|  |  |
| --- | --- |
| Datei mit Funktion | Funktion |
| ReferencepointsScript | float[] CopyPlaneAndMoveAlongNormal(Vector3 normal, Vector3 pointOnOriginalPlane, int factor) |
|  | Vector3 DetermineDirectionVectorForSymmetryLine(UpperJawPlaneData upperJawPlaneData, Vector3 direction) |
|  | Vector3 OrthogonalProjectionPointOnPlane(UpperJawPlaneData upperJawPlaneData, Vector3 point) |
|  | float DistancePointToStraightLine(Vector3 point, Vector3 localVec, Vector3 distanceVec) |
| DentalArchMarginManager | float EuclideanDistance(float[] p, float[] q) |
| OcclusionSurfaceHelper | Vector3 DetermineIntersectionForLineAndPlane(float[] planeCoords, Vector3 localVec, Vector3 dirVecPlus,  Vector3 dirVecMinus) |
|  | Float[] ConvertParamTooCoordForm(Vector3 localVec, Vector3 dirVecOne, Vector3 dirVecTwo) |
| OcclusionSurfaceManager | double normalizedVector(double[] parameters) |
|  | double DistancePointToPlane(double[] parameters, float[] point) |
|  | double ConvertNormalFormToHesseForm(double[] parameters) |
| UpperJawPlaneHelper | double CalculateMeanValue(double[] values) |
|  | double[] CalculateCenterOfGravity(Vector3[] vertices) *(Eigentlich eher für eine physikalische Lib)* |
|  | double[] PolarCoordinates(Vector3 coord, float pi) |
|  | double[,] CovarianceMatrix(Vector3[] values) |
|  | double CalculateVariance(double[] allValues, double meanValue) |
|  | double CalculateCovariance(double[] firstValues, double[] secondValues, double firstMean, double secondMean) |
|  | Matrix<double> DetermineEigenvectors(double[,] inertiaMatrixJ) |
|  | Vector<Complex> DetermineEigenValues(double[,] inertiaMatrixJ) |
|  | CheckDotProduct(double[] first, double[] second) |

**RED: DONE**