

# LINUX W SZKOLE PODSTAWOWEJ

Dominika Wikiera  
ul. 22 Stycznia 4, 86-200 Chełmno  
[d.wikiera@sp2chelmno.pl](mailto:d.wikiera@sp2chelmno.pl)

*Abstract. Linux is now the most popular system on mobile devices and it is also very friendly operating system for desktops. In this paper we show how simply, without much money, it can be used in primary school.*

## 1. Wstęp

Idea zmiany systemu na Szkolny Remiks Ucznia pojawiła się, gdy firma Microsoft w kwietniu 2014 r. zdecydowała o zaprzestaniu wspierania systemu XP. Poszukując rozwiązań na miarę możliwości finansowych naszej szkoły i sprzętu, którym dysponowaliśmy: zestaw 10 komputerów Dell OptiPlex 760 (Core 2 Quad) procesor Intel C2Q 2,4GHz/1GB/160GB wraz z urządzeniami peryferyjnymi, 5 komputerów w salach, w których zostały zainstalowane tablice multimedialne – już wtedy, co najmniej 5 letnie, rozsądnym rozwiązaniem wydawał się system Linux. Był tylko jeden mały problem, nigdy go nie instalowałam, nigdy na nim nie pracowałam i ledwie znałam nazwy najpopularniejszych. Po poszukiwaniach w sieci okazało się, że istnieje prężnie działająca społeczność wielbicieli tego systemu. Strona linux.pl czy jakilinux.org oferuje przegląd różnych dystrybucji systemu. Linuxiarze.pl/dystrybucje-szukaj daje narzędzie do wyboru systemu, który będzie nam najbardziej odpowiadał. Możemy wpisać parametry sprzętu, na którym pracujemy, kraj pochodzenia systemu, wersję pulpitu, wskazać, do czego będziemy go najczęściej wykorzystywali. Moimi typami były od początku, Ubuntu oraz Mint. Przy czym okazuje się, że oba wywodzą się z tego samego źródła (Debiana) i dla mnie, jako laika, różnią się jedynie interfacem. Ubuntu ma wiele wersji np. Edubuntu czy Linux-EduCD. Ciekawym rozwiązaniem wydawała się chińska kompilacja przypominająca Windows XP. Jednak w trakcie poszukiwań trafiłam na stronę fwioo.pl Fundacji Wolnego i Otwartego Oprogramowania, która realizując projekt SWOI, czyli Strategia Wolnych i Otwartych Implementacji [1] wypracowała Szkolny Remiks Ucznia (SRU). Jest on przeznaczony do nauczania od II do IV etapu edukacyjnego. Dodatkowo, oprócz wersji Desktop (dla jednostek uczniowskich), zawiera implementację Serwer do zarządzania pracownią opartą na SRU. Obecnie korzystam

z wersji Marble Toy [2]. Zawiera ona programy niezbędne do realizacji podstawy programowej. Po instalacji kilku dodatkowych aplikacji, bardzo dobrze sprawdza się podczas zajęć komputerowych realizowanych na I etapie edukacyjnym.

## 2. FWIOO

Fundacja Wolnego i Otwartego Oprogramowania, oprócz łączenia sił i wspólnego działania w imię popularyzacji idei wolnego i otwartego oprogramowania ma na celu wspieranie:

- rozwoju demokratycznego państwa i społeczeństwa obywatelskiego, którego integralną częścią jest jego informatyzacja,
- transferu osiągnięć w dziedzinie wolnego oprogramowania do gospodarki [3]

Przeciwdziała wykluczeniu cyfrowemu poprzez promowanie swobodnego dostępu do narzędzi informatycznych. Postuluje wyrównywanie szans przez wybieranie wolnych rozwiązań technologicznych, m.in. w komunikacji administracja–obywatel.

Państwa takie jak Estonia czy jeden z kantonów w Szwajcarii, już dawno odkryły drzemiący w wolnym oprogramowaniu potencjał i korzystają na tym ich gospodarki. Natomiast wykorzystanie systemu Linux w edukacji powinno być jednym z priorytetów MEN, ponieważ już od najmłodszych lat powinniśmy promować darmowe (nie wykluczają i promują działania etyczne), pozwalające na realizację celów, takich jak rozrywka, nauka, praca narzędzi.

