בס"ד
טליה שמואליאן 211378658 אודליה בן ארי 315439554

דו"ח פרויקט חלק 2

הסבר על שיפורי הזמן ריצה

תהליכונים:

שינינו את מחלקת render שתעבוד עם מספר תהליכונים, וכל תהליכון יחשב צבע של פיקסל אחר.

הוספנו מחלקה פנימית שתייצג את כל הפיקסלים בצורה לינארית כדי שיהיה ניתן לפצל את הפיקסלים בין התהליכונים השונים.

BVH:

שלב 1 נציין שניתן לכבות ולהדליק את השיפור על ידי משתנה בוליאני סטטי שנמצא במחלקת Intersectable .

אנחנו עשינו את השיפור שמתבסס על הרעיון של BVH-boundary volume hierarchy.

הרעיון הוא שבמקום לחפש את נקודות החיתוך בין קרן לכל הגיאומטריות בסצנה, נתחום את הגאומטריות בתוך קופסאות, ונאחד קופסאות קרובות, והבדיקה הראשונית של הקרן תהיה חיתוך עם הקופסא העוטפת. החיסכון הוא שאם אין בכלל חיתוך בין הקרן לקופסא, אין צורך לחפש נקודות חיתוך עם כל הצורות שבתוכה.

את הקופסא העוטפת ייצגנו באמצעות גבולות קופסא שמקבילים לצירים- AABB-axis aligned bounded box.

```java
public class AABB {
    /**
     * the minimum point in the box
     */
    final Point3D _min;
    /**
     * the maximum point in the box
     */
    final Point3D _max;
    /**
     * constructor
     *
     * @param min the minimum point in the box
     * @param max the maximum point in the box
     */
    public AABB(Point3D min, Point3D max) {
        _min = min;
        _max = max;
    }
```

כפי שניתן לראות בקוד, את גבולות הקופסא שמרנו באמצעות נקודת מינימום ומקסימום.

בס"ד
טליה שמואליאן 211378658 אודליה בן ארי 315439554

את הבדיקה האם הקרן חותכת את הקופסא הוספנו לפונקציה שבודקת את הנקודות חיתוך עם קרן ורשימת גאומטריות-במחלקת Geometries. (שורה 62)

```
49
50  /**
51   * this methode calculate the intersections between ray and geometries
52   * and has an information about the geometry.
53   *
54   * @param ray
55   * @param maxDistance the maxDistance between the start of the ray and the geometries
56   * @return list of geoPoint intersections
57   */
58  @Override
59  public List<GeoPoint> findGeoIntersections(Ray ray, double maxDistance) {
60      List<GeoPoint> result = null;
61      for (Intersectable geo : _listOfGeometries) {
62          if (improvementBVHIsOff || geo._box.hit(ray)) {//if the ray don't hit the box,no need to find intersection points
63              List<GeoPoint> geoPoints = geo.findGeoIntersections(ray, maxDistance);
64              if (geoPoints != null) {
65                  if (result == null) {
66                      result = new LinkedList<>();
67                  }
68                  result.addAll(geoPoints);
69              }
70          }
71      }
72      return result;
73  }
```

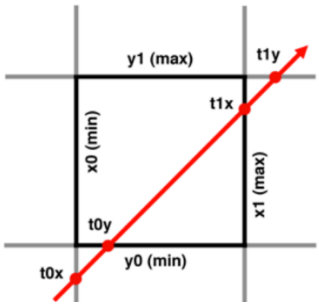
IntelliJ IDEA 2020.3.4 available

בס"ד
טליה שמואליאן 211378658 אודליה בן ארי 315439554

זאת הפונקציה hit שבודקת האם הקרן חותכת את הקופסא.

הפונקציה היא בוליאנית ומחזירה true או false, ולא את נקודת החיתוך עצמה.

