

Elementy bufora ramki

W OpenGL w skład bufora ramki (pamięci obrazu) wchodzi następujące elementy:

- bufor **koloru** (ang. color buffer),
- bufor **głębokości**, nazywany także buforem głębi (ang. depth buffer),
- bufor **szablony**, nazywany także buforem szablonu (ang. stencil buffer),
- bufor **akumulacyjny** (ang. accumulation buffer).

Wybór trybu działania buforu jest wykonywany na początku działania programu:

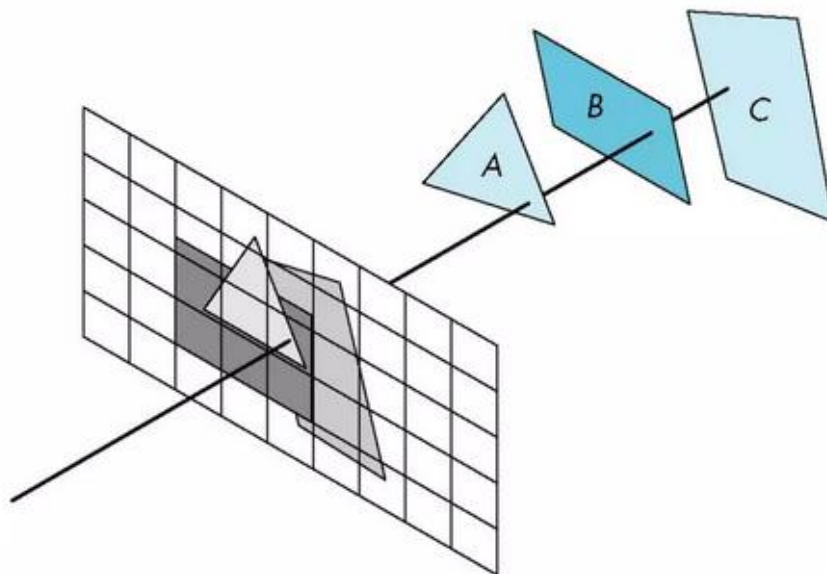
glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);

Czyszczenie buforów w OpenGL:

glClear(argument);

Bufor głębokości

Działanie **bufora głębokości** polega w uproszczeniu na przechowywaniu wartości współrzędnej **z** (głębokości) dla każdego piksela obrazu. Przy rysowaniu kolejnych pikseli danego obiektu obliczana jest wartość współrzędnej **z** i porównywana z wartością znajdującą się w buforze. Jeżeli porównanie wskazuje, że rysowany piksel „przesłania” to, co zostało narysowane wcześniej, zmieniana jest zarówno zawartość bufora głębokości jak i bufora koloru.



Bufor głębokości

- Standardowo bufor ramki nie używa bufora głębokości. Aby program mógł korzystać z z-bufora na etapie inicjalizacji OpenGL należy użyć stałej **GLUT_DEPTH**.
glutInitDisplayMode(GLUT_DOUBLE| GLUT_RGB| GLUT_DEPTH);
- Dołączenie bufora głębokości do bufora ramki nie powoduje automatycznie jego użycia. Testowanie głębokości włączamy ustawiając odpowiednią zmienną stanu:
glEnable(GL_DEPTH_TEST);
- Chwilowe włączanie i wyłączanie wykonywania testów głębokości realizuje funkcja:
glDepthMask(GLboolean flag);
gdzie **flag** może przyjmować wartości **GL_FALSE** i **GL_TRUE**.
- Przed rozpoczęciem rysowania z-bufor należy wyczyścić:
glClear(GL_DEPTH_BUFFER_BIT);
Bufor czyszczony jest wartością 1.

Blending – mieszanie kolorów

OpenGL umożliwia miksowanie renderowanych obiektów z tłem - na poziomie przetwarzania poszczególnych pikseli. Umożliwia to tworzenie efektów szkła, iluminacji, rozmycia itp.



Blending

- ◆ Przetwarzanie przezroczystości materiałów należy włączyć wywołaniem:
glEnable(GL_BLEND);
- ◆ Dodatkowo miksowanie pikseli wymaga zawieszenia rejestrowania obiektów w buforze głębokości:
glDepthMask(GL_FALSE);
- ◆ Funkcja definiująca parametry techniki blending:
glBlendFunc(GLenum źródło, GLenum cel)

Funkcja umożliwia zdefiniowanie mnożników wartości koloru piksela źródła i celu, przy czym cel to już istniejąca zawartość bufora ramki, a źródło to kolor właśnie rysowanego obiektu. Stosowany wzór, produkujący nową wartość piksela w buforze ramki jest następujący:

$$\text{wynik} = \text{kolor_nowego_obiektu} * \text{źródło} + \text{kolor_w_buforze} * \text{cel}$$

```
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
```

