

המכללה להוראת טכנולוגיה

עבודת גמר לתואר ראשון B.Ed. Bachelor of Education במחשבים וטכנולוגיות מידע

בנושא: משחק תלת-מימדי DOOM 4D

מגיש: דניס אסטחוב ת.ז. 0000000000
מנחה: פרופ' ערמוני אביעד

תאריך הגשה

המכללה להוראת טכנולוגיה מיסודה של המכללה למינהל (ע"ר)

מתחם אנטיגנוס רח' אנטיגנוס 8, ת"ד 22674, תל-אביב 62684. טל: 03-5441370 פקס: 03-5466264

מתחם קלישר רח' קלישר 5, תל-אביב 65257. טל: 03-5165535 פקס: 03-5165536

נלדיה לאמנות עכשווית נלדיה רחל וישראל פולק של המכללה להוראת טכנולוגיה, רח' קלישר 5 תל-אביב 65257

www.mictec.ac.il

הצעת פרויקט

המכללה להוראת טכנולוגיה מיסודה של המכללה למינהל (ע"ר)

מתחם אנטיגנוס רח' אנטיגנוס 8, ת"ד 22674, תל-אביב 62664. **טל:** 03-5441370 **פקס:** 03-5466264

מתחם קלישר רח' קלישר 5, תל-אביב 65257. **טל:** 03-5165535 **פקס:** 03-5165536

גלריה לאמנות עכשווית גלריה רחל ישראל פולק של המכללה להוראת טכנולוגיה, רח' קלישר 5 תל-אביב 65257

www.mictec.ac.il

המכללה להוראת טכנולוגיה

רח' אנטיגונוס 6, ת"ד 22674, תל-אביב 62664

מגמה: מחשבים וטכנולוגיות מידע

מסלול: השלמת הנדסאים לתואר ראשון

שם סטודנט : דניס אסטחוב
תעודת זהות : 000000000
כתובת : פתח-תקוה
טלפונים : 03-0000000, 054-0000000
תואר הנדסאי : בית הספר להנדסאים של אוניברסיטת תל-אביב
סיום הנדסאי : שנת 2001
סיום B.Ed. : שנת 2004

המכללה להוראת טכנולוגיה מיסודה של המכללה למינהל (ע"ר)

מתחם אנטיגונוס רח' אנטיגונוס 6, ת"ד 22674, תל-אביב 62664. טל: 03-5441370 פקס: 03-5466264

מתחם קלישר רח' קלישר 5, תל-אביב 65257. טל: 03-5165535 פקס: 03-5165536

גלריה לאמנות עכשווית גלריה רחל ישראל מולק של המכללה להוראת טכנולוגיה, רח' קלישר 5 תל-אביב 65257

www.mitectec.ac.il

שם פרוייקט:

משחק תלת-מימדי DOOM Fourth Dimension

מטרת פרוייקט:

- כתיבת משחק תלת-מימדי המציג יכולות גרפיקה של טכנולוגיה Microsoft DirectX7.
- פיתוח ומימוש אלגוריתמים בגרפיקה ממוחשבת.
- מימוש כל הידע הנרכש במהלך הלימודים לקראת תואר ראשון.

המכללה להוראת טכנולוגיה מיסודה של המכללה למינהל (ע"ר)

מתחם אנטיגונוס רח' אנטיגונוס 8, ת"ד 22674, תל-אביב 62664. טל: 03-5441370 פקס: 03-5466264

מתחם קלישר רח' קלישר 5, תל-אביב 65257. טל: 03-5165535 פקס: 03-5165536

נלדיה לאמנות עכשווית נלדיה רחל ישראל פולק של המכללה להוראת טכנולוגיה, רח' קלישר 5 תל-אביב 65257

www.mictec.ac.il

חלוקת עבודה:

1. לימוד מתקדם של Visual Basic.
2. לימוד מתקדם של DirectX 7.0.
 - (א) שימוש ב-DirectDraw.
 - (ב) שימוש ב-DirectInput.
 - (ג) שימוש ב-DirectSound/DirectMusic.
 - (ד) שימוש ב-Direct3D.
3. תכנון GUI ומסמכים.
4. כתיבת תוכנת המשחק.
5. בדיקות וחיפוש שגויות.
6. כתיבת ספר הפרוייקט.

המכללה להוראת טכנולוגיה מיסודה של המכללה למינהל (ע"ר)

מתחם אניגנוס רח' אניגנוס 8, ת"ד 22674, תל-אביב 62664. טל: 03-5441370 פקס: 03-5466264

מתחם קלישר רח' קלישר 5, תל-אביב 65257. טל: 03-5165535 פקס: 03-5165536

גלריה לאמנות עכשווית גלריה רחל ישראל פולק של המכללה להוראת טכנולוגיה, רח' קלישר 5 תל-אביב 65257

www.mitect.ac.il

לוח זמנים לביצוע פרויקט:

| תקופה | שעות עבודה | נושא |
|------------------|------------|------|
| יוני | 40 | 1 |
| יולי | 100 | 2 |
| יולי | 20 | 3 |
| אוגוסט - אוקטובר | 350 | 4 |
| נובמבר | 50 | 5 |
| דצמבר | 40 | 6 |

ביצוע פרויקט בשנת 2004

המכללה להוראת טכנולוגיה מיסודה של המכללה למינהל (ע"ר)

מתחם אנטיגונוס רח' אנטיגונוס 6, ת"ד 22674, תל-אביב 62664. טל: 03-5441370 פקס: 03-5466264

מתחם קלישר רח' קלישר 5, תל-אביב 65257. טל: 03-5165535 פקס: 03-5165536

גלריה לאמנות עכשווית גלריה רחל ישראל מולק של המכללה להוראת טכנולוגיה, רח' קלישר 5 תל-אביב 65257

www.mitectec.ac.il

ציוד לביצוע פרויקט:

ציוד חומרה:

- מחשב PC עם מעבד AMD Duron 1200Mhz
- זיכרון פנימי 352Mb RAM
- דיסק קשיח 40 Gb
- כרטיס מסך ATI Radeon 7500 עם מעבד 3D פנימי.
- דיסק 32Mb USB Disk On-Key
- מסך, מקלדת, עכבר, Disk on-key 32Mb
- מדפסת Lexmark X1130
- חיבור לאינטרנט

ציוד תוכנה:

- מערכת הפעלה Windows XP
- שפת פיתוח Visual Basic 6.0 / NET
- ספריות גרפיקה DirectX 9.0b RunTime
- Microsoft Word 2002
- תוכנה לעיבוד תמונות PV 2.78
- Microsoft Paint 5.1
- WinRar 3.41

המכללה להוראת טכנולוגיה מיסודה של המכללה למינהל (ע"ר)

מתחם אניגנוס רח' אניגנוס 8, ת"ד 22674, תל-אביב 62664. טל: 03-5441370 פקס: 03-5466264

מתחם קלישר רח' קלישר 5, תל-אביב 65257. טל: 03-5165535 פקס: 03-5165536

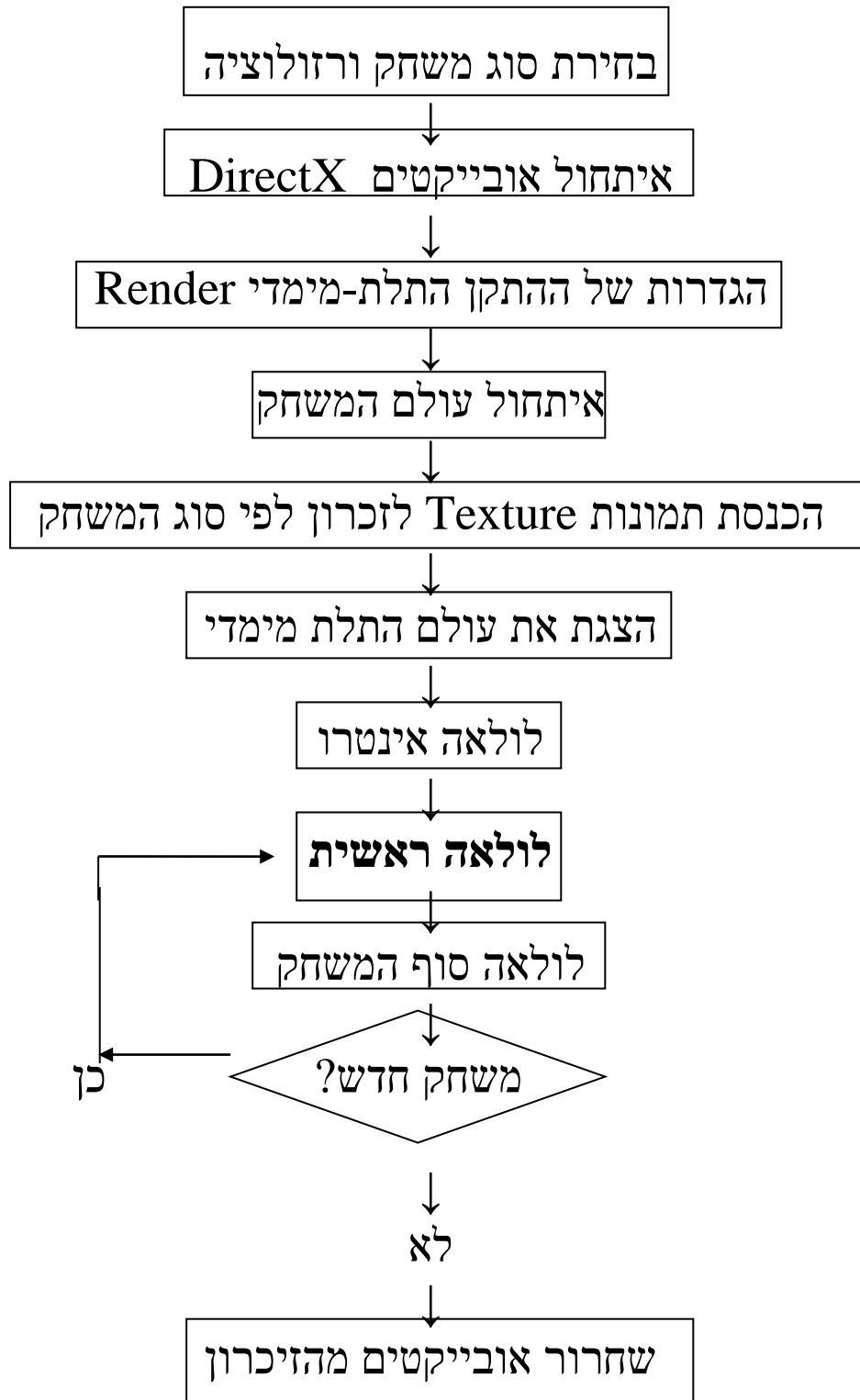
גלריה לאמנות עכשווית גלריה רחל ישראל פולק של המכללה להוראת טכנולוגיה, רח' קלישר 5 תל-אביב 65257

www.mitect.ac.il

תוכן עניינים

| | |
|-----|--|
| 8 | 1. תרשים זרימה של התוכנת המשחק |
| 10 | 2. מבוא ל-DirectX |
| 12 | • גרפיקה תלת-מימדית |
| 14 | • מטריצה (Matrix) |
| 17 | 3. מבנה עולם של משחק DOOM 4D |
| 18 | • אופן בניה של עולם תלת-מימדי |
| 19 | • מרקמים (Textures) |
| 23 | • אופן בניה של עצים |
| 24 | • אופן בניה של אפקט אשן/אש |
| 25 | • אופן הזזה של שחקן בעולם 3D |
| 26 | • אלגוריתם פגיה למטרה |
| 28 | 4. מהלך וחוקים של המשחק |
| 30 | 5. מדריך למשתמש |
| 31 | • דרישות מינימליות למשחק |
| 32 | • התקנת משחק |
| 35 | • התחלת משחק |
| 38 | • סיום משחק |
| 39 | • החלפת מוסיקה במשחק |
| 41 | 6. משתנים של התוכנה |
| 50 | 7. פונקציות של התוכנה |
| 61 | 8. גירסה מסחרית של Doom Fourth Dimension |
| 62 | 9. קוד מלא של התוכנה |
| 105 | 10. סקירת ספרות וביבליוגרפיה |

תרשים זרימה של התוכנת המשחק



מבוא ל-DirectX

DirectX זה ספריית של חברת **Microsoft** שנבנה לשם בניית משחקים ותוכנות עם ביצועי מולטימדיה בעלי פונקציות לגישה ישירה לחומרת המחשב, הכוללים תמיכה לגרפיקה דו-מימדית, תלת-מימדית, צלילים ומוסיקה, קלט כמו עכבר, מקלדת ועוד, כמו כן תקשורת בין רשתות מחשבים לשם יצירת משחקים ותוכנות רבי משתתפים.

DirectX מכיל מספר ספריות שונות, כגון:

DirectDraw – הוא אוסף של פונקציות המאפשר גישה ישירה להתקני כרטיס המסך, תוך שמירה על תאימות עם ממשק הגרפיקה של חלונות שנקרא:
(Graphics Device Interface) GDI

DirectInput – מאפשר קלט מהתקני שליטה כגון: עכבר, מקלדת, מוט-היגוי והתקני שליטה אחרים.

DirectSound/DirectMusic – מאפשר השמעת מוסיקה וצלילים בזמן אמת, על ידי גישה ישירה לכרטיס הקול שמותקן במחשב.

Direct3D – פותח לשם יצירת משחקים אינטראקטיביים ברמה גבוהה עם גרפיקה תלת-מימדית בסביבת Windows. הוא מאפשר גישה מותנית ביחס לכרטיס המסך שנמצא במחשב. גישה תלויה בכרטיס מסך!!!

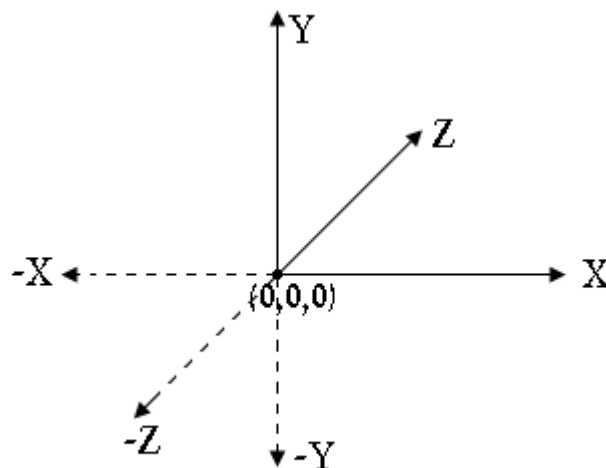
DirectPlay – מאפשר ליצור משחקים מרובי משתתפים
(MultiPlayer Games), שרתי משחק (Game Servers),
תוכנות למסרים מיידיים ועוד.

גרפיקה תלת-מימדית

במערכת צירים תלת-מימדית יש לנו 3 סוגי צירים:
ציר X – מתאר מיקום אנכי במרחב.
ציר Y – מתאר מיקום אופקי במרחב.
ציר Z – מתאר עומק במרחב.

ראשית הצירים ממוקמת בנקודה $(0,0,0)$ ונקודה זו נמצאת במרכז של המסך.

נקודה במערכת הצירים תלת-מימדים נקראת Vector.



ווקטורים ומקומם במערכת XYZ:

| Direction | Vector |
|-----------|------------------|
| ימינה | $X=1, Y=0, Z=0$ |
| שמאלה | $X=-1, Y=0, Z=0$ |
| למעלה | $X=0, Y=1, Z=0$ |
| למטה | $X=0, Y=-1, Z=0$ |
| פנימה | $X=0, Y=0, Z=1$ |
| החוצה | $X=0, Y=0, Z=-1$ |

פוליגון (Polygon) הוא צורה גרפית המורכבת מכמה
ווקטורים, בדרך כלל מ-3 וקטורים.

מטריצה (Matrix)

מטריצה היא אוסף של מספרים מסודרים בעמודות ושורות:

$$\begin{bmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{bmatrix}$$

מטריצה זו נקראת 4X4 מכיון שיש לה 4 שורות ו-4 עמודות.
זו מטריצה ש-DirectX משתמש בה.

באמצעות מטריצות נוכל לסובב, להזיז, להגדיל ולהקטין
וקטורים ופוליגונים.

Identify Matrix – משתמשים במטריצה זו לשם נקודת
התחלה לפני שמבצעים שינויים (Transformations):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Move Matrix – משתמשים במטריצה זו כדי למקם אובייקטים במסך. הערכים vX , vY , vZ הם ערכי המיקום החדש:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ vX & vY & vZ & 1 \end{bmatrix}$$

Zoom Matrix – משתמשים במטריצה זו כדי לשנות קנה מידה של אובייקטים במסך. הערכים vX , vY , vZ הם ערכי קני המידה החדשים:

$$\begin{bmatrix} vX & 0 & 0 & 0 \\ 0 & vY & 0 & 0 \\ 0 & 0 & vZ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

RotateX Matrix – משתמשים במטריצה זו כדי לסובב אובייקטים סביב ציר X:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(a) & \sin(a) & 0 \\ 0 & -\sin(a) & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

RotateY Matrix – משתמשים במטריצה זו כדי לסובב
אובייקטים סביב ציר Y:

$$\begin{bmatrix} \cos(a) & 0 & -\sin(a) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(a) & 0 & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

RotateZ Matrix – משתמשים במטריצה זו כדי לסובב
אובייקטים סביב ציר Z:

$$\begin{bmatrix} \cos(a) & \sin(a) & 0 & 0 \\ -\sin(a) & \cos(a) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

מבנה עולם של משחק DOOM 4D

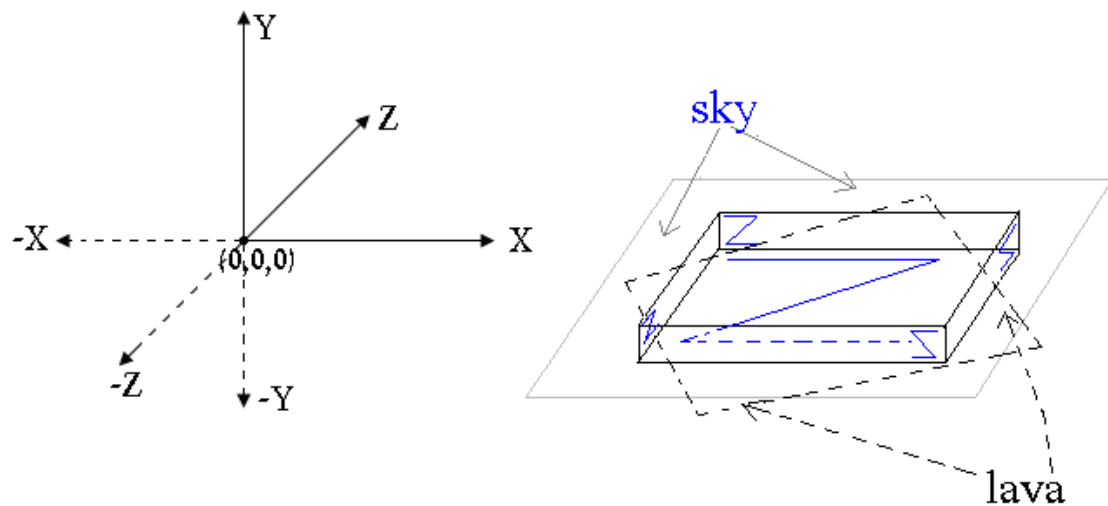
עולם בנוי במערכת הצירים תלת-מימדי XYZ.
בהנחה שפוליגון אחד בנוי מ-3 ווקטורים בעולם יש כ-10000
פוליגונים.

