



**WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI**
POLITECHNIKI RZESZOWSKIEJ

Kacper Rudź

Prównanie silników Unreal Engine i Unity pod kątem tworzenia
gier

Praca dyplomowa magisterska

Opiekun pracy:

dr inż. Sławomir Samolej prof. PRz

Rzeszów, 2024

Spis treści

Wykaz symboli, oznaczeń i skrótów (opcjonalny)	5
1. Cel pracy	6
2. Wstęp	7
3. Teoretycznie o silnikach graficznych	9
3.1. Założenia silników graficznych używanych w grach	9
3.2. Architektura silnika Unity	9
3.3. Architektura silnika Unreal Engine	9
3.4. Podsumowanie różnic między silnikami	9
4. Tekst zasadniczy – II	10
4.1. Formatowanie tekstu. Należy pamiętać, że na końcu tytułu rozdziału, podrozdziału i zakresu nie umieszcza się kropki	10
4.1.1. Marginesy i akapity	10
4.1.2. Zalecenia co do sposobu pisania jednostek i symboli wielkości fizycznych	11
4.1.3. Rysunki i tabele	13
4.1.4. Wzory matematyczne	14
4.1.5. Listingi programów	16
4.1.6. Numerowanie i punktowanie	16
4.2. Wykaz literatury	17
4.3. Wydruk pracy	17
5. Podsumowanie i wnioski końcowe	19
Załączniki	20
Literatura	21

Wykaz symboli, oznaczeń i skrótów (opcjonalny)

1 ÷ 2 stron wykaz ważniejszych symboli i oznaczeń (jeśli jest potrzebny).

1. Cel pracy

Tematem pracy jest porównanie silników Unreal Engine i Unity pod kątem tworzenia gier. W tym celu należy wskazać różnice w budowie silnika. Zostaną wskazane różnice w pisaniu skryptów – użyte języki programowania i inne narzędzia – różnice w architekturze silników oraz różnice związane z kolejnością wykonywania metod i funkcji.

Ważnym elementem gier jest stabilność stworzonej aplikacji oraz ilość zasobów wymaganych do prawidłowego funkcjonowania gry. W tym celu zostanie przeprowadzony test polegający na rozłożeniu w silniku kilku tysięcy kopii jednej postaci i poruszaniu się kamery po ustalonej wcześniej ścieżce przez kilkadziesiąt minut. Kopie postaci będą wykonywać zapętlone animacje – będzie 10 animacji i każda kopia będzie wykonywała jedną z tych animacji. W trakcie testu będą zbierane informacje na temat ilości klatek na sekundę, zapotrzebowaniu na pamięć RAM oraz pamięci VRAM i ile czasu w danej klatce aplikacja używała procesora i procesora graficznego. Dane będą zebrane z kilku różnych komputerów – komputery osobiste i laptopy - o różnych specyfikacjach, a następnie dane zostaną przeanalizowane. Analizowane dane będą oceniane pod względem różnych kryteriów.

2. Wstęp

Branża gier jest bardzo szeroka i obejmuje wiele różnych urządzeń. Głównymi platformami, na które są wydawane gry to telefony, komputery osobiste oraz różne konsole. Każda z tych platform ma swoje wymagania i posiada różne ograniczenia. Przykładowo telefony nie mają dużej możliwości dołączania urządzeń peryferyjnych takich jak klawiatury czy myszki. Konsole są na tyle grupą platformą, że każda konsola stanowi osobną platformę. Komputery natomiast mogą posiadać różne systemy operacyjne – różne dystrybucje systemów Linux, Windows i MacOS – oraz różne podzespoły.

Rozwój gier komputerowych zawsze szedł w parze z rozwojem branży filmowej, a szczególnie w gałęzi filmów animowanych, co spowodowało rozwój narzędzi do modelowania, animowania i tekstuowania. Początkowo silniki graficzne nie były one zbyt rozwinięte – posiadały proste systemy kolizji, bardzo proste systemy cząsteczkowe, obsługa animacji szkieletowych, modeli i dźwięku. Wraz z czasem silniki graficzne były coraz bardziej optymalizowane i rozwijane. Rozwój oprogramowania szedł w parze z rozwojem sprzętu.

