

דו"ח פרויקט

פרוייקט זה מרנדר סימולציית צמיחה של אלמוגים ב-Blender.
בדו"ח זה אני אסביר את הצד הטכני של הפרוייקט – מעט על הפלטפורמה עצמה (Blender),
האלגוריתם בו השתמשתי, בחירות מימוש ואופן המימוש.

תוכן

2:Blender הפלטפורמה
3:L-system הסבר של
5:מימוש
5:Follow Food
6:Coral System
6:L-system
6:Wave Animation
6:Distance
6:Thickness
7:Distortion
7:Growth Animation
7:Voxel
8:Coral (מראה האלמוג)

הפלטפורמה Blender:

בלנדר זו תוכנת עיצוב המשמשת למידול, רנדור, אנימציה ועוד. המערכת כולה נבנתה בבלנדר, בלי assets חיצוניים, ולמעשה כל כולה היא הקובץ corals.blend.

ספציפית, עיקר המערכת נבנתה בעזרת פיצ'ר ה-**Geometry Nodes** שהתוכנה מציעה, אשר מאפשר לבצע אוטומציה של בניית מודלים באופן של פייפליין, כלומר: יש קלטים מסוימים, הם עוברים טרנספורמציה דרך אוסף של פעולות (nodes) שמופעלות בפייפליין, ובסיום מקבלים את התוצר.

חלק מהכוח של בלנדר מגיע מהעובדה שניתן "לשנות" את התוכנה עצמה, כלומר להוסיף או לגרוע ממנה פונקציונליות. מה שעשיתי היה להוסיף אוסף "סליידרים" שהשתמש יכול לשנות, ובכך "מתחת למכסה המנוע" לשנות את הקלטים ל-Geometry Nodes, ובכך לייצר אלמוגים שונים.



הסבר של L-system:

L-system זהו אלגוריתם לג'נרט פרצדורלי המשמש בעולמות הגרפיקה לייצור של מבנים בעלי אופי [פרקטלי](#), כמו צמחים ועצים. הוא נוצר ב-1968 על ידי הביולוג והבוטנאי אריסטיד לינדנהיימר, על מנת למדל את אופן הצמיחה של צמחים שחקר. ניתן לקרוא עוד על L-system [בא](#).

אלגוריתם זה משמש בפרייקט כדי לג'נרט באופן איטרטיבי את האלמוגים. הוא מורכב מאלפבית (המורכב ממשתנים וקבועים), כללי יצירה, ואקסיומה (מחרוזת התחלתית). אנו מתחילים עם האקסיומה, ומפעילים עליה באופן איטרטיבי את כללי יצירה. בכל איטרציה אנו מפעילים את כל כללי היצירה שאפשר, בו-זמנית.

דוגמה: ניקח את המשתנים "1", "0"; הקבועים "[", "]", ";"; כללי היצירה $0 \rightarrow 11$, $1 \rightarrow 0[0]0$; והאקסיומה "0". כעת נוכל לבנות מחרוזות באופן איטרטיבי:

האקסיומה: 0.

האיטרציה הראשונה: $1[0]0$

האיטרציה השנייה: $11[1[0]0]1[0]0$

האיטרציה השלישית: $1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0$

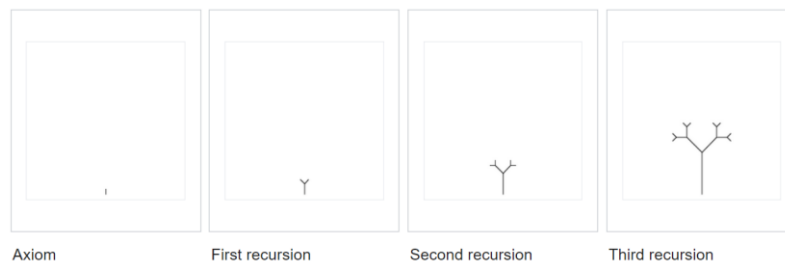
כעת, כדי **להמיר את המחרוזות הללו לתמונה גרפית**, נפרש אותן באופן הבא: נחשוב על מעין cursor במרחב, ונקרא משמאל לימין, כך שכל אות תגרום ל-cursor לבצע פעולה מסוימת. נפרט את הפעולה שכל אות מבצעת:

0 – צייר קו שנגמר בעלה.