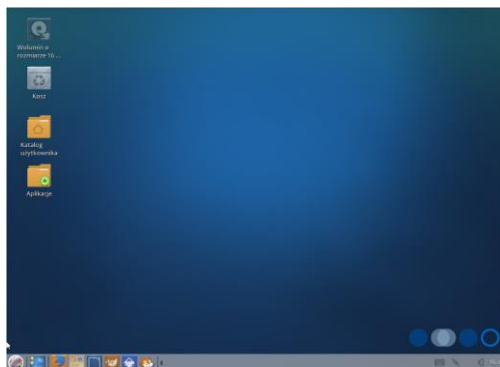
## 3. Twórca Szkolnego Remiksu Ucznia (SRU)

Mówiąc o SRU należy wspomnieć o Adamie Jurkiewiczu, który skompilował remiks oraz propaguje ideę Hakerów Edukacji [4]. Hakerzy rozwiązują problemy i tworzą rzeczy wierząc w wolność i wzajemną pomoc. Prowadzi firmę ABIX, oferującą rozwiązania oparte na Linuxie dla firm i edukacji oraz jest członkiem grupy Superbelfrzy.pl. Odpowiada również na zasadnicze pytanie: jak na czymś, co jest darmowe i każdy ma do tego dostęp – zarobić.

## 4. Praca nad zmianą systemu

Zaczęłam od instalacji SRU Desktop na komputerze uczniowskim, aby przekonać się, czy ta dystrybucja zostanie uruchomiona na tak archaicznym sprzęcie. Okazało się, iż nie ma przeciwwskazań. Następnie, po instalacji na kolejnych stacjach, przesłałam do konfigurowania serwera. Zainstalowanie go nie stanowiło najmniejszego problemu, jednak, gdy chciałam dokonać bardziej gruntownych zmian (było do tego konieczne użycie konta root), napotkałam na przeszkody. Dlatego zdecydowałam się na instalację pseudo serwera na zwykłym Ubuntu. Wybrałam ten

system ze względu na długie wsparcie i aktualizację kolejnych wersji, nie bez znaczenia było również środowisko Unity, które nadawało lekkości tej wersji.



**Rysunek 1.** Pulpit Szkolnego Remiksu Uczenia

Instalacja zarówno Ubuntu jak i systemów na komputery uczniowskie przebiega bardzo intuicyjnie i wymaga jedynie klikania przycisku Dalej oraz wyboru odpowiedniego języka.

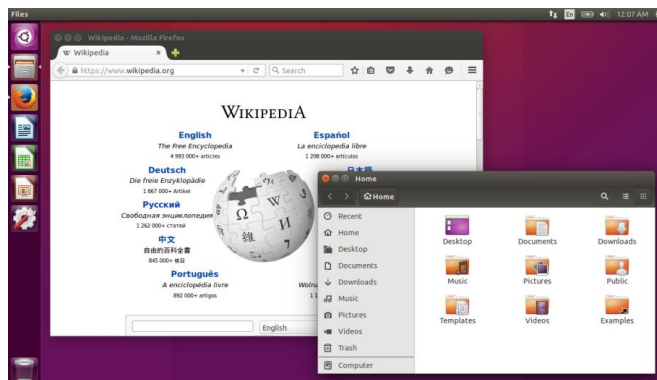
Pseudoserwer, był mi potrzebny do kontroli ruchu w sieci i konfiguracji programu Epopetes [5], który służy do nadzorowania pracy uczniów. Dzięki temu darmowemu rozwiązaniu:

- w każdym momencie nauczyciel widzi, co robi dany uczeń,
- może również prezentować swój ekran uczniom,
- opcja łączenia się z komputerem ucznia, umożliwia to zdalną pomoc,
- zablokowanie i odblokowanie pulpitu,
- oraz zdalne włączenie i wyłączenie komputera.



**Rysunek 3.** Okno programu Epopetes Server

Ze względu na typologię sieci (ruch „przepuszczany” jest przez serwer, lecz nie korzystam z innych jego funkcji) komputer nauczycielski nazywam pseudo-serwerem. W przyszłości będzie to pełnowartościowy serwer z możliwością przesyłania obrazu dysku na stacje uczniowskie bez konieczności używania tradycyjnych „skrzynek”.



Rysunek 4. Pulpit Ubuntu

## 5. Realizacja podstawy programowej

Wśród wielu programów wspomagających pracę na I etapie edukacyjnym skupię się na dwóch zestawach: GCompis oraz pySioGame, które zawierają pakiety ćwiczeń językowych, matematycznych, z dziedzin przyrodniczych i artystycznych. Uczeń kończący III klasę powinien umieć posługiwać się myszką i klawiaturą. Programy te oferują zestaw różnorodnych ćwiczeń do kształcenia tych kompetencji.