עולם בנוי מחלקים הבאים:

1. שטח (רצפה) שבו אנחנו יכולים לזוז
2. קירות מסביב שמוגבלות הזזה של שחקן.
3. מתחת לרצפה יש פלטה מסתובבת הנותנת אפקט של לאבה.
4. למעלה מעל הקירות יש שמיים.
5. על שטח משחק מפוזרים עצים בעופן מיקרי (random).
6. על שטח משחק מפוזרים דמויות של אש.
7. על שטח משחק מפוזרים דמויות של אשן.
8. דמות של אייב זזה בשטח.

אופן בניה של עולם תלת-מימדי

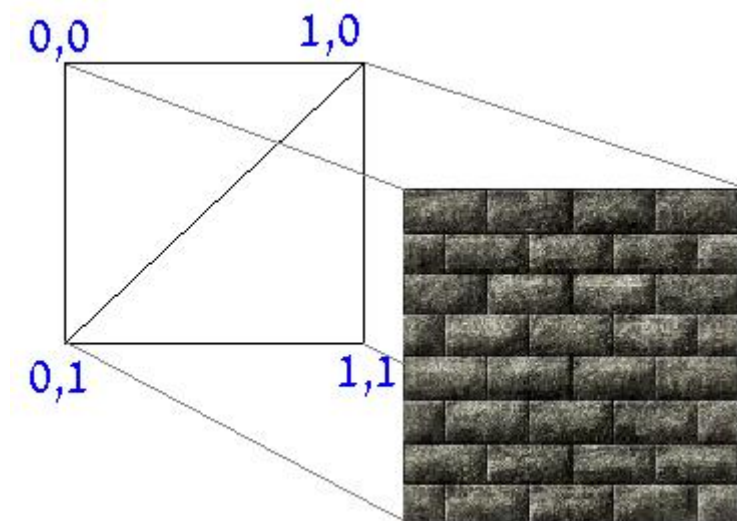
בגרף ניתן לראות עופן בניה של עולם תלת-מימדי:



sky – שמיים הנמצאים מעל הקירות.
lava – לאבה הנמצאת מתחת לרצפה ומסתובבת.
Z – סדר בניית נקודות של פוליגונים ל-Texture.

מרקמים (Textures)

במשחק משתמשים במרקמים כדי ליצור יופי ולהראות עיצוב מדהים של המשחק, את המרקמים מודבקים לכל 2 פוליגונים בעופן הבאה:



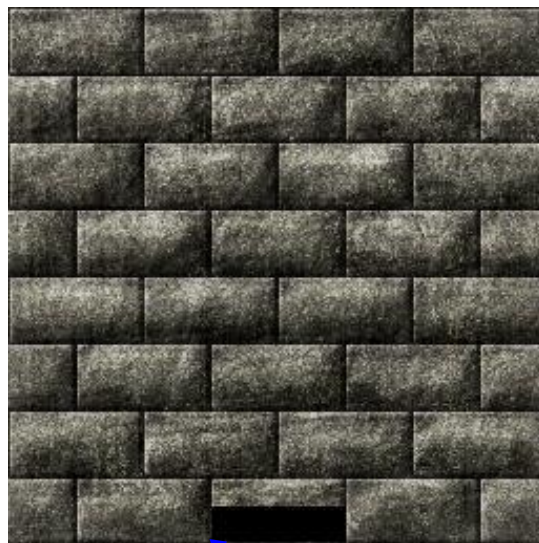
מרקם שמראה את השטח המשחק נראה כך:



מרקם שמראה את הקיר בלי חלק שקוף במשחק נראה כך:

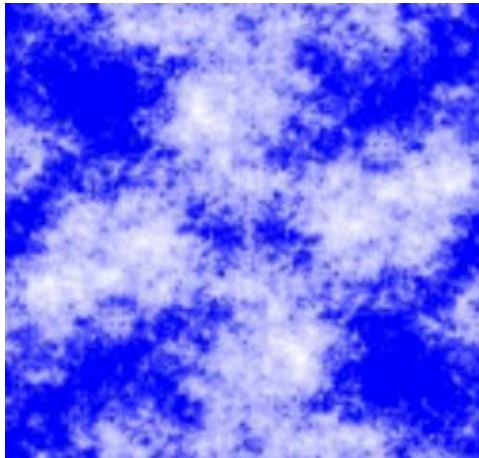
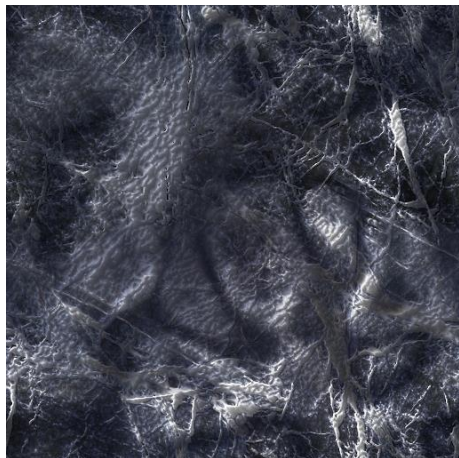


מרקם שמראה את הקיר עם חלק שקוף במשחק נראה כך:

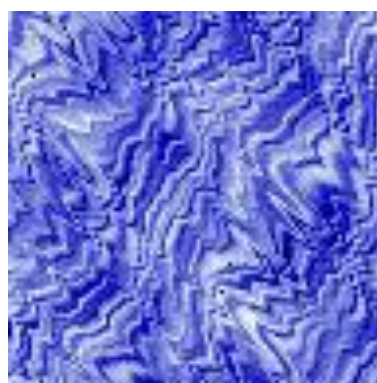


חלק שקוף במרקם הוא חלק בצבע שחור $RGB(0,0,0)$
חצי מהאבן תחתון מרכזי הוא חלק שקוף במשחק.

מרקם שמראה את שמיים במשחק נראה כך:



מרקם שמראה את הלאבה במשחק נראה כך:



מרקמים שמראים את סוגי של עצים במשחק נראים כך:



חלק בצבע שחור $RGB(0,0,0)$ הוא חלק שקוף במשחק
אז רואים רק את העצים בלי רקע שחור מסביב.

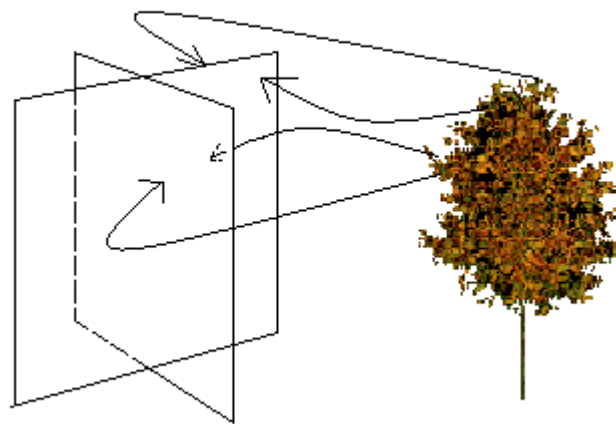
אופן בניה של עצים

דמות של עץ בנוי ככה שלא משנה מאיזה כיוון אנחנו מסתקלים אל העץ, אנחנו תמיד רואים עץ שלם למרות שבסה"כ עץ זה תמונה דו-מימדית.

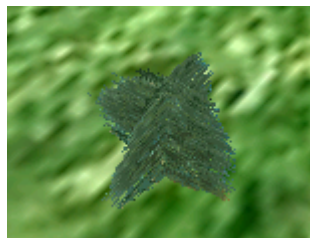
קודם כל מאחורי תמונה אנחנו מציגים אוד פעם את התמונה כדי שנוכל לראות אותה אם ניסתקל מאחוריה.

דבר שני, אנחנו אושים הצלבה של תמונות פירושו של דבר זה שאנחנו מציגים את התמונה שוב פעם אל התמונה המקורית רק בזווית-90% ממנה.

בגרף הבאה ניתן לראות את אופן בניה של עצים:



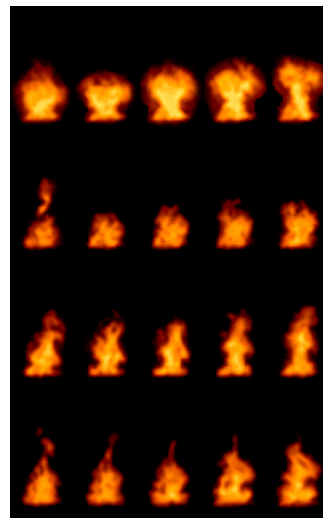
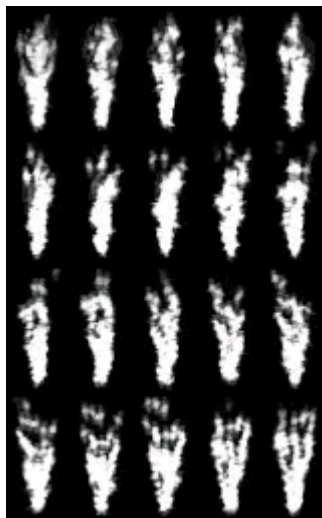
כתוצאה מכך כדי להציג עץ אחד במערכת תלת-מימדית אנחנו חייבים להציג 4 פעמים תמונה של עץ, בתמונה הבאה ניתן לראות עצ מלמאלה:



אופן בניה של אפקט אשן/אש

אפקט אשן ואש בנוי מתמונת מטריצה שבטוחה יש סרטון תזוזה של אשן או אש.

תמונות מטריצה שמיצגות אפקט אש ו אשן בנויות באופן הבא:



אנחנו משתמשים בפונקציה שגוזרת מהתמונה רק את החלק הנדרש להצאיגו, אחרי פרק זמן קצר גוזרים חלק אחר (Next Frame) וכך רואים סרט זז.

אופן הזזה של שחקן בעולם 3D

השחקן, זה באצם דמות שלנו, אנחנו רואים בעזרת עיניים שלו. עיניים של שחקן זה Camera (מצלמה) שאותה אנחנו מזזים במערכת תלת-מימדית.

השחקן זז בעזרת 4 מקשים:

חץ שמאלה – הזזה שמאלה

חץ ימינה – הזזה ימינה

חץ למאלה – הזזה קדימה

חץ למעלה – הזזה אחורה

ניתן גם לשלב מקשים כדי ליצור תזוזה באלכסון.

המצלמה בנויה משני חלקים של קורדינטות:

חלק ראשון: מקום שבו מצלמה אומדת. $Z Y X$

חלק שני: מקום שאליו מצלמה מסתכלת. $X1 Y1 Z1$

אז לתזוזה במערכת 3D אנחנו חייבים לשנות את שני קורדינטות של מצלמה

אלגוריתם פגיה למטרה

מטרה שלנו זה דמות של המחשב (עוייב) שיש לו מיקום שלו במערכת 3D.
אצל מצלמה יש מיקום משלה ומקום שאליו מצלמה מסתכלת מקום שאליו מסתכלים שימושו נוסף זה הסימן של מקום שבו אנחנו יורים.

בזמן יריה כדי לבדוק העם מטרה הייתה בשטח שבין נקודה שאליה מסתכלים לבים מיקום של מצלמה בנוסף צריך לבדוק העם זווית בין מצלמה למטרה לבין מצלמה ומיקום הסטכלות נימצא בזווית כבערך 4 ראדיאן תלוי גם במרחק ממצלמה למטרה כי כמה שמטרה יותר קרובה היא יותר גדולה ולהפך.

אז נניח:

XM - מיקום X של מטרה

ZM - מיקום Z של מטרה

Dist2focus – מרחק בין מצלמה למקום שאליו מסתכלים

Dist2target – מרחק בין מצלמה למטרה

בהתחלה בודקים העם מרחק:

If (dist2target <= dist2focus) Then

אם כן אז מטרה נימצאת באזור שבו אנחנו יכולים לפגוע במטרה, אכשיו צריך לבדוק את הזווית בין קוים: מצלמה-מקום הסטכלות לבין מצלמה-מטרה את זה אנחנו יכולים לחשב שפי נוסחה מאלגברה אנליטית:

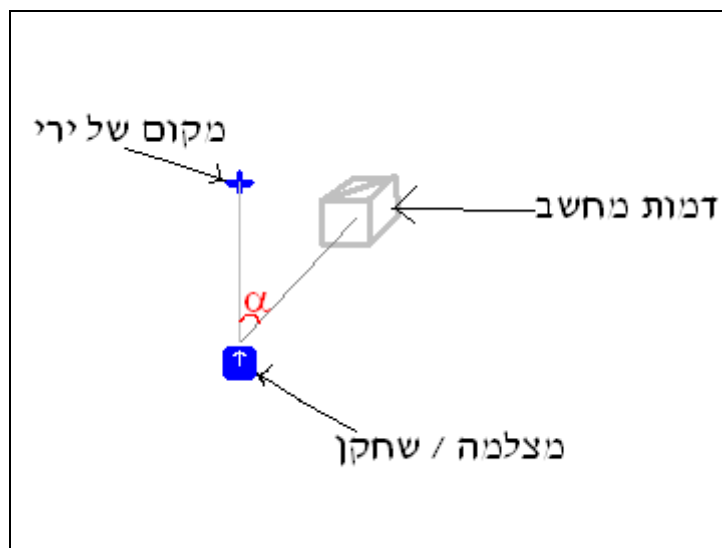
$$\cos \alpha = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} = \frac{a_x b_x + a_y b_y + a_z b_z}{\sqrt{a_x^2 + a_y^2 + a_z^2} \sqrt{b_x^2 + b_y^2 + b_z^2}}$$

$\cos a$ – זווית בין שני וקטורים, בין וקטור \vec{a} לבין וקטור b
 אחרי זה בודקים העם זווית שקיבלנו הוא גדול מזווית
 של $\cos(4/\text{dist2target})$

If ($\cos\alpha \geq \cos(4 / \text{dist2target})$) Then

אם כן, אז זאת פגיה במטרה.

בגרף הבאה ניתן לראות איזה זווית α אנחנו מחשבים:



מהלך וחוקים של המשחק

אחרי התקנה והארצה של משחק משתמש יכול לבחור את רצולוציה שבה משחק יעבוד.
ברירת מחדל 800x600x16Bit.
כדי להתחיל משחק אצמו למשתמש ניתנת אפשרות לבחור באיזה אולם לשחק:
בקדור הארץ או באולם של מוות.

במהלך המשחק אנחנו נכנסים לארפל FOG בפרק זמן כ-60 שניות ארפל אובר ללילה ואחר כך שוב ליום.

אם משתמש מנסה ללכת מעבר לקיר אז מקבלים הודעה קולים.
אם משתמש פוגע במטרה אז רואים פיצוץ ומקבלים ראש של פגיה.

כאשר מחשב פוגע במשתמש אז יש צאקה של משתמש.

בהתחלת המשחק אצל שחקן ומחשב יש 100 נקודות של אנרגיה, בכל פגיה אנרגיה יורדת.
אם אנרגיה של המחשב ירדה ל-0 אז שומעים צאקה ועוברים לאולם אחר להרוג את המחשב.
אם אנרגיה של שחקן ירדה ל-0 אז מקבלים תמונה של סוף משחק ונתונים של משחק:

- כמות פעמים שהספקנו להרוג את המחשב
- כמות דקות "זמן" שבמהלכו שיחקנו.

המטרה במשחק היא: להרוג כמה שיותר דמויות של המחשב.
בכל שבל מחשב זו יותר ויותר מהר, את הנקודה שאליה הוא זו
ניבחרת ספונטני, אבל ליפעמים מחשב בוחר את נקודה שבו
אנחנו אומדים, לכן כדי לא לאמוד אלא תמיד להיות בתזוזה.

לכל אירוע יש תגובה קולית במשחק:

התחלת קרב

פגיע למטרה

מחשב פוגע בשחקן

התקרבות לקיר

ירי

מוות של מחשב

טלפורטציה

כל הקולות אלו משמעים באזרת DirectSound

כל המוסיקה במשחק משמעים באזרת ספרייה MCI32.OCX
Microsoft Multimedia Control 6.0

מדריך למשתמש

Welcome to the...



