

Wykorzystanie systemu planowania powtórek z elementami gamifikacji w nauczaniu podstaw programowania komputerów

The use of a spaced repetition system with gamification elements in teaching the basics of computer programming

Jakub Swacha

Uniwersytet Szczeciński | jakubs@uoo.univ.szczecin.pl

Abstract: The difficulty in teaching computer programming is the reason for trying various novel techniques and tools in this field. This paper describes an experiment in which the Flippi web platform, merging functionality of a spaced repetition system, a social network, and containing a number of gamification elements was used. Although the experiment failed to attain the planned research aims, it eventually led to drawing conclusions that may be useful when a similar tool (having elements of gamification) is introduced for supporting education.

Keywords: gamification, flashcards, spaced repetition system, computer programming education

1. Wstęp

Nauczanie programowania komputerów uważa się za szczególnie trudne (Robins, Rountree, Rountree, 2003; zob. też prace tam cytowane), co jest powodem wypróbowywania wielu rozmaitych podejść. Żadne z nich, jak dotąd, nie okazało się optymalne, choć nie brakowało rozwiązań wykorzystujących – jako przedmiot programowania – gry służące zdobyciu (Leutenegger, Edgington, 2007) lub utrzymaniu (Swacha, 2010) zainteresowania uczniów.

Nową szansę na poprawę tego stanu rzeczy przynosi gamifikacja, która, wybiórczo stosując rozwiązania znane z gier, wspiera zarówno motywację zewnętrzną (poprzez umożliwienie gromadzenia punktów, osiągania kolejnych poziomów rozwoju oraz zdobywania odznaczeń i czołowych pozycji w rankingu), jak i wewnętrzną (poprzez wyzwania, rozbudzanie ciekawości, dawanie poczucia sprawowania kontroli i wprowadzanie aspektu fantastycznego). Gamifikacja nierzadko wywołuje w uczestniku zadowolenie z sukcesu, bogactwa, pozycji czy sławy – uczucie, którego wielu osobom brakuje w życiu codziennym (Malone, Lepper, 1987, s. 230, 242).

Nie powinno zatem dziwić rosnące zainteresowanie gamifikacją w edukacji (Nah, Zeng, Telaprolu, Ayyappa, Eschenbrenner, 2014; zob. też prace tam cytowane). Wyrazem tego zainteresowania jest również niniejszy artykuł, w którym opisano eksperyment z wykorzystaniem w nauczaniu podstaw programowania – na Wydziale Nauk Ekonomicznych i Zarządzania Uniwersytetu Szczecińskiego – systemu Flippi, w oryginalny sposób łączącego elementy gamifikacji i naukę z użyciem fiszek (Florczak, 2014), oraz zebrane doświadczenia i wynikające z nich wnioski.

Artykuł podzielono na pięć sekcji. Po niniejszym wstępie pokrótce przybliżona zostanie koncepcja gamifikacji, jej walory użyteczne z perspektywy edukacji i dotychczasowy zakres wykorzystania we wspomaganiu nauczania programowania komputerów. Dalej opisana będzie platforma Flippi, użyta w eksperymencie, który z kolei omówi sekcja czwarta. Ostatnią poświęcono zaś doświadczeniom uzyskanym w tym przedsięwzięciu i wnioskom z nich płynącym; odnoszą się one nie tylko do gamifikacji nauczania programowania, ale także do gamifikacji nauczania w ogóle.

2. Gamifikacja w nauczaniu programowania

W literaturze polskojęzycznej gamifikacja znana jest także jako grywalizacja (Tkaczyk, 2012) i gryfikacja (Starzyński, 2012), choć posługiwanie się tym ostatnim terminem odradza się – ze względu na jego niezgodność z terminologiczną zasadą jednorodności, według której nazwa powinna być utworzona w całości albo ze źródłosłów rodzimych, albo obcych (Mazur, 1961).

Według najbardziej rozpowszechnionej definicji, sformułowanej przez Sebastiana Deterdinga i współpracowników, gamifikacja polega na „użyciu elementów projektowych charakterystycznych dla gier w kontekstach innych niż gry” (Deterding, Dixon, Khaled, Nacke, 2011). Z kolei Andrzej Marczewski (2012) definiuje ją nieco inaczej – jako „zastosowanie metafor [pochodzących – J. S.] z gier w zadaniach [wykonywanych – J. S.] w prawdziwym życiu, tak by wpłynąć na zachowanie, poprawić motywację oraz zwiększyć zaangażowanie”. Podobnie według Karla M. Kappa (2012) gamifikacja to „użycie mechanizmów, estetyki i sposobu myślenia zaczerpniętych z gier, by angażować ludzi, motywować do działania, pobudzać do nauki i rozwiązywać problemy”. Wreszcie według definicji procesowej Kevina Werbacha (2014) gamifikacja to „proces czynienia aktywności bardziej przypominającymi gry”.

