

Programowanie gier 3D w HTML5

Andrzej P. Urbański
Politechnika Poznańska

Moje marzenie

- Od dawna jest znany pakiet Open GL napisany w C++ i bardzo ułatwiający tworzenie gier 3D
- Zaproponowałem kiedyś jako pracę dyplomową stworzenie plug-in do wybranej przeglądarki, który by umożliwiał użycie Open GL z poziomu JavaScript
- Innymi słowy zaraz po wejściu na stronę internetową można by grać w grę 3D

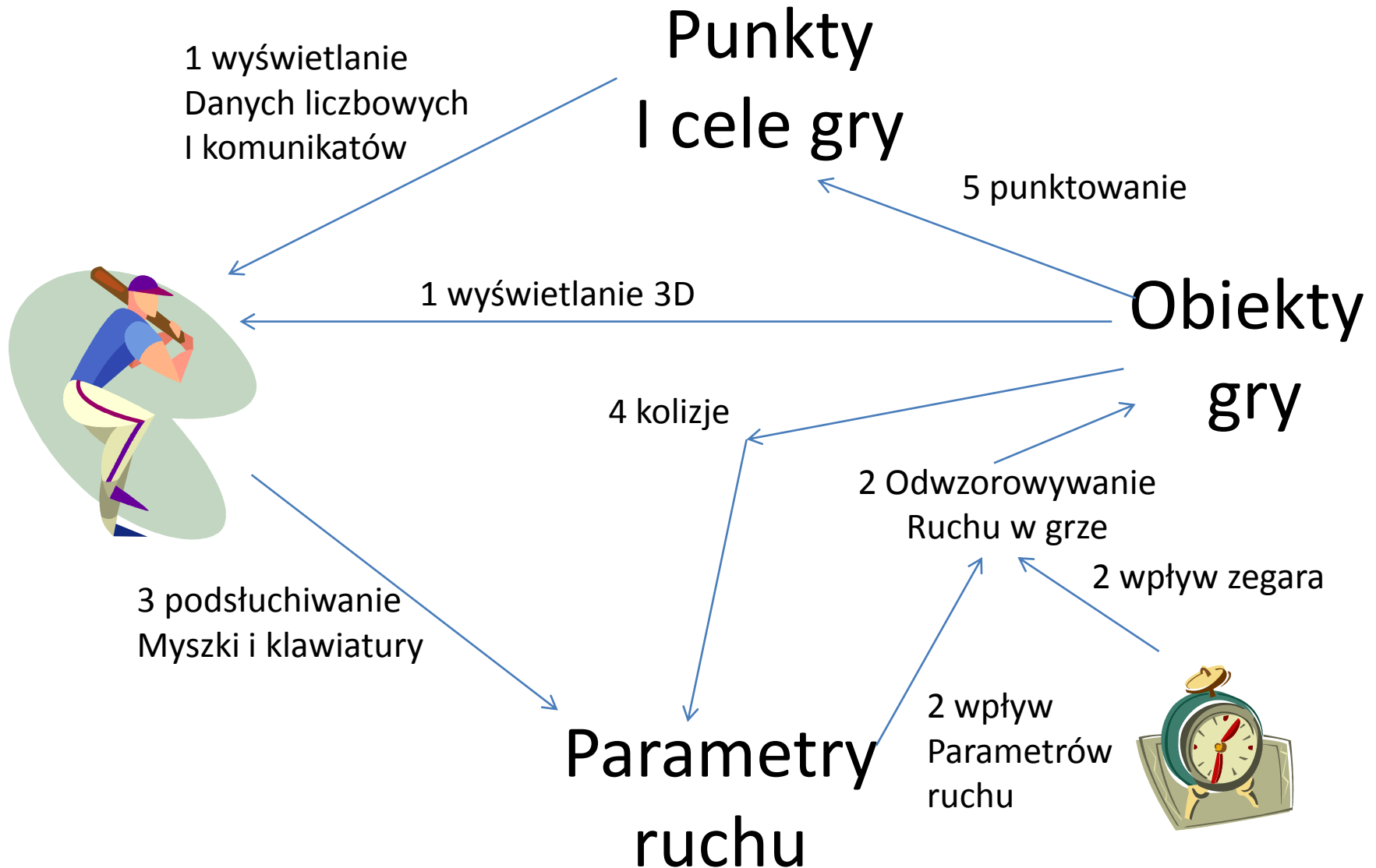
Moja Gwiazdka

- W trakcie ustalania standardu HTML5 podobno pod naciskiem takich lobby jak Stevie Jobs i Microsoft, chcących usunąć plug-in Flash z rynku przyjęto standard rozszerzenia HTML o część odpowiedzialną za renderowanie 3D. Założono użycie w tym celu przez producentów przeglądarek internetu kodu OpenGL.
- To więcej niż chciałem – Open GL dostępne bez żadnych we wszystkich przeglądarkach

Jak jest zaprogramowana gra?

1. Potrafi wyświetlać pojedynczą ramkę 3D
2. Potrafi animować! – biorąc wskazania zegara i parametrów ruchu zmienia obiekty gry
3. Jest interaktywna! - czyta ruchy manipulatorów gracza i odpowiednio zmienia parametry ruchu
4. Jest wiarygodna i ciekawa! - traktuje kolizje obiektów zgodnie z fizyką i regułami gry
5. Jest wymagająca i emocjonująca! - punktuje cele pośrednie i ostateczny

Powiązanie poszczególnych elementów gry



Struktura programu 3D w HTML5

Moduł ... te unikalne dla konkretnej gry podkreślono

1. obsługi karty graficznej w języku GLCS
2. obsługi karty graficznej w języku JavaScript
3. odwołujący się do biblioteki WebGL celem wyświetlania obiektów 3D
4. nad słuchający klawiaturę/mysz
5. przeliczania parametrów ruchu i zegara na ramki 3D
6. odwzorowujący klawiaturę na parametry ruchu
7. rozpoznający i rozstrzygający kolizje
8. aktualizujący punkty i cele wg. stanu obiektów gry

Zalety WebGL w HTML5

- Dzięki (1) i (2) programy w WebGL efektywnie wykorzystują możliwości karty graficznej
- Biblioteka WebGL (3) oparta na rozwijalnym od lat OpenGL ułatwia programowanie w 3D
- Możliwości (4) programowania obsługi klawiatury, a zwłaszcza myszy i kanału dźwiękowego są dużo większe niż bez WebGL

Struktura dokumentu z grą

```
<!DOCTYPE html>
```

```
<html>
```

```
<script id="shader-fs" type="x-shader/x-fragment">
```

```
Kod w języku GLCS </script>
```

```
<script id="shader-vs" type="x-shader/x-vertex">
```

```
Kod w języku GLCS </script>
```

```
<script type="text/javascript">
```

```
Kod w języku JavaScript wt ym funkcji: startujMojaGre();</script>
```

```
<body onload='startujMojaGre();'>
```

```
<canvas width='400' height='400'>
```

```
</canvas>
```

```
</body>
```

```
</html>
```


Jak się przechowuje obiekty 3D?

- Projektujemy obiekt w siatce 3D o współrzędnych w trójwymiarowym układzie kartezjańskim XYZ
- Łączymy wszystkie trójkąty na tej powierzchni w większe niekoniecznie płaskie powierzchnie
- Każdą taką powierzchnię umieszczamy w odpowiednim kształcie i proporcjach na płaszczyźnie w jednym pliku z teksturami
- Każdą powierzchnię z pliku tekstur mapujemy na powierzchnię na trójwymiarowej siatce obiektu

Jak to wygląda w praktyce?

- Najprostsze gry można tworzyć w czystym HTML-u przetwarzanym wyłącznie w przeglądarce na własnym komputerze
- Chrome wymaga by co najmniej obrazy z teksturami były pobierane ze strony internetowej, a nie lokalnego komputera, ale można było to włączyć dodając *–allow-file-access-from-files*
- Przeglądarki opornie adaptują nowości 3D. Na dziś dzień tylko Chrome i FireFox.
- Wymagania dla kart graficznych(np.nie Intel 3150)
- HTML5 z WebGL to już jednak standard!

Czym jest WebGL w HTML5?

- OpenGL
 - tradycyjnie był pakietem bibliotecznym dla C++
 - WebGL to OpenGL ES 2.0 dla JavaScript
- Renderowanie w Open GLES 2.0 wymaga shaderów karty graficznej, które muszą być:
 - Ładowane w kodzie źródłowym (shaderSource),
 - Kompilowane wywołaniem z JS (compileShader)
 - Przyłączone do programu (attachShader)
 - Linkowane (linkProgram)
 - Używane (useProgram).
-