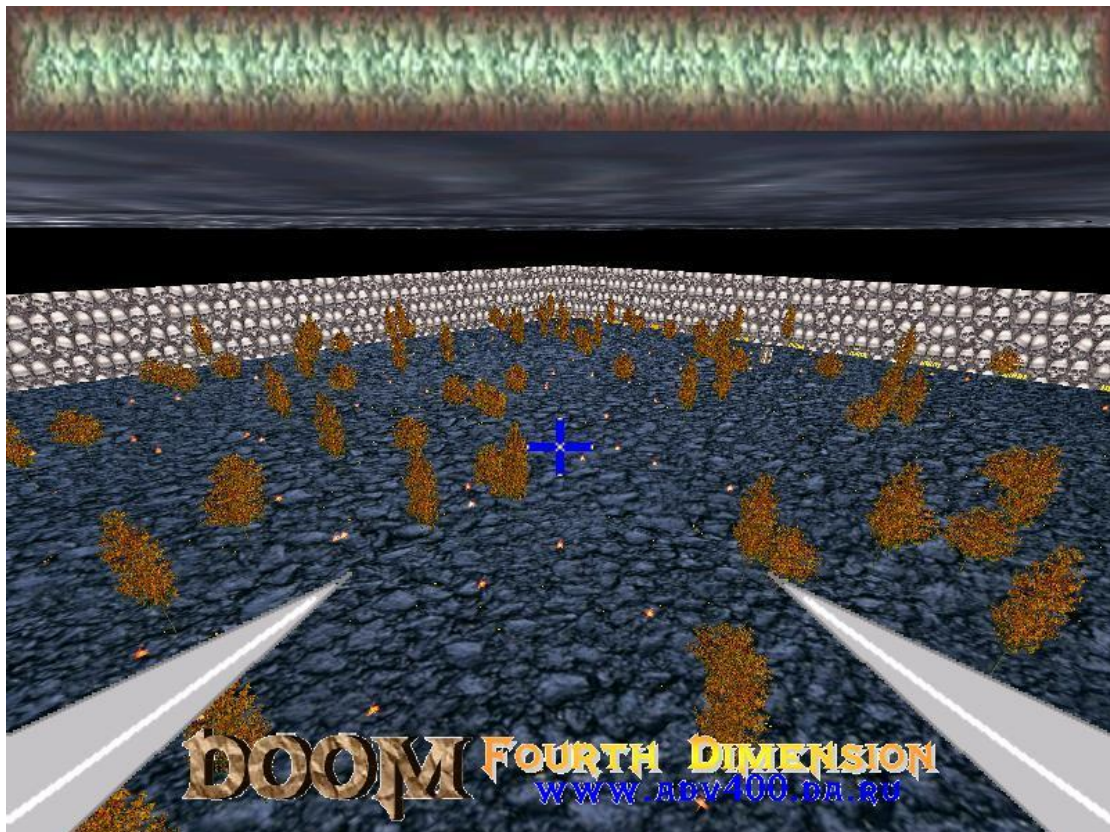
מדריך מחולק לכמה חלקים:

- דרישות מינימליות למשחק
- התקנת משחק
- התחלת משחק
- סיום משחק
- החלפת מוסיקה במשחק



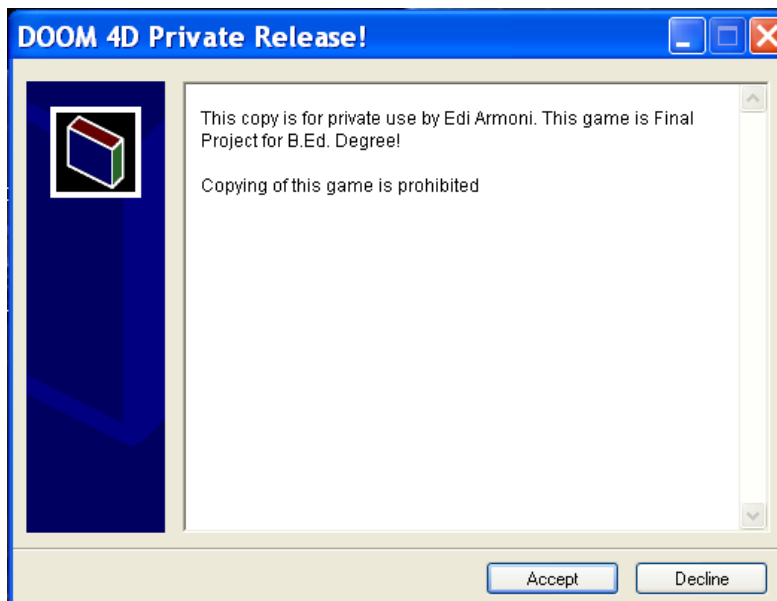
דרישות מינימליות למשחק:

- Pentium III 1000Mhz or AMD Duron 1200Mhz or higher
- Video Card with 32Mb RAM, (3D Accellerator optional)
- Windows 98, Me, XP.
- DirectX 7.0a Graphics Drivers
- 100Mb free on the Hard Disk
- Color Monitor with supporting VESA SuperVGA Modes
- Sound Blaster Card with supporting DirectSound 7.0
- Keyboard, Speakers.

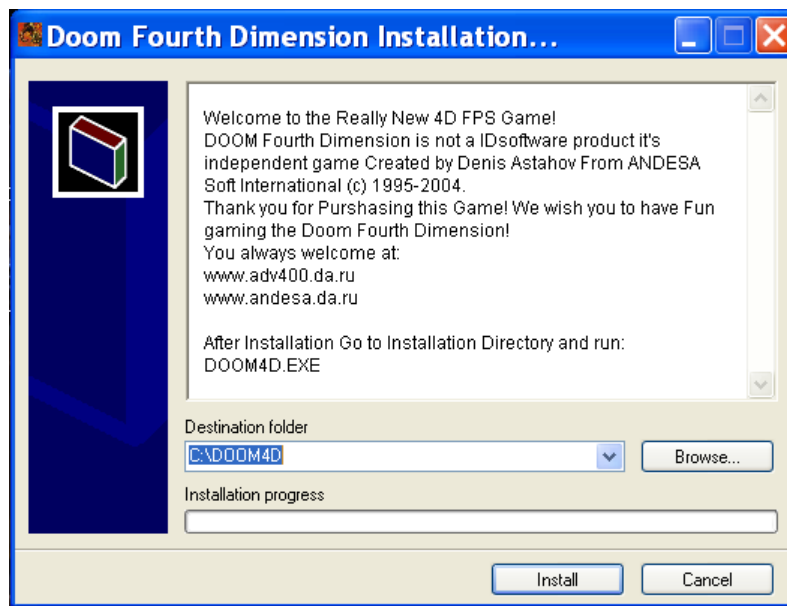


התקנת משחק:

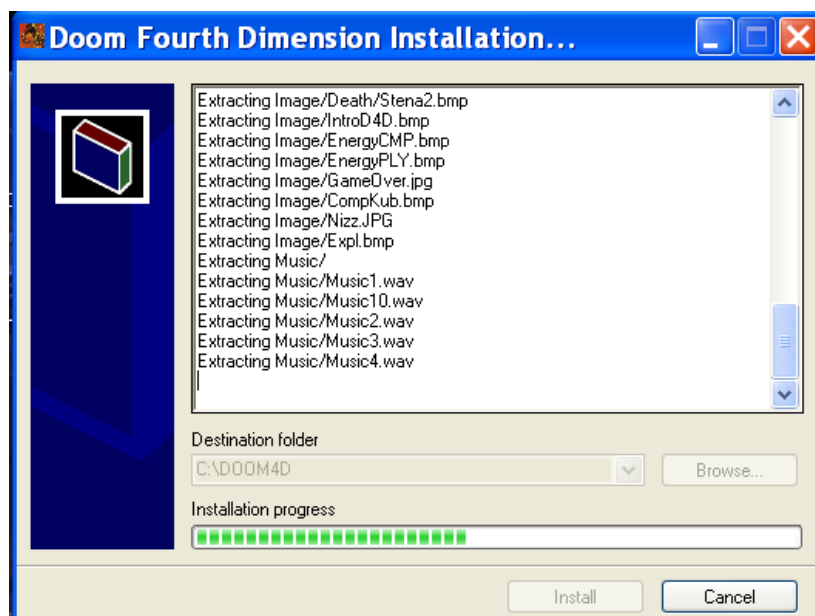
אם הכנסת דיסק התקנה ל-CDROM מערכת הפעלה Windows תפעיל אוטומטית קובץ התקנה של משחק. חלון ראשון שאנחנו רואים בתהליך התקנה הוא:



פז צריך ללחוץ אל כפתור **Accept** בחלון הבאה שאנחנו רואים אפשר לבחור סיפרייה שבה משחק יהיה הותקן, כברירת מחדל סיפרייה C:\DOOM4D\



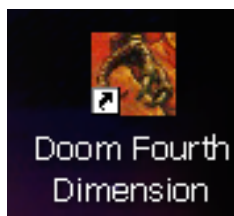
בוחרים ספרייה ולוחצים אל כפתור **Install** תהליך התקנה נמשך כדקה והוא תלוי במהירות של כונן דיסק קשיח וCDROM. בזמן התקנה אנחנו רואים את "פרוגרס" של התקנה:



אחרי התקנת המשחק אל שולחן עבודה "Desktop" תייצר קישור להארצת משחק.

כדי להריץ משחק Doom Fourth Dimension

צריך ללחוץ פעמיים על קישור:



מהלך משחק:

בחלון ראשון שאנחנו רואים אחרי הארצה של המשחק
אנחנו רואים כמה אפשרויות:



יש אפשרות לבחור רזולוציה שבה משחק יעבוד.

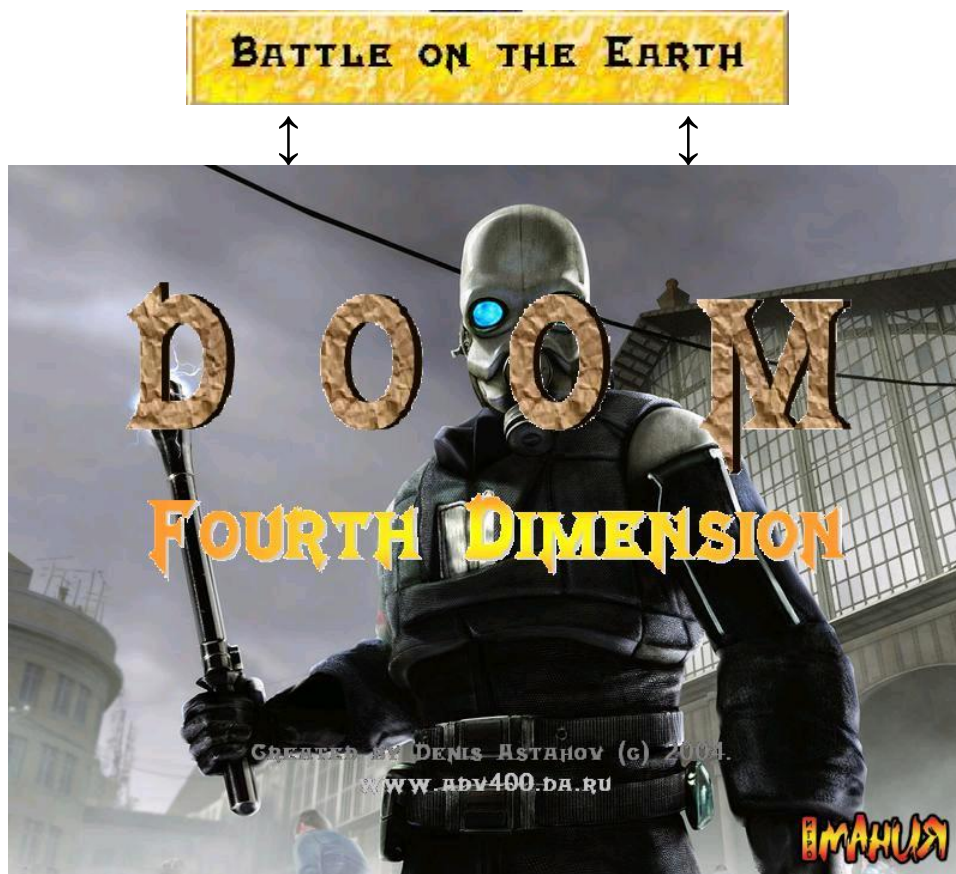
יש לזכור: רזולוציה יותר גבוהה דורשת חומרה יותר מהירה.
ברירת מחדל 800x600x16Bit

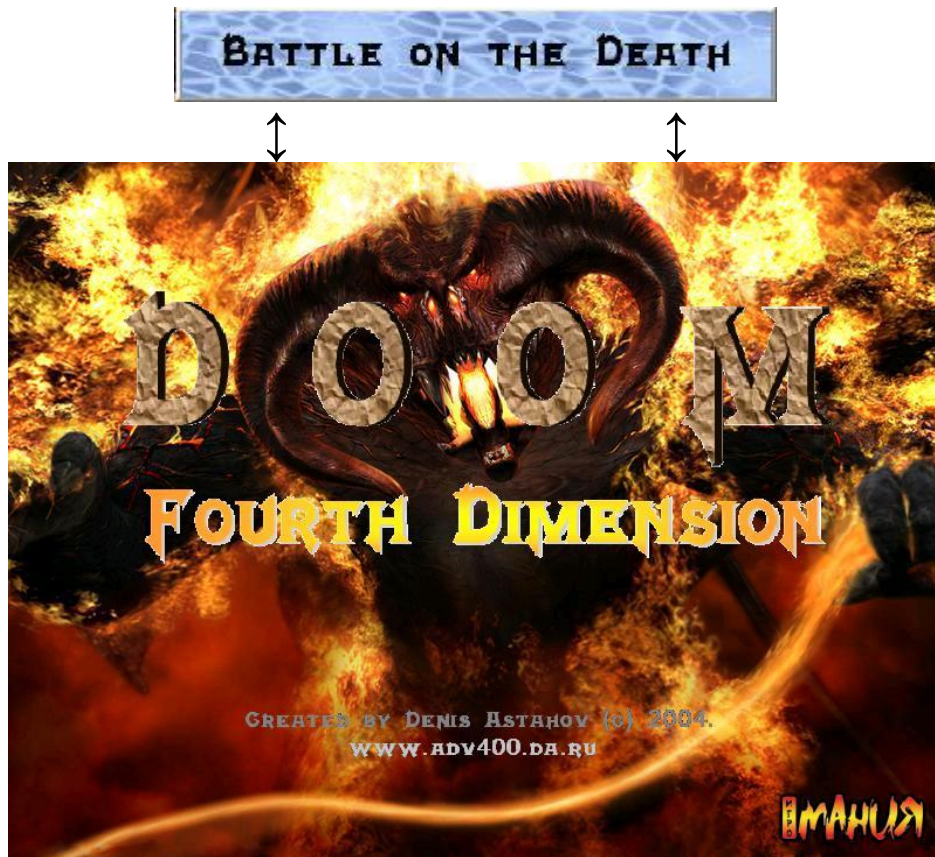
בנוסף אנחנו יכולים לבחור באיזה אולם נשחק,

מעביר עותנו לעולם
BATTLE ON THE EARTH
כפתור של קדור הארץ.

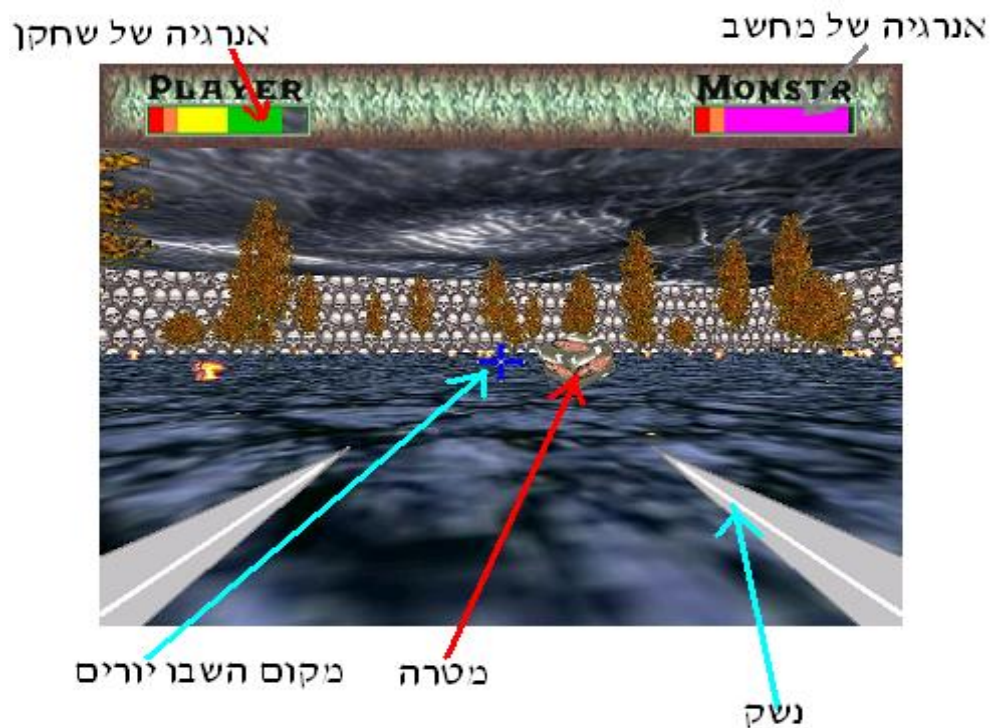
מעביר אותנו לעולם
BATTLE ON THE DEATH
כפתור של מוות.

אחרי בחירת סוג האולם ותלוי איזה סוג בחרנו
אנחנו ניכנסים לDEMO של המשחק
ומסך אחרון לפני התחלת המשחק לפי סוג המשחק נירה כך:





להתחלה של המשחק צריך ללחוץ מקש כלשהו (Any key)
החלון של משחק אצמו ותפקידי מרכיבם:



מקשים שימושיים במשחק:

← - להסתובב שמאלה
→ - להסתובב ימינה
↑ - לזוז קדימה
↓ - לזוז אחורה
Space – ירי

לתחילת המוסיקה יש ללחוץ מקשים מ **F1** עד **F10**
להפסקת מוסיקה יש ללחוץ על מקש **F11**
ליציאה מהמשחק יש ללחוץ על מקש **ESC**

מקשים נוספים סודיים:

F12 – טלפורטציה בין העולמות
Q – להפעיל ארפל לילה
W – להפעיל ארפל יום
1-2-3 – סוגים של ארפל
A – שקיפות של אולם

סיום משחק:

- חלון סיום של המשחק מגיא בשני המקרים:
- אנרגיה של שחקן ירדה ל-0.
- משתמש לחץ מקש ESC.

בחלון סוף המשחק:



יש נתונים הבאים:

- כמות דמויות מחשבים שהספקתם להרוג
- זמן שיחקתם בדקות

לפניכם שתי אפשרויות:

- להקיש מקש **N** למשחק חדש.
- להקיש מקש **ESC** ליציאה מתוכנת משחק.

החלפת מוסיקה במשחק

כל המוסיקה במשחק שמורה בסיפריית \Music בסיפריית ברירת מחדל C:\DOOM4D

כל קובץ מוסיקה זה קובץ MP3 עם הגדרות של קובץ WAV כדי להחליף הגדרות של הקובץ MP3 שלכם בסיפרייה \Music יש תוכנה **wavemp3.exe** דוגמא להחלפת מוסיקה:

1. מעתיקים קובץ MP3 שלכם לספרייה \Music במשחק
2. מריצים תוכנה **wavemp3.exe**
3. לוחצים על כפתור ! Wave כמו בתמונה



4. בוחרים את הקובץ MP3 שלכם וליחצים OPEN
5. מוחקים מוסיקה ישנה למשל קובץ Music3.wav
6. משנים שם של קובץ שלכם שכבר עם סיומת WAV ל שם של קובץ שמחקנו כלומר Music3.wav

עכשיו במשחק עם נלחץ F3 אז נישמע מוסיקה חדשה שלכם!