Twórców gier można podzielić na kilka grup:

- Niezależni twórcy – jedna lub kilka osób tworzy gry i wydaje je zwykle samodzielnie. Często korzystają z gotowych silników graficznych lub używają swoich bardzo prostych silników. Gry wykonane przez twórców niezależnych: Factorio, Stardew Valley i pierwsze wersje gry Minecraft
- Małe studia gier – niewielkie firmy zatrudniające do kilkunastu osób. Skończony produkt jest wydawany z pomocą wydawcy, rzadko samodzielnie. Bardzo rzadko te firmy korzystają z własnych silników graficznych. Przykładowe gry zrobione przez małe studia: Celeste, Ori and The Blind forest, Terraria
- Średnie studia gier – już większe firmy zatrudniające nawet kilkaset osób. W przeszłości spora część korzystała z własnych silników graficznych. Przykładowymi grami stworzonymi przez średnie studia są: Biomutant, Baldurs Gate 3.
- Duże i bardzo duże studia gier – posiadają nawet do kilku tysięcy pracowników. Bardzo często tworzą gry na własnych silnikach graficznych. Studia takie potrafią pracować nad kilkoma grami jednocześnie. Przykładowe gry stworzone przez duże studia: Wiedźmin 3 Dziki Gon, gry serii Far Cry i Assassins Creed.

Rzeczą, która łączy wszystkie te grupy ludzi są silniki graficzne, a właściwie wymóg ich użycia w projekcie. Często brany pod uwagę przy wyborze silnika graficznego do projektu przez małych i średnich twórców są możliwości modyfikacji silnika, jak i samego silnika oraz wielkość społeczności zbudowanej wokół silnika graficznego. Dodatkowym czynnikiem jest brany pod uwagę jest jakość dokumentacji silnika graficznego. Dla dużych twórców koszt używania zewnętrznych silników graficznych może być wiele większy niż modernizowanie swojego silnika. Koszty licencji silników graficznych zwykle są zależne od zysku wygenerowanego przez grę. Dlatego duże firmy pokroju Electronic Arts lub Ubisoft tworzą swoje gry w oparciu o własne silniki graficzne. Gry tych firm generują zyski liczone w miliardach, a nie rzadko sięgają bilionów.

Obecnie do najbardziej popularnymi silnikami graficznymi z dużymi społecznościami, z pomocą których powstają gry są: Unity, Unreal Engine oraz GoDot. Z innych mniej znanych silników graficznych używanych w produkcji gier są Frostbite, RPG Maker, CryEngine, GameMaker Studio, Construct i wiele innych. Pewne silniki specjalizują się w bardzo specyficznej części gier, inne są wykorzystywane poza branżą gier – symulacje, rendering filmów czy wizualizacja danych.

3. Teoretycznie o silnikach graficznych

Gry komputerowe nie są tworzone jedynie za pomocą silników graficznych. Większość gier wymaga udźwiękowienia w postaci muzyki i różnego rodzaju dźwięków, grafiki – modele, animacje, tekstury w różnych postaciach oraz elementy interfejsu graficznego – oraz „kodu” – zaprogramowanie elementów świata gry, aby te elementy działały w taki sposób jaki chce tego twórca. Nie wszystkie gry muszą posiadać te elementy.. Do każdego elementu używane jest zwykle wyspecjalizowane oprogramowanie w celu przyspieszenia pracy.

Silniki graficzne są używane głównie przy tworzeniu modeli, animacji i tekstur oraz jako solidna podstawa pod kod gry. Przykładowo program do grafiki komputerowej 3D „Blender” posiada 3 silniki graficzne – EEVEE, Cycles i Workbench. Każdy z nich jest wyspecjalizowany w własnej dziedzinie – EEVEE jako renderer casu rzeczywistego, Cycles jest używany do renderowania obrazów w wysokiej jakości, Workbench jest używany do podglądów, modelowania i animacji. Jako podstawę do kodu gry można użyć silnika graficznego podobnego do Unity lub Unreal Engine.