1 – צייר קו (פנימי).

[– push למחסנית את המיקום והזווית הנוכחית, ותפנה שמאלה בזווית של 45 מעלות.
] – pop ותעדכן בהתאם את המיקום והזווית הנוכחית, ותפנה ימינה בזווית של 45 מעלות.

נקבל מכך את התמונות הבאות:



הערה: L-system הוא כמו שפה פורמלית, אך השוני הוא שבאלגוריתם זה מפעילים את כל כללי היצירה "בו-זמנית" על המחרוזת הנוכחית, במקום אחד-אחד.

אז אם יש לנו כלל יצירה $S \rightarrow SS$ ומחרוזת S; אנו נקבל באיטרציה הראשונה SS, ובאיטרציה השנייה SSSS (ולא SSS), כי כל S במחרוזת הפך ל-SS.

בפרוייקט זה:

- (1) המשתנים הינם: A,B.
- (2) הקבועים הינם: !, ", /, &, [,], F.
- (3) האקסיומה הינה המחרוזת A.
- (4) כללי היצירה הינם:

A->!"[B]////[B]////B

B->&FFFA

כמו מקודם, נחשוב על cursor במרחב, כך שכל קבוע יגרום לו לבצע פעולה מסוימת, ונתרגם כל קבוע באופן הבא:

F – תנוע קדימה.

! – מעכשיו, תצייר עם עובי קצת קטן יותר.

" – מעכשיו, בכל תנועה, תנוע קצת יותר (מגדיל את אורך הקו).

/ – תפנה שמאלה בזווית של 45 מעלות.

& – תפנה בזווית של 25 מעלות "לתוך המסך" (כזכור, יש שלושה צירים בפרוייקט שלנו).

[– push למחסנית (מתחילים ענף חדש).

] – pop למחסנית (מסיימים את הענף הנוכחי).

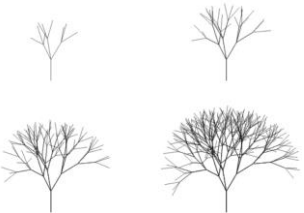
הערה: במקרה זה אנו מתרגמים רק קבועים, ללא משתנים.

כלומר, כאשר קוראים את המחרוזת משמאל לימין, המשתנים לא מתורגמים לפעולה כלשהי.

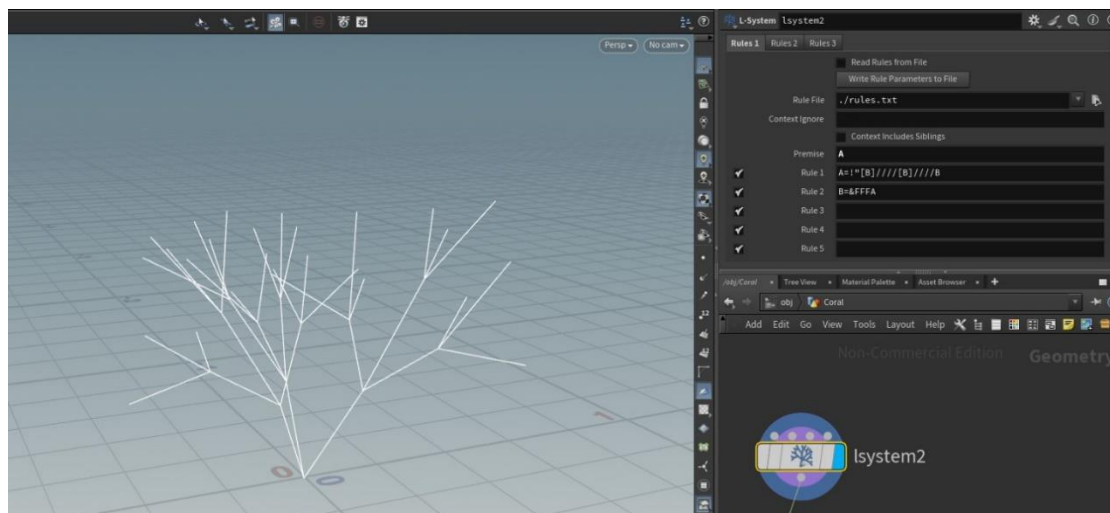
יש עוד כללי יצירה שיכולים לייצר צורת אלמוג, כמו למשל:

A->"[&FFFA]////[&FFFA]////[&FFFA]

כאשר אופן התרגום נשאר זהה.



הכללים נבדקו בתוכנת Houdini (תוכנת אנימצית 3D) לפני שהובאו לשימוש בפרוייקט הנוכחי.

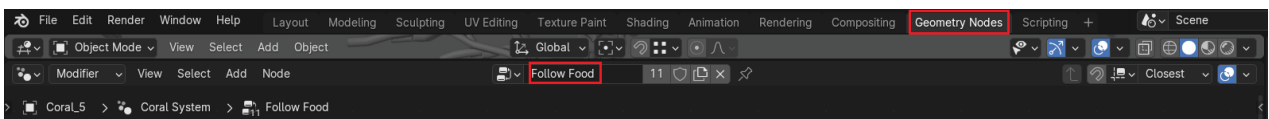


מימוש:

בחלק זה אני אסביר באופן כללי כיצד מימשתי את המערכת, ובפרט את החלקים השונים של ה-Geometry Nodes שיצרתי.
אני אעבור על הפונקציונליות שאני הוספתי לבלנדר, ולא אכסה שינויים שעשיתי בפונקציונליות שמגיעה built in (למשל איך שיניתי את הרקע מאחורי האלמנטים לכחול).

:Follow Food

כדי להגיע ל-nodes המממשים את מכניקת ה"אוכל" (אובייקט שלכיוונו האלמנטים צומחים), נלך בסרגל למעלה ל-Geometry Nodes, ואז ללשונית ה-Follow Food:

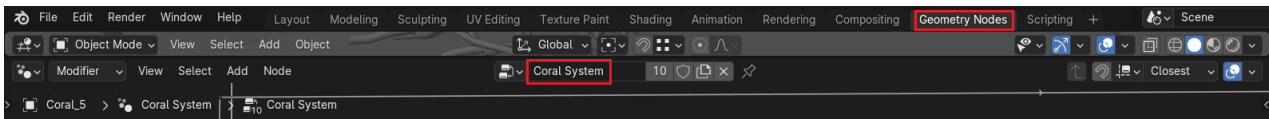


הסבר של ה-nodes: לוקחים את המיקום של כל נקודה בבסיס (ב"גזע") האלמנט, ולוקחים את הנקודה בה נמצא האוכל, ומחשבים את וקטור המרחק ביניהם. לאחר מכן, משתמשים בוקטור הנ"ל כדי לבצע scaling של נקודות האלמנט בכיוון של וקטור המרחק. בכך אנחנו גורמים לנקודות הללו "להתקרב" למיקום של האוכל, שזה בדיוק מה שאנחנו רוצים. ככל שהאלמנט יצמח, הנקודות ישאפו לאותו מיקום.

כמו כן, יש שם Switch node שעוזר לממש תכונות גלובליות, כפי שמוסבר במדריך למשתמש המצורף. בעזרת זה אנו בוחרים האם לקחת כקלט את התכונה הלוקאלית או הגלובלית.