שם של קובץ ומספר בסוף אומר איזה מקש F יפעיל את המוסיקה לדוגמא:

F1 – מופעיל מוסיקה \Music\Music1.wav

F5 – מופעיל מוסיקה \Music\Music5.wav

משתנים של התוכנה

Public GameType As Byte

סוג משחק / סוג אולם

Public Area As String ' Death or Earth

מילת פתח לסוג משחק / אולם

Public ScreenW As Integer ' Weidth of Screen

Public ScreenH As Integer ' Height of Screen

Public BitPP As Byte ' Bit Per Pixel

משתנים מיאצגים רזולוציה של המשחק

ניתנים לבחירה רזולוציות הבאות:

640x480x16 800x600x16 1024x768x16 1280x1024x16

640x480x32 800x600x32 1024x768x32 1280x1024x32

Const COMPSTEP = 100' Number of Computer's STEPS
in one Direction Movement

מספר צאדים שמחשב אושה בכל תזוזה ממקום למקום

Const TREE_MAX = 40 ' Counter of Tree -1

כמות עצים מכל סוג שמפוזרים אל השטח

Const FIRESMOKE = 40 ' Counter of maximum Fire
and Smoke Animations -1

כמות אפקטים של עשן ואש המפוזרים אל השטח

Const PI = 3.14159265358979

מספר פי קבוע

Const Radians = PI / 180

מיועד המרה זווית מרדיאנים למאלות

Const POLE = 200' Size of Area from Center to
WALL

גודל שטח ממרכז לכיבון קירות

'-----Main DirectX Objects-----

Dim objDX As DirectX7

Dim objDD As DirectDraw7

Dim objD3D As Direct3D7

Dim objDI As DirectInput

Dim objDS As DirectSound

הגדרת אובייקטים של DirectX

'-----Direct Draw Objects-----

Dim objFront As DirectDrawSurface7 ' Front Buffer

Dim objBack As DirectDrawSurface7 ' Back Buffer

Dim objZBuffer As DirectDrawSurface7 ' Z-Buffer

הגדרת אובייקטים של DirectDraw

'----- Intro Pictures-----

Dim Intro1 As DirectDrawSurface7 ' Intro Picture

תמונה של אינטרו

Dim Intro5 As DirectDrawSurface7 ' LOGO Picture

תמונה של לוגו Doom Fourth Dimension

Dim demodsc As DDSURFACEDESC2 ' LOGO Demo
Description

מעפינים של תמונת לוגו

Dim intrdsc As DDSURFACEDESC2 ' Intro Picture
Description

מעפינים של תמונת אינטרו

Dim GameOver As DirectDrawSurface7 'GameOver
Picture

תמונה של סוף המשחק

Dim overdsc As DDSURFACEDESC2 'GameOver
Picture description

מעפינים של תמונת סוף המשחק

Dim PicShowed As Byte ' Number of picture to be
Showed

מספר תמונה להצגה באינטרו

Dim ckey As DDCOLORKEY ' Color Key

צבע שקוף בתמונה

Dim StartYES As Boolean ' Start New Game

להתחיל משחק חדש

'----- Direct Sound Objects

Dim DSBuffer As DirectSoundBuffer

Dim BufferDesc As DSBUFFERDESC

Dim WaveFileFormat As WAVEFORMATEX

הגדרת אובייקטים של DirectSound

'-----DirectX 3D Objects-----

Dim objDevice As Direct3DDevice7 ' ForUsing3DScreen

Dim objD3DEnum As Direct3DEnumDevices ' For Z-
Buffer Settings

Dim ViewPort As D3DVIEWPORT7 ' For View Port

הגדרת אובייקטים של Direct3D

Dim Texture(10) As DirectDrawSurface7 ' For Texture
Pictires

מרקמים של עולם

Dim Animate1 As DirectDrawSurface7 ' For Animation

Dim Animate2 As DirectDrawSurface7 ' For Animation

מרקמים של אפקט עשן / אש

Dim Craft As DirectDrawSurface7 ' For Craft
Interfaca Picture

Dim Craff As DirectDrawSurface7 ' For Craft Fired
Interface

תמונות של ה- Space Craft במצב רגיל ומצב ירי

Dim CraftDsc As DDSURFACEDESC2

מעפינים של תמונת Space Craft

Dim EnergyCMP As DirectDrawSurface7 'Energy
Picture of Computer

תמונת מונה חיים של מחשב

Dim EnergyPLY As DirectDrawSurface7 'Energy
Picture of Player

תמונת מונה חיים של שחקן

Dim CMP_Energy As Byte

'Energy of Computer
מונה מספרי של מחשב

Dim PLY_Energy As Byte

'Energy of Player
מונה מספרי של שחקן

Dim Player_Win As Integer
WINNINGs

'Counter of Player
מספר ניצחונות של שחקן

Dim Expl As DirectDrawSurface7 ' For Exploid
Pictures

תמונת פיצוץ בזמן פגיע

Dim Expldsc As DDSURFACEDESC2

פעיפינים של תמונת פיצוץ

Dim VertexWALL1(40) As D3DLVERTEX

Dim VertexWALL2(40) As D3DLVERTEX

Dim VertexWALL3(40) As D3DLVERTEX

Dim VertexWALL4(40) As D3DLVERTEX

קואורדינטות של קירות שטח העולם 3D

Dim VertexGRND(4) As D3DLVERTEX

רצפה בעולם 3D

Dim VertexNEBO(4) As D3DLVERTEX

שמיים בעולם 3D

Dim VertexNIZZ(4) As D3DLVERTEX

לאבה בעולם 3D

Dim VertexCOMP(24) As D3DLVERTEX ' Computer
Moving Object

דמות של מחשב בעולם 3D

Dim CompNewX As Integer

Dim CompNewZ As Integer

קואורדינטות מיקום חשד של מחשב

```
Dim CompOldX As Integer
Dim CompOldZ As Integer
```

קואורדינטות מיקום ישנות של מחשב

```
Dim cstep As Integer ' Current state of STEP <- counter
```

צאד נוככי של מחשב

```
Dim dx, dz, xm, zm As Long ' Current X,Z and STEP of
Computer
```

מיקום נוככי של מחשב ו שוני תזוזה

```
Dim dist2target As Long ' Distantion to TARGET
```

מרחק ממצלמה למטרה (מחשב)

```
Private Type SPRITE_3D ' For Tree, Explouse & ect...
```

```
Vert1(4) As D3DLVERTEX
```

```
Vert2(4) As D3DLVERTEX
```

```
Vert3(4) As D3DLVERTEX
```

```
Vert4(4) As D3DLVERTEX
```

```
Vist As Integer ' Visota Sprite
```

```
Shir As Integer ' SAhirina Sprite
```

```
x As Integer ' X of Sprite
```

```
z As Integer ' Z of Sprite
```

```
End Type
```

מבנה לבניה נוחה של אובייקט 3D כמו עץ, אפקט עשן / אש

Dim Tree1(TREE_MAX) As SPRITE_3D ' Make MAX
Trees objects

Dim Tree2(TREE_MAX) As SPRITE_3D ' Make MAX
Trees objects

אובייקטים עצים

Dim Smoke(FIRESMOKE) As SPRITE_3D ' For Smoke
Animation

Dim Faire(FIRESMOKE) As SPRITE_3D ' For Fair
Animation

אובייקטים עשן / אש

' For Fire & Smoke Animation

Dim CelX As Integer ' For TILE Animation

Dim CelY As Integer ' For TILE Animation

Const CelMaxX = 32

Const CelMaxY = 64

Const FrameX = 5

Const FrameY = 4

Dim Wait As Long ' For TILE Animation

משתנים לאנימציה של אפקט עשן / אש

'-----Direct Input-----

Dim Keyboard As DirectInputDevice

Dim KeyState As DIKEYBOARDSTATE

הגדרת אובייקטים של DirectInput

'-----Other-----

Dim ddsd1 As DDSURFACEDESC2 ' For Front/Back
Buffers

Dim ddsd2 As DDSURFACEDESC2 ' For Z-Buffer

Dim ddscaps As DDSCAPS2

משתנים של DirectX נוספים

Dim Quit As Boolean ' For Quit From Programm

ליציאה ממצב משחק נוכחי

Dim Continue As Boolean

המשך למצב הבאה

'-----For 3D Transformations-----

Dim RotateX As Long

Dim RotateY As Long

Dim RotateZ As Long

זווית סיבובי

Dim WorkMatrix As D3DMATRIX ' Working Matrix

Dim MatGrn As D3DMATRIX ' Changes for GROUND

Dim MatNiz As D3DMATRIX ' For Nizz Rotation

Dim MatCmp As D3DMATRIX ' For Computer Moving

איבוד גאומטריה

'-----For Player Movement-----
 Dim ViewMatrix As D3DMATRIX
 Dim VecCamLok As D3DVECTOR ' Vector Camera
 Loketed for MOVE
 Dim VecCamSee As D3DVECTOR ' Vector Camera
 Looking for MOVE
 Dim dist2focus As Integer ' Focus Lenght
 Dim step As Long ' Step
 Dim alfa As Double ' Ugol kuda smotril = 0 pri starte
 Dim delta As Double ' Ugol na skolko povara4ivaem

משתנים מיועדים לתזוזה של השחקן

Dim PlayTime As Long ' Playing Time

זמן משחק

Dim CosAlfa As Double 'Ugol mejdy PRICELOM i
 CELIYU

זווית בין מטרה לבין מקום שאליו יורים

Dim Ax, Ay, Az As Double ' For HIT Target!
 Dim Bx, By, Bz As Double ' For HIT Target!

וקטורים לחישוב פגיע

Dim VistrelNOW As Boolean

הגבלת ירי

פונקציות של התוכנה

Private Sub Init_Geometry()

פונקציה אושה בניה של עולם במערכת תלת מימדית כגון: שטח, שמיים, קירות, עצים, אשן, אש.

Private Sub Init_ComputerOBJ()

הגדרה ובניה אובייקט דמות מחשב

Private Sub Create_SPRITE3D(x, z, Vist, Shir, OBJ As SPRITE_3D)

פונקציה המקבלת מיקום במערכת תלת מימדית כאשר y קבוע, גובה ורוחב של אובייקט, ובונה במקום פוליגונים להצגת Texture בצורת הצלבה.

Function DDRect(x1, y1, x2, y2) As RECT

פונקציה מחזירה אובייקט שמכיל ערכים של מלבן

פונקציות של VB Form :

Private Sub **Form_KeyDown**(KeyCode, Shift As Integer)
פונקציה של VB המופעלת בזמן לחיצה על מקש

Private Sub **Form_KeyUp**(KeyCode , Shift As Integer)
פונקציה של VB המופעלת ברגע שמקש משתחרר, שימוש בפונקציה
לצורך הגבלת ירי של השחקן

Private Sub **Form_Unload**(Cancel As Integer)
פונקציה של VB המופעלת כאשר תוכנה מסיימת עבודתה, שימוש
לצורך שחרור זיכרון

Private Sub **Form_Load**()
פונקציה של VB המופעלת בהתחלת המשחק בכניסה למצב Full
Screen

Function LoadJPG(FileName, desc) As
DirectDrawSurface7

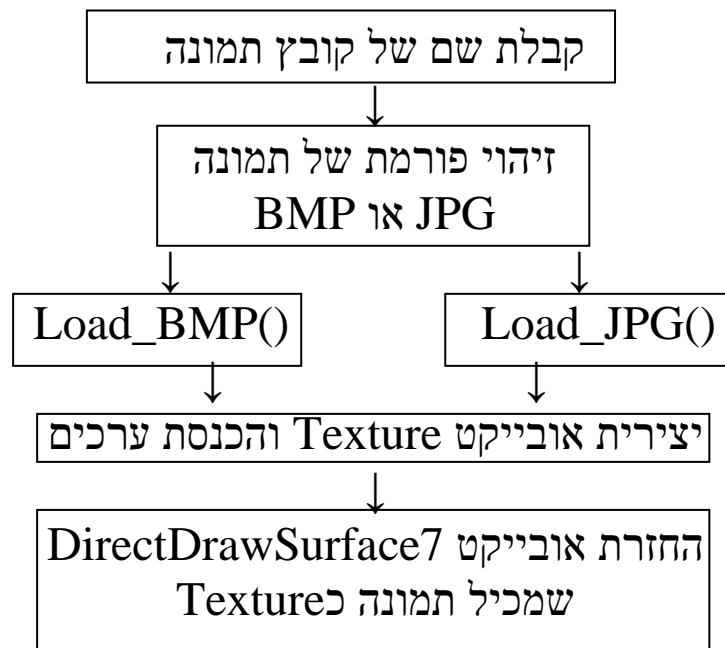
פונקציה מקבלת שם של קובץ תמונה ומחזירה אובייקט
DirectDrawSurface7

Function LoadBMP(FileName, desc) As
DirectDrawSurface7

פונקציה מקבלת שם של קובץ תמונה ומחזירה אובייקט
DirectDrawSurface7

Function MakeTexture(FileName) As
DirectDrawSurface7

פונקציה מקבלת שם של קובץ ומחזירה אובייקט מרקם Texture



Private Sub Init_RenderMode()' Setup Rendering States
פונקציה מגדירה את כל המצבים של ההתקן התלת-מחמדי

בעזרת פונקציה זאת אפשר לבטל Z-Buffer
אפשר להפעיל אפקט Alfa Blending – שקופות
ניתן להפעיל אפקט של ארפל
אפשרויות של הפונקציה רבות מאוד, אני השתמשתי בחלק גדול מהם.

משתני מצבי ההתקן הנפוצים ביותר:

D3DRENDERSTATE_FOGDENSITY
D3DRENDERSTATE_FOGVERTEXMODE
D3DRENDERSTATE_FOGCOLOR
D3DRENDERSTATE_FOGENABLE
D3DRENDERSTATE_LIGHTING
D3DRENDERSTATE_SHADEMODE
D3DRENDERSTATE_ZENABLE
D3DRENDERSTATE_COLORKEYENABLE
D3DRENDERSTATE_DESTBLEND
D3DRENDERSTATE_ALPHABLENDENABLE
D3DRENDERSTATE_FILLMODE

Private Sub Init_DirectX()
פונקציה אושה הגדרות של כל האובייקטים של DirectX

Private Sub Init_3DMatixes 'TransformationMatrices
הגדרת מטריצות עיקריות ב 3D גרפיקה

WorldMatrix – מטריצה זו שולטת במיקום ובאוריינטציה של
הפוליגונים.
אם נבצע שינוי במטריצה נוכל לסובב, להזיז, להגדיל או להקטין
אובייקטים תלת-מימדיים.

ProjectionMatrix – מטריצה זו מגדירה כיצד הסצנה התלת-מימדית תוקרן על המשטח האחורי (Back Buffer). במטריצה אנחנו קובעים את מישור החיתוך הקרוב והרחוק ואת זווית שדה הראייה (Field of view).

ViewMatrix – מטריצה זו קובעת מיקום המצלמה, גלגול המצלמה והמיקום שהמצלמה תסתכל אליו.

מטריצה זו לא מוגדרת בפונקציה זו כי אנחנו מגדירים את המטריצה ומשנים את ערכים שלה מהפונקציה **PrimaryLoop()**. מטריצה זו זה באצם עיניים של השחקן.

Private Sub Load_Textures()

פונקציה זו בודקת ערך של משתנה **GameType** ולפי ערכו מכניסה תמונות מתאימות לזכרון והופכת אותם ל**Textures**.

Private Sub MainGameLoop()

פונקציה זו קוראת לשתי פונקציות יקרות בתוכנה:

PrimaryLoop()
EndGameLoop()

פונקציות אלו מוסברות בהמשך...

Private Sub ShowDoom4DWorld()

פונקציה מעתיקה את כל משיש במשטח האחורה ל משטח קידמי ובמילים אחרות מציגה את כל אל המסך של משתמש.

Private Sub Clear_Device() ' Clear 3D Device

פונקציה שתפקידה לנקות את המשטח האחורי,
כלומר "clear screen"

Private Function DXVector(x , y , z) As D3DVECTOR

פונקציה המקבלת שלושה פרמארים ומחזירה אובייקט מסוג וקטור
בנוי משלושה פרמטרים שהתקבלו

Private Sub MoveMatrix(Matrix As D3DMATRIX,
Vector As D3DVECTOR)

פונקציה שמשתמשים בה כדי להזיז אובייקטים תלת מימדיים במשחק
למשל הזזה דמות של מחשב.
X, Y, Z – של Vector זה המיקום החדש של אובייקט שאותו רוצים
להזיז.

Private Sub ZoomMatrix(Matrix As D3DMATRIX,
Vector As D3DVECTOR)

פונקציה שמשתמשים בה כדי להגדיל או להקטין אובייקטים תלת
מימדיים במשחק

X, Y, Z – של Vector זה ערך הגדלה או הקטנה של אובייקט
שאותו רוצים להגדיל/להקטין.

Private Sub Show_Expl() ' Show 3D Expl Texture

פונקציה שמציגה את תמונת הפיצוף בזמן פגיע למטרה

Private Function GetRandom(FrNum,TNum)As Integer
פונקציה מקבלת שני ערכים מספרים שלמים ומחזירה מספר אקראי
שבין המספרים המתקבלים

Private Sub Draw_Nizz() ' Lava or Water Only
פונקציה שתפקידה לצייר לבה מתחת לשטח של המשחק

Private Sub Draw_Computer() ' Draw by Ground
פונקציה המציירת דמות של המשחק במקום נוכחי

Private Sub Draw_SPRITE3D(OBJ As SPRITE_3D)
פונקציה מציירת אובייקט תלת מימדי מסוג Sprite4D

Private Sub Draw_Ground() ' Draw 3D WORLD
פונקציה המציירת רצפה / שטח של המשחק

Private Sub Read_Keys()' Read DirectInput Keys
פונקציה המבעצת קליטה נתונים מהמקלדת דרך DirectInput

Private Sub Change_Area()
פונקציה שתפקידה להחליף סוג של המשחק
פונקציה בודקת מה סוג קייב ומחליפה אותו בסוג אחר

Private Sub Set_CameraEye() ‘ Set View Matrix

פונקציה שמגדירה מטריצה יקרית במשחק

ViewMatrix – מטריצה זו קובעת מיקום המצלמה, גלגול המצלמה והמיקום שהמצלמה תסתכל אליו.

מטריצה זו זה באצם עיניים של השחקן.
כל משאנחנו רואים על המסך זה כל משמצלמה רואה.