3.1. Założenia silników graficznych używanych w grach

3.2. Architektura silnika Unity

3.3. Architektura silnika Unreal Engine

3.4. Podsumowanie różnic między silnikami

4. Tekst zasadniczy – II

Ponad 50% objętości pracy – część autorska:

- a) założenia – dane,
- b) opis zastosowanej metody rozwiązania lub analizy,
- c) opis proponowanego rozwiązania, wyniki analizy teoretycznej, obliczenia, projekt konstrukcyjny, procesowy, technologiczny,
- d) wyniki badań analitycznych, symulacyjnych lub eksperymentalnych itp.

Przy stosowaniu podziału na rozdziały i podrozdziały zaleca się unikać podziału więcej niż trzystopniowego. Podział tekstu, szczególnie na rozdziały główne, wynikać powinien z zakresu i charakterystyki realizowanej pracy.

4.1. Formatowanie tekstu. Należy pamiętać, że na końcu tytułu rozdziału, podrozdziału i zakresu nie umieszcza się kropki

4.1.1. Marginesy i akapity

Marginesy deklaruje się jako „lustrzane” i ustawia na 2 cm, na oprawę 1,5 cm. Nagłówek i stopka 1,25 cm. Tekst podstawowy akapitu: czcionka szeryfowa, styl Times (Times New Roman, Liberation Serif itp.), rozmiar 12 punktów, interlinia 1,5 wiersza. Akapit wyjustowany, wcięcie pierwszego wiersza 1,25 cm.

Na końcu każdego akapitu, którego tekst zaczerpnięto z literatury, musi znajdować się odnośnik do właściwej pozycji w wykazie literatury. W pracy nie stosuje się odnośników w formie przypisów. Liczby w nawiasie kwadratowym oznaczają kolejny numer pozycji w wykazie, np. [1] lub [1, 4, 7] lub [1, 6-8] itp.

Cytaty (dosłowne przytoczenie obcego tekstu w pracy) pisze się czcionką pochylą (kursywą) i ujmuje w cudzysłów. Przykład: „*Współpracując z jednostkami gospodarczymi działającymi w kraju, kształci wysokokwalifikowaną kadrę inżynierów*”.

Fragmenty kodów programów pisze się czcionką o stałej szerokości, styl Courier (Courier New, Liberation Mono itp.) o rozmiarze 10 punktów.

4.1.2. Zalecenia co do sposobu pisania jednostek i symboli wielkości fizycznych

Poniższy podrozdział opracowano na podstawie [5]. W trakcie pisania pracy należy zwracać uwagę na sposób oznaczania jednostek i symboli wielkości fizycznych. Przy zapisywaniu jednostek i symboli wielkości fizycznych można wyróżnić zapis w postaci kursywy (pismo pochyle) oraz antykwy (pismo proste).

1) Kursywę należy stosować w następujących przypadkach:

- symboli wielkości fizycznych niezależnie od tego czy jest to litera alfabetu greckiego (np. przenikalność magnetyczna μ) czy też łacińskiego (np. rezystancja R). Należy przestrzegać tej zasady niezależnie od miejsca, w którym pojawia się symbol tj. tekst, wzory matematyczne, rysunki, tabele,
- ogólny symbol zapisu funkcji czyli np. f , a nie f . Nie dotyczy to jednak zapisu konkretnych funkcji np. $\cos\omega t$ a nie $\cos\omega t$,
- macierze, wektory, których elementami są wielkości fizyczne należy zapisywać dodatkowo czcionką półgrubą (bold) np. $\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$, $\mathbf{U} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$,
- wskaźnik dolny, górny, prawo- i lewostronny, ale tylko gdy odnosi się do konkretnej wielkości fizycznej, czyli np. składowa x -owa indukcji magnetycznej B_x , a nie B_x ,
- wskaźniki górne i dolne oznaczające dowolną liczbę np. R_j , I^k , ale nie R_1 , I^2 .

