



POLITECHNIKA WROCŁAWSKA
Instytut Informatyki, Automatyki i Robotyki
Zakład Systemów Komputerowych

**Grafika komputerowa i komunikacja
człowiek - komputer**

Kurs: INEK00012L

Sprawozdanie z ćwiczenia nr 5

TEMAT ĆWICZENIA
OpenGL – oświetlanie scen 3-D

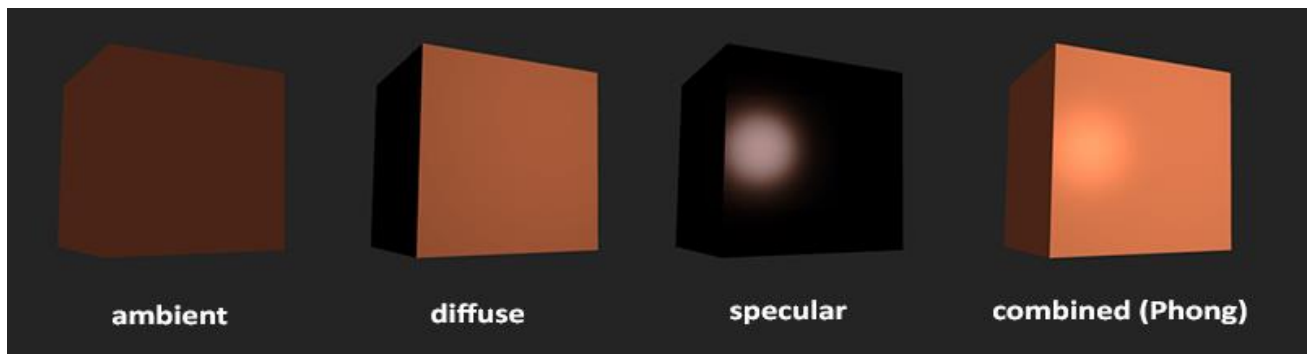
Wykonał:	Karol Pastewski 252798
Termin:	WT/TP 7.30-10.30
Data wykonania ćwiczenia:	23.11.2021r.
Data oddania sprawozdania:	30.11.2021r.
Ocena:	

Uwagi prowadzącego:

1. Wstęp teoretyczny

1.1. Model Phong

Oświetlenie w prawdziwym życiu jest skomplikowane i zależne od wielu czynników, a obliczenie ich jest zbyt czasochłonne oraz zbyt zasobożerne, dlatego w OpenGL można zastosować modele oświetlenia, które odwzorowują w przybliżeniu prawdziwe oświetlenie. Jednym z takich modeli jest model Phong. Główne elementy modelu Phong to światło zwierciadlane (ang. specular), rozproszone (ang. diffuse) oraz otoczenia (ang. ambient). Poniżej rysunek prezentujący jak te elementy wyglądają na obiekcie pojedynczo oraz w połączeniu:



Rysunek 1 Elementy modelu Phong

Opis szczegółowy elementów modelu Phong:

- światło otoczenia – obiekty prawie nigdy na świecie nie są całkowicie ciemne, bo gdzieś nadal jest jakieś światło (księżyc, odległe światło). Aby to zasymulować, używamy stałej oświetlenia otoczenia, które zawsze nadaje obiektowi jakiś kolor.
- światło rozproszone – symuluje oddziaływanie kierunkowe światła na obiekt. Objawia się to tym na przykład, że im bliżej obiekt jest źródła światła, tym staje się on jaśniejszy.
- światło zwierciadlane – symuluje jasny punkt, który się pojawia na błyszczących obiektach. Światło zwierciadlane jest bardziej nastawione na kolor światła niż kolor obiektu.