:Coral System

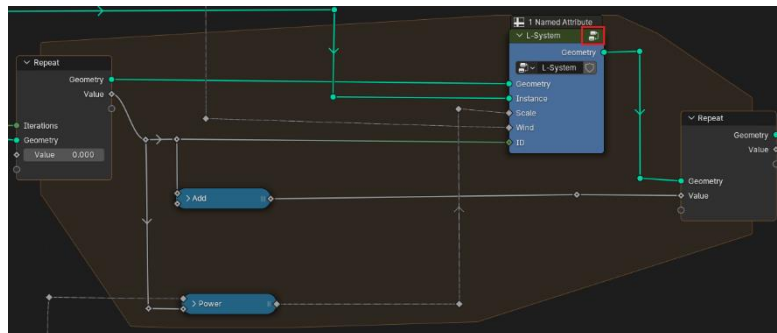
כדי להגיע ל-nodes המממשים את האלמוגים עצמם,
נלך שוב בסרגל למעלה ל-Geometry Nodes, והפעם ללשונית Coral System:



הסבר של ה-nodes: לשם קריאות וארגון, ה-nodes מחולקים לקטגוריות בקובץ עצמו, ולפי זה נסביר אותן כאן:

:L-system

כדי להגיע ל-nodes בקטגוריה הזו, נלך ל-nodes L-System ונלחץ על הכפתור המסומן באדום:



מה שאנו עושים כאן זה, באופן רפטיבי (בעזרת ה-Repeat node), מבצעים איטרציות של L-system, כלומר הולכים לעלים, יוצרים לכל אחד בן, מוסיפים אותו למערך הצמתים, וכן הלאה. בכל רגע נתון מוחזקים רק העלים, בלי צמתים פנימיים, ובכך האלגוריתם מתבצע ביעילות.

:Wave Animation

אנו מממשים את התנודתיות של האלמוגים בכך שאנו לוקחים ערך רנדומי ואת הזמן הנוכחי, מפעילים עליהם מניפולציה אלגברית מסויימת, ופולטים ערך seed מסויים שתלוי בזמן.

הערך הזה יהיה קלט ל-nodes של ה-L-system.

:Distance

מאוד פשוט – מחשבים את המרחק של כל צומת באלמוג מצומת הבסיס (הצומת הראשון של האלמוג), בעזרת אלגוריתם shortest path כלשהו.

:Thickness

מקבלים כקלט את המרחקים שחישבנו ב-Distance, ובעזרת Attribute Statistic Node לוקחים את המקסימלי. כעת, בעזרת Divide Node מחלקים את המרחק שכל צומת שומר באותו מרחק מקסימלי. קיבלנו מעין גרדיאנט, שבו כל צומת מחזיק ערך בין 0 ל-1, כך שצומת מחזיק ערך קרוב יותר ל-0 ככל שהוא קרוב יותר לצומת הבסיס.

לאחר מכן, בעזרת Subtract Node, עבור כל צומת אנו מחסרים את הערך שלו מקבוע מסוים, ובכך אנו הופכים את כיוון הגרדיאנט – כעת הצמתים הקרובים לצומת הבסיס הם אלו שיחזיקו ערכים גדולים יותר.

כל מה שעשינו עד כה היה על מנת לתת משקל גדול יותר לצמתים הקרובים לצומת הבסיס. לבסוף, מעבירים את הצמתים דרך RGB Curve Node, שמגדיל את עובי ה-mesh לפי אותם משקלים, כך שהאלמוג יהיה עבה יותר בבסיס שלו.

:Distortion

לוקחים את mesh הקלט ומעבירים אותו דרך Noise Texture Node, שלוקח את הצורה ה"שטוחה" שלו ומעקל אותה מעט במקומות אקראיים, לפי שיטת אקראיות מוגדרת מראש. לאחר מכן עוברים ב-Subtract Node כדי לקזז את העיקולים בבסיס האלמוג, כך שלא יקרה מצב בו הוא זז ממקומו המקורי. לבסוף עוברים ב-Scaling Node כדי לקבוע באיזה עוצמה נרצה לעקל.