Rysunek 5. Okna GCompis przedstawiające zadania z klawiaturą i myszą

Zadania ułożone są od najprostszych do bardziej złożonych. Pierwsze ćwiczenie wymaga koordynacji i równoczesnego naciskania klawisza Shift, co może być okazją do omówienia jego funkcji. Grając w Proste litery warto najpierw zmienić język programu na angielski (brak polskich liter ułatwia zadanie w młodszych kla-

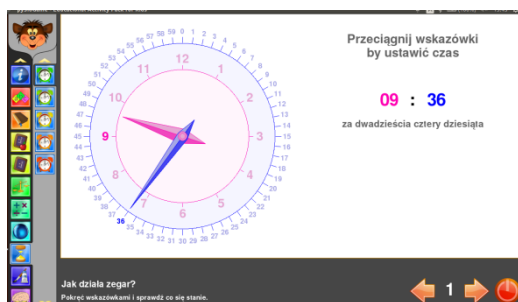
sach). Uczniowie ćwiczą umiejętność poruszania się po menu programu, wybierają odpowiednie opcje. W starszych klasach można pracować z polskimi literami. Korzystając z myszy uczeń ćwiczy: pojedyncze kliknięcie, przesuwanie myszki, trzymanie klawisza i przesuwanie myszy, klikanie wskazanego obiektu i dwuklik. W czasie, gdy uczniowie pokonują kolejne poziomy, nauczycielka może zaobserwować i skorygować postawę ucznia, trzymanie przez niego myszki.

Wśród wielu Zabaw w odkrywaniu są takie, jak: Poznaj system Braille'a, Zadania z kolorami, Labirynty, Ćwiczenia pamięciowe, Zabawy dźwiękami. Przedstawiona historia Louisa Braille'a umożliwia przekazanie treści wychowawczych oraz ćwiczenie pojęcia chronologii.



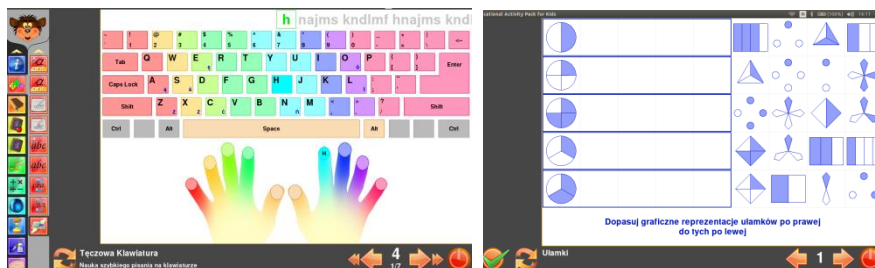
**Rysunek 6.** Okno programu GCompis, kategoria Zabawy w odkrywaniu

Przed tymi zajęciami zwykle przeprowadzam ćwiczenia w programie pySioGame, który zawiera zadania służące do lepszego zrozumienia czasu. Najpierw dzieci mogą przeciągać wskazówki, aby ustawić czas, do wyboru jest tarcza z liczbami arabskimi i rzymskimi. Kolejne to: odczytywanie czasu, najpierw pełne godziny, pół godziny i kwadrans do określenia, co do minuty. Ustawianie właściwego czasu na zegarze analogowym ze wskazania na cyfrowym, kolejny poziom to odczytanie zapisanej słownie godziny. Na zakończenie gra pt. Zegarkowa dopasowanka.



**Rysunek 7.** Okno programu pySioGame: Czas

Ćwiczenia z "pysia" wykorzystują również w klasie V podczas wprowadzenia do pisania na klawiaturze. Kolorowa klawiatura ułatwia pisanie bezwzrokowe. Ćwiczenia matematyczne, zarówno z algebry jak i geometrii, umożliwiają utrwalanie pojęć. Można ćwiczyć ułamki, procenty, działania pisemne na dużych liczbach. Co prawda uczniowie niezbyt chętnie podchodzą do "dziecięcej" stylistyki programu, jednak później sami wracają do niego i odkrywają kolejne zadania.