1.2. Wektor normalny

Aby móc użyć modelu Phong'a do oświetlenia obiektu, nasz obiekt musi posiadać informacje nie tylko o swoim położeniu w przestrzeni, ale także o geometrii powierzchni. Potrzebne nam są wektory normalne do powierzchni dla każdego punktu, z którego złożony jest obiekt. Obiekt czajnika ma obliczone te wartości, więc musimy je obliczyć dla jajka. Model jajka jest budowany na podstawie równań parametrycznych w postaci:

$$\begin{aligned}x(u, v) &= (-90u^5 + 225u^4 - 270u^3 + 180u^2 - 45u) \cos(\pi v) \\y(u, v) &= 160u^4 - 320u^3 + 160u^2 \\z(u, v) &= (-90u^5 + 225u^4 - 270u^3 + 180u^2 - 45u) \sin(\pi v) \\0 &\leq u \leq 1 \\0 &\leq v \leq 1\end{aligned}$$

Wektor normalny do punktu leżącego na takiej powierzchni można znaleźć dzięki temu wzorowi:

$$\begin{aligned}N(u, v) &= \begin{bmatrix} y_u & z_u \\ y_v & z_v \end{bmatrix} \cdot \begin{bmatrix} z_u & x_u \\ z_v & x_v \end{bmatrix} \cdot \begin{bmatrix} x_u & y_u \\ x_v & y_v \end{bmatrix} = \\&= [y_u \cdot z_v - z_u \cdot y_v, \quad z_u \cdot x_v - x_u \cdot z_v, \quad x_u \cdot y_v - y_u \cdot x_v] \neq 0\end{aligned}$$

przy czym:

$$\begin{aligned}x_u &= \frac{\partial x(u, v)}{\partial u}, & x_v &= \frac{\partial x(u, v)}{\partial v} \\y_u &= \frac{\partial y(u, v)}{\partial u}, & y_v &= \frac{\partial y(u, v)}{\partial v} \\z_u &= \frac{\partial z(u, v)}{\partial u}, & z_v &= \frac{\partial z(u, v)}{\partial v}\end{aligned}$$

Po różniczkowaniu otrzymuje się wzory pozwalające na obliczenie wektora normalnego:

$$\begin{aligned}x_u &= \frac{\partial x(u, v)}{\partial u} = (-450u^4 + 900u^3 - 810u^2 + 360u - 45) \cdot \cos(\pi v) \\x_v &= \frac{\partial x(u, v)}{\partial v} = \pi \cdot (-90u^5 + 225u^4 - 270u^3 + 180u^2 - 45u) \cdot \sin(\pi v) \\y_u &= \frac{\partial y(u, v)}{\partial u} = 640u^3 - 960u^2 + 320u \\y_v &= \frac{\partial y(u, v)}{\partial v} = 0 \\z_u &= \frac{\partial z(u, v)}{\partial u} = (-450u^4 + 900u^3 - 810u^2 + 360u - 45) \cdot \sin(\pi v) \\z_v &= \frac{\partial z(u, v)}{\partial v} = -\pi \cdot (-90u^5 + 225u^4 - 270u^3 + 180u^2 - 45u) \cdot \cos(\pi v)\end{aligned}$$

Po wyliczeniu wektorów normalnych, z podanych na poprzedniej stronie wzorów, należy je znormalizować, czyli doprowadzić do sytuacji, gdzie długość wektorów będzie wynosił 1. Normalizacja polega na podzieleniu każdej składowej wektora przez jej długość, a długość jest pierwiastkiem z sumy kwadratów jego składowych.

2. Nowe polecenia OpenGL

- **glMaterialfv**(GLenum face,
GLenum pname,
const GLfloat *params));

Funkcja określa parametry materiału obiektu dla modelu oświetlenia. Parametr face oznacza powierzchnię (jedną lub wiele), która jest określana (możliwe są GL_FRONT albo/i GL_BACK). Zmienna pname jest nazwą parametru, który ma być zaktualizowany. Używane w programie parametry oraz ich opis:

GL_SPECULAR – składowe RGBA określające stopień odbicia światła odbitego;

GL_AMBIENT – składowe RGBA określające stopień odbicia światła otaczającego;

GL_DIFFUSE – składowe RGBA określające stopień odbicia światła rozproszonego;

GL_SHININESS – stała z przedziału [0, 128] regulująca stopień występowania efektu rozbłysku obiektu;

Zmienna params oznacza wartość zmienianego parametru.