Private Sub MMControl_Done(NotifyCode As Integer)

פונקציה המופעלת בסיום מנגינה של המוסיקה ועוצרת אובייקט

Microsoft Multimedia Control 6.0

Private Sub PlayMusic(MusicFile As String)

פונקציה המקבלת שם של קובץ מוסיקה ומתחילה לנגן אותו בעזרת

Microsoft Multimedia Control 6.0

Private Sub PlaySound(FileName As String)

פונקציה המקבלת שם של קובץ Wav ומתחילה לנגן אותו בעזרת

אובייקט DirectSound

Private Sub TimerIntro_Timer()

פונקציה של תימר שמופעלת אוטומטית כל 3 שניות בהתחלת האינרו

שימושה להציג תמונה אינטרו בחלקים נפרדים כל 3 שניות להציג

חלק נוסף של התמונה

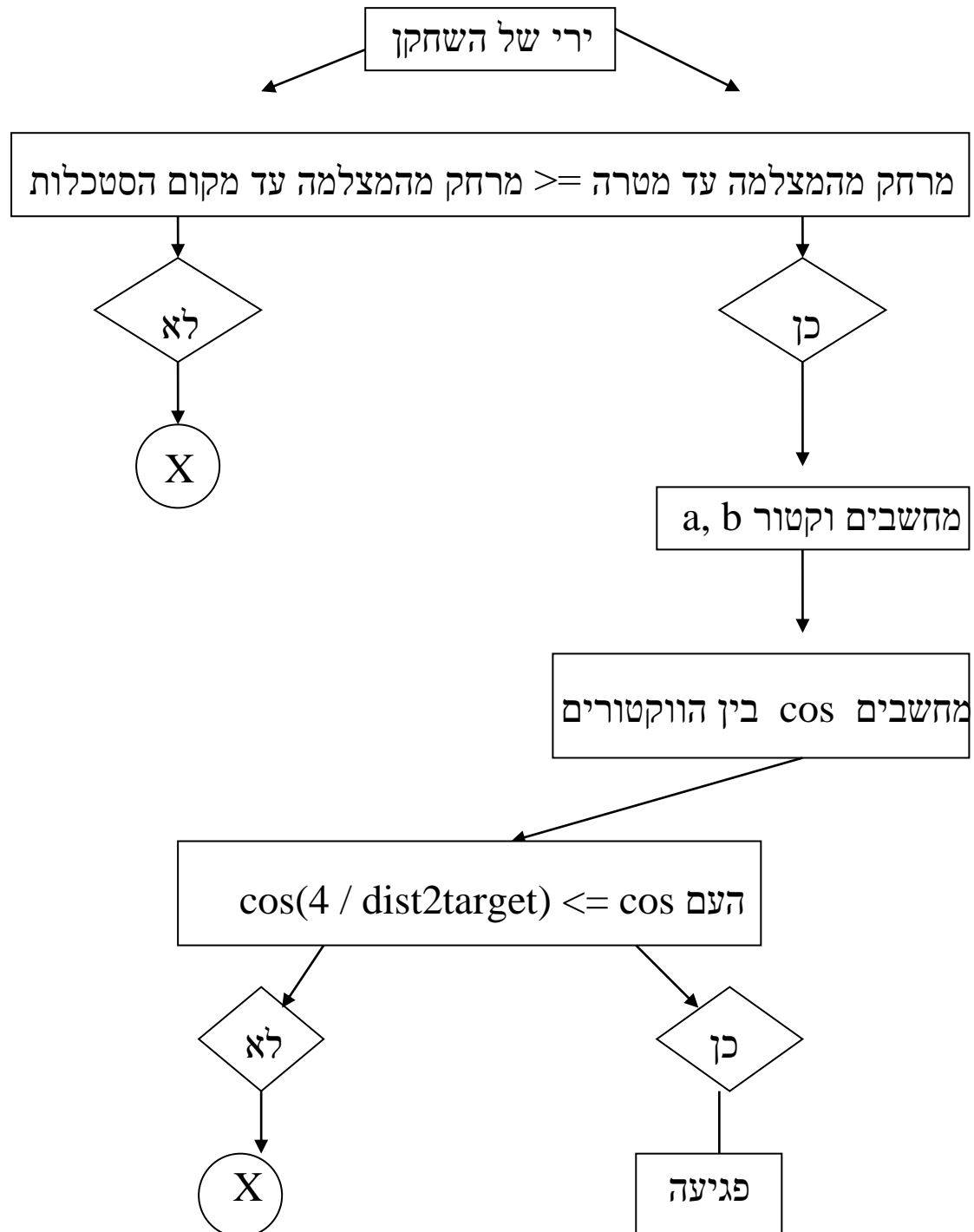
Private Sub TimerPlayTime_Timer()

פונקציה שמופעלת אוטומטית כל שניה אחרי תחילת משחק עצמו

מימושה ספירה זמן משחק והפעלת ארפל.

Private Sub CheckHitTarget()

פונקציה המופעלת בכל ירי של השחקן לצורך בדיקת פגיעה למטרה



Private Sub Move_Computer()

פונקציה מחשבת תזוזה של המשחב

פונקציה בוחרת מקום חדש בשטח מחשבת מרחק ומחלקת אותו לכמות צעדים, ובכל קריאה של פונקציה מזיזה את הדמות של מחשב לצעד אחת.

Private Sub CheckDamage() ' Check Collision

פונקציה מחשבת העם יש פגיעה עם מחשב כלומר נעם מחשב פוגעה בשחקן

מחשבים מרחק בין מצלמה למטרה
עם מרחק פחות או שווה ל-4 אז יש פגיעה, אחרת אין
4 זה מרחק אופטימלי

Public Sub Make_Move()

פונקציה הזזה של השחקן במשחק
פונקציה בודקת איזה מקש משתמש לוחץ לפי זה
מבצעת הזזה של השחקן בשטח של המשחק.

Private Sub DemonstLoop()

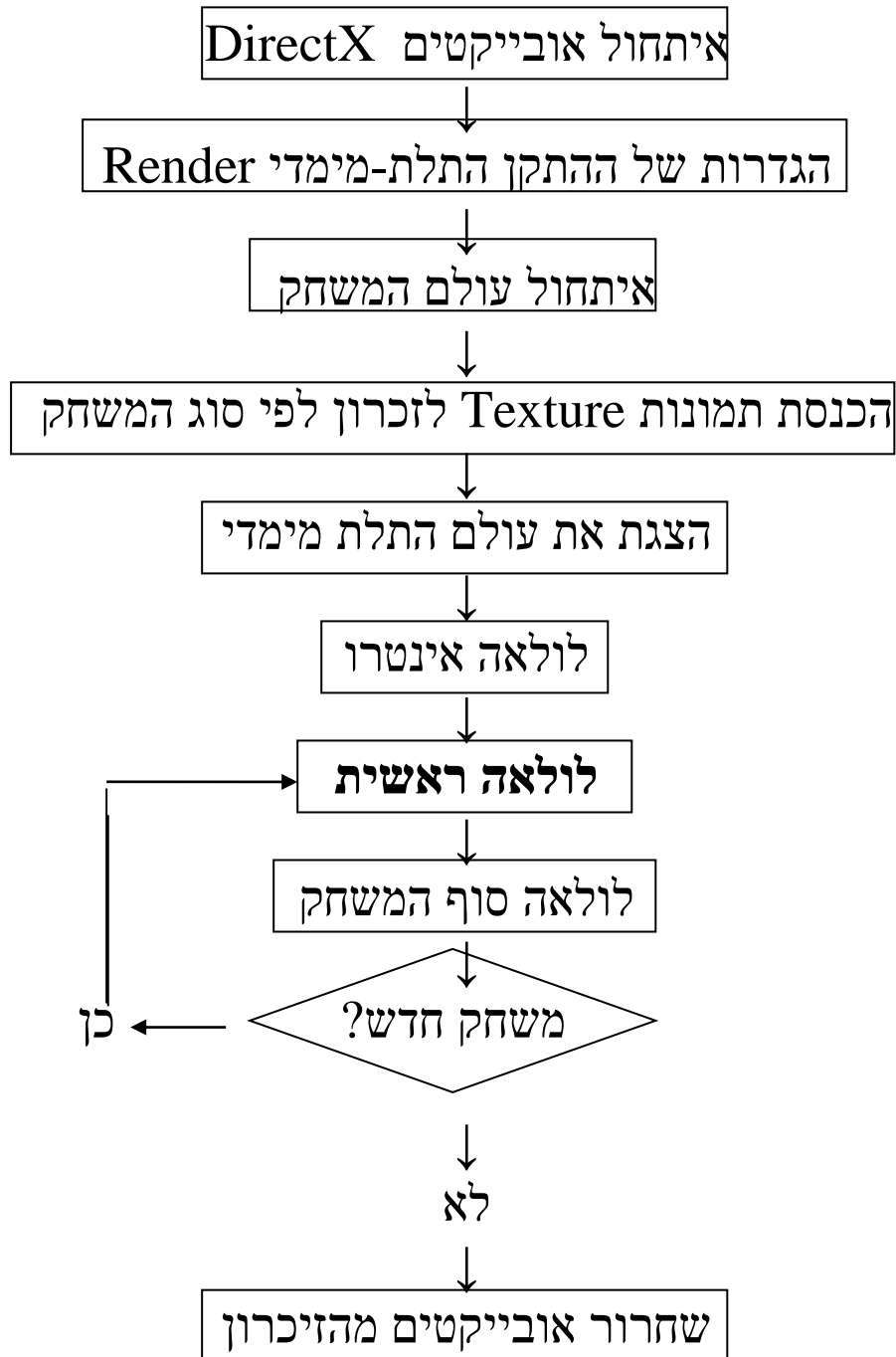
פונקציה המציגה את האינטרו של המשחק,
מתיגה את תמונה ראשונה ואת הלוגו מקטתן.

Private Sub SelectsLoop()

פונקציה אוצרת כל המוזיקה ומכינה כניסה ללולאה ראשית

Private Sub PrimaryLoop()

לולאה ראשית במשחק
סדר פעולות הוא כגון:



Private Sub EndGameLoop()

פונקציה מציגה את תמונת סוף המשחק ונתונים של המשחק.

גירסה מסחרית של DOOM 4D

בגירסה מסחרית צפוי לראות יותר סצנות / רמות

דמות של המחשב יהיה בנוי כמודל 3D Studio

דמות של מחשב תוכל גם לבצע ירי לכיוון שחקן

באינטרו של המשחק צפוי להציג וידאו (Video Intro)

בגירסה מסחרית יהיה אפשרות לשחק נגן עוד שחקן דרך

רשת אינטרנט או אנטראנט בעזרת פרוטוקול TCP/IP

קוד מלא של התוכנה

Form Main

```
'-----  
'DirectX 3D Graphics by Denis Astahov  
'DirectX 7 Graphics Library Project of First B.Ed. Degree!!!  
'DOOM 4D Version 1.00 Alfa 15.12.2004 (C)2004  
'-----
```

```
Const COMPSTEP = 100 ' Computer's STEPS Direction Movement  
Const TREE_MAX = 40 ' Counter of TREes -1 My FOREST :)  
Const FIRESMOKE = 40 ' Counter of maximum Fire and Smoke
```

```
Const PI = 3.14159265358979  
Const Radians = PI / 180  
Const POLE = 200 ' Size of Area from Center to WALLs
```

```
'-----Main DirectX Objects-----
```

```
Dim objDX As DirectX7  
Dim objDD As DirectDraw7  
Dim objD3D As Direct3D7  
Dim objDI As DirectInput  
Dim objDS As DirectSound
```

```
'-----Direct Draw Objects-----
```

```
Dim objFront As DirectDrawSurface7 ' Front Buffer  
Dim objBack As DirectDrawSurface7 ' Back Buffer  
Dim objZBuffer As DirectDrawSurface7 ' Z-Buffer
```

```
'----- Intro Pictures-----
```

```
Dim Intro1 As DirectDrawSurface7 ' Intro Picture  
Dim Intro5 As DirectDrawSurface7 ' LOGO Picture  
Dim demodsc As DDSURFACEDESC2 ' LOGO Demo Description  
Dim intrdsc As DDSURFACEDESC2 ' Intro Picture Description
```

```
Dim GameOver As DirectDrawSurface7 'GameOver Picture  
Dim overdsc As DDSURFACEDESC2'GameOver Picture dsc
```



```

Dim PicShowed As Byte ' Number of picture to be Showed
Dim ckey As DDColorKey ' Color Key
Dim StartYES As Boolean ' Start New Game

'----- Direct Sound Objects
Dim DSBuffer As DirectSoundBuffer
Dim BufferDesc As DSBUFFERDESC
Dim WaveFileFormat As WAVEFORMATEX

'-----DirectX 3D Objects-----
Dim objDevice As Direct3DDevice7 ' For Using 3D Screen
Dim objD3DEnum As Direct3DEnumDevices ' For Z-Buffer
Settings
Dim ViewPort As D3DVIEWPORT7 ' For View Port
Dim Texture(10) As DirectDrawSurface7 ' For Texture Pictures
Dim Animate1 As DirectDrawSurface7 ' For Animation
Dim Animate2 As DirectDrawSurface7 ' For Animation

Dim Craft As DirectDrawSurface7 ' For Craft Interfaca Picture
Dim Craff As DirectDrawSurface7 ' For Craft Fired Interface
Dim CraftDsc As DDSURFACEDESC2
Dim EnergyCMP As DirectDrawSurface7 'Energy Picture of
Computer
Dim EnergyPLY As DirectDrawSurface7 'Energy Picture of Player
Dim CMP_Energy As Byte 'Energy of Computer
Dim PLY_Energy As Byte 'Energy of Player
Dim Player_Win As Integer 'Counter of Player WINNINGs

Dim Expl As DirectDrawSurface7 ' For Exploid Pictures
Dim Expldsc As DDSURFACEDESC2

Dim VertexWALL1(40) As D3DLVERTEX
Dim VertexWALL2(40) As D3DLVERTEX
Dim VertexWALL3(40) As D3DLVERTEX
Dim VertexWALL4(40) As D3DLVERTEX
Dim VertexGRND(4) As D3DLVERTEX
Dim VertexNEBO(4) As D3DLVERTEX
Dim VertexNIZZ(4) As D3DLVERTEX

```

```

Dim VertexCOMP(24) As D3DLVERTEX ' Computer Moving
Object
Dim CompNewX As Integer
Dim CompNewZ As Integer
Dim CompOldX As Integer
Dim CompOldZ As Integer
Dim cstep As Integer ' Current state of STEP <- counter
Dim dx, dz, xm, zm As Long ' Current X,Z and STEP of Computer
Dim dist2target As Long ' Distantion to TARGET

```

```

Private Type SPRITE_3D ' For Tree, Explouse & ect...

```

```

    Vert1(4) As D3DLVERTEX
    Vert2(4) As D3DLVERTEX
    Vert3(4) As D3DLVERTEX
    Vert4(4) As D3DLVERTEX
    Vist As Integer ' Visota Sprite
    Shir As Integer ' SAhirina Sprite
    x As Integer ' X of Sprite
    z As Integer ' Z of Sprite

```

```

End Type

```

```

Dim Tree1(TREE_MAX) As SPRITE_3D ' Make MAX Trees
objects

```

```

Dim Tree2(TREE_MAX) As SPRITE_3D ' Make MAX Trees
objects

```

```

Dim Smoke(FIRESMOKE) As SPRITE_3D ' For Smoke Animation

```

```

Dim Faire(FIRESMOKE) As SPRITE_3D ' For Fair Animation

```

```

' For Fire & Smoke Animation

```

```

Dim CelX As Integer ' For TILE Animation

```

```

Dim CelY As Integer ' For TILE Animation

```

```

Const CelMaxX = 32

```

```

Const CelMaxY = 64

```

```

Const FrameX = 5

```

```

Const FrameY = 4

```

```

Dim Wait As Long ' For TILE Animation

```

```

'-----Direct Input-----

```

```

Dim Keyboard As DirectInputDevice

```

Dim KeyState As DIKEYBOARDSTATE

'-----Other-----

Dim ddsd1 As DDSURFACEDESC2 ' For Front/Back Buffers

Dim ddsd2 As DDSURFACEDESC2 ' For Z-Buffer

Dim ddscaps As DDSCAPS2

Dim Quit As Boolean ' For Quit From Programm

Dim Continue As Boolean

'----For 3D Transformations-----

Dim RotateX As Long

Dim RotateY As Long

Dim RotateZ As Long

Dim WorkMatrix As D3DMATRIX ' Working Matrix

Dim MatGrn As D3DMATRIX ' All Changes for GROUND

Dim MatNiz As D3DMATRIX ' For Nizz Rotation

Dim MatCmp As D3DMATRIX ' For Computer Moving

'-----Dlya Dviveniya Personaja-----

Dim ViewMatrix As D3DMATRIX

Dim VecCamLok As D3DVECTOR ' Vector Camera Loketed for MOVE

Dim VecCamSee As D3DVECTOR ' Vector Camera Looking for MOVE

Dim dist2focus As Integer ' Focus Lenght

Dim step As Long ' Step

Dim alfa As Double ' Ugol kuda smotril = 0 pri starte

Dim delta As Double ' Ugol na skolko povara4ivaem

Dim PlayTime As Long ' Playing Time

Dim CosAlfa As Double 'Ugol mejdy PRICELOM i CELIYU

Dim Ax, Ay, Az As Double ' For HIT Target!

Dim Bx, By, Bz As Double ' For HIT Target!