Blending

Wartości parametrów funkcji **glBlendFunc**

parametr	współczynnik mieszania
GL_ZERO	$(0, 0, 0, 0)$
GL_ONE	$(1, 1, 1, 1)$
GL_SRC_COLOR	(R_s, G_s, B_s, A_s)
GL_ONE_MINUS_SRC_COLOR	$(1, 1, 1, 1) - (R_s, G_s, B_s, A_s)$
GL_DST_COLOR	(R_d, G_d, B_d, A_d)
GL_ONE_MINUS_DST_COLOR	$(1, 1, 1, 1) - (R_d, G_d, B_d, A_d)$
GL_SRC_ALPHA	(A_s, A_s, A_s, A_s)
GL_ONE_MINUS_SRC_ALPHA	$(1, 1, 1, 1) - (A_s, A_s, A_s, A_s)$
GL_DST_ALPHA	(A_d, A_d, A_d, A_d)
GL_ONE_MINUS_DST_ALPHA	$(1, 1, 1, 1) - (A_d, A_d, A_d, A_d)$
GL_CONSTANT_COLOR	(R_c, G_c, B_c, A_c)
GL_ONE_MINUS_CONSTANT_COLOR	$(1, 1, 1, 1) - (R_c, G_c, B_c, A_c)$
GL_CONSTANT_ALPHA	(A_c, A_c, A_c, A_c)
GL_ONE_MINUS_CONSTANT_ALPHA	$(1, 1, 1, 1) - (A_c, A_c, A_c, A_c)$
GL_SRC_ALPHA_SATURATE	$(f, f, f, 1), f = \min(A_s, 1 - A_d)$

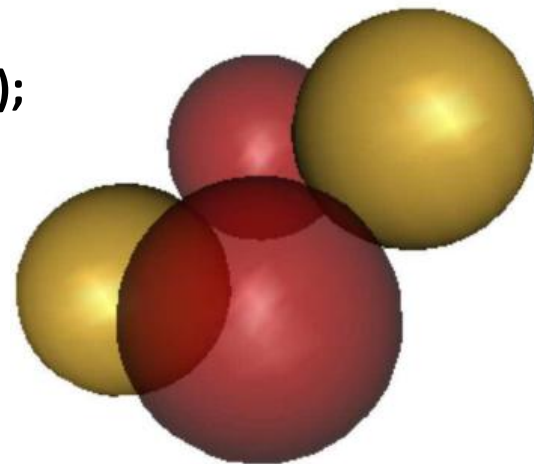
Blending

- ◆ **Kanał alfa** (ang. alpha channel) – dodatkowy kanał, definiujący przezroczystość wyświetlanych informacji graficznych. Jest on zapisywany razem z trzema wartościami barw składowych RGB (ang. red, green, blue – czerwony, zielony i niebieski). Wartość zerowa tego kanału oznacza pełną przezroczystość, zaś maksymalna – całkowitą nieprzezroczystość (czyli barwę jak bez kanału alfa).
- ◆ Przykład – **Fragment Shader**:

```
//przezroczysty zielony  
gl_FragColor = vec4(0.0f, 1.0f, 0.0f, 0.5f);
```

Uzyskanie efektu przezroczystości

- 1) Narysowanie obiektów nieprzezroczystych
- 2) Włączenie mechanizmu mieszania kolorów (blendingu)
glEnable(GL_BLEND);
- 3) Określenie w jaki sposób obiekt, który chcemy narysować ma mieszać się z tłem (tzn tym co już jest na scenie) , realizujemy ten etap za pomocą tzw funkcji mieszającej, przykładowe zastosowanie:
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
- 4) 'Wyłączenie' chwilowe bufora głębokości
glDepthMask(GL_FALSE);
- 5) Narysowanie obiektu, który ma być przezroczysty
- 6) Wyłączenie mechanizmu blendingu
glDisable(GL_BLEND);
- 7) 'Włączenie' ponowne bufora głębokości
glDepthMask(GL_TRUE);
- 8) Dodatkowo należy pamiętać o odpowiednim ustawieniu kanału alfa (to jest czwarta składowa koloru, którą do tej pory przyjmowaliśmy jako 1.0) we fragment shaderze, np.:
gl_FragColor =vec4(1.0,1.0,0.0,0.5);



Przezroczystość tekstur

Jeżeli chcemy uzyskać przezroczystą teksturę należy wcześniej pobrać z niej kolor i dopiero odpowiednio ustawić wartość kanału alfa (we fragment shaderze oczywiście), np.:

```
vec4 blend_text;  
blend_text = texture( tex0, texcoord0 );  
blend_text[3]=0.5;  
gl_FragColor = blend_text;
```