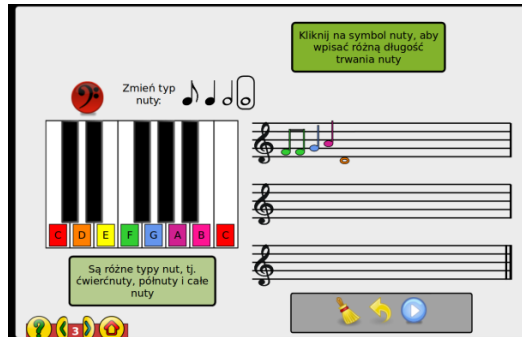
Rysunek 8. PySiOGame: Tęczowa klawiatura i Matematyka

Po przygotowaniu przez uczniów prezentacji na temat ciekawych miejsc w Polsce (jednym z nich jest Kanał Augustowski) i zaprezentowaniu ich, przechodzimy do zadań związanych z obsługą śluzy. Uczniowie mają za zadanie przeprowadzić Tuxa na drugi brzeg. Oszczędna grafika umożliwia skupienie się na istocie zadania, uczeń kształci umiejętność logicznego myślenia oraz praktycznie uczy się, na czym polega praca śluzy. Wśród zabaw eksperymentalnych znajdziemy również: Spadochroniarza, Dowiedz się o obiegu wody lub jak działa system produkcji prądu oparty o odnawialne źródła energii, Wstęp do grawitacji, Sterowanie rakieta lub łodzią podwodną.



Rysunek 8. GCompris: Zabawy eksperymentalne – Obsługa śluzy

Programy umożliwiają również twórcze wykorzystanie narzędzi do rysowania, tworzenia animacji i muzyki. Przy czym proste, intuicyjne menu oraz szybki efekt pracy zachęcają do podejmowania kreatywnych działań.



**Rysunek 9.** GCompis: Zabawy w odkrywaniu: Zabawy dźwiękami, Kompozycja dla pianina

Sądzę, że wspieranie, rozszerzanie, urozmaicanie przekazywanych na lekcji treści (oprócz realizacji wymagań zawartych w podstawie programowej) powinno być głównym zadaniem zajęć komputerowych.

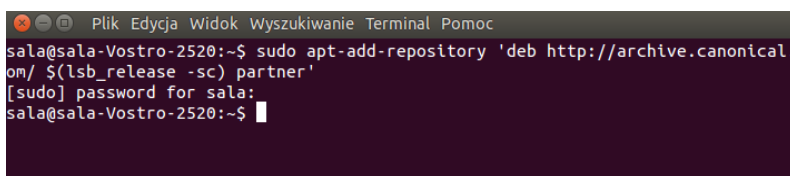
Kształtując umiejętności wymagane na II etapie edukacyjnym warto równocześnie uświadamić uczniom wartość, jaką jest przestrzeganie prawa autorskiego. Zastąpienie w przestrzeni szkolnej płatnych programów darmową alternatywą, daje uczniom możliwość pracy w domu na takim samym oprogramowaniu jak w szkole, bez konieczności łamania prawa. Powinno również wpłynąć na proces nauczania w ten sposób, że nauczyciel zwraca bacniejszą uwagę na konieczność kształcenia umiejętności a nie wybranego programu. Kształcąc uczniów dla ich dobrego funkcjonowania w przyszłości nie jesteśmy w stanie przewidzieć, z jakim interfejsem przyjdzie pracować uczniowi. Mamy natomiast wpływ na to, czy jego praca w przyszłości będzie starannie wykonana (umiejętność ewaluacji swojej pracy), czytelna, wykonana zgodnie z wymaganiami, z wykorzystaniem legalnych treści, z poszanowaniem/uznaniem autorstwa. Dlatego korzystajmy np. z pakietu LibreOffice. KolorPaint i GIMP w zupełności wystarczą dla realizacji podstawy programowej. Z mojego doświadczenia, dzieci bez zastrzeżeń pracują i realizują program za pomocą darmowych programów, bez problemu wykorzystując zdobyte umiejętności w innych programach. Największą przeszkodą jest przyzwyczajony do określonego menu nauczyciel, niechętnie zmieniający raz przygotowane materiały lub ćwiczenia.

Proponowane obecnie zmiany w podstawie programowej zakładają wprowadzenie programowania w trakcie zajęć komputerowych. Pracując z SRU otrzymujemy program Scratch w wersji 1.4, dostępna natomiast jest już wersja 2.0. Zawiera ona więcej klocków do tworzenia skryptów oraz edytor grafiki rastrowej i wektorowej (można go wykorzystać do pokazania uczniom różnicy między rodzajami obrazów).