Jako szczególnie nadające się do zastosowania w nauczaniu wymienić można – za platformą Knewton (2008) – takie elementy gamifikacji jak:

- punkty – pokazujące bieżący postęp uczestnika i pomagające kierować jego wysiłki na bardziej właściwe (zdaniem projektanta) tory (poprzez zwiększanie lub zmniejszanie nagród punktowych przysługujących za różne rodzaje aktywności);
- poziomy – w zwężły i przejrzysty sposób odzwierciedlające całość dotychczasowych osiągnięć użytkownika i pozwalające odblokowywać kolejne treści w miarę jego postępów;
- rankingi – oddające relatywny postęp uczestnika względem innych;
- osiągnięcia i odznaczenia za nie – stanowiące wyraz docenienia postępów uczestnika;
- ustalone terminy – zachęcające do regularnego i konsekwentnego uczenia się;

- współpraca – służąca kształceniu umiejętności współdziałania w grupie, pracy w warunkach podziału zadań i odpowiedzialności oraz myślenia zespołowego;
- wzmacnianie znaczenia – nadawanie powszednim działaniom i sukcesom aury wielkości i wyjątkowości;
- wiralność – polegająca na stymulowaniu uczestników do angażowania innych potencjalnych użytkowników;
- premie – wprowadzające element zaskoczenia poprzez dodatkowe, niespodziewane nagrody;
- limity czasowe – uczące rozwiązywania problemów w sytuacji stresu, tak jak w realnym życiu;
- odkrywanie – uczestnik, w miarę zdobywania doświadczenia, dowiaduje się o istnieniu nowych dróg i sposobów działania;
- lęk przed utratą – stanowiący osobny powód do kontynuowania rozgrywki;
- gra bez końca – dająca możliwość dalszego szlifowania własnych umiejętności nawet uczestnikom, którzy osiągnęli już poziom ekspercki;
- synteza – wymagająca wykorzystania umiejętności różnego rodzaju do rozwiązania pojedynczego problemu.

Historycznie pierwszy, poprzedzający nawet pojawienie się samego terminu, przykład posłużenia się elementami charakterystycznymi dla gamifikacji (w postaci np. rankingów opartych na liczbie poprawnych i niepoprawnych zgłoszeń oraz turniejów rozgrywanych w ustalonych terminach) w nauczaniu programowania komputerów stanowią systemy automatycznej oceny poprawności programów, nazywane też sędziami online, takie jak np. polski Sphere Online Judge (Kosowski, Małafiejski, Noiński, Pomykalski, 2005).

W roku 2010 Microsoft Research udostępnił platformę Pex4Fun (Xie, de Halleux, Tillman, Schulte, 2010), będącą swego rodzaju rozwinięciem koncepcji „sędziego *online*” o elementy dydaktyczne (w postaci kompendium podstaw języka C# i zasad programowania kontraktowego), a także o (ograniczone) wsparcie dla nauczycieli (poprzez możliwość podglądu osiągnięć uczniów). Jej następcą jest platforma CodeHunt (Tillmann, de Halleux, Xie, Bishop, 2014), w której zadania programistyczne zorganizowano w sektory

i poziomy o rosnącej trudności (nadając całemu kursowi formę przypominającą grę), proces testowania (wykonywania) programu zyskał na interaktywności, a uczestnik otrzymuje bardziej szczegółowe wskazówki dotyczące koniecznych poprawek. Zmiany te uczyniły z CodeHunt swoisty samouczek programowania.

Taki właśnie charakter od początku miała Codecademy – spośród omawianych platform najbardziej popularna, bo już w kwietniu 2014 wykorzystywana przez ponad 24 miliony osób (Colao, 2014). W chwili pisania tych słów były tam dostępne kursy podstaw programowania w dziewięciu językach oraz kilkanaście kursów specjalistycznych. Użytkownicy nagradzani są punktami za poprawne wykonanie ćwiczeń, a także odznakami za ukończenie kursów lub wybranych lekcji, za zebranie pewnej liczby punktów czy też za długość okresu, w którym codziennie korzystają z platformy.

Codecademy od niedawna udostępnia nauczycielom materiały edukacyjne oraz umożliwia śledzenie postępów uczniów, jednak oryginalnie pomyślana była jako miejsce zdobywania i doskonalenia umiejętności przez zawodowych programistów. Z zamiarem wspierania nauczania programowania w szkołach stworzono natomiast platformę Code Avengers (2012). Jej grupę docelową stanowią nastolatki, stąd utrzymana w konwencji komiksowej grafika oraz specyficzny język. Z kolei dla młodszych dzieci przygotowano kurs programowania Khan Academy (2014), zawierający m.in. takie elementy gamifikacji jak punkty energii, które otrzymuje się za ukończenie zadań, i odznaki, zdobywane wraz z osiągnięciem kolejnych poziomów.

