

דו"ח מיני פרויקט בהנדסת תוכנה 2023



מגישות: לאה גולובנצ'יק ונעמי רייצר

הקדמה:

במיני פרויקט זה המטרה הייתה לייצר תמונה בצורה כמה שיותר דומה לאופן שבה העין שלנו רואה.

התמונה שלנו נבנתה בתהליכים רבים. נעזרנו הרבה בתוכנה geogebra בשביל למקם את האובייקטים, המצלמה ומקורות התאורה.

הבניה הייתה עם הרבה ניסוי ותהיה עד שהגענו לתמונה הרצויה.

בניה של התמונה ללא שיפורים:

בעזרת התוכנה geogebra מיקמנו את ההרים והשלגים המורכבים ממשולשים, עננים מספרות, ציפורים ממשולשים, צמחים מספרות, שמים, מים, ושני מקורות אור: spot ו-directional



לאחר מכן, הוספנו את השמש (point light) ומיקמנו את המצלמה בזווית שונה כדי שיהיה ניתן לראות את השתקפות ההרים על המים:



שיפור התמונה - glossy surface and diffuse glass

שמנו לב לכך שההשתקפות על המים מושלמת מידי ולא כמו שהיא במציאות – מעומעמת יותר. הוספנו לחישוב של שקיפות והשתקפות קוד שבנה רשת והצבע היה הממוצע בין הצבעים של מספר קרני שקיפות או השתקפות סביב הנקודה המקורית ברשת.

```
int num = 1;

Color ans = calcColor(gp, ray, level: level - 1, kxx); // Calc the color in the specific point
double gap = (double) ACCURACY / RAYS; // The gap between each point in the grid
Vector v = ray.getDirection();
// Vector that orthogonal to the ray
// in order to create a plane that include the ray's point and find more points in the area
Vector dir1 = new Vector(new Double3(-1, -1, ((v.getX() + v.getY()) / v.getZ()))).normalize();
Vector dir2 = ray.getDirection().crossProduct(dir1).normalize();
Point startGrid = gp.point;
Point temp;
Color c;

// Create the grid on the target
// Go over the points in the area of the original point and calc their color
for (int i = 0; i < RAYS / 2d; i++) {
    for (int j = 0; j < RAYS / 2d; j++) {
        if (j == 0 && i != 0) { // If the point is on dir1 - there are 2 points for every +-i
            temp = startGrid.add(dir1.scale( scalar: -i * gap));
            c = improvementHelper(startGrid, temp, ray, level, kxx);
            if (c != Color.BLACK) {
                ans = ans.add(c);
                temp = startGrid.add(dir1.scale( scalar: i * gap));
                ans.add(improvementHelper(startGrid, temp, ray, level, kxx));
                num = num + 2;
            }
        } else if (i == 0 && j != 0) { // If the point is on dir2 - there are 2 points for every +-j
            temp = startGrid.add(dir2.scale( scalar: -j * gap));
            c = improvementHelper(startGrid, temp, ray, level, kxx);
            if (c != Color.BLACK) {
                ans = ans.add(c);
                temp = startGrid.add(dir2.scale( scalar: j * gap));
                ans.add(improvementHelper(startGrid, temp, ray, level, kxx));
                num = num + 2;
            }
        } else if (j != 0 && i != 0) { // Else there are 4 points for every +-i, +-j
            temp = startGrid.add(dir1.scale( scalar: i * gap)).add(dir2.scale( scalar: j * gap));
            c = improvementHelper(startGrid, temp, ray, level, kxx);
            if (c != Color.BLACK) {
                ans = ans.add(c);
                temp = startGrid.add(dir1.scale( scalar: -i * gap)).add(dir2.scale( scalar: j * gap));
                ans = ans.add(improvementHelper(startGrid, temp, ray, level, kxx));
                temp = startGrid.add(dir1.scale( scalar: i * gap)).add(dir2.scale( scalar: -j * gap));
                ans = ans.add(improvementHelper(startGrid, temp, ray, level, kxx));
                temp = startGrid.add(dir1.scale( scalar: -i * gap)).add(dir2.scale( scalar: -j * gap));
                ans = ans.add(improvementHelper(startGrid, temp, ray, level, kxx));
                num = num + 4;
            }
        }
    }
}

return ans.scale(kx).reduce(num); // Return the average color of all the rays in the grid
```

לאחר הוספת השיפורים ניתן לראות שההשתקפות על המים מעומעמת יותר
וכתוצאה מכך מציאותית יותר, ובזאת השלמנו את התמונה שלנו!



קרדיט ליעל גלוברמן על קטעי קוד מסויימים