Połączenie wersji desktopowej z kontem na serwerach `scratch.mit.edu` daje możliwość pracy bez dostępu do internetu oraz przenoszenia programu na nośnikach zewnętrznych. Warto więc mieć dostęp do nowszej wersji, jednak instalacja zalecana jest tylko dla niektórych wersji systemu Linux 32-bitowego (na SRU działa bez zarzutu) i wymaga podjęciu kilku niezbędnych działań.

## 5.1. Instalacja Adobe Air

Aby przeprowadzić instalację Scratch 2.0 musimy mieć zainstalowane środowisko Adobe Air. W tym celu należy uruchomić Terminal – wybierając ikonę lub naciśkając kombinację klawiszy `Ctrl+Alt+T`.



```
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
sala@sala-Vostro-2520:~$ sudo apt-add-repository 'deb http://archive.canonical.
om/ $(lsb_release -sc) partner'
[sudo] password for sala:
sala@sala-Vostro-2520:~$
```

Rysunek 9. Emulator terminala dla środowiska GNOME

Wpisujemy (wszystko w jednej linii, istotne są małe i wielkie litery oraz spacje) lub kopiujemy ze strony [6]:

```
sudo apt-add-repository 'deb
http://archive.canonical.com/ $(lsb_release -sc)
partner'
```

następnie

```
sudo apt-get update && sudo apt-get install
flashplugin-installer acroread
```

Użycie komendy `sudo` wymaga podania hasła administratora, które ustaliliśmy podczas instalacji systemu dla konta np. nauczyciela. Podczas wpisywania hasła nie widać liter ani gwiazdek – zatwierdzamy enterem.

Pobieramy Adobe Air wpisując:

```
wget
http://airdownload.adobe.com/air/lin/download/latest/Ad
obeAIRInstaller.bin
```



zmieniamy uprawnienia dla pliku:

```
chmod +x AdobeAIRInstaller.bin
```

i instalujemy:

```
sudo ./AdobeAIRInstaller.bin
```

W przypadku błędu, wyłączamy terminal i włączamy go ponownie. Wpisujemy:

```
sudo ln -s /usr/lib/i386-linux-gnu/libgnome-  
keyring.so.0 /usr/lib/libgnome-keyring.so.0
```

ponownie wpisujemy hasło i komendę:

```
sudo ln -s /usr/lib/i386-linux-gnu/libgnome-  
keyring.so.0.2.0 /usr/lib/libgnome-keyring.so.0.2.0
```

Ponownie instalujemy środowisko Adobe Air:

```
sudo ./AdobeAIRInstaller.bin
```

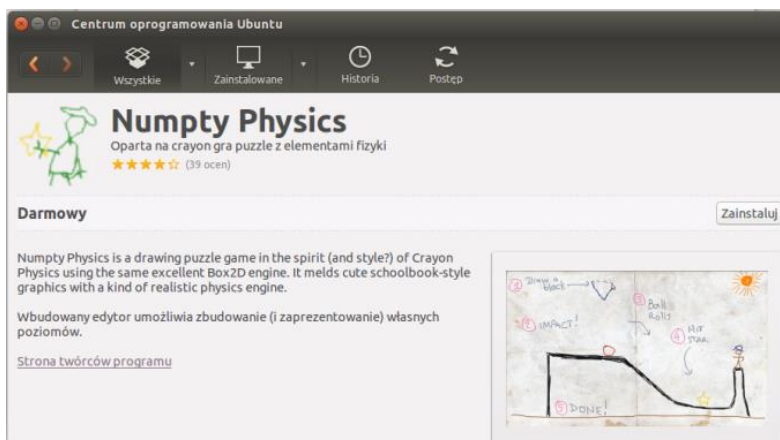
## 5.2. Instalacja Scratch 2.0

Pobieramy program ze strony [scratch.mit.edu/scratch2download](http://scratch.mit.edu/scratch2download) (dostępne są wersje dla systemu Mac OS X i starszych, Windows i Linux). Jesteśmy już w punkcie 2. ponieważ Adobe Air zainstalowaliśmy. Pobrany plik znajduje się w katalogu Pobrane. Ale to za mało informacji, aby przeprowadzić instalację. Struktura plików w systemie Linux wygląda inaczej niż w Windows, musimy więc dowiedzieć się, jaka jest ścieżka dostępu do pliku. W przeglądarce, przy pobraniu pliku, wybieramy opcję Pokaż w folderze. Na pliku klikamy prawym przyciskiem myszy i wybieramy Właściwości. Kopiujemy Położenie np. /home/sala/Pobrane. Wpisujemy komendę zmieniając /napisz/pełną/ścieżkę/dostępu na tą z Położenie. Należy zwrócić uwagę na wersję programu Scratch, którą pobraliśmy i ewentualnie zmienić numer.

```
sudo Adobe\ AIR\ Application\ Installer  
/napisz/pełną/ścieżkę/dostępu/Scratch-446.air
```

### 5.3. Centrum oprogramowania Ubuntu

Proces instalacji programów w Ubuntu przebiega zwykle w bardziej przyjazny sposób. W Centrum oprogramowania Ubuntu możemy wyszukać potrzebny nam program według kategorii lub wpisując jego nazwę. Zaznaczamy go i wybieramy Zainstaluj.