Dim VistrelNOW As Boolean

Private Sub Init_Geometry()
'Init 3D Object in Space

Dim size As Integer
Dim i As Integer
size = 20 'Size of ONE sprite of WALL

'Make Ground
Call objDX.CreateD3DLVertex(-100, 0, 100,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexGRND(0))
Call objDX.CreateD3DLVertex(100, 0, 100,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexGRND(1))
Call objDX.CreateD3DLVertex(-100, 0, -100,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexGRND(2))
Call objDX.CreateD3DLVertex(100, 0, -100,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexGRND(3))

'Make TREE & FOREST-----

For i = 1 To TREE_MAX ' Rastavit my FOREST by RANDOM!!!

x = GetRandom(-95, 95)
z = GetRandom(-95, 95)
Call Create_SPRITE3D(x, z, 20, 3, Tree1(i))

x = GetRandom(-95, 95)
z = GetRandom(-95, 95)
Call Create_SPRITE3D(x, z, 12, 5, Tree2(i))
"---Create_SPRITE3D(X, Z, Vist, Shir, OBJ_Sprite3D)

Next i

'-Make Smoke-----
'----Create_SPRITE3D(X, Z, Vist, Shir, OBJ_Sprite3D)
For i = 1 To FIRESMOKE
x = GetRandom(-98, 98)
z = GetRandom(-98, 98)
Call Create_SPRITE3D(x, z, 3, 1, Smoke(i))

x = GetRandom(-98, 98)

```

z = GetRandom(-98, 98)
Call Create_SPRITE3D(x, z, 3, 1, Faire(i))
Next i

```

```

'-----

```

```

'Make NEBO

```

```

Call objDX.CreateD3DLVertex(250, 50, 250,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexNEBO(0))
Call objDX.CreateD3DLVertex(-250, 50, 250,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexNEBO(1))
Call objDX.CreateD3DLVertex(250, 50, -250,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexNEBO(2))
Call objDX.CreateD3DLVertex(-250, 50, -250,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexNEBO(3))

```

```

'Make NIZZ

```

```

Call objDX.CreateD3DLVertex(-500, -10, 500,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexNIZZ(0))
Call objDX.CreateD3DLVertex(500, -10, 500,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexNIZZ(1))
Call objDX.CreateD3DLVertex(-500, -10, -500,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexNIZZ(2))
Call objDX.CreateD3DLVertex(500, -10, -500,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexNIZZ(3))

```

```

'Make WALLs

```

```

For i = 0 To 9

```

```

'Back WALL1 - Ptyamo

```

```

Call objDX.CreateD3DLVertex(-100 + i * size, 30, 100,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexWALL1(0 + i * 4))
Call objDX.CreateD3DLVertex(-80 + i * size, 30, 100,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexWALL1(1 + i * 4))
Call objDX.CreateD3DLVertex(-100 + i * size, 0, 100,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexWALL1(2 + i * 4))
Call objDX.CreateD3DLVertex(-80 + i * size, 0, 100,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexWALL1(3 + i * 4))

```

'Front WALL2 - Zzadi

```
Call objDX.CreateD3DLVertex(-80 + i * size, 30, -100,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexWALL2(0 + i * 4))  
Call objDX.CreateD3DLVertex(-100 + i * size, 30, -100,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexWALL2(1 + i * 4))  
Call objDX.CreateD3DLVertex(-80 + i * size, 0, -100,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexWALL2(2 + i * 4))  
Call objDX.CreateD3DLVertex(-100 + i * size, 0, -100,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexWALL2(3 + i * 4))
```

'Left WALL3 - Sprava

```
Call objDX.CreateD3DLVertex(100, 30, -80 + i * size,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexWALL3(0 + i * 4))  
Call objDX.CreateD3DLVertex(100, 30, -100 + i * size,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexWALL3(1 + i * 4))  
Call objDX.CreateD3DLVertex(100, 0, -80 + i * size,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexWALL3(2 + i * 4))  
Call objDX.CreateD3DLVertex(100, 0, -100 + i * size,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexWALL3(3 + i * 4))
```

'Right WALL4 - Sleva

```
Call objDX.CreateD3DLVertex(-100, 30, -100 + i * size,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexWALL4(0 + i * 4))  
Call objDX.CreateD3DLVertex(-100, 30, -80 + i * size,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexWALL4(1 + i * 4))  
Call objDX.CreateD3DLVertex(-100, 0, -100 + i * size,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexWALL4(2 + i * 4))  
Call objDX.CreateD3DLVertex(-100, 0, -80 + i * size,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexWALL4(3 + i * 4))
```

Next i

Call Init_ComputerOBJ 'Computerniy KUB

End Sub

Private Sub Init_ComputerOBJ()

'Init 3D Computer Object in Space

'Front side

```
Call objDX.CreateD3DLVertex(-10, -10, -10,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexCOMP(0))  
Call objDX.CreateD3DLVertex(-10, 10, -10,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexCOMP(1))  
Call objDX.CreateD3DLVertex(10, -10, -10,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexCOMP(2))  
Call objDX.CreateD3DLVertex(10, 10, -10,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexCOMP(3))
```

'Right side

```
Call objDX.CreateD3DLVertex(10, -10, -10,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexCOMP(4))  
Call objDX.CreateD3DLVertex(10, 10, -10,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexCOMP(5))  
Call objDX.CreateD3DLVertex(10, -10, 10,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexCOMP(6))  
Call objDX.CreateD3DLVertex(10, 10, 10,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexCOMP(7))
```

'Back side

```
Call objDX.CreateD3DLVertex(10, -10, 10,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexCOMP(8))  
Call objDX.CreateD3DLVertex(10, 10, 10,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexCOMP(9))  
Call objDX.CreateD3DLVertex(-10, -10, 10,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexCOMP(10))  
Call objDX.CreateD3DLVertex(-10, 10, 10,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexCOMP(11))
```

'Left side

```
Call objDX.CreateD3DLVertex(-10, -10, 10,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexCOMP(12))  
Call objDX.CreateD3DLVertex(-10, 10, 10,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexCOMP(13))  
Call objDX.CreateD3DLVertex(-10, -10, -10,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexCOMP(14))  
Call objDX.CreateD3DLVertex(-10, 10, -10,  
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexCOMP(15))
```

'Top side

```
Call objDX.CreateD3DLVertex(-10, 10, -10,  
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexCOMP(16))
```

```

Call objDX.CreateD3DLVertex(-10, 10, 10,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexCOMP(17))
Call objDX.CreateD3DLVertex(10, 10, -10,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexCOMP(18))
Call objDX.CreateD3DLVertex(10, 10, 10,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexCOMP(19))
'Bottom side
Call objDX.CreateD3DLVertex(-10, -10, 10,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, VertexCOMP(20))
Call objDX.CreateD3DLVertex(-10, -10, -10,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, VertexCOMP(21))
Call objDX.CreateD3DLVertex(10, -10, 10,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, VertexCOMP(22))
Call objDX.CreateD3DLVertex(10, -10, -10,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, VertexCOMP(23))

```

End Sub

```

Private Sub Create_SPRITE3D(ByVal x As Integer, ByVal z As
Integer, ByVal Vist As Integer, ByVal Shir As Integer, OBJ As
SPRITE_3D)

```

```

OBJ.Vist = Vist
OBJ.Shir = Shir
OBJ.x = x
OBJ.z = z

```

```

Call objDX.CreateD3DLVertex(x - Shir, Vist, z,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, OBJ.Vert1(0))
Call objDX.CreateD3DLVertex(x + Shir, Vist, z,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, OBJ.Vert1(1))
Call objDX.CreateD3DLVertex(x - Shir, 0, z,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, OBJ.Vert1(2))
Call objDX.CreateD3DLVertex(x + Shir, 0, z,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, OBJ.Vert1(3))

```

```

Call objDX.CreateD3DLVertex(x + Shir, Vist, z,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, OBJ.Vert2(0))

```



```

Call objDX.CreateD3DLVertex(x - Shir, Vist, z,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, OBJ.Vert2(1))
Call objDX.CreateD3DLVertex(x + Shir, 0, z,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, OBJ.Vert2(2))
Call objDX.CreateD3DLVertex(x - Shir, 0, z,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, OBJ.Vert2(3))

```

```

Call objDX.CreateD3DLVertex(x, Vist, z - Shir,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, OBJ.Vert3(0))
Call objDX.CreateD3DLVertex(x, Vist, z + Shir,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, OBJ.Vert3(1))
Call objDX.CreateD3DLVertex(x, 0, z - Shir,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, OBJ.Vert3(2))
Call objDX.CreateD3DLVertex(x, 0, z + Shir,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, OBJ.Vert3(3))

```

```

Call objDX.CreateD3DLVertex(x, Vist, z + Shir,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 0, OBJ.Vert4(0))
Call objDX.CreateD3DLVertex(x, Vist, z - Shir,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 0, OBJ.Vert4(1))
Call objDX.CreateD3DLVertex(x, 0, z + Shir,
objDX.CreateColorRGB(1, 1, 1), 1, 0, 1, OBJ.Vert4(2))
Call objDX.CreateD3DLVertex(x, 0, z - Shir,
objDX.CreateColorRGB(1, 1, 1), 1, 1, 1, OBJ.Vert4(3))

```

End Sub

```

Private Function DDRect(ByVal x1 As Integer, ByVal y1 As
Integer, ByVal x2 As Integer, ByVal y2 As Integer) As RECT
    DDRect.Left = x1
    DDRect.Top = y1
    DDRect.Right = x2
    DDRect.Bottom = y2
End Function

```

```

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)

```

Continue = True 'if Any key pressed

```

Select Case KeyCode

```

```

Case vbKeyEscape: Quit = True And Continue = True
' Case vbKeyReturn: Continue = True
' Case vbKeySpace: Continue = True
Case vbKeyN: StartYES = True

Case vbKeyF1: PlayMusic (App.Path & "\\Music\\Music1.wav")
Case vbKeyF2: PlayMusic (App.Path & "\\Music\\Music2.wav")
Case vbKeyF3: PlayMusic (App.Path & "\\Music\\Music3.wav")
Case vbKeyF4: PlayMusic (App.Path & "\\Music\\Music4.wav")
Case vbKeyF5: PlayMusic (App.Path & "\\Music\\Music5.wav")
Case vbKeyF6: PlayMusic (App.Path & "\\Music\\Music6.wav")
Case vbKeyF7: PlayMusic (App.Path & "\\Music\\Music7.wav")
Case vbKeyF8: PlayMusic (App.Path & "\\Music\\Music8.wav")
Case vbKeyF9: PlayMusic (App.Path & "\\Music\\Music9.wav")
Case vbKeyF10: PlayMusic (App.Path & "\\Music\\Music10.wav")
Case vbKeyF11: MMControl.Command = "Close"

End Select
If (KeyCode >= vbKeyF1 And KeyCode <= vbKeyF10) Then 'Start
NEW Music
    PlaySound (App.Path & "\\Sound\\Switch.wav")
End If

End Sub

Private Function LoadJPG(ByVal FileName As String, desc As
DDSURFACEDESC2) As DirectDrawSurface7

Dim pictemp As Picture ' Temporary picture

Set pictemp = LoadPicture(FileName)          ' Load as *.JPG file
SavePicture pictemp, App.Path & "\\pictemp.bmp" ' Save as *.BMP
file
Set LoadJPG = objDD.CreateSurfaceFromFile(App.Path &
"\\pictemp.bmp", desc)
Kill App.Path & "\\pictemp.bmp"                ' Delete TEMP file

Set pictemp = Nothing

```

End Function

Private Function LoadBMP(ByVal FileName As String, desc As
DDSURFACEDESC2) As DirectDrawSurface7

Set LoadBMP = objDD.CreateSurfaceFromFile(FileName, desc)

End Function

Private Function MakeTexture(ByVal FileName As String, Optional
ByVal Width As Integer, Optional ByVal Height As Integer) As
DirectDrawSurface7

Dim picdsc As DDSURFACEDESC2

'Dim TextureEnum As Direct3DEnumPixelFormat

'Set TextureEnum = objDevice.GetTextureFormatsEnum()

'TextureEnum.GetItem 1, picdsc.ddpfPixelFormat

'-----Texture settings-----

If (FileName <> "") Then ' if Texture in FILE

picdsc.ddscaps.lCaps = DDSCAPS_TEXTURE

If (Right(LCase(FileName), 4) = ".jpg") Then ' If JPG file

Set MakeTexture = LoadJPG(FileName, picdsc) ' Load *.JPG file

End If

If (Right(LCase(FileName), 4) = ".bmp") Then ' If BMP file

Set MakeTexture = LoadBMP(FileName, picdsc) ' Load *.BMP

file

End If

Else

picdsc.lFlags = DDSD_CAPS Or DDSD_TEXTURESTAGE Or
DDSD_WIDTH Or DDSD_HEIGHT

picdsc.lFlags = DDSD_WIDTH Or DDSD_HEIGHT

picdsc.lWidth = Width

picdsc.lHeight = Height

picdsc.ddpfPixelFormat.lFlags = DDPF_RGB Or
DDPF_ALHAPIXELS

```

picdsc.ddscaps.lCaps = DDSCAPS_TEXTURE
picdsc.ddscaps.lCaps2 = DDSCAPS2_D3DTEXTUREMANAGE
picdsc.lTextureStage = 0
Set MakeTexture = objDD.CreateSurface(picdsc) ' Create EMPTY
texture
End If

```

```

'---Set Invisible Color-----
Dim ckey As DDCOLORKEY
ckey.high = 0 ' Black
ckey.low = 0 ' Black
MakeTexture.SetColorKey DDCKEY_SRCBLT, ckey

End Function

```

```

Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
If (KeyCode = vbKeySpace) Then
    VistrelNOW = False
End If
End Sub

```

```

Private Sub Init_RenderMode()
'----- Setup Rendering States/Modes -----

'Set FOG
objDevice.SetRenderStateSingle
D3DRENDERSTATE_FOGDENSITY, 0.004
objDevice.SetRenderState
D3DRENDERSTATE_FOGVERTEXMODE, D3DFOG_EXP
'Type FOG By Default
'-----
objDevice.SetRenderState D3DRENDERSTATE_LIGHTING,
False
objDevice.SetRenderState D3DRENDERSTATE_SHADEMODE,
D3DSHADE_GOURAUD
objDevice.SetRenderState D3DRENDERSTATE_ZENABLE,
D3DZB_TRUE
objDevice.SetRenderState
D3DRENDERSTATE_COLORKEYENABLE, True

```

```
'objDevice.SetRenderState D3DRENDERSTATE_DESTBLEND, 2
'objDevice.SetRenderState
D3DRENDERSTATE_ALPHABLENDENABLE, True
```

```
objDevice.SetRenderState D3DRENDERSTATE_FILLMODE,
D3DFILL_SOLID 'Warning!!!!!!
```

```
'-- Let use Bilinear filtering for more RULEZzzzz!
'- MAGFILTER if texture is Stretched
objDevice.SetTextureStageState 0, D3DTSS_MAGFILTER,
D3DTFG_LINEAR
'- MINFILTER if texture is made smaller
objDevice.SetTextureStageState 0, D3DTSS_MINFILTER,
D3DTFN_LINEAR
```

```
End Sub
```

```
Private Sub Init_DirectX()
Set objDX = New DirectX7          'Create DirectX Object
Set objDD = objDX.DirectDrawCreate("") 'Create DirectDraw
Object
```

```
'-----Set Full Screen Mode-----
Call objDD.SetCooperativeLevel(Me.hWnd,
DDSCL_ALLOWREBOOT Or _
                        DDSCL_EXCLUSIVE Or _
                        DDSCL_FULLSCREEN)
Call objDD.SetDisplayMode(ScreenW, ScreenH, BitPP, 0,
DDSDM_DEFAULT)
```

```
'----Create Front/Back Buffers-----
ddsd1.lFlags = DDSD_CAPS Or DDSD_BACKBUFFERCOUNT
ddsd1.lBackBufferCount = 1
```

```
'--- Front Buffer Settings---
ddsd1.ddscaps.lCaps = DDSCAPS_COMPLEX Or
DDSCAPS_FLIP Or DDSCAPS_3DDEVICE Or
DDSCAPS_PRIMARYSURFACE
```

```
'--- Back Buffer Settings ---
```

```
ddscaps.lCaps = DDSCAPS_BACKBUFFER Or  
DDSCAPS_3DDEVICE
```

```
Set objFront = objDD.CreateSurface(ddsd1)      ' Create Front  
Buffer
```

```
Set objBack = objFront.GetAttachedSurface(ddscaps) ' Create Back  
Buffer
```

```
'--- Init 3D Device -----
```

```
Set objD3D = objDD.GetDirect3D()
```

```
Set objD3DEnum = objD3D.GetDevicesEnum
```

```
Dim GUID As String
```

```
GUID = objD3DEnum.GetGuid(objD3DEnum.GetCount)
```

```
'---- Create Z-Buffer -----
```

```
Dim ZEnum As Direct3DEnumPixelFormat
```

```
Dim PixelFormat As DDPIXELFORMAT
```

```
Set ZEnum = objD3D.GetEnumZBufferFormats(GUID)
```

```
Dim i As Long
```

```
'Loop until we find Z-Buffer with at least Depth of 16 Bits
```

```
For i = 1 To ZEnum.GetCount
```

```
    Call ZEnum.GetItem(i, PixelFormat)
```

```
    If (PixelFormat.lFlags = DDPF_ZBUFFER And  
PixelFormat.lZBufferBitDepth >= 16) Then
```

```
        Exit For
```

```
    End If
```

```
Next i
```

```
'--- Prepare and Cerate Z-Buffer Surface -----
```

```
ddsd2.lFlags = DDSD_CAPS Or DDSD_WIDTH Or  
DDSD_HEIGHT Or DDSD_PIXELFORMAT
```

```
ddsd2.ddscaps.lCaps = DDSCAPS_ZBUFFER
```

```
ddsd2.lWidth = ScreenW
```

```
ddsd2.lHeight = ScreenH
```

```
ddsd2.ddpfPixelFormat = PixelFormat
```

```

'Hardware Accelerate Device : Z-Buffer in Video Memory
'Software Accelerate Device : Z-Buffer in System Memory
If GUID = "IID_IDirect3DRGBDevice" Then
    ddsd2.ddscaps.lCaps = ddsd2.ddscaps.lCaps Or
    DDSCAPS_SYSTEMMEMORY
Else
    ddsd2.ddscaps.lCaps = ddsd2.ddscaps.lCaps Or
    DDSCAPS_VIDEOMEMORY
End If

```

```

Set objZBuffer = objDD.CreateSurface(ddsd2) ' Create Z-Buffer
Surface
objBack.AddAttachedSurface objZBuffer      ' Attach Z-Buffer to
Back Buffer

```

```

Set objDevice = objD3D.CreateDevice(GUID, objBack) ' Set the 3D
Device

```

```

'-----Define Direct Sound -----

```

```

Set objDS = objDX.DirectSoundCreate("")
objDS.SetCooperativeLevel FormMain.hWnd, DSSCL_PRIORITY

```

```

'----- Define View Port -----

```

```

ViewPort.lX = 0
ViewPort.lY = 0
ViewPort.lWidth = ScreenW
ViewPort.lHeight = ScreenH
ViewPort.minz = 0#
ViewPort.maxz = 1#

```

```

'-----Init Keyboard-----

```

```

Set objDI = objDX.DirectInputCreate()
Set Keyboard = objDI.CreateDevice("GUID_SysKeyboard")
Keyboard.SetCommonDataFormat DIFORMAT_KEYBOARD
Keyboard.SetCooperativeLevel Me.hWnd,
DISCL_NONEXCLUSIVE Or DISCL_BACKGROUND
Keyboard.Acquire

```

Randomize

Quit = False

Continue = False

End Sub

Private Sub Init_3DMatixes()

'----- Define Transformation Matrices -----

'-----

' ~~ WORLD MATRIX ~~

Dim WorldMatrix As D3DMATRIX

objDX.IdentityMatrix WorldMatrix

objDevice.SetTransform D3DTRANSFORMSTATE_WORLD,
WorldMatrix

'-----

' ~~ PROJECTION MATRIX ~~

Dim ProjectionMatrix As D3DMATRIX

Call objDX.ProjectionMatrix(ProjectionMatrix, 1, 1000, PI / 2)

objDevice.SetTransform

D3DTRANSFORMSTATE_PROJECTION, ProjectionMatrix

'-----

' ~~ VIEW MATRIX ~~

' Moved to PRIMARY LOOP :-)

Dim Viewmatrix As D3DMATRIX

'Call objDX.ViewMatrix(ViewMatrix, DXVector(0, -100, -100),
DXVector(0, 0, 0), DXVector(0, 1, 0), 0)

'objDevice.SetTransform D3DTRANSFORMSTATE_VIEW,
ViewMatrix

End Sub

Private Sub Load_Textures()

'Load Intro Pictures

If (GameType = 1) Then

Set Intro1 = MakeTexture(App.Path & "\\Image\\Earth\\Intro1.jpg")

'Main screen

End If

If (GameType = 2) Then

Set Intro1 = MakeTexture(App.Path & "\\Image\\Death\\Intro1.jpg")

'Main screen

End If

Set Intro5 = MakeTexture(App.Path & "\\Image\\IntroD4D.bmp")

Intro1.GetSurfaceDesc intrdsc ' Read Description of file

Intro5.GetSurfaceDesc demodsc ' Read Description of file

'-- Load Area Textures -----

If (GameType = 1) Then Area = "Earth"

If (GameType = 2) Then Area = "Death"

Set Texture(0) = MakeTexture(App.Path & "\\Image\\" + Area +
"\\Ground.jpg")

