



!-(~^X >^@-(-

THE ENGINE OF FOURTH DIMENSION

# XENON CORE

version 0.7.5.100

Dokumentacja

*Copyright: © 2007-2008 Xenon Core by PsichiX  
All rights reserved*

Moduły i ich aktualny stan:

• **Fizyka punktowa:**

<Gotowy>

88% | | | | | | | | | |

• **Rendering:**

<W budowie>

20% | | | | | | | | | |

• **Multiplayer:**

<Testy>

46% | | | | | | | | | |

• **Inteligentne Obiekty Fizyczne:**

<W budowie>

10% | | | | | | | | | |

• **Ruchomy czwarty wymiar (czas):**

<W budowie>

5% | | | | | | | | | |

# SPIS TREŚCI

O silniku słów kilka...	3
Funkcje:	
Camera	4
Elements:	5
Black Hole	6
Covalence Force	7
Force	8
Global Force	9
Particle	10
Substance	11
Surface	12
Wave Force	14
Miscellaneous	15
Physic	16
System	17

**Dokumentacja:** *XenoN Core version 0.7.5.100*

**Wersja dokumentu:** *1.0*

**Dokument napisany przez:** *PsichiX*

**Data utworzenia:** *29.04.2008*

**Strona internetowa:** *www.xenon-core.yoyo.pl*

## O SILNIKU SŁÓW KILKA...

XenoN Core to silnik do gier i aplikacji 3D (oraz 2D). W założeniu prosty w obsłudze, prezentujący trójwymiarowy świat i jego zjawiska możliwie jak najrealniej. Silnik ten powstał z mojej inicjatywy, gdyż używając programu Game Maker zapragnąłem, by gry w nim robione były bardziej realistyczne. Skupiłem się więc z początku na fizyce, gdyż w głównej mierze od niej zależeć miała jakość zjawisk panujących w grze/aplikacji.

Silnik w pełnej wersji posiadać ma rendering grafiki w OpenGL (obecnie jest emulowany za pomocą Direct3D w programie Game Maker), szybką obliczeniowo i realistyczną fizykę punktową z dokładnym wyliczeniem kolizji, obsługę trybu Multiplayer do gier sieciowych, system modeli Inteligentnych Obiektów Fizycznych oraz (co chyba najważniejsze) - ruchomy czwarty wymiar, a dokładnie czas. Możliwość przemieszczania się w czasie i dowolna zmiana układów fizycznych na razie pozostanie tajemnicą twórcy, gdyż technologia ta (nazywana przeze mnie "TimeBridge") jest w tej chwili nie zbyt stabilna by mówić o jej działaniu.

Mam nadzieję, że krótko i zwięźle przedstawiłem Tobie zalety silnika XenoN Core i ujrzę go kiedyś w Twoim projekcie ;).

# CAMERA

Kamera - dzięki niej widzimy nasz trójwymiarowy świat w pełnej okazałości. Poniżej opisuję funkcje odpowiedzialne za kamerę.

**xenon\_core\_camera\_set**(*alpha, beta, gamma, x, y, z, length, type*)

Ustawia właściwości kamery. **alpha, beta, gamma** - kąty obrotu kamery. **x, y, z** - pozycja kamery. **length** - długość linii łączącej oko kamery z samą kamerą. **type** - typ kamery. Gdy wartość jest *true*, oko jest zamienione miejscem z kamerą.

**xenon\_core\_camera\_from\_x**()

**xenon\_core\_camera\_from\_y**()

**xenon\_core\_camera\_from\_z**()

Funkcje zwracają pozycję kamery - punkt z którego patrzy oko kamery.

**xenon\_core\_camera\_to\_x**()

**xenon\_core\_camera\_to\_y**()

**xenon\_core\_camera\_to\_z**()

Funkcje zwracają pozycję oka kamery - punkt na które patrzy oko kamery.

**xenon\_core\_camera\_up\_x**()

**xenon\_core\_camera\_up\_y**()

**xenon\_core\_camera\_up\_z**()

Funkcje zwracają wartość up-wektora kamery. Up-wektor określa gdzie jest „góra” kamery.

# ELEMENTS

Fizykę Xenon Core budują elementy takie jak: substancje, powierzchnie, cząsteczki oraz różnego rodzaju siły (z których najważniejszymi są siły kowalencyjne, łączące poszczególne substancje).

- Black Hole - czarne dziury
- Covalence Force - siły kowalencyjne
- Force - siły
- Global Force - siły globalne
- Particle - cząsteczki
- Substance - substancje
- Surface - powierzchnie
- Wave Force - siły o naturze falowej

## **xenon\_core\_element\_activate**(*type, index*)

Aktywuje element *index* danego typu. **type** - typ elementu. Dostępne wartości dla każdego typu: *XE\_BLACK\_HOLE*, *XE\_COVALENCE\_FORCE*, *XE\_FORCE*, *XE\_GLOBAL\_FORCE*, *XE\_PARTICLE*, *XE\_SUBSTANCE*, *XE\_SURFACE*, *XE\_WAVE\_FORCE*. **index** - identyfikator danego elementu.

## **xenon\_core\_element\_deactivate**(*type, index*)

Deaktywuje element *index* danego typu.

Argumenty takie same jak w poprzedniej funkcji.

## **xenon\_core\_element\_export**(*src\_type, dest\_type, src\_index, dest\_index*)

Eksportuje (dokładnie kopiuje) odnośnik elementu w elemencie *src\_index* typu *src\_type* do innego odnośnika w elemencie *dest\_index* typu *dest\_type*. Trochę to niezrozumiałe :p. Tak czy inaczej, należy uważać z tą funkcją, gdyż nadal jest testowana i nie zawsze zachowuje się jak należy.

# BLACK HOLE

Czarne dziury są pewnego rodzaju siłami, służą niejako do wchłaniania substancji. Mogą też „teleportować” je do innych czarnych dziur.

## **xenon\_core\_black\_hole\_create()**

Funkcja tworzy czarną dziurę i zwraca jej identyfikator.

## **xenon\_core\_black\_hole\_destroy(*index*)**

Funkcja niszczy czarną dziurę z danym identyfkatorem *index*.

## **xenon\_core\_black\_hole\_set\_anchor(*index, anchor*)**

Ustala z którą czarną dziurą ta będzie połączona. Jeśli wartość jest różna od *XE\_NULL*, wtedy możliwy będzie teleport substancji. **index** - index czarnej dziury. **anchor** - index czarnej dziury, z którą się połączy.

## **xenon\_core\_black\_hole\_set\_force(*index, force*)**

Ustala siłę danej czarnej dziury. Jeśli wartość jest większa niż 0, możliwy jest teleport substancji. **force** - wartość siły.

## **xenon\_core\_black\_hole\_set\_position(*index, x, y, z*)**

Przenosi czarną dziurę na daną pozycję. **x, y, z** - pozycja czarnej dziury.

## **xenon\_core\_black\_hole\_set\_range(*index, range*)**

Ustala zasięg oddziaływania czarnej dziury. Jeśli wartość jest większa niż 0, możliwy jest teleport substancji. **range** - zasięg działania.

## **xenon\_core\_black\_hole\_get(*index, type*)**

Zwraca wartość zmiennej danej czarnej dziury. **type** - typ zmiennej. Dostępne wartości: *XE\_BLACK\_HOLE\_X, XE\_BLACK\_HOLE\_Y, XE\_BLACK\_HOLE\_Z, XE\_BLACK\_HOLE\_FORCE, XE\_BLACK\_HOLE\_RANGE, XE\_BLACK\_HOLE\_ANCHOR*.

# COVALENCE FORCE

Siły kowalencyjne (międzycząsteczkowe) - najważniejsze z sił w Xenonie. One utrzymują poszczególne substancje między sobą. Doskonałe do ciał szkieletowych (ruchome części).

## **xenon\_core\_covalence\_force\_create()**

Funkcja tworzy siłę kowalencyjną i zwraca jej identyfikator.

## **xenon\_core\_covalence\_force\_destroy(*index*)**

Funkcja niszczy siłę kow. z danym identyfkatorem *index*. **index** - identyfikator siły kow.

## **xenon\_core\_covalence\_force\_set\_force(*index, force*)**

Nadaje sile kow. daną wartość siły. Siła ta jest również miarą odległości między połączonymi substancjami. **force** - wartość siły.

## **xenon\_core\_covalence\_force\_set\_resilience(*index, resilience*)**

Ustala sprężystość siły kow., która pozwala na jej rozciąganie. Domyślnie wartość równa 1. **resilience** - wartość sprężystości.

## **xenon\_core\_covalence\_force\_set\_substance(*index, type, substance*)**

Ustala sile kow. substancję z jaką będzie połączona. Aby siła działa ustawione muszą być oba odnośniki do substancji. **type** - typ substancji. Dostępne wartości: *XE\_COVALENCE\_FORCE\_SUBSTANCE\_1* oraz *XE\_COVALENCE\_FORCE\_SUBSTANCE\_2*. **substance** - index substancji z którą łączymy siłę kow.

## **xenon\_core\_covalence\_force\_get(*index, type*)**

Zwraca wartość zmiennej danej siły kow. **type** - typ zmiennej. Dostępne wartości: *XE\_COVALENCE\_FORCE\_SUBSTANCE\_1*, *XE\_COVALENCE\_FORCE\_SUBSTANCE\_2*, *XE\_COVALENCE\_FORCE\_FORCE*, *XE\_COVALENCE\_FORCE\_RESILIENCE*.

# FORCE

Zwykle siły kierunkowe o zasięgu lokalnym (dokładnie globalnym, lecz widoczne skutki oddziaływań są w obrębie siły). Możliwe jest „przymocowanie” siły do jakiejś substancji. Wtedy siła ta będzie działać tylko i wyłącznie na daną substancję.

## **xenon\_core\_force\_create()**

Funkcja tworzy siłę i zwraca jej identyfikator.

## **xenon\_core\_force\_destroy(*index*)**

Funkcja niszczy siłę z danym identyfkatorem *index*.

## **xenon\_core\_force\_set\_force(*index*, *force*)**

Ustala wartość siły. **force** - wartość siły.

## **xenon\_core\_force\_set\_alpha(*index*, *alpha*)**

Ustala wartość kątu alfa danej siły. **alpha** - kąt alfa.

## **xenon\_core\_force\_set\_beta(*index*, *beta*)**

Ustala wartość kątu beta danej siły. **beta** - kąt beta.

## **xenon\_core\_force\_set\_position(*index*, *x*, *y*, *z*)**

Ustawia siłę na daną pozycję. **x**, **y**, **z** - pozycja siły.

## **xenon\_core\_force\_set\_speed(*index*, *speed*)**

Nadaje prędkość danej siły. **speed** - prędkość siły.

## **xenon\_core\_force\_set\_anchor(*index*, *anchor*)**

Ustala odnośnik do substancji z którą siła będzie połączona. Wartość XE\_NULL oznacza brak połączenia.

## **xenon\_core\_force\_get(*index*, *type*)**

Zwraca wartość zmiennej danej siły. **type** - typ zmiennej. Dostępne wartości: XE\_FORCE\_X, XE\_FORCE\_Y, XE\_FORCE\_Z, XE\_FORCE\_ALPHA, XE\_FORCE\_BETA, XE\_FORCE\_FORCE, XE\_FORCE\_SPEED, XE\_FORCE\_ANCHOR.



# GLOBAL FORCE

Siły o zasięgu globalnym. Podczas obliczeń fizycznych siły te są sumowane.

**xenon\_core\_global\_force\_create()**

Funkcja tworzy siłę globalną i zwraca jej identyfikator.

**xenon\_core\_global\_force\_destroy(*index*)**

Funkcja niszczy siłę globalną z danym identyfkatorem *index*.

**xenon\_core\_global\_force\_set\_alpha(*index*, *alpha*)**

Ustala wartość kątu alfa. **alpha** - wartość kątu alfa.

**xenon\_core\_global\_force\_set\_beta(*index*, *beta*)**

Ustala wartość kątu alfa. **beta** - wartość kątu beta.

**xenon\_core\_global\_force\_set\_force(*index*, *force*)**

Ustala wartość siły. **force** - wartość siły.

**xenon\_core\_global\_force\_get(*index*, *type*)**

Zwraca wartość zmiennej danej siły globalnej. **type** - typ zmiennej. Dostępne wartości: *XE\_GLOBAL\_FORCE\_ALPHA*, *XE\_GLOBAL\_FORCE\_BETA*, *XE\_GLOBAL\_FORCE\_FORCE*.

# PARTICLE

Cząsteczki, zastępujące billboardy w renderingu. Każda musi być połączona z jakąś substancją, by była wyświetlana.

## **xenon\_core\_particle\_create()**

Funkcja tworzy cząsteczkę i zwraca jej identyfikator.

## **xenon\_core\_particle\_destroy(*index*)**

Funkcja niszczy cząsteczkę z danym identyfkatorem *index*.

## **xenon\_core\_particle\_set\_size(*index, width, height*)**

Ustala rozmiar danej cząsteczki. **width** - szerokość. **height** - wysokość.

## **xenon\_core\_particle\_set\_substance(*index, substance*)**

Łączy cząsteczkę z daną substancją. **substance** - index substancji, z którą łączymy cząsteczkę.

## **xenon\_core\_particle\_set\_texture(*index, texture, type*)**

Ustala cząsteczce teksturę o danym indeksie. *texture* - index tekstury. *type* - typ textury. Dostępna wartość: *XE\_2D*.

## **xenon\_core\_particle\_get(*index, type*)**

Zwraca wartość zmiennej danej cząsteczki. **type** - typ zmiennej. Dostępne wartości: *XE\_PARTICLE\_SUBSTANCE*, *XE\_PARTICLE\_WIDTH*, *XE\_PARTICLE\_HEIGHT*, *XE\_PARTICLE\_TEXTURE*, *XE\_PARTICLE\_TEXTURE\_TYPE*.

# SUBSTANCE

Substancje są najważniejszymi elementami XenoNa. One odpowiadają za całokształt zjawisk fizycznych oraz za kształt świata gry.

## **xenon\_core\_substance\_create()**

Funkcja tworzy substancję i zwraca jej identyfikator.

## **xenon\_core\_substance\_destroy(*index*)**

Funkcja niszczy substancję z danym identyfkatorem *index*.

## **xenon\_core\_substance\_set\_position(*index*, *x*, *y*, *z*)**

Ustawia substancję na danej pozycji. **x**, **y**, **z** - pozycja substancji.

## **xenon\_core\_substance\_set\_mass(*index*, *mass*)**

Nadaje substancji masę. **mass** - masa substancji.

## **xenon\_core\_substance\_set\_range(*index*, *range*)**

Ustala zasięg substancji, jej wielkość. **range** - promień substancji.

## **xenon\_core\_substance\_set\_energy(*index*, *energy*)**

Nadaje substancji energię. W obecnej wersji silnika wartość niekonieczna. **energy** - energia substancji.

## **xenon\_core\_substance\_add\_inertion(*index*, *x*, *y*, *z*)**

Dodaje substancji siłę bezwładności o wektorze **x**, **y**, **z**. **x**, **y**, **z** - wektor siły bezwładności.

## **xenon\_core\_substance\_reset\_inertion(*index*)**

Zeruje wartość bezwładności danej substancji.

## **xenon\_core\_substance\_get(*index*, *type*)**

Zwraca wartość zmiennej danej substancji. **type** - typ zmiennej. Dostępne wartości: *XE\_SUBSTANCE\_X*, *XE\_SUBSTANCE\_Y*, *XE\_SUBSTANCE\_Z*, *XE\_SUBSTANCE\_MASS*, *XE\_SUBSTANCE\_RANGE*, *XE\_SUBSTANCE\_ENERGY*, *XE\_SUBSTANCE\_INERT\_X*, *XE\_SUBSTANCE\_INERT\_Y*, *XE\_SUBSTANCE\_INERT\_Z*.

# SURFACE

Dzięki powierzchniom możliwe jest ukazanie świata gry oraz wyliczanie dokładnych kolizji.

## **xenon\_core\_surface\_create()**

Funkcja tworzy powierzchnię i zwraca jej identyfikator.

## **xenon\_core\_surface\_destroy(*index*)**

Funkcja niszczy powierzchnię z danym identyfkatorem *index*.

## **xenon\_core\_surface\_set\_substance(*index, type, substance*)**

Łączy substancję z daną powierzchnią. **type** - typ substancji. Dostępne wartości: *XE\_SURFACE\_SUBSTANCE\_1*, *XE\_SURFACE\_SUBSTANCE\_2*, *XE\_SURFACE\_SUBSTANCE\_3*. **substance** - index substancji, którą łączymy z powierzchnią.

## **xenon\_core\_surface\_set\_texture(*index, texture, type*)**

Ustala teksturę dla danej powierzchni. *texture* - index tekstury. *type* - typ tekstury. Dostępna wartość: *XE\_2D*.

## **xenon\_core\_surface\_set\_texture\_coord(*index, type, x, y, z*)**

Nadaje koordynaty danemu wierzchołkowi (substancji) tekstury. **type** - typ wierzchołka. Dostępne wartości: *XE\_SURFACE\_TEX\_SUBSTANCE\_1*, *XE\_SURFACE\_TEX\_SUBSTANCE\_2*, *XE\_SURFACE\_TEX\_SUBSTANCE\_3*. **x**, **y**, **z** - pozycja tekstury. Wartości od 0 do 1.

## **xenon\_core\_surface\_set\_orientation(*index, orientation*)**

Nadaje powierzchni orientację. **orientation** - typ orientacji. Dostępne wartości: *XE\_SURFACE\_ORIENTATION\_FORWARD*, *XE\_SURFACE\_ORIENTATION\_BACKWARD*.

## **xenon\_core\_surface\_set\_mirror(*index, mirror*)**

Nadaje powierzchni właściwości odbijania światła niczym lustro. *mirror* - natężenie odbicia. Wartości od 0 do 1.

## **xenon\_core\_surface\_set\_diffusion(*index, diffusion*)**

Ustala poziom rozproszenia światła padającego na powierzchnię. **diffusion** - poziom rozproszenia.

# SURFACE

**xenon\_core\_surface\_get**(*index*, *type*)

Zwraca wartość zmiennej danej powierzchni. **type** - typ zmiennej. Dostępne wartości: *XE\_SURFACE\_SUBSTANCE\_1*, *XE\_SURFACE\_SUBSTANCE\_2*, *XE\_SURFACE\_SUBSTANCE\_3*, *XE\_SURFACE\_TEXTURE*, *XE\_SURFACE\_TEXTURE\_TYPE*, *XE\_SURFACE\_TEX\_SUBSTANCE\_1\_X*, *XE\_SURFACE\_TEX\_SUBSTANCE\_1\_Y*, *XE\_SURFACE\_TEX\_SUBSTANCE\_1\_Z*, *XE\_SURFACE\_TEX\_SUBSTANCE\_2\_X*, *XE\_SURFACE\_TEX\_SUBSTANCE\_2\_Y*, *XE\_SURFACE\_TEX\_SUBSTANCE\_2\_Z*, *XE\_SURFACE\_TEX\_SUBSTANCE\_3\_X*, *XE\_SURFACE\_TEX\_SUBSTANCE\_3\_Y*, *XE\_SURFACE\_TEX\_SUBSTANCE\_3\_Z*, *XE\_SURFACE\_TEX\_DIFFUSION*, *XE\_SURFACE\_TEX\_MIRROR*, *XE\_SURFACE\_TEX\_ORIENTATION*.

# WAVE FORCE

Siły falowe, najlepiej odzwierciedlające eksplozje.

**xenon\_core\_wave\_force\_create()**

Funkcja tworzy siłę falową i zwraca jej identyfikator.

**xenon\_core\_wave\_force\_destroy(*index*)**

Funkcja niszczy siłę falową z danym identyfkatorem *index*.

**xenon\_core\_wave\_force\_set\_force(*index*, *force*)**

Ustala wartość siły. **force** - wartość siły.

**xenon\_core\_wave\_force\_set\_length(*index*, *length*)**

Ustala długość fali. **force** - długość.

**xenon\_core\_wave\_force\_set\_orientation(*index*, *orientation*)**

Ustala kierunek fali. Dostępne wartości: *XE\_WAVE\_FORCE\_ORIENTATION\_OUT*, *XE\_WAVE\_FORCE\_ORIENTATION\_IN*, *XE\_WAVE\_FORCE\_ORIENTATION\_OUTER*, *XE\_WAVE\_FORCE\_ORIENTATION\_INNER*.

**xenon\_core\_wave\_force\_set\_position(*index*, *x*, *y*, *z*)**

Ustawia siłę falową na danej pozycji. **x**, **y**, **z** - pozycja.

**xenon\_core\_wave\_force\_set\_range(*index*, *range*)**

Ustala zasięg fali. **range** - zasięg.

**xenon\_core\_wave\_force\_set\_speed(*index*, *speed*)**

Nadaje fali prędkość. **speed** - prędkość fali.

**xenon\_core\_wave\_force\_get(*index*, *type*)**

Zwraca wartość zmiennej danej siły falowej. **type** - typ zmiennej. Dostępne wartości: *XE\_WAVE\_FORCE\_X*, *XE\_WAVE\_FORCE\_Y*, *XE\_WAVE\_FORCE\_Z*, *XE\_WAVE\_FORCE\_X*, *XE\_WAVE\_FORCE\_FORCE*, *XE\_WAVE\_FORCE\_RANGE*, *XE\_WAVE\_FORCE\_LENGTH*, *XE\_WAVE\_FORCE\_SPEED*, *XE\_WAVE\_FORCE\_ORIENTATION*.

## MISCELLANEOUS

**xenon\_core\_direction\_alpha**( $x1, y1, z1, x2, y2, z2$ )

Funkcja zwraca kąt alfa między dwoma punktami w przestrzeni 3D.

**x1, y1, z1** - pozycja pierwszego punktu. **x2, y2, z2** - pozycja drugiego punktu.

**xenon\_core\_direction\_beta**( $x1, y1, z1, x2, y2, z2$ )

Funkcja zwraca kąt beta między dwoma punktami w przestrzeni 3D.

**x1, y1, z1** - pozycja pierwszego punktu. **x2, y2, z2** - pozycja drugiego punktu.

**xenon\_core\_lengthdir\_x**( $alpha, beta, length$ )

Zwraca wartość osi X wektora o kierunku  $alpha, beta$  i długości  $length$ .

**alpha, beta** - kierunek wektora. **length** - długość wektora.

**xenon\_core\_lengthdir\_y**( $alpha, beta, length$ )

Zwraca wartość osi Y wektora o kierunku  $alpha, beta$  i długości  $length$ .

**alpha, beta** - kierunek wektora. **length** - długość wektora.

**xenon\_core\_lengthdir\_z**( $alpha, beta, length$ )

Zwraca wartość osi Z wektora o kierunku  $alpha, beta$  i długości  $length$ .

**alpha, beta** - kierunek wektora. **length** - długość wektora.

# PHYSIC

## **xenon\_core\_physic\_update()**

Najważniejsza funkcja fizyki XenonN Core. Aktualizuje ona cały świat gry zbudowany z elementów.

## **xenon\_core\_physic\_set(*type*, *index*)**

Kopiuje odnośnik elementu *index* typu *type* do odnośnika elementu bieżącego. Bezsprzecznie ważna i przydatna funkcja w optymalizacji gry. Dzięki odnośnikom elementów bieżących, ustalanie i pobieranie wartości dla poszczególnych elementów przy ich dużej liczbie zachodzi bardzo szybko. Jeśli potem zechcemy w funkcji jakiegoś elementu odnieść się do odnośnika elementu bieżącego, za *index* elementu podajemy wartość XE\_NULL.

**type** - typ elementu. Dostępne wartości: XE\_SUBSTANCE, XE\_COVALENCE\_FORCE, XE\_FORCE, XE\_SURFACE, XE\_PARTICLE, XE\_GLOBAL\_FORCE, XE\_BLACK\_HOLE, XE\_WAVE\_FORCE. **index** - index elementu który kopiujemy do odnośnika elementu bieżącego.

## **xenon\_core\_physic\_back(*type*)**

Ustawia odnośnik elementu bieżącego o jedną pozycję wstecz.

Wartości argumentu *type* takie jak w poprzedniej funkcji.

## **xenon\_core\_physic\_next(*type*)**

Ustawia odnośnik elementu bieżącego o jedną pozycję w przód.

Wartości argumentu *type* takie jak w poprzedniej funkcji.

## **xenon\_core\_physic\_reset(*type*)**

Zeruje odnośnik elementu bieżącego (ustawia go na pierwszym elemencie typu *type*).

Wartości argumentu *type* takie jak w poprzedniej funkcji.



**xenon\_core\_get\_char**(*var*)

Zwraca wartość typu znakowego zmiennej *var* maszyny stanu Xenon Core. Dostępne wartości: *XE\_ENGINE\_NAME*, *XE\_ENGINE\_CREATOR*, *XE\_ENGINE\_VERSION*, *XE\_ENGINE\_DESCRIPTION*, *XE\_ENGINE\_COPYRIGHT*, *XE\_ENGINE\_OWNER*.

**xenon\_core\_get\_double**(*var*)

Zwraca wartość typu liczbowego zmiennej *var* maszyny stanu Xenon Core. Dostępne wartości: *XE\_WINDOW\_HANDLE*, *XE\_DISPLAY\_CONTEXT*, *XE\_RENDER\_CONTEXT*, *XE\_VIRTUAL\_CONTEXT*, *XE\_RENDER\_WIDTH*, *XE\_RENDER\_HEIGHT*, *XE\_RENDER\_ANGLE*, *XE\_RENDER\_NEAR*, *XE\_RENDER\_FAR*, *XE\_RENDER\_DITHER*, *XE\_RENDER\_TEXTURING*, *XE\_PHYSICAL\_FACTOR*, *XE\_PHYSICAL\_GRAVITY\_MODE*, *XE\_PHYSICAL\_SURFACE\_COLLISION\_MODE*, *XE\_PHYSICAL\_COVALENCE\_FORCE\_MODE*, *XE\_PHYSICAL\_COVALENCE\_FORCE\_FIRST\_STREAM*, *XE\_PHYSICAL\_COVALENCE\_FORCE\_LAST\_STREAM*, *XE\_PHYSICAL\_FORCE\_MODE*, *XE\_PHYSICAL\_GLOBAL\_FORCE\_MODE*, *XE\_PHYSICAL\_BLACK\_HOLE\_MODE*, *XE\_PHYSICAL\_RANGE\_COLLISION\_MODE*, *XE\_PHYSICAL\_INERTION\_MODE*, *XE\_PHYSICAL\_INERTION\_FRICTION*, *XE\_PHYSICAL\_INERTION\_RANGE*, *XE\_PHYSICAL\_INERTION\_FORCE*, *XE\_PHYSICAL\_INERTION\_GLOBAL\_FORCE*, *XE\_PHYSICAL\_INERTION\_WAVE\_FORCE*, *XE\_PHYSICAL\_WAVE\_FORCE\_MODE*, *XE\_SUBSTANCE\_COUNT*, *XE\_RAY\_COUNT*, *XE\_COVALENCE\_FORCE\_COUNT*, *XE\_FORCE\_COUNT*, *XE\_SURFACE\_COUNT*, *XE\_PARTICLE\_COUNT*, *XE\_GLOBAL\_FORCE\_COUNT*, *XE\_BLACK\_HOLE\_COUNT*, *XE\_WAVE\_FORCE\_COUNT*, *XE\_SUBSTANCE\_CURRENT*, *XE\_RAY\_CURRENT*, *XE\_COVALENCE\_FORCE\_CURRENT*, *XE\_FORCE\_CURRENT*, *XE\_SURFACE\_CURRENT*, *XE\_PARTICLE\_CURRENT*, *XE\_GLOBAL\_FORCE\_CURRENT*, *XE\_BLACK\_HOLE\_CURRENT*, *XE\_WAVE\_FORCE\_CURRENT*, *XE\_TRANSFORM\_ROTATION\_ALPHA*,

*XE\_TRANSFORM\_ROTATION\_BETA*, *XE\_TRANSFORM\_ROTATION\_GAMMA*, *XE\_TRANSFORM\_SCALING\_X*, *XE\_TRANSFORM\_SCALING\_Y*, *XE\_TRANSFORM\_SCALING\_Z*, *XE\_TRANSFORM\_TRANSLATION\_X*, *XE\_TRANSFORM\_TRANSLATION\_Y*, *XE\_TRANSFORM\_TRANSLATION\_Z*.

**xenon\_core\_set\_double**(*var*, *value*)

Nadaje wartość *value* zmiennej *var* maszyny stanu Xenon Core. **var** - zmienna maszyny stanu. Dostępne wartości: *XE\_TRANSFORM\_RESET*, *XE\_RENDER\_WIDTH*, *XE\_RENDER\_HEIGHT*, *XE\_RENDER\_ANGLE*, *XE\_RENDER\_NEAR*, *XE\_RENDER\_FAR*, *XE\_RENDER\_DITHER*, *XE\_RENDER\_TEXTURING*, *XE\_PHYSICAL\_COVALENCE\_FORCE\_MODE*, *XE\_PHYSICAL\_COVALENCE\_FORCE\_FIRST\_STREAM*, *XE\_PHYSICAL\_COVALENCE\_FORCE\_LAST\_STREAM*, *XE\_PHYSICAL\_FORCE\_MODE*, *XE\_PHYSICAL\_GRAVITY\_MODE*, *XE\_PHYSICAL\_SURFACE\_COLLISION\_MODE*, *XE\_PHYSICAL\_FACTOR*, *XE\_PHYSICAL\_GLOBAL\_FORCE\_MODE*, *XE\_PHYSICAL\_BLACK\_HOLE\_MODE*, *XE\_PHYSICAL\_RANGE\_COLLISION\_MODE*, *XE\_PHYSICAL\_INERTION\_MODE*, *XE\_PHYSICAL\_INERTION\_FRICTION*, *XE\_PHYSICAL\_INERTION\_RANGE*, *XE\_PHYSICAL\_INERTION\_FORCE*, *XE\_PHYSICAL\_INERTION\_GLOBAL\_FORCE*, *XE\_PHYSICAL\_INERTION\_WAVE\_FORCE*, *XE\_PHYSICAL\_WAVE\_FORCE\_MODE*.

**xenon\_core\_header**()

Funkcja nagłówkowa. Musi być wykonana przynajmniej raz w każdym obiekcie wykorzystującym funkcje Xenon Core, by uniknąć błędów.

**xenon\_core\_initiation**(*dllfilename*, *flag*)

Funkcja inicjująca silnik Xenon Core. **dllfilename** - ścieżka pliku DLL silnika Xenon Core. **flag** - flagi inicjacji silnika. Dostępne wartości: *XE\_NULL*, *XE\_DRAW\_TO\_WINDOW*.

**xenon\_core\_closure**()

Funkcja wyłączająca działanie silnika i zwalnająca DLL z pamięci.