Rysunek 11. Okno Centrum oprogramowania Ubuntu

Polecam grę Numpty Physics czy Multiplication Puzzle. Natomiast aplikacje, takie jak Audacity, OpenShot, Pencil i ffDiaporama pomogą w realizacji zagadnień medialnych.

Zastosowanie systemu Linux i dostępnych dla niego darmowych programów zapewnia bogate środowisko nauki dla ucznia i stwarza możliwość realizacji podstawy programowej dla nauczyciela.

## 6. Zakończenie

Linux i szkoła podstawowa. Można zapytać: dlaczego, ale jak, po co? Bardzo złożony system operacyjny dla profesjonalnych informatyków, jak tego nauczyć małe dzieci? Tak samo jak systemu Windows – są okienka, podobne menu. Ma wiele zalet min:

- możliwość pracy na starszych komputerach,
- dostosowania go do swoich potrzeb,
- darmowe oprogramowanie.

W 2007 roku, 99, 29% urządzeń komputerowych łączących się z Internetem miało zainstalowany system Windows, ale świat urządzeń się zmienił. Teraz z Windows jest to tylko 67.53%. Upowszechnienie się tabletów i telefonów zmieniło proporcję na rzecz systemów Unix. Wzrost z 0.70% do 32.35% [7] Dla każdego, kto

jest zainteresowany karierą w technologii, znajomość Linuxa będzie zdecydowanym atutem. Poza tym, Linux jest wszechobecny:

- wykorzystują go międzynarodowe stacje kosmiczne,
- jest w nowych samochodach, takich jak Tesla i Cadillac,
- Google, Facebook, Twitter - oni wszyscy używają Linuksa,
- większość superkomputerów na świecie jest uruchomionych na Linuksie.

Spółeczeństwo bazujące na wiedzy, do którego dążymy, powinno uczyć się na otwartym oprogramowaniu, dającym możliwość kreowania swojego świata, tworzeniu go i dostosowywaniu do swoich potrzeb. Uczmy dzieci różnorodności poczynając od używania słów: edytor tekstu i arkusz kalkulacyjny – bez wpajania im konkretnej marki. Szkoła powinna uczyć i wychowywać – nie przygotowywać konsumenta. Wyrównywać szanse – nie zmuszać do piractwa. Używajmy, więc w szkołach otwartego i darmowego oprogramowania.

## Literatura

1. SWOI, strona przedstawiająca ideę i materiały wypracowane w ramach Strategii Wolnych i Otwartych Implementacji, <http://e-swoi.pl>
2. Linux.Portal.pl, portal poświęcony dyskusji nad dystrybucjami i programami dla systemu Linux, <http://www.linuxportal.pl/newsbar/sru-marble-toy-czyli-szkolny-remiks-uczni-dla-szkol-id177483>
3. Fundacja Wolnego i Otwartego Oprogramowania, <https://fwioo.pl>
4. Cyfrowa Szkoła – robotyka i programowanie, serwisy prowadzone przez państwa Jurkiewiczów, promujące *open source* w edukacji, <http://otwarta-edukacja.pl> i <http://cyfrowaszkoła.waw.pl>
5. Epoptes, strona twórców programu, zawierająca instrukcję instalacji oraz dokumentację programu, <http://www.epoptes.org>
6. Ubuntu-pomoc.org, portal udzielający praktycznych porad i oferujący sprawdzone rozwiązania dla systemu Ubuntu, [www.ubuntu-pomoc.org/ubuntu-12-04-instalacja-oprogramowania-adobe-air-flash-reader](http://www.ubuntu-pomoc.org/ubuntu-12-04-instalacja-oprogramowania-adobe-air-flash-reader)
7. Gemius, serwis międzynarodowej firmy badawczej, prekursora w dziedzinie badań Internetu i przez Internet w Europie, <http://www ranking.pl/pl/rankings/operating-systems-families.html>