Wszystkie wspomniane wyżej platformy charakteryzują się brakiem warstwy fabularnej – a w najlepszym razie szcztątkową jej formą. Przykład odmiennego podejścia stanowi platforma Code Combat (2013), będąca połączeniem kursu programowania i gry RPG; gracz steruje w niej bohaterem, pisząc krótkie programy określające jego zachowanie na danym poziomie. Dostępny jest też interesujący tryb rozgrywki dla dwóch osób („Dungeon Arena”), w którym każda z nich steruje bohaterem i sługami jednej z walczących ze sobą stron, zdobywając punkty decydujące o pozycji w rankingu. W odróżnieniu od wcześniej wymienionych platforma ta powstaje jako oprogramowanie o otwartym kodzie źródłowym, a wśród udostępnionych wersji językowych znajduje się także polska.

Gamifikację nauczania programowania często realizuje się z użyciem innych narzędzi niż wspomniane już platformy edukacyjne, takich jak specjalnie rozszerzone (zgamifikowane) wersje oprogramowania wykorzystywanego wcześniej (zob. np. Neve, Livingstone, Hunter, Edwards, Alsop, 2014) lub nowe narzędzia tworzone specjalnie w tym celu (zob. np. Paspallis, 2014). Do tej drugiej grupy zaliczyć należy zgamifikowaną platformę e-learningową wspomagającą nauczanie programowania obmyśloną przeze mnie, której implementację pod moim kierunkiem rozpoczęła grupa studentów Wydziału Technicznego Państwowej Wyższej Szkoły Zawodowej im. Jakuba z Paradyża w Gorzowie Wielkopolskim (Swacha, Baszuro, 2013). Choć z powodów pozamerytorycznych projekt ten musiał zostać zawieszony (planuje się jego wznowienie), w tejże jednostce powstało narzędzie służące temu samemu celowi i także wykorzystujące elementy gamifikacji. Jest nim platforma Flippi, zaprojektowana i zaimplementowana przez Rafała Florczaka jako część praktyczna pracy inżynierskiej, nad którą sprawowałem opiekę promotorską (2014).

3. Platforma Flippi

Główny cel platformy Flippi stanowi wspomaganie nauki języka programowania poprzez posłużenie się zestawami fiszek – kart zawierających pary: pytanie–odpowiedź. Wykorzystanie tego rodzaju przedmiotów przy uczeniu się szeroko rozpropagował Sebastian Leitner (2011), a do znaczącego rozwoju komputerowych implementacji takiego sposobu zapamiętywania danych wielce przyczynił się Piotr A. Woźniak, twórca algorytmu SuperMemo (Woźniak, Gorzelańczyk, 1994), służącego jako podstawa działania wielu systemów planowania powtórek (zob. np. Elmes, 2014).

Jako główne funkcje platformy Flippi wymienić należy tworzenie kursów (dotyczących wybranego przez użytkownika tematu) i przypisanych do nich fiszek oraz tryb nauki, w którym wyświetlane są pytania, następnie odpowiedzi, a student ocenia stan swojej wiedzy. Wyboru kolejnej fiszki do prezentacji dokonuje algorytm wyznaczania powtórek. W przypadku Flippi jest to autorski algorytm Florczaka; starał się on dostosować mechanizm do specyfiki przedmiotu – programowania komputerów – zwracając uwagę (2014, s. 22–23) m.in. na to, że:

- nauka programowania ma bardzo intensywny przebieg (dużo nowej wiedzy w krótkim czasie);
- proces nauczania części informacji (fiszki) musi być szybko zakończony, gdyż kolejne partie materiału wymagają doskonałego opanowania poprzednich;
- zawartość fiszek może być złożona, np. przykłady implementacji algorytmów w postaci schematów blokowych i pseudokodu;
- opanowanie istotnej części materiału nie polega na zapamiętaniu przykładów, ale raczej na zrozumieniu kryjących się za nimi zasad.

W rezultacie zaimplementowany algorytm charakteryzuje się uproszczonym, binarnym modelem decyzyjnym – użytkownik, widząc pytanie, odpowiada: „Umiem” lub „Nie umiem”, a zasady wyznaczania powtórek (Florczak, 2014, s. 24) są następujące:

- jeżeli udzielono odpowiedzi „Nie umiem”:
 - po raz pierwszy, drugi lub trzeci w danym dniu: pytanie zostanie zadane użytkownikowi ponownie jeszcze tego samego dnia;
 - po raz czwarty w danym dniu: pytanie zostanie zadane użytkownikowi ponownie następnego dnia;
- jeżeli udzielono odpowiedzi „Umiem”:
 - po raz pierwszy, drugi lub trzeci: pytanie zostanie zadane użytkownikowi ponownie po upływie odpowiednio jednego, dwóch lub trzech dni;