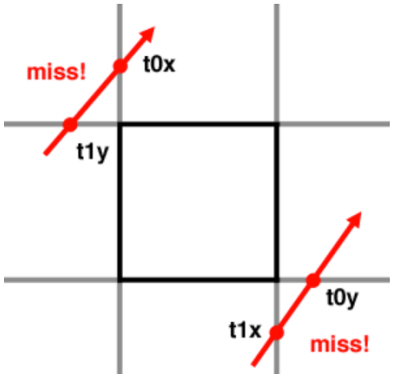
בפונקציה אנו מחשבים את ה t מינימום ומקסימום עבור כל ציר, שהם מבטאים את המרחק בין נקודת תחילת הקרן לקווים המקבילים לגבולות הקופסא כפי שמתואר בציור הבא.




```java
/**
 * this function check if there is intersection between the box and the ray
 * @param ray
 * @return
 */
public boolean hit(Ray ray) {
    Point3D p0 = ray.getP0();
    double p0X = p0.getX();
    double p0Y = p0.getY();
    double p0Z = p0.getZ();
    Vector direction = ray.getDir();
    double dirX = direction.getX();
    double dirY = direction.getY();
    double dirZ = direction.getZ();

    double tmin ;
    double tmax ;

    if (dirX >= 0) {
        tmin = (_min.getX() - p0X) /dirX;
        tmax = (_max.getX() - p0X) / dirX;
    }
    else {//if the directionX is negative, then the min tx in the box is maximal
        tmin = (_max.getX() -p0X) /dirX;
        tmax = (_min.getX() - p0X) /dirX;
    }

    double tymin ;
    double tymax ;
    if (dirY >= 0) {
        tymin = (_min.getY() - p0Y) / dirY;
        tymax = (_max.getY() - p0Y) / dirY;
    }
    else {//if the directionY is negative, then the min tY in the box is maximal
        tymin = (_max.getY() - p0Y) / dirY;
        tymax = (_min.getY() - p0Y) / dirY;
    }
```

בס"ד

טליה שמואליאן 211378658 אודליה בן ארי 315439554



במקרה והקרן באה מהכיוונים שמצוירים באדום אז הקרן לא חותכת את הקופסא.

זה קורה אם tminX גדול מ tmaxY או אם tmaxX קטן tminY

```
//        |         |
//miss!!  *t0x      |
//     *  |         |
//--*t1y--|---------|-------------
//        |         |
//        |         |
//--------|---------|---*t0y----------
//        |         | *   miss!!
//        |         *t1x
//        |         |
if ((tmin > tymax) || (tymin > tmax))
    return false;
```

חוזרים על אותן הפעולות גם עבור ציר Z.

```
    if (tymin > tmin)
        tmin = tymin;

    if (tymax < tmax)
        tmax = tymax;

    double tzmin ;
    double tzmax ;

    if (dirZ >= 0) {
        tzmin = (_min.getZ() - p0Z) / dirZ;
        tzmax = (_max.getZ() - p0Z) / dirZ;
    }
    else {//if the directionZ is negative, then the min tZ in the box is maximal
        tzmin = (_max.getZ()- p0Z) /dirZ;
        tzmax = (_min.getZ() - p0Z) / dirZ;
    }

    if ((tmin > tzmax) || (tzmin > tmax))
        return false;

    return true;
}
```

בס"ד
טליה שמואליאן 211378658 אודליה בן ארי 315439554

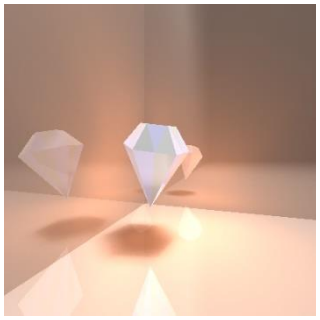
שלב 2

בנינו את עץ ההיררכיה בצורה ידנית

```
Geometries geo3=new Geometries(geo1,geo2);
geometries.add(geo3);
```

הזמן שלוקח להריץ את התמונה לפני השיפור של התהליכונים וBVH הוא: 32 דקות

הזמן שלוקח עם השיפורים הוא 4 דקות.



שלב 3

בניית העץ ההיררכי בצורה אוטומטית.

הוספנו במחלקת Geometries את הפונקציה createBVHTree

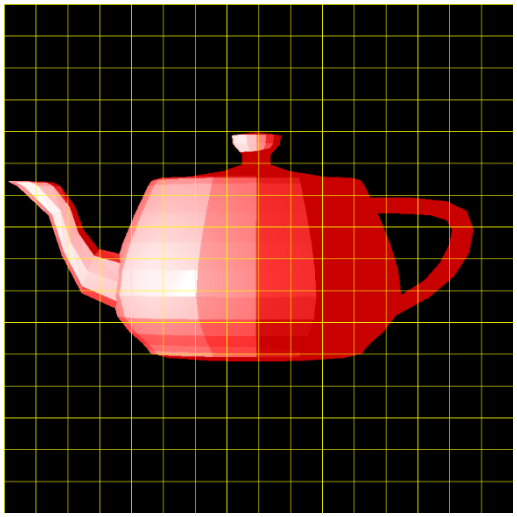
שעוברת על רשימת הגאומטריות ומעבירה זמנית את הצורות האין סופיות לרשימה אחרת.

