



Transformacje
wierzchołków

Kamera

Współrzędne
biegunowe

Obsługa
klawiatury

Diagram klas

Zadanie

Gry Komputerowe

Interaktywna kamera FPP

Michał Chwesiuk

Zachodniopomorski Uniwersytet Technologiczny w Szczecinie
Wydział Informatyki

28 Marzec 2018



Kamera

Transformacje wierzchołków

$$\begin{bmatrix} v'_x \\ v'_y \\ v'_z \\ v'_w \end{bmatrix} = \begin{bmatrix} Projection \\ matrix \end{bmatrix} \times \begin{bmatrix} View \\ matrix \end{bmatrix} \times \begin{bmatrix} Model \\ matrix \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \\ v_z \\ v_w \end{bmatrix}$$

Transformacje w OpenGL

$$\begin{bmatrix} v'_x \\ v'_y \\ v'_z \\ v'_w \end{bmatrix} = [GL_PROJECTION] \times [GL_MODELVIEW] \times \begin{bmatrix} v_x \\ v_y \\ v_z \\ v_w \end{bmatrix}$$

Transformacje
wierzchołków

Kamera

Współrzędne
biegunowe

Obsługa
klawiatury

Diagram klas

Zadanie



Kamera

Transformacje
wierzchołków

Kamera

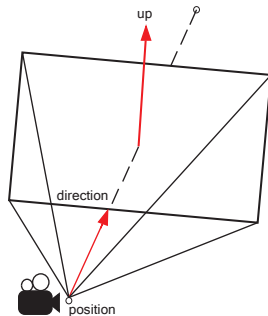
Współrzędne
biegunowe

Obsługa
klawiatury

Diagram klas

Zadanie

- Atrybuty kamery :
 - Pozycja (**position**)
 - Punkt patrzenia (**target**)
 - Wektor góry (**up**)





Macierz transformacji widoku

Macierz transformacji widoku

$gluLookAt(eye_x, eye_y, eye_z, target_x, target_y, target_z, up_x, up_y, up_z)$

$$\begin{bmatrix} left_x & left_y & left_z & -left_x eye_x - left_y eye_y - left_z eye_z \\ up_x & up_y & up_z & -up_x eye_x - up_y eye_y - up_z eye_z \\ f_x & f_y & f_z & -f_x eye_x - f_y eye_y - f_z eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Gdzie :

- $\vec{f} = |e\vec{y}e - tar\vec{get}|$, $left = |\vec{u}\vec{p} \times \vec{f}|$
- $e\vec{y}e$ - pozycja kamery
- $tar\vec{get}$ - punkt patrzenia kamery ($e\vec{y}e + \vec{dir}$)
- $\vec{u}\vec{p}$ - wektor góry kamery

Transformacje
wierzchołków

Kamera

Współrzędne
biegunowe

Obsługa
klawiatury

Diagram klas

Zadanie



Poruszanie kamerą

Transformacje
wierzchołków

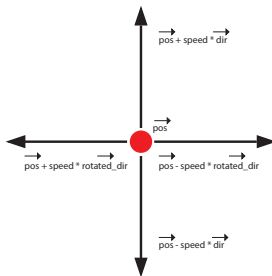
Kamera

Współrzędne
biegunowe

Obsługa
klawiatury

Diagram klas

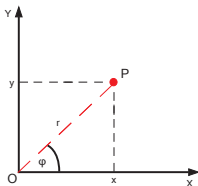
Zadanie



- \vec{pos} - pozycja kamery
- \vec{dir} - kierunek patrzenia kamery
- $speed$ - prędkość przemieszczania kamery
- $\vec{rotated_dir}$ - wektor prostopadły do kierunku patrzenia kamery



Współrzędne biegunowe



- Najczęściej podczas opisywania punktu w przestrzeni dwuwymiarowej wykorzystujemy współrzędne kartezjańskie, które opisują punkt za pomocą dwóch zmiennych : X i Y . Współrzędne biegunowe polegają na opisanu tych współrzędnych za pomocą dwóch innych atrybutów :
 - promień wodzący r - odległość punktu P od środka układu współrzędnych O
 - amplituda punktu ϕ - kąt pomiędzy osią OX , a wektorem \vec{OP}



Współrzędne biegunowe

Transformacje punktów do układów

- Z układu kartezjańskiego do układu biegunowego

$$r = \sqrt{X^2 + Y^2}$$

$$\phi = \arctan\left(\frac{Y}{X}\right)$$

- Z układu biegunowego do układu kartezjańskiego

$$X = r * \cos(\phi)$$

$$Y = r * \sin(\phi)$$

Transformacje
wierzchołków

Kamera

Współrzędne
biegunowe

Obsługa
klawiatury

Diagram klas

Zadanie



Obsługa klawiatury

Transformacje
wierzchołków

Kamera

Współrzędne
biegunowe

Obsługa
klawiatury

Diagram klas

Zadanie

- Wejście klawiatury za pomocą GLUT odbywa się za pomocą dwóch callbacków :
 - *glutKeyboardFunc(f)* - wciśnięcie klawisza
 - *glutKeyboardUpFunc(f)* - puszczenie klawisza
- W GLUT występuje zjawisko **repetycji wciśnięcia klawisza!** Podczas jego trzymania odczytywane jest wielokrotnie jego wciśnięcie.
- GLUT pozwala tylko na odczyt wydarzeń wciśnięcia i puszczenia klawisza, a **nie na odczyt jego stanu**. Rozwiązanie tego problemu leży po stronie programisty.
 - Należy stworzyć tablice zmiennych typu bool.
 - Ustawić wartości elementów tablicy na true lub false w odpowiednich funkcjach obsługi klawiatury.
 - Odwoływanie się do elementów tablicy za pomocą wartości kodów ASCII.



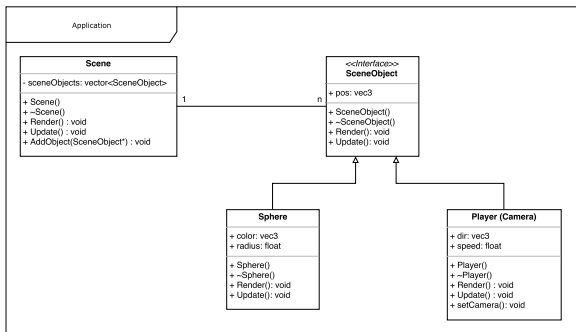
Tablica ASCII

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]



Diagram klas



Należy zwrócić uwagę na rozdzielone metody do
Renderowania i Aktualizowania!



Zadanie

Transformacje
wierzchołków

Kamera

Współrzędne
biegunowe

Obsługa
klawiatury

Diagram klas

Zadanie

Wersja Podstawowa

Zadaniem na dzisiejszych zajęciach jest zaimplementowanie interaktywnej kamery FPP :

- Sterowanie kamerą w czterech kierunkach za pomocą klawiatury do przodu, do tyłu i na boki
- Obracanie kierunku kamery wokół jednej osi za pomocą klawiatury
- Mile widziana bezwładność - zamiast natychmiastowego zatrzymania się, zmniejszanie prędkości w czasie

Wersja Rozszerzona

Do poruszania kamerą należy dodać :

- Obracanie wokół dwóch osi (podpowiedź : Układ współrzędnych sferycznych)
- Sterowanie myszą - przydatne funkcje : *glutWarpPointer(x, y)*, *glutPassiveMotionFunc(f)*, *glutSetCursor(GLUT_CURSOR_NONE)*