:Growth Animation

כאן מסתתרת אופטימיזציה יפה שאנו עושים במימוש האנימציה. בהינתן כל הפרמטרים שהמשתמש נותן, הצורה הסופית של האלמוג כבר נקבעת באופן יחיד, על ידי לקיחת התוצר הסופי של ה-L-system. כלומר, כשמגיעים לכאן, כבר סיימנו את כל האיטרציות. בפרט, אנו לא מג'נרטים כל איטרציה ב-real-time עבור המשתמש. אפשר לחשוב על זה כמעין preprocessing שעושים, כך שבהרצת התוכנית המידע כבר קיים. אז כדי לדמות צמיחה, אנו בפריים הראשון מסתירים את כל האלמוג, ואז בפריים שאחריו מסתירים קצת פחות (החל מהבסיס), ובפריים שאחרי קצת פחות וכן הלאה, עד שאנחנו חושפים את כל האלמוג – מלמטה למעלה. אבל האובייקט עצמו של האלמוג היה קיים במלואו עוד מההתחלה.

בנוסף, בעזרת Scene Time node, ומעט אריתמטיקה עליו, אנו ממשים את הסליידר של Growth Time ואת האנימציה עצמה.

:Voxel

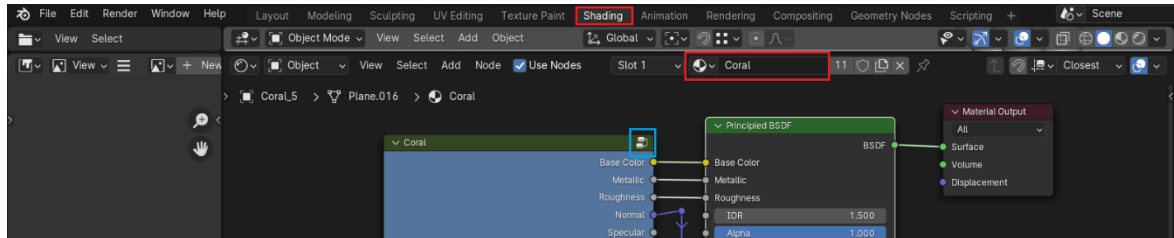
זהו השלב האחרון בפיפליין. עד כה עבדנו עם curves כי זה יותר מהיר, ובחלק הזה אנו לבסוף ממירים ל-mesh שהמשתמש יכול לראות.

אם אופציית ה-voxel מסומנת, מה שיקרה זה שה-mesh יעבור לגוף בעל נפח, ואז בחזרה ל-mesh, מה שמייצר את הצורה ה-"voxelized" המחוספסת. זה קורה על ידי ה-Mesh To Volume node וה-Volume To Mesh node.

הערה: יש הרבה Switch nodes בשלל הקטגוריות הללו, והם כולם ממלאים אותה פונקציה שראינו [בפרק על Follow Food](#).

Coral (מראה האלמוג):

כדי להגיע ל-nodes המממשים את שינוי מראה האלמוג (בעיקר צבעים וגודל נקבוביות), נלך בסרגל למעלה ל-Shading, נוודא שלשונית ה-Coral נבחרה, ואז נלחץ על הכפתור המסומן בכחול:



הסבר של ה-nodes: ה-nodes הללו יותר פשטניים, ועוסקים באסתטיקה יותר מאשר בלוגיקה.

את Bump Scale, או צפיפות הנקבוביות, אנו מממשים על ידי שימוש בדיאגרמת וורוני, בעזרת ה-Voroni Texture node, Color Ramp node, Bump Node.

את Noise Scale, או "מידת העיקול" של הנקבוביות, אנו מממשים על ידי Noise Texture node, שלוקח טקסטורה ומייצר בה עיקולים באופן אקראי.

את Base Color, Secondary Color אנו מקבלים כבר כקלט, ורק צריכים להעביר אותם דרך הפייפליין כדי שבסוף הם ישקפו את השינויים ששאר הסליידרים, כמו Hue, Saturation, Value, מפעילים.

את Random Value, או השוני בבהירות בין האלמוגים השונים, אנו מממשים על ידי Object Info ו-Color Ramp node, שמאפשר לנו להוריד/להנמיך את הבהירות.

את Hue, Saturation, Value, או גוון, עוצמת ובהירות האלמוג, אנו מממשים על ידי Hue/Saturation/Value node.

את Normal Strength, או העומק של הנקבוביות, אנו מממשים על ידי Bump node.