- **glLightfv**(GLenum light,
GLenum pname,
const GLfloat *params));

Funkcja określa parametry materiału obiektu dla modelu oświetlenia. Parametr light oznacza identyfikator źródła światła (nazwa źródła światła to GL_LIGHT*i*, gdzie *i* może być wartością od 0 do GL_MAX_LIGHTS – 1 (w tym programie do 7)).

Zmienna pname jest nazwą parametru, który ma być zaktualizowany. Używane w programie parametry oraz ich opis:

GL_SPECULAR – wartości składowych RGBA światła odbitego;

GL_AMBIENT – wartości składowych RGBA światła otaczającego;

GL_DIFFUSE – wartości składowych RGBA światła rozproszonego;

GL_CONSTANT_ATTENUATION – stały współczynnik tłumienia światła;

GL_LINEAR_ATTENUATION – liniowy współczynnik tłumienia światła;

GL_QUADRATIC_ATTENUATION – kwadratowy współczynnik tłumienia światła;

Zmienna params oznacza wartość zmienianego parametru.

- **glShadeModel**(GLenum mode);

Polecenie służy do wybrania, albo płaskiego (GL_FLAT), albo gładkiego (GL_SMOOTH) modelu cieniowania. Kiedy zostanie wybrany model płaski, to dla każdego prymitywu (w tym programie dla każdego trójkąta) zostanie obliczony i wyznaczony jeden kolor, a w przypadku modelu gładkiego każdy piksel prymitywu będzie miał obliczony kolor.

3. Rozwiązanie zadania

```
265 x_u = (-450 * uPow4 + 900 * uPow3 - 810 * uPow2 + 360 * u - 45) * cos(M_PI * v);
266 x_v = M_PI * (90 * uPow5 - 225 * uPow4 + 270 * uPow3 - 180 * uPow2 + 45) * sin(M_PI * v);
267 y_u = 640 * uPow3 - 960 * uPow2 + 320 * u;
268 y_v = 0;
269 z_u = (-450 * uPow4 + 900 * uPow3 - 810 * uPow2 + 360 * u - 45) * sin(M_PI * v);
270 z_v = -M_PI * (90 * uPow5 - 225 * uPow4 + 270 * uPow3 - 180 * uPow2 + 45) * cos(M_PI * v);
271
272 vector_x = y_u * z_v - z_u * y_v;
273 vector_y = z_u * x_v - x_u * z_v;
274 vector_z = x_u * y_v - y_u * x_v;
275
276 float vector_len = sqrt(pow(vector_x, 2) + pow(vector_y, 2) + pow(vector_z, 2));
277 if (vector_len == 0) {
278     vector_len = 1;
279 }
280
281 vector_x = vector_x / vector_len;
282 vector_y = vector_y / vector_len;
283 vector_z = vector_z / vector_len;
```

W liniach 265-274 obliczane są składowe wektorów według wzorów podanych w punkcie 1.2 sprawozdania. W linii 277 utworzone jest zabezpieczenie przed dzieleniem składowych przez 0. Linie 281-283 odpowiadają za normalizację składowych wektorów

```
412 /*****
413 // Definicja materiału, z jakiego zrobiony jest obiekt
414
415 GLfloat mat_ambient[] = {0.2, 0.2, 0.2, 0.2};
416 // współczynniki ka =[kar,kag,kab] dla światła otoczenia
417
418 GLfloat mat_diffuse[] = {1.0, 1.0, 1.0, 1.0};
419 // współczynniki kd =[kdr,kdg,kdb] światła rozproszonego
420
421 GLfloat mat_specular[] = {1.0, 1.0, 1.0, 1.0};
422 // współczynniki ks =[ksr,ksg,ksb] dla światła odbitego
423
424 GLfloat mat_shininess = {20.0};
425 // współczynnik n opisujący połysk powierzchni
426
427 /*****
428 // Ustawienie parametrów materiału
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Definicje zmiennych opisujących materiał, z jakiego zrobiony jest obiekt, a następnie ustawienie odpowiednich parametrów materiału zdefiniowanymi wartościami. Szczegółowy opis każdego parametru znajduje się przy opisie funkcji `glMaterialfv()`.