אח"כ היא עוברת על הרשימה שנשארה ומאחדת כל פעם בין הקופסא של האיבר הראשון ברשימה, לבין הקופסא הקרובה אליו ביותר, עד שנשאר רק איבר אחד ברשימה, ואז מוסיפים את הצורות האינסופיות לרשימה.

```java
 * create BVH tree from the geometries
 */
public void createBVHTree() {
    //remove all the infinity geometries from  _listOfGeometries, and add them in the end
    List<Intersectable> planeList = new LinkedList<>();
    for (Intersectable geo : _listOfGeometries) {
        if (geo instanceof Plane) {
            planeList.add(geo);
        }
    }
    _listOfGeometries.removeAll(planeList);

    //seek every time the closest boxes, and union them into a big box
    //repeat until there is one box left
    double distance = 0;
    Intersectable son1 = null;
    Intersectable son2 = null;
    while (_listOfGeometries.size() > 1) {//
        double minDistance = Double.POSITIVE_INFINITY;
        Intersectable geo1 = _listOfGeometries.get(0);
            for (Intersectable geo2 : _listOfGeometries) {
                //find the minimum distance between 2 boxes
                if (geo1 != geo2 && (distance = distance(geo1, geo2)) < minDistance) {
                    minDistance = distance;
                    son1 = geo1;
                    son2 = geo2;
                }
            }

        Geometries tempGeometries = new Geometries(son1, son2);//union the two closest geometries
        _listOfGeometries.remove(son1);
        _listOfGeometries.remove(son2);
        _listOfGeometries.add(tempGeometries);

    }

    _listOfGeometries.addAll(planeList);//add the infinity geometries to the list
```

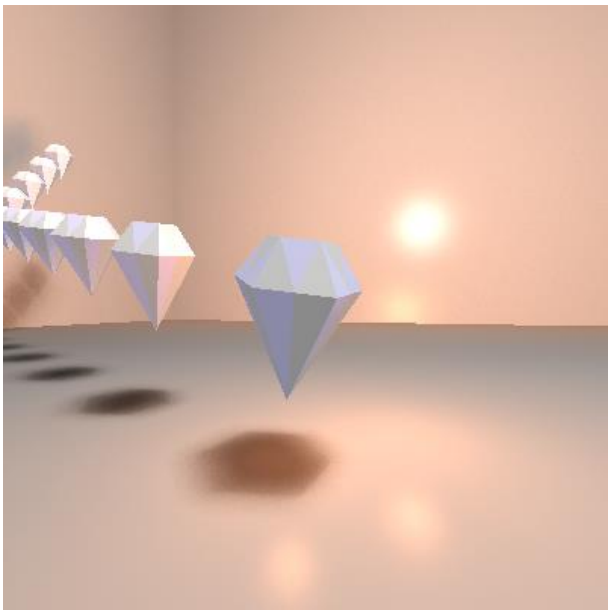
בס"ד

טליה שמואליאן 211378658 אודליה בן ארי 315439554



הרצת הטסט של הקומקום ללא שום שיפור לקחה 4 דקות,

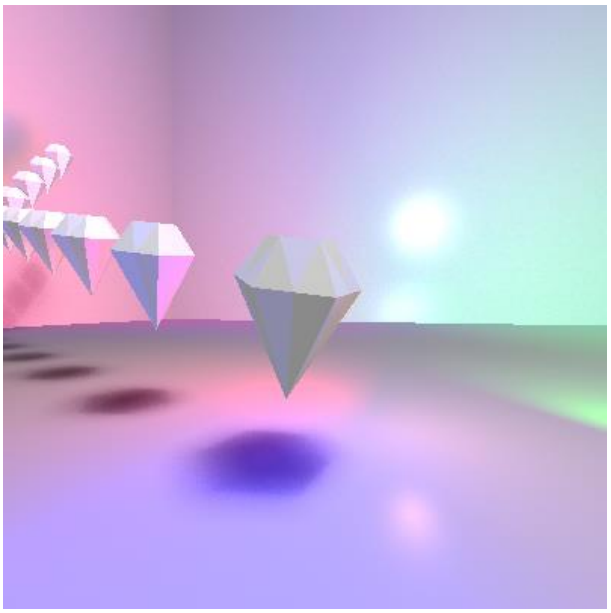
ולאחר תוספת השיפורים ההרצה לקחה 3.5 שניות.



בתמונה הבאה יש 333 גופים ו5 גופי תאורה, בלי שיפורי הזמן

ריצה זה לקח שעתיים ו5 דקות.

עם תוספת של שיפורים זה לקח 4.5 דקות.



בתמונה הבאה יש 333 גופים ו8 גופי תאורה, בלי שיפורי

הזמן ריצה זה לקח 6 שעות ו40דקות.

עם תוספת של שיפורים זה לקח 17 דקות.

בס"ד
טליה שמואליאן 211378658 אודליה בן ארי 315439554

בונוסים:

5 נקודות על הוספת שיפור נוסף (soft shadow).

1 נק' על תמונה בתרגיל 7 שכוללת יותר מ10 גופים.

1 נק' על הגשת תרגיל 6 תוך שבוע.

1 נק' על פיתרון בעית מרחק מהצללה בתרגיל 7 בדרך 2 .