Set Texture(1) = MakeTexture(App.Path & "\\Image\\" + Area +
"\\Stena1.bmp")

Set Texture(2) = MakeTexture(App.Path & "\\Image\\" + Area +
"\\Stena2.bmp")

Set Texture(3) = MakeTexture(App.Path & "\\Image\\" + Area +
"\\Nebo.jpg")

Set Texture(4) = MakeTexture(App.Path & "\\Image\\" + Area +
"\\Nizz.jpg")

If (GameType = 1) Then 'Earth <- Smoke Only

Set Texture(5) = MakeTexture(App.Path & "\\Image\\" + Area +
"\\Tree1.bmp")

Set Texture(6) = MakeTexture(App.Path & "\\Image\\" + Area +
"\\Tree1.bmp")

```

Set Texture(8) = MakeTexture(App.Path & "\Image\" + Area +
"\Smoke.bmp")
Set Texture(9) = MakeTexture(App.Path & "\Image\" + Area +
"\Smoke.bmp")
End If

```

```

If (GameType = 2) Then 'Death <- Faire Only
Set Texture(5) = MakeTexture(App.Path & "\Image\" + Area +
"\Tree2.bmp")
Set Texture(6) = MakeTexture(App.Path & "\Image\" + Area +
"\Tree2.bmp")
Set Texture(8) = MakeTexture(App.Path & "\Image\" + Area +
"\Faire.bmp")
Set Texture(9) = MakeTexture(App.Path & "\Image\" + Area +
"\Faire.bmp")
End If

```

'-- Load Other Textures -----

```

Set Expl = MakeTexture(App.Path & "\Image\Expl.bmp")
Expl.GetSurfaceDesc Expldsc ' Read Description of file

```

```

Set Texture(7) = MakeTexture(App.Path &
"\Image\CompKub.bmp")
Set Animate1 = MakeTexture("", CelMaxX, CelMaxY) ' Empty for
ANIMATION
Set Animate2 = MakeTexture("", CelMaxX, CelMaxY) ' Empty for
ANIMATION

```

```

Set GameOver = MakeTexture(App.Path &
"\Image\GameOver.jpg")
GameOver.GetSurfaceDesc overdesc 'Read Description of file

```

```

Set EnergyCMP = MakeTexture(App.Path &
"\Image\EnergyCMP.bmp")
Set EnergyPLY = MakeTexture(App.Path &
"\Image\EnergyPLY.bmp")

```

```

Set Craft = MakeTexture(App.Path & "\\Image\\CraftStain.bmp")
Set Craff = MakeTexture(App.Path & "\\Image\\CraftFired.bmp")
Craft.GetSurfaceDesc CraftDsc
Craff.GetSurfaceDesc CraftDsc

```

```

End Sub

```

```

'===== MAIN START FUNCTION =====

```

```

Private Sub Form_Load()

```

```

    ShowCursor False      ' Hide Mouse Cursor
    CMP_Energy = 100      ' Computer Energy
    PLY_Energy = 100      ' Player Energy
    Player_Win = 0        ' Player WINNINGs
    Call Init_DirectX      ' Init DirectX, DirectDraw, Sound, Input
    Call Init_3DMatixes    ' InitWorld Matrix, Projection Matrix
    Call Init_RenderMode    ' Init Render Mode, FOG
    Call Init_Geometry      ' Init Doom 4D World
    Call Load_Textures      ' Load All Texture pictures
    Call Clear_Device      ' Clear 3D Screen to BLACK
    Call ShowDoom4DWorld    ' Show DOOM 4D World
    Call DemonstLoop        ' Intro Pictures and DOOM LOGO
    Call SelectsLoop        ' After Intro, STOP Sounds and Music
    Call MainGameLoop      ' Primary Loop & EndGameLoop
    ShowCursor True        ' Show Mouse Cursor
    Unload Me ' Exit from Programm!!!!!!!!!!!!111  bye bye...
End Sub

```

```

Private Sub MainGameLoop()

```

```

    Call PrimaryLoop      ' Main Game Loop.....
    Call EndGameLoop      ' Game Over and Exit From DOOM
    4D.....

```

```

End Sub

```

```

Private Sub ShowDoom4DWorld()

```

```

    objFront.Flip Nothing, DDFLIP_WAIT ' Show DOOM 4D World
End Sub

```

Private Sub SelectsLoop()

MMControl.Command = "Close"

DSBuffer.Stop

End Sub

Private Sub EndGameLoop()

Quit = False

StartYES = False

PlaySound (App.Path & "\\Sound\\Death_end.wav")

z = 0

Do

z = z + 1

objBack.Blt DDRect((ScreenW / 2 - 100) - z, (ScreenH / 2) - z,
(ScreenW / 2 + 100) + z, (ScreenH / 2) + z), GameOver, DDRect(0,
0, overdsc.lWidth, overdsc.lHeight), DDBLT_KEYSRC

Call ShowDoom4DWorld 'FLIPPPPPPPPPPPPPPPPPPPPPPP

Loop Until (z > ScreenH / 2)

Do ' Press ESC to exit from the game

objBack.Blt DDRect(0, 0, ScreenW, ScreenH), GameOver,
DDRect(0, 0, overdsc.lWidth, overdsc.lHeight), DDBLT_WAIT
objBack.SetForeColor (RGB(255, 0, 0)) ' Green

'objBack.SetFont (Arial)

Call objBack.DrawText(40, 45, "You Kill Monster -> " &
Player_Win & " Times!!!", False)

Call objBack.DrawText(40, 60, "You Play time is -> " &
Int(PlayTime / 60) & " Minutes", False)

objBack.SetForeColor (RGB(128, 70, 255)) ' Green

Call objBack.DrawText(40, 80, "Come Back Soon!!!", False)

objBack.SetForeColor (RGB(255, 255, 0)) 'Yellow

Call objBack.DrawText(30, 20, "The G A M E is O V E R", False)

Call objBack.DrawText(ScreenW / 2 - 100, 10, "Created By Denis Astahov (c) 2004.", False)

Call objBack.DrawText(ScreenW - 150, 20, "www.adv400.da.ru", False)

objBack.SetForeColor (RGB(255, 0, 255)) 'Magenta

Call objBack.DrawText(20, ScreenH - 40, "Press N to New Game!", False)

Call objBack.DrawText(20, ScreenH - 20, "Press ESC to Exit...", False)

Call ShowDoom4DWorld 'FLIPPPPPPPPPPPPPPPPPPPPPPP

DoEvents

Loop Until Quit = True Or StartYES = True

If (StartYES = True) Then ' START NEW GAME

CMP_Energy = 100 ' Computer Energy

PLY_Energy = 100 ' Player Energy

Player_Win = 0 ' Player WINNINGs

Call MainGameLoop ' Start NEW GAME

End If

'Unload Me ' Exit from Programm!!!!!!!!!!!!111 bye bye...

End Sub

Private Sub DemonstLoop()

PicShowed = 0

'LoadPAKFile (App.Path & "\Doom4d.pak")

'GetPAKFile ("Sound\Intro0.wav")

'Close #80

PlayMusic (App.Path & "\Music\Music1.wav")

PlaySound (App.Path & "\Sound\Intro0.wav")

TimerIntro.Enabled = True

x = Y = 0

Do

Call Clear_Device

If (PicShowed >= 1) Then

objBack.Blt DDRect(0, 0, ScreenW / 2, ScreenH / 2), Intro1,
DDRect(0, 0, intrdsc.lWidth / 2, intrdsc.lHeight / 2), DDBLT_WAIT
End If

If (PicShowed >= 2) Then

objBack.Blt DDRect(ScreenW / 2, 0, ScreenW, ScreenH / 2),
Intro1, DDRect(intrdsc.lWidth / 2, 0, intrdsc.lWidth, intrdsc.lHeight
/ 2), DDBLT_WAIT
End If

If (PicShowed >= 3) Then

objBack.Blt DDRect(0, ScreenH / 2, ScreenW / 2, ScreenH), Intro1,
DDRect(0, intrdsc.lHeight / 2, intrdsc.lWidth / 2, intrdsc.lHeight),
DDBLT_WAIT
End If

If (PicShowed >= 4) Then

objBack.Blt DDRect(ScreenW / 2, ScreenH / 2, ScreenW,
ScreenH), Intro1, DDRect(intrdsc.lWidth / 2, intrdsc.lHeight / 2,
intrdsc.lWidth, intrdsc.lHeight), DDBLT_WAIT
End If

If (PicShowed = 6) Then

x = x + 0.3

Y = Y + 0.3

If (x > 80) Then x = 80

If (Y > 80) Then Y = 80

objBack.Blt DDRect(0, 0, ScreenW, ScreenH), Intro1, DDRect(0, 0,
intrdsc.lWidth, intrdsc.lHeight), DDBLT_WAIT
objBack.Blt DDRect(x, Y, ScreenW - x, ScreenH - Y), Intro5,
DDRect(0, 0, demodsc.lWidth, demodsc.lHeight - 5),
DDBLT_KEYSRC

End If

'-----

If (PicShowed >= 1) Then Call ShowDoom4DWorld '
FLIPPPPPPPPPPPPPPPPP

DoEvents

Loop Until Continue = True ' Press ENTER to Continue

Set Intro1 = Nothing ' Clear Video Memory

Set Intro5 = Nothing ' Clear Video Memory

TimerIntro.Enabled = False

Quit = False

End Sub

Private Sub Clear_Device() ' Clear 3D Device

Dim ClearRect(0 To 0) As D3DRECT

ClearRect(0).x1 = 0

ClearRect(0).y1 = 0

ClearRect(0).x2 = ScreenW

ClearRect(0).y2 = ScreenH

Call objDevice.Clear(1, ClearRect, D3DCLEAR_TARGET Or
D3DCLEAR_ZBUFFER, RGB(0, 0, 0), 1, 0)

End Sub

Private Function DXVector(ByVal x As Single, ByVal Y As Single,
ByVal z As Single) As D3DVECTOR

DXVector.x = x

DXVector.Y = Y

DXVector.z = z

End Function

Private Sub MoveMatrix(Matrix As D3DMATRIX, Vector As
D3DVECTOR)

objDX.IdentityMatrix Matrix

```
Matrix.rc41 = Vector.x  
Matrix.rc42 = Vector.Y  
Matrix.rc43 = Vector.z
```

```
End Sub
```

```
Private Sub ZoomMatrix(Matrix As D3DMATRIX, Vector As  
D3DVECTOR)
```

```
objDX.IdentityMatrix Matrix  
Matrix.rc11 = Vector.x  
Matrix.rc22 = Vector.Y  
Matrix.rc33 = Vector.z
```

```
End Sub
```

```
Private Sub Draw_Nizz()
```

```
'Draw NIZZ  
objDevice.SetTexture 0, Texture(4)  
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,  
D3DFVF_LVERTEX, VertexNIZZ(0), 4, D3DDP_DEFAULT)
```

```
End Sub
```

```
Private Sub Draw_Computer() ' Draw by Ground
```

```
objDevice.SetTexture 0, Texture(7)  
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,  
D3DFVF_LVERTEX, VertexCOMP(0), 24, D3DDP_DEFAULT)
```

```
End Sub
```

```
Private Sub Draw_SPRITE3D(OBJ As SPRITE_3D)
```

```
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,  
D3DFVF_LVERTEX, OBJ.Vert1(0), 4, D3DDP_DEFAULT)  
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,  
D3DFVF_LVERTEX, OBJ.Vert2(0), 4, D3DDP_DEFAULT)
```



```
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,  
D3DFVF_LVERTEX, OBJ.Vert3(0), 4, D3DDP_DEFAULT)  
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,  
D3DFVF_LVERTEX, OBJ.Vert4(0), 4, D3DDP_DEFAULT)
```

```
End Sub
```

```
Private Sub Draw_Ground() ' Draw by Ground
```

```
'Draw Ground
```

```
objDevice.SetTexture 0, Texture(0)  
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,  
D3DFVF_LVERTEX, VertexGRND(0), 4, D3DDP_DEFAULT)
```

```
'Draw TREEEE My FOREST :)
```

```
objDevice.SetTexture 0, Texture(5) ' First TREE type
```

```
For i = 1 To TREE_MAX
```

```
    Call Draw_SPRITE3D(Tree1(i))
```

```
Next i
```

```
objDevice.SetTexture 0, Texture(6) ' Second TREE type
```

```
For i = 1 To TREE_MAX
```

```
    Call Draw_SPRITE3D(Tree2(i))
```

```
Next i
```

```
'Draw NEBO
```

```
objDevice.SetTexture 0, Texture(3)
```

```
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,  
D3DFVF_LVERTEX, VertexNEBO(0), 4, D3DDP_DEFAULT)
```

```
'Draw WALL1
```

```
objDevice.SetTexture 0, Texture(1)
```

```
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,  
D3DFVF_LVERTEX, VertexWALL1(0), 40, D3DDP_DEFAULT)
```

```
'Draw WALL2
```

```
objDevice.SetTexture 0, Texture(1)
```

```
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,  
D3DFVF_LVERTEX, VertexWALL2(0), 40, D3DDP_DEFAULT)
```

```
'Draw WALL3
objDevice.SetTexture 0, Texture(2)
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,
D3DFVF_LVERTEX, VertexWALL3(0), 40, D3DDP_DEFAULT)
```

```
'Draw WALL4
objDevice.SetTexture 0, Texture(2)
Call objDevice.DrawPrimitive(D3DPT_TRIANGLESTRIP,
D3DFVF_LVERTEX, VertexWALL4(0), 40, D3DDP_DEFAULT)
```

```
'Fire & smoke-----
If (objDX.TickCount > Wait + 40) Then
    Wait = objDX.TickCount
```

```
    CelX = CelX + 1
    If (CelX > FrameX - 1) Then
        CelX = 0
        CelY = CelY + 1
    End If
```

```
    If (CelY > FrameY - 1) Then
        CelX = 0
        CelY = 0
    End If
```

```
    Dim Rect1 As RECT
    Dim Rect2 As RECT
```

```
    Rect1.Left = CelX * CelMaxX
    Rect1.Right = Rect1.Left + CelMaxX
    Rect1.Top = CelY * CelMaxY
    Rect1.Bottom = Rect1.Top + CelMaxY
    Animate1.Blt Rect2, Texture(8), Rect1, DDBLT_WAIT
    Animate2.Blt Rect2, Texture(9), Rect1, DDBLT_WAIT
End If
```

```
objDevice.SetRenderState D3DRENDERSTATE_DESTBLEND, 2
```

```
objDevice.SetRenderState  
D3DRENDERSTATE_ALPHABLENDENABLE, True  
'Prozra4nost
```

```
objDevice.SetTexture 0, Animate1 ' Animated TILE Texture  
For i = 1 To FIRESMOKE  
    Call Draw_SPRITE3D(Smoke(i))  
Next i  
objDevice.SetTexture 0, Animate2 ' Animated TILE Texture  
For i = 1 To FIRESMOKE  
    Call Draw_SPRITE3D(Faire(i))  
Next i
```

```
objDevice.SetRenderState  
D3DRENDERSTATE_ALPHABLENDENABLE, False  
'-----
```

End Sub

```
Private Sub Read_Keys()  
    Keyboard.GetDeviceStateKeyboard KeyState 'Read Keyboard Keys  
  
    If (KeyState.Key(DIK_F12) <> 0) Then ' Teleportation Tunnel  
        PlaySound (App.Path & "\\Sound\\Teleport.wav")  
        Call Change_Area  
    End If
```

End Sub

```
Private Sub Change_Area()
```

```
Select Case (GameType)  
    Case 1: GameType = 2  
    Case 2: GameType = 1  
End Select  
Call Load_Textures
```

End Sub

```
Private Sub Set_CameraEye()
```

```
'-----Glaza Personaja === Glaza Cameri-----
Call objDX.ViewMatrix(ViewMatrix, VecCamLok, VecCamSee,
DXVector(0, 1, 0), 0)
objDevice.SetTransform D3DTRANSFORMSTATE_VIEW,
ViewMatrix
'-----
```

End Sub

Private Sub PrimaryLoop()

```
PlayTime = 0 ' Star Time
TimerPlayTime.Enabled = True
```

```
' Ustanovki CAMERI-PERSONAJA-----
dist2focus = 70      ' Focus Cameri kuda smotrit
step = 0.5001         ' Razmer Shaga
alfa = 0              ' Ugol kuda smotril = 0 Default
delta = 0.8 * Radians ' Ugol na skoko povara4ivaem
zzz = POLE / 100      ' For ZOOM Igrovoe Pole
```

```
' Camera First Lokated!
VecCamLok.x = 0
VecCamLok.Y = 6 ' Constanta!!!
VecCamLok.z = -12
' Camera First Looking!
VecCamSee.x = 0
VecCamSee.Y = VecCamLok.Y
VecCamSee.z = dist2focus ' Smotret Vglub Vpered na 10
```

```
'-----
PlaySound (App.Path & "\Sound\Lets_go.wav")
Do
```

```
Call Clear_Device 'Clear Back Buffer Screen
Call Read_Keys    'Read Keyboard Keys
Call Make_Move    'Make Virtual Movement of my Creature-
Camera
```

Call Set_CameraEye 'Set Camera-Eye Updated Polojenie

```
'----->>> TRANSFORMATION MAKES HERE <<<<--  
Call ZoomMatrix(MatGrn, DXVector(zzz, zzz, zzz)) ' Must to do  
Zoom!!!!  
Call ZoomMatrix(MatNiz, DXVector(zzz, zzz, zzz)) ' Must to do  
Zoom!!!!  
Call ZoomMatrix(MatCmp, DXVector(0.25, 0.25, 0.25)) ' Must to  
do Zoom!!!!  
'-----
```

```
RotateX = RotateX + 1  
RotateY = RotateY + 4  
RotateZ = RotateZ + 1  
If (RotateX > 360) Then RotateX = 0  
If (RotateY > 360) Then RotateY = 0  
If (RotateZ > 360) Then RotateZ = 0
```

```
'Rotate Computer OBJ  
objDX.RotateXMatrix WorkMatrix, RotateX * Radians ' Rotate by  
X  
objDX.MatrixMultiply MatCmp, WorkMatrix, MatCmp ' Update  
Matrix
```

```
objDX.RotateZMatrix WorkMatrix, RotateZ * Radians ' Rotate by Z  
objDX.MatrixMultiply MatCmp, WorkMatrix, MatCmp ' Update  
Matrix
```

```
objDX.RotateYMatrix WorkMatrix, RotateY * Radians ' Rotate by  
Y  
objDX.MatrixMultiply MatCmp, WorkMatrix, MatCmp ' Update  
Matrix
```

```
'Rotate NIZZ  
objDX.RotateYMatrix WorkMatrix, RotateX * Radians 'Rotate  
NIZZ by -Y-
```

objDX.MatrixMultiply MatNiz, WorkMatrix, MatNiz 'Rotate NIZZ
by -Y-

Call Move_Computer

'Begin Scene-----
objDevice.BeginScene

objDevice.SetTransform D3DTRANSFORMSTATE_WORLD,
MatGrn
Call Draw_Ground

objDevice.SetTransform D3DTRANSFORMSTATE_WORLD,
MatCmp
Call Draw_Computer

objDevice.SetTransform D3DTRANSFORMSTATE_WORLD,
MatNiz
Call Draw_Nizz

objDevice.EndScene
'End Scene-----

'-----Put Craft Interface -----

'Call PlayerShot

If (KeyState.Key(DIK_SPACE) <> 0) Then

 SetShip = Craft ' Return by DEFAULT Craft Stain

 If (VistrelNOW = False) Then

 VistrelNOW = True 'Sey4as strelaem!!! ne delat povtorniy

vistrel

 SetShip = Craff ' Craft Fired Picture

 PlaySound (App.Path & "\Sound\Laser.wav")

 Call CheckHitTarget ' Proverka popadeniya

 End If

Else

 SetShip = Craft ' Craft Stain Picture

```

End If
objBack.Blt DDRect(0, 0, ScreenW, ScreenH), SetShip, DDRect(0,
0, CraftDsc.lWidth, CraftDsc.lHeight), DDBLT_KEYSRC
'
'-----

```

'Show Energy Pictures According to SCREEN and CRAFT
Resolution!!!

```

objBack.Blt DDRect(ScreenW / CraftDsc.lWidth * 30, ScreenH /
CraftDsc.lHeight * 30, ScreenW / CraftDsc.lWidth * (PLY_Energy
+ 30), ScreenH / CraftDsc.lHeight * 50), EnergyPLY, DDRect(0, 0,
PLY_Energy, 20), DDBLT_KEYSRC
objBack.Blt DDRect(ScreenW / CraftDsc.lWidth * 370, ScreenH /
CraftDsc.lHeight * 30, ScreenW / CraftDsc.lWidth * (CMP_Energy
+ 370), ScreenH / CraftDsc.lHeight * 50), EnergyCMP, DDRect(0,
0, CMP_Energy, 20), DDBLT_KEYSRC

```

'-----Copy all From BACK to FONT buffer
Call ShowDoom4DWorld 'FLIPPPPPPPPPPPPPPPPPPPPP

```

DoEvents
Loop Until Quit = True
'-----
TimerPlayTime.Enabled = False
End Sub

```

Private Sub CheckHitTarget()

'=====>>>> Proverka padeniya=====

```

'xm <- X_Computer_Current location
'zm <- Z_Computer Current location
'dist2focus <- Rastoyanie do focusa kameri

dist2target = Sqr((xm - VecCamLok.x) ^ 2 + (zm -
VecCamLok.z) ^ 2)
If (dist2target <= dist2focus) Then

    Ax = VecCamSee.x - VecCamLok.x ' Vector -> Ax

```

```

Ay = VecCamSee.Y - VecCamLok.Y ' Vector -> Ay
Az = VecCamSee.z - VecCamLok.z ' Vector -> Az

```

```

Bx = xm - VecCamLok.x ' Vector -> Bx
By = 0 ' Vector -> By
Bz = zm - VecCamLok.z ' Vector -> Bz

```

```

CosAlfa = ((Ax * Bx + Ay * By + Az * Bz) / ((Sqr(Ax ^ 2 +
Ay ^ 2 + Az ^ 2)) * (Sqr(Bx ^ 2 + By ^ 2 + Bz ^ 2))))

```

```

If (CosAlfa >= Cos(4 / dist2target)) Then
    Call Show_Expl ' Show Eplosion
    PlaySound (App.Path & "\\Sound\\Killcomp.wav")
    CMP_Energy = CMP_Energy - 1
    If (CMP_Energy = 0) Then ' If computer DEAD
        PlaySound (App.Path & "\\Sound\\Dead.wav")
        Player_Win = Player_Win + 1
        Call Change_Area ' Change Area
        CMP_Energy = 100
    End If
End If
End If
End If

```

```

'=====>>>> Proverka popadeniya =====
End Sub

```

```

Private Sub Show_Expl() ' Show 3D Expl Texture
objBack.Blt DDRect(0, 0, ScreenW, ScreenH), Expl, DDRect(0, 0,
Expldsc.lWidth, Expldsc.lHeight), DDBLT_KEYSRC
End Sub

```

```

Private Sub Move_Computer()
Randomize
If (cstep = COMPSTEP) Then ' kogda doshli do to4ki, vzyat
novuyu

```

```

CompNewX = GetRandom(-90, 90)
CompNewZ = GetRandom(-90, 90)

```



```

    If (CompNewX < -90 Or CompNewX > 90) Then CompNewX =
0
    If (CompNewZ < -90 Or CompNewZ > 90) Then CompNewZ = 0

    Call objBack.SetForeColor(RGB(255, 255, 255))
    Call objBack.DrawText(100, 100, "X= " & CompNewX, False)
    Call objBack.DrawText(100, 130, "Z= " & CompNewZ, False)

    dx = (CompOldX - CompNewX) / COMPSTEP ' rashitat shag
    dz = (CompOldZ - CompNewZ) / COMPSTEP

    CompOldX = CompNewX
    CompOldZ = CompNewZ

    cstep = 0
End If

    Call MoveMatrix(WorkMatrix, DXVector(xm, VecCamLok.Y -
2, zm))
    objDX.MatrixMultiply MatCmp, MatCmp, WorkMatrix '<--
Problema marganiya zdes

    xm = xm + dx
    zm = zm + dz

    cstep = cstep + 1

    Call CheckDamage ' Check Collision Player & Computer

End Sub
Private Sub CheckDamage() ' Check Collision Player & Computer

    'proverka stolkonoveniya
    dist2target = Sqr((xm - VecCamLok.x) ^ 2 + (zm -
VecCamLok.z) ^ 2)

    If (dist2target <= 4) Then ' Stolknovenie
        PlaySound (App.Path & "\\Sound\\Pain.wav")
        PLY_Energy = PLY_Energy - 2 ' Minus Energy
        If (PLY_Energy = 0) Then Quit = True ' Quit

```

End If

End Sub

Public Sub Make_Move()

'-----Dvijenje Personaja-----

' Nadobi izmenit na Switch!!! :)

If (KeyState.Key(DIK_UP) <> 0) Then

VecCamLok.z = VecCamLok.z + step * Cos(alfa)

VecCamLok.x = VecCamLok.x + step * Sin(alfa)

VecCamSee.z = VecCamLok.z + (step + dist2focus) *
Cos(alfa)

VecCamSee.x = VecCamLok.x + (step + dist2focus) *
Sin(alfa)

End If

If (KeyState.Key(DIK_DOWN) <> 0) Then

VecCamLok.z = VecCamLok.z - step * Cos(alfa)

VecCamLok.x = VecCamLok.x - step * Sin(alfa)

VecCamSee.z = VecCamLok.z + (dist2focus - step) *
Cos(alfa)

VecCamSee.x = VecCamLok.x + (dist2focus - step) * Sin(alfa)

End If

If (KeyState.Key(DIK_RIGHT) <> 0) Then

alfa = alfa + delta

If (alfa = 360 Or alfa = -360) Then alfa = 0

VecCamSee.z = VecCamLok.z + dist2focus * Cos(alfa)

VecCamSee.x = VecCamLok.x + dist2focus * Sin(alfa)

End If

If (KeyState.Key(DIK_LEFT) <> 0) Then

alfa = alfa - delta

If (alfa = 360 Or alfa = -360) Then alfa = 0

```

    VecCamSee.z = VecCamLok.z + dist2focus * Cos(alfa)
    VecCamSee.x = VecCamLok.x + dist2focus * Sin(alfa)
End If

'-----FOG Control!!!!!!!!!!!!-----
If (KeyState.Key(DIK_Z) <> 0) Then ' Swith FOG to -> OFF
objDevice.SetRenderState D3DRENDERSTATE_FOGENABLE,
False
End If

If (KeyState.Key(DIK_W) <> 0) Then ' Swith FOG to -> WHITE
ON
FogColor = objDX.CreateColorRGBA(1, 1, 1, 1) 'RED GREEN
BLUE ALFA 'By Default!
objDevice.SetRenderState D3DRENDERSTATE_FOGCOLOR,
FogColor
objDevice.SetRenderState D3DRENDERSTATE_FOGENABLE,
True
End If

If (KeyState.Key(DIK_Q) <> 0) Then ' Swith FOG to -> OF
BLACK
FogColor = objDX.CreateColorRGBA(0, 0, 0.1, 1) 'RED GREEN
BLUE ALFA 'By Default!
objDevice.SetRenderState D3DRENDERSTATE_FOGCOLOR,
FogColor
objDevice.SetRenderState D3DRENDERSTATE_FOGENABLE,
True
End If

If (KeyState.Key(DIK_1) <> 0) Then
objDevice.SetRenderState
D3DRENDERSTATE_FOGVERTEXMODE, D3DFOG_LINEAR
End If

If (KeyState.Key(DIK_2) <> 0) Then

```

```

objDevice.SetRenderState
D3DRENDERSTATE_FOGVERTEXMODE, D3DFOG_EXP
End If

If (KeyState.Key(DIK_3) <> 0) Then
objDevice.SetRenderState
D3DRENDERSTATE_FOGVERTEXMODE, D3DFOG_EXP2
End If
'-----

'-----ALFA Blending Effect-----
If (KeyState.Key(DIK_A) <> 0) Then ' Swith ALFABLENDING ->
ON
objDevice.SetRenderState D3DRENDERSTATE_DESTBLEND, 2
objDevice.SetRenderState
D3DRENDERSTATE_ALPHABLENDENABLE, True
End If

If (KeyState.Key(DIK_S) <> 0) Then ' Swith ALFABLENDING ->
OFF
objDevice.SetRenderState
D3DRENDERSTATE_ALPHABLENDENABLE, False
End If

'-----
'If Player try to go OUT of wall say NOTHINK!!!!!!!!!!1

If (VecCamLok.x > POLE - 2) Then
    VecCamLok.x = POLE - 2
    PlaySound (App.Path & "\\Sound\\Nothink.wav")
End If
If (VecCamLok.z > POLE - 2) Then
    VecCamLok.z = POLE - 2
    PlaySound (App.Path & "\\Sound\\Nothink.wav")
End If

If (VecCamLok.x < -POLE + 2) Then
    VecCamLok.x = -POLE + 2
    PlaySound (App.Path & "\\Sound\\Nothink.wav")
End If

```

```

If (VecCamLok.z < -POLE + 2) Then
    VecCamLok.z = -POLE + 2
    PlaySound (App.Path & "\\Sound\\Nothink.wav")
End If

```

```

'-----

```

```

End Sub

```

```

Private Function GetRandom(ByVal FromNumber As Integer,
ByVal ToNumber As Integer) As Integer
    GetRandom = Int(FromNumber + (Rnd() * (Abs(ToNumber -
FromNumber) + 1)))
End Function

```

```

Private Sub Form_Unload(Cancel As Integer)
    Close          ' Close all OPENed files
    ShowCursor True    ' Show Mouse Cursor
    TimerIntro.Enabled = False ' Stop TIMER
    Keyboard.Unacquire    ' Stop Read Keyboard
    DSBuffer.Stop        ' Stop all SOUND
    MMControl.Command = "Close" ' Stop all MUSIC
    Set objDS = Nothing
    Set objDI = Nothing
    Set objD3D = Nothing
    Set objDD = Nothing
    Set objDX = Nothing
    End
End Sub

```

```

'=====
=====

```

```

Private Sub MMControl_Done(NotifyCode As Integer)
    MMControl.Command = "Close"
End Sub

```

```

Private Sub PlayMusic(MusicFile As String) ' Play MP3 -> WAV
File by MCI32

```

```

MMControl.Command = "Close"
MMControl.FileName = MusicFile
MMControl.Command = "Open"
MMControl.Command = "Play"
End Sub

```

```

Private Sub PlaySound(FileName As String) ' Play WAV File by
DirectSound
    BufferDesc.lFlags = (DSBCAPS_CTRLFREQUENCY Or
DSBCAPS_CTRLPAN Or DSBCAPS_CTRLVOLUME) Or
DSBCAPS_STATIC
    Set DSBuffer = objDS.CreateSoundBufferFromFile(FileName,
BufferDesc, WaveFileFormat)
    DSBuffer.Play DSBPLAY_DEFAULT ' or ->
DSBPLAY_LOOPING
End Sub

```

```

Private Sub TimerIntro_Timer()
    PicShowed = PicShowed + 1
    Select Case PicShowed

        Case 1: PlaySound (App.Path & "\\Sound\\Intro1.wav")
        Case 2: PlaySound (App.Path & "\\Sound\\Intro2.wav")
        Case 3: PlaySound (App.Path & "\\Sound\\Intro3.wav")
        Case 4: PlaySound (App.Path & "\\Sound\\Intro4.wav")
        Case 6: PlaySound (App.Path & "\\Sound\\Intro0.wav")
            TimerIntro.Enabled = False
    End Select

```

```

End Select

```

```

End Sub

```

```

Private Sub TimerPlayTime_Timer()
    PlayTime = PlayTime + 1 'Plus ONE Seconds

```

```

If (PlayTime = 60 Or PlayTime = 300) Then ' Swith FOG to -> ON
WHITE
objDevice.SetRenderState D3DRENDERSTATE_FOGENABLE,
False

```

```

FogColor = objDX.CreateColorRGBA(1, 1, 1, 1) 'RED GREEN
BLUE ALFA 'By Default!
objDevice.SetRenderState
D3DRENDERSTATE_FOGVERTEXMODE, D3DFOG_EXP2
objDevice.SetRenderState D3DRENDERSTATE_FOGCOLOR,
FogColor
objDevice.SetRenderState D3DRENDERSTATE_FOGENABLE,
True
End If

If (PlayTime = 180 Or PlayTime = 400) Then ' Swith FOG to -> ON
BLACK
objDevice.SetRenderState D3DRENDERSTATE_FOGENABLE,
False
FogColor = objDX.CreateColorRGBA(0, 0, 0.1, 1) 'RED GREEN
BLUE ALFA 'By Default!
objDevice.SetRenderState
D3DRENDERSTATE_FOGVERTEXMODE, D3DFOG_EXP2
objDevice.SetRenderState D3DRENDERSTATE_FOGCOLOR,
FogColor
objDevice.SetRenderState D3DRENDERSTATE_FOGENABLE,
True
End If

If (PlayTime = 120 Or PlayTime = 500) Then ' Swith FOG to ->
OFF
objDevice.SetRenderState D3DRENDERSTATE_FOGENABLE,
False
End If

End Sub

```

Form Menu

```
()Private Sub Command1_Click
GameType = 1 ' Game type is -> EARTH
Call SetSelectedMode ' Set Video Settings
("PlaySound (App.Path & "\\Sound\\Select.wav
FormMenu.Hide
FormMain.Show
End Sub
```

```
()Private Sub Command2_Click
GameType = 2 ' Game type is -> DEATH
Call SetSelectedMode ' Set Video Settings
("PlaySound (App.Path & "\\Sound\\Select.wav
FormMenu.Hide
FormMain.Show
End Sub
```

```
()Private Sub Form_Load
GameType = 0
```

```
VideoMode.AddItem "640x480x16Bit", 0
VideoMode.AddItem "640x480x32Bit", 1
VideoMode.AddItem "800x600x16Bit", 2
VideoMode.AddItem "800x600x32Bit", 3
VideoMode.AddItem "1024x768x16Bit", 4
VideoMode.AddItem "1024x768x32Bit", 5
VideoMode.AddItem "1280x1024x16Bit", 6
VideoMode.AddItem "1280x1024x32Bit", 7
VideoMode.Text = VideoMode.List(2) ' Default Video Mode
```

```
("PlaySound (App.Path & "\\Sound\\Start.wav
End Sub
```

```
()Private Sub SetSelectedMode
```

```
(Select Case (VideoMode.Text
(Case "640x480x16Bit": Call SetVideoSettings(640, 480, 16
(Case "640x480x32Bit": Call SetVideoSettings(640, 480, 32
(Case "800x600x16Bit": Call SetVideoSettings(800, 600, 16
```



```
(Case "800x600x32Bit": Call SetVideoSettings(800, 600, 32
(Case "1024x768x16Bit": Call SetVideoSettings(1024, 768, 16
(Case "1024x768x32Bit": Call SetVideoSettings(1024, 768, 32
(Case "1280x1024x16Bit": Call SetVideoSettings(1280, 1024, 16
(Case "1280x1024x32Bit": Call SetVideoSettings(1280, 1024, 32
End Select
```

```
End Sub
```

```
()Private Sub Form_Terminate
End
End Sub
```

```
(Private Sub MMControl_Done(NotifyCode As Integer
"MMControl.Command = "Close
End Sub
```

```
Private Sub PlaySound(MusicFile As String) ' Play MP3 -> WAV
File by MCI32
"MMControl.Command = "Close
MMControl.FileName = MusicFile
"MMControl.Command = "Open
"MMControl.Command = "Play
End Sub
```

Module Main

```
API Function to Show or Hide Mouse Cursor ---'
Declare Function ShowCursor Lib "user32" (ByVal bShow As
Long) As Long
ShowCursor True <--- Show Cursor '
ShowCursor False <--- Hide Cursor '
-----'

.....Type of Textures to Loading ----'
Public GameType As Byte ' 1 -> Earth
Hell <- 2 '
-----'

Public Area As String ' Death or Earth
-----'

-----Video Mode Settings'
Public ScreenW As Integer ' Weidth of Screen
Public ScreenH As Integer ' Height of Screen
Public BitPP As Byte ' Bit Per Pixel

Set Selected Video Mode Settings'
Resolution of Video Mode'
:Test PASSED '
x480x16 800x600x16 1024x768x16 1280x1024x16640 '
x480x32 800x600x32 1024x768x32 1280x1024x32640 '

Public Sub SetVideoSettings(W As Integer, H As Integer, BPP As
(Byte
ScreenW = W
ScreenH = H
BitPP = BPP
End Sub
-----'
```

סקירת ספרות וביבליוגרפיה:

ספרות:

Visual Basic 6.0 ,(2001), Michael Reitinger, Gerald Mutch -
Mastering Visual Basic.NET,(2002) ,Evangelos Petroutsos -
DirectX עם Visual Basic 6.0 ,(2002), אורן קרדי
Visual Basic mit DirectX, (2000), Christian Schmelzer -

מקורות אינטרנט:

<http://francis.dupont.free.fr/coindex/english> -
<http://msdn.microsoft.com/vbasic> -
גאומטריה אנליטית: (רוסית)
http://www.ssga.ru/AllMetodMaterial/metod_mat_for_ioot/metodichki/piroch/analit.html



DOOM **FOURTH DIMENSION**
WWW.ADV400.DA.RU

Have a nice Doom!