```

427 /*****
428 // Definicja stałych odpowiadających za osłabienie światła dla obu źródeł
429
430     GLfloat att_constant = {1.0};
431     // składowa stała ds dla modelu zmian oświetlenia w funkcji
432     // odległości od źródła
433
434     GLfloat att_linear = {0.05};
435     // składowa liniowa dl dla modelu zmian oświetlenia w funkcji
436     // odległości od źródła
437
438     GLfloat att_quadratic = {0.001};
439     // składowa kwadratowa dq dla modelu zmian oświetlenia w funkcji
440     // odległości od źródła
441
442 /*****
443 // Definicja pierwszego źródła światła
444
445     GLfloat light_ambient1[] = {0.1, 0.0, 0.0, 1.0};
446     // składowe intensywności świecenia źródła światła otoczenia
447     // Ia = [Iar,Iag,Iab]
448
449     GLfloat light_diffuse1[] = {1.0, 0.0, 0.0, 1.0};
450     // składowe intensywności świecenia źródła światła powodującego
451     // odbicie dyfuzyjne Id = [Idr,Idg,Idb]
452
453     GLfloat light_specular1[] = {0.85, 0.85, 0.85, 0.85};
454     // składowe intensywności świecenia źródła światła powodującego
455     // odbicie kierunkowe Is = [Isr,Isg,Isb]
456
457 /*****
458 // Definicja drugiego źródła światła
459
460     GLfloat light_ambient2[] = {0.0, 0.0, 0.1, 1.0};
461     // składowe intensywności świecenia źródła światła otoczenia
462     // Ia = [Iar,Iag,Iab]
463
464     GLfloat light_diffuse2[] = {0.0, 0.0, 1.0, 1.0};
465     // składowe intensywności świecenia źródła światła powodującego
466     // odbicie dyfuzyjne Id = [Idr,Idg,Idb]
467
468     GLfloat light_specular2[] = {0.85, 0.85, 0.85, 0.85};
469     // składowe intensywności świecenia źródła światła powodującego
470     // odbicie kierunkowe Is = [Isr,Isg,Isb]
471

```

Definicje stałych dla źródeł światła oraz definicje źródeł światła. Szczegółowy opis każdego parametru znajduje się przy opisie funkcji `glLightfv()`.

```

482  /*****
483  // Ustawienie parametrów pierwszego źródła światła
484
485      glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient1);
486      glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse1);
487      glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular1);
488
489      glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, att_constant);
490      glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, att_linear);
491      glLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, att_quadratic);
492
493  /*****
494  // Ustawienie parametrów pierwszego źródła światła
495
496      glLightfv(GL_LIGHT1, GL_AMBIENT, light_ambient2);
497      glLightfv(GL_LIGHT1, GL_DIFFUSE, light_diffuse2);
498      glLightfv(GL_LIGHT1, GL_SPECULAR, light_specular2);
499
500      glLightf(GL_LIGHT1, GL_CONSTANT_ATTENUATION, att_constant);
501      glLightf(GL_LIGHT1, GL_LINEAR_ATTENUATION, att_linear);
502      glLightf(GL_LIGHT1, GL_QUADRATIC_ATTENUATION, att_quadratic);
503
504

```

Ustawienie parametrów źródeł światła. Pierwsze źródło światła jest identyfikowane przez GL_LIGHT0, a drugie przez GL_LIGHT1.

```

505  /*****
506  // Ustawienie opcji systemu oświetlania sceny
507
508      glShadeModel(GL_SMOOTH); // włączenie łagodnego cieniowania
509      glEnable(GL_LIGHTING);    // włączenie systemu oświetlania sceny
510      glEnable(GL_LIGHT0);     // włączenie źródła o numerze 0
511      glEnable(GL_LIGHT1);     // włączenie źródła o numerze 1
512      glEnable(GL_DEPTH_TEST); // włączenie mechanizmu z-bufora

```

Ustawienie oświetlenia sceny. W linii 512 aktywujemy testowanie głębi, które ma nam zapobiegać sytuacjom, w których fragmenty obiektu, które nie powinny być widoczne, są rysowane na ekranie.