הראשי. כאשר משתמש מבטל את המשחק, היריב יקבל הודעה על ביטול המשחק כאשר ינסה לשחק מולו.
לאחר ביטול משחק ניתן לפתוח משחק חדש מול אותו יריב.
3. **Notifications** – כאשר נשלח את השחק duel חדש, הוא מקבל במכשירו notification (עם צליל ורטט) שמודיע לו שנשלח אליו duel חדש וממי הוא נשלח. המשתמש יכול ללחוץ על ה notification ולהגיע ישירות למסך הניחוש.
4. **שימוש ב GCM (שירות הענן של גוגל)** – על מנת ליצור חווית משחק טובה יותר, נעשה שימוש בשירות הענן של גוגל כדי לשלוח הודעות push למכשירים. הדבר לבצע את עיבוד התמונה ברקע בלי שהמשתמש מודע לכך, לכן כאשר המשתמש מקבל את ה notification התמונה כבר עובדה ומוכנה למשחק (הוא לא נדרש לחכות).
כמו כן, המשתמש מיודע כאשר מגיע אליו משחק חדש, והוא לא נדרש לפתוח את האפליקציה בכל פעם כדי לבדוק.
5. **סיכום פרופיל משתמש** – במסך הראשי של האפליקציה מוצג למשתמש סיכום של הניקוד שצבר עד כה וכן מספר הניצחונות הכולל שלו (וכן התמונה שלו במידה והתחבר עם הפייסבוק).
6. **Top Players** – המשתמש יכול לראות את רשימת 5 השחקנים הטובים ביותר על פי הניקוד הכולל שלהם ועל פי מספר הנצחונות.
7. **משחק מול שחקן אקראי** – בנוסף לאפשרויות לשחק מול חבר פייסבוק ומול שחקן שנבחר לפי כתובת מייל, ישנה אפשרות גם לשחק מול שחקן אקראי מתוך כל המשתמשים של האפליקציה.

הצעות לשיפור

1. הוספת עוד אלגוריתמים לעיבוד תמונה.
2. אפשרות לצרף הודעת טקסט קצרה כאשר שולחים duel חדש ליריב.
3. הוספת אפשרויות נוספות לחיפוש אנשים למשחק (חשבון ה gmail, אנשי קשר וכדומה).
4. הוספת אפשרות לחסום שחקנים (במידה ומשתמש רוצה שמשתמש אחר לא יפתח מולו משחקים).
5. הוספת אפשרות ל"נדנוד", כלומר במידה ושחקן 1 שלח לשחקן 2 תמונה, ושחקן 2 לא מגיב אליה, שחקן 1 יוכל לשלוח לו "נדנוד" ולהתריע לשחקן 2 שהוא עדיין מחכה לתשובה.
6. לאפשר להמיר את הניקוד שנצבר בפתיחת פי'צרים נוספים באפליקציה למשל להמיר ניקוד באלגוריתמי עיבוד מתקדמים נוספים או הוספת זמן לניחוש וכדומה.