2) Antykwę należy stosować w następujących sytuacjach:

- wszystkie cyfry,
- symbole konkretnych funkcji np. $\tan\omega t$, a nie $\tan\omega t$,
- operatory operacji matematycznych np. pochodne zwyczajne $\frac{dx}{dt}$, a nie $\frac{dx}{dt}$,
- symbole liczb o konkretnej wartości np. przenikalność elektryczna próżni $\varepsilon_0 = 8,8542 \cdot 10^{-12} \text{ F} \cdot \text{m}^{-1}$, a nie $\varepsilon_0 = 8,8542 \cdot 10^{-12} \text{ F} \cdot \text{m}^{-1}$,
- indeksy, jeżeli odnoszą się do: obiektów (fizycznych, geometrycznych), czyli, np. natężenie pola elektrycznego w punkcie A to E_A , a nie E_A , zjawisk lub stanów fizycznych, np. moment obciążenia to T_L , a nie T_L , do nazwisk czy też

oznaczeń pierwiastków, np. straty w miedzi to P_{Cu} a nie P_{Cu} , do charakteru wielkości symbolizowanej przez literę źródłową, np. wartość maksymalna siły to F_{\max} , a nie F_{max} , oznaczeń jednostek miary np. $M\Omega$, a nie $M\Omega$.

3) W przypadku jednostek miar (które zawsze należy pisać antykwą) zapisując konkretną wartość liczbą należy podać jej wartość i jednostkę z zachowaniem następujących zasad:

- zapisując wartość liczbową wielkości fizycznej po spacji należy podać jej jednostkę, ale nie nazwę jednostki np. 10A, ale nie 10 amper czy też 10 amperów,
- zapisując wartość liczbową słownie należy w tej konwencji podać też jednostkę np. dziesięć omów, ale nie dziesięć Ω
- do oznaczeń jednostek nie wolno dopisywać indeksów, np. moc wyjściowa silnika wynosi $P = 100 \text{ kW}_{\text{out}}$. W takim przypadku należy zapisać $P_{\text{out}} = 100 \text{ kW}$,
- jednostek nie należy umieszczać w nawiasach kwadratowych, np. $I = 1 \text{ [A]}$. Odstępstwem od tej zasady mogą być tabele, nagłówki kolumn, opisy osi na wykresach oraz w sporadycznych sytuacjach we wzorach matematycznych (ale tylko wówczas, gdy zależność matematyczna nie wskazuje w jakiej jednostce wystąpi wartość liczbowa). Przykłady odstępstw zamieszczono w podrozdziale 4.1.3.

4) W trakcie zapisu symboli wielkości matematycznych można stosować również szereg znaków diakrytycznych, jak również należy przestrzegać następujących zaleceń:

- wartości chwilowe podstawowych wielkości fizycznych używanych np. w elektrotechnice należy zapisać małymi literami, np. u , i , lub stosować zapis np. $u(t)$, lub stosować indeks „ t ” przy wielkości, np. U_t ,
- wartości skuteczne wielkości okresowych należy zapisać dużą literą np. U , I ,
- wartości szczytowe funkcji zmiennej, amplitudę funkcji sinusoidalnej czasu należy zapisać jako np. U_m ,

- podkreślenie symboli reprezentujących wielkości fizyczne, których wartość liczbową jest liczbą zespoloną, przy czym podkreślenie dotyczy tylko literki źródłowej np. \underline{Z}_1 , a nie \underline{Z}_1 ,
- kreska nad literą źródłową oznacza wartość średnią, np. \bar{I} co jest równoważne I_{av} .

4.1.3. Rysunki i tabele

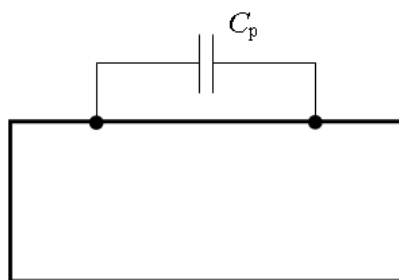
Tekst podstawowy w tabeli pisze się czcionką o rozmiarze 10 punktów, pojedyncza interlinia. Dane liczbowe – wyśrodkowane, dane tekstowe – wyrównane do lewej. Rysunki i tabele zamieszcza się wyśrodkowane na stronie, bez wcięcia pierwszego wiersza.

W akapicie poprzedzającym rysunek lub tabelę musi znajdować się krótki opis, czego dotyczy dany rysunek/tabela (odniesienie do rysunku/tabeli). Tytuły numeruje się zgodnie z kolejnością w danym rozdziale: numer_rozdziału.numer_tabeli/rysunku (np. rys. 2.1, tabela 3.5). W tytule rysunku/tabeli, zaczerpniętych z literatury, podaje się odnośnik do właściwej pozycji. Należy zadbać o to, aby opisy na rysunkach były czytelne (czcionka 8 punktów lub większa). Staraj się nie wymuszać numeracji, pozwól aby robił to za ciebie L^AT_EX. Stosuj `\label` do znakowania obiektów, do których być może w tekście się będziesz odwoływał (rozdziały, rysunki, tabele, wzory, listingi ...). Odwołuj się do nich w tekście za pomocą funkcji `\ref{NazwaObiektu}`. Pamiętaj, że L^AT_EX korzystając z polecenia `latex` nie odczytuje z plików .jpg, .png ich wielkości. Polecenie `latex` generuje plik DVI. Jeżeli chcesz go używać zgłosi stosowny błąd. Aby się go pozbyć zdefiniuj wielkość natywną pliku grafiki. Polecamy jednak używanie zamiast polecenia `latex`, polecenie `pdflatex`, wówczas problem nie wystąpi.

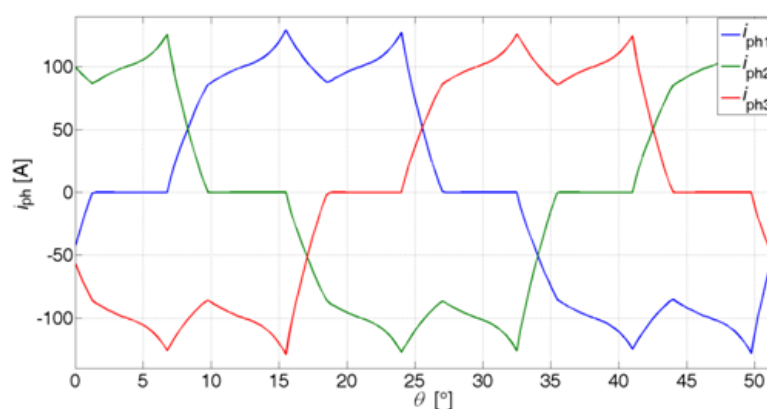
Przykład: [...] co umożliwia wyznaczenie wartości napięcia. Na rys. 4.1 przedstawiono schemat obwodu z równolegle dołączoną pojemnością C_p .

Przykład: [...] Na rysunku 4.2 pokazano przykładową zależność prądów pasmowych i_{ph} bezszczotkowego silnika prądu stałego z magnesami trwałymi w funkcji położenia wirnika θ .

Przykład: [...] oraz indukcyjności wzajemnej. W tabeli 4.1 przedstawiono podstawowe parametry obwodu nieliniowego, zasilanego napięciem trójfazowym.



Rysunek 4.1: Tytuł rysunku, rozmiar 11 pkt., pojedyncza interlinia, akapit wyśrodkowany, bez wcięcia pierwszego wiersza. Na końcu tytułu rysunku/tabeli nie stawia się kropki [8]



Rysunek 4.2: Tytuł rysunku, rozmiar 11 pkt., pojedyncza interlinia, akapit wyśrodkowany, bez wcięcia pierwszego wiersza. Na końcu tytułu rysunku/tabeli nie stawia się kropki [8]

4.1.4. Wzory matematyczne

Zmienne we wzorach pisze się czcionką pochyłą (styl edytora równań „Matematyka”) natomiast symbole, nie będące zmiennymi, czcionką prostą (styl „Tekst”). Rozmiary czcionek: normalny 12 punktów, indeks dolny/górny 9 pkt., indeks podrzędny 7 pkt., symbol 24 pkt., podsymbol 12 pkt. Separatorem dziesiętnym w liczbach jest przecinek, a nie kropka (dotyczy to również liczb pisanych w tekście akapitu). Poddaj się w tym zakresie $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ’owi - pisz wzór, a poprawnie się utworzy.

Pod wzorem należy zamieścić objaśnienia użytych symboli (chyba, że znajdują się w wykazie na początku pracy). Wzory umieszcza się wyśrodkowane i numeruje

Tabela 4.1: Tytuł tabeli, rozmiar 11 pkt., pojedyncza interlinia, akapit wyrównany do lewej

U [V]	I [mA]	R , [k Ω]	L [mH]	R/R_{20}
13,6	7,29	3,94	100	1,25

zgodnie z kolejnością w danym rozdziale: (numer_rozdziału.numer_wzoru). Numery wzorów wyrównuje się do prawego marginesu. W akapicie poprzedzającym wzór musi znajdować się krótki opis, czego dotyczy dany wzór i – jeżeli potrzeba – odwołanie do literatury.

Przykład: [...] wyznacza się, na podstawie wyrażenia (4.1). W nawiasach podano rozmiary czcionek używanych we wzorach

$$A(12) = \sum (24) m_{s(9)} N^{k_p(7)} \quad (4.1)$$

gdzie: m_s – masa próbki, N – natężenie oświetlenia, k_p – wykładnik potęgi ($k_p = 1,3 - 2,1$).

4.1.5. Listingi programów

W pracy dyplomowej możesz umieszczać fragmenty programów. Pamiętaj, aby umieszczać krótkie, tylko najważniejsze fragmenty kodów źródłowych. Zawsze je komentuj w treści pracy dyplomowej. Typowo w L^AT_EX kody źródłowe umieszczane są w środowisku `verbatim` (`\begin{verbatim}...\end{verbatim}`). Obecnie istnieje jednak bardziej nowoczesne i bardziej funkcjonalne środowisko `lstlisting` (wymaga zainstalowanego w systemie pakietu `listings`). Zwróć uwagę, że możesz kolorować składnię automatycznie za pomocą parametru `language`. W niniejszym dokumencie przedstawiono dwa przykłady listingów, Listing 1 to przykład kodu źródłowego Matlaba, a poniżej Listing 2 dla Perl'a.

```
1 i = 1
2 p = 3
3 for i = 1:10
4     if i > 3
5         i=i+p
6     else
7         i=i+1
8     end
9 end
```

Listing 1: Listing programu Matlab

```
1 my $url = 'http://pei.prz.edu.pl';
2 use LWP::Simple;
3 my $content = get $url;
4 die "Couldn't get $url" unless defined $content;
5 print $content;
6 print "\n";
7 print "Length " + length($content)
```

Listing 2: Listing programu Perl

Z pewnością przeglądając źródło tego dokumentu zobaczysz, że kody źródłowe powinny mieć zdefiniowane parametry `label`, aby łatwo w tekście do nich się odwoływać. Numeracja linii jest w stylu domyślnie włączona (to przydatne, bo w treści pracy łatwo odwołać się dzięki temu do konkretnego wiersza w kodzie źródłowym), możesz je wyłączyć podając jako parametr `numbers=none`. Więcej szczegółów możesz odnaleźć w sekcji `\lstset` pliku arkusza stylu.

4.1.6. Numerowanie i punktowanie

- 1) Pierwszy poziom (stosuje się numerowanie lub punktowanie). Formatowanie: akapit wyjustowany, wcięcie od lewej 0,75 cm, wysunięcie co 0,5 cm.

- 2) Znakiem numerowania jest liczba (z kropką lub nawiasem).
- drugi poziom (stosuje się wyłącznie punktowanie). Formatowanie: akapit wyjustowany, wcięcie od lewej 1,25 cm, wysunięcie co 0,5 cm,
 - znakiem punktowania jest łącznik lub mała litera alfabetu (z nawiasem). Nie zaleca się stosowania kropek, strzałek itp.,
 - punktowane akapity rozpoczyna się minuskulą (małą literą), na końcu akapitu stawia się przecinek, ostatni punktowany akapit kończy się kropką.
- 3) Numerowane akapity rozpoczyna się majuskulą (wielką literą) i kończy kropką.
- 4) Należy zwrócić uwagę, aby nie rozdzielać numerowania/punktowania pomiędzy kolejnymi stronami tekstu.

4.2. Wykaz literatury

W wykazie literatury zamieszcza się wyłącznie pozycje, na które powołano się w pracy. Kolejność numerów w wykazie – zgodna z kolejnością pojawiania się danej pozycji w tekście.

Format akapitu: akapit wyjustowany, wysunięcie 0,75 cm. Prawidłowo opracowany wykaz został zaprezentowany w niniejszym dokumencie w odpowiednim rozdziale, oznaczonym jako „Literatura” (pozycja nr [1] to zasoby internetowe, [2] – książka, [3] – artykuł w czasopiśmie, [4] – karta katalogowa).

4.3. Wydruk pracy

Przed wydrukiem należy usunąć ewentualne błędy literowe i sprawdzić prawidłową interpunkcję. Przykładowo, łącznik zapisuje się za pomocą krótkiego minusa (np. badawczo-rozwojowy) natomiast myślnik – stosowany w zdaniach wtrąconych – zapisuje się za pomocą długiej pauzy. Dzielenie wyrazów według uznania Autora (można podzielić długie wyrazy, powodujące duże „rozstrzelenie” tekstu w poprzedzającym wierszu. Zaleca się usunięcie pojedynczych znaków na końcu wiersza oraz podwójnych spacji w tekście. Dla przedrostka „mikro” należy unikać stosowania litery „u” zamiast „μ”. Znak „μ” można otrzymać przytrzymując lewy Alt i wpisując na klawiaturze numerycznej 0181 (podobnie „stopień”: Alt-0176). W celu uniknięcia „rozstrzelenia” liczb i ich jednostek zaleca się używanie „twardej” spacji pomiędzy liczbą i jednostką. Należy

sprawdzić, czy tytuły podrozdziałów/zakresów nie zostały jako pojedyncze wiersze na poprzedniej stronie oraz czy rysunki/tabele i ich tytuły nie zostały rozdzielone pomiędzy kolejnymi stronami.

Pracę drukuje się dwustronnie. Zaleca się wydruk w kolorze. Przed wydrukiem należy ponumerować strony (czcionka 10 pkt., dół strony, akapit wyśrodkowany). Strony tytułowej oraz strony z podziękowaniem nie numeruje się. Spis treści rozpoczyna się od strony numer 3 (lub 5, jeżeli zamieszczono podziękowania).

5. Podsumowanie i wnioski końcowe

1 ÷ 3 stron merytorycznie podsumowanie najważniejszych elementów pracy oraz wnioski wynikające z osiągniętego celu pracy. Proponowane zalecenia i modyfikacje oraz rozwiązania będące wynikiem realizowanej pracy.

Ostatni akapit podsumowania musi zawierać wykaz własnej pracy dyplomanta i zaczynać się od sformułowania: „Autor za własny wkład pracy uważa: ...”.

Załączniki

Według potrzeb zawarte i uporządkowane uzupełnienie pracy o dowolny materiał źródłowy (wydruk programu komputerowego, dokumentacja konstrukcyjno-technologiczna, konstrukcja modelu – makiety – urządzenia, instrukcja obsługi urządzenia lub stanowiska laboratoryjnego, zestawienie wyników pomiarów i obliczeń, informacyjne materiały katalogowe itp.).

Literatura

- [1] <http://weii.portal.prz.edu.pl/pl/materialy-do-pobrania>. Dostęp 5.01.2015.
- [2] Jakubczyk T., Klette A.: Pomiary w akustyce. WNT, Warszawa 1997.
- [3] Barski S.: Modele transmitancji. Elektronika praktyczna, nr 7/2011, str. 15-18.
- [4] Czujnik S200. Dokumentacja techniczno-ruchowa. Lumel, Zielona Góra, 2001.
- [5] Pawluk K.: Jak pisać teksty techniczne poprawnie, Wiadomości Elektrotechniczne, Nr 12, 2001, str. 513-515.

STRESZCZENIE PRACY DYPLOMOWEJ MAGISTERSKIEJ
PRÓWNANIE SILNIKÓW UNREAL ENGINE I UNITY POD
KĄTEM TWORZENIA GIER

Autor: Kacper Rudź, nr albumu: EF-16045

Opiekun: dr inż. Sławomir Samolej prof. PRz

Słowa kluczowe: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)

Treść streszczenia po polsku

MSC THESIS ABSTRACT
COMPARISON OF UNREAL ENGINE AND UNITY SOFTWARE
FOR GAME DEVELOPMENT

Author: Kacper Rudź, nr albumu: EF-16045

Supervisor: PhD Eng Sławomir Samolej

Key words: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)

Treść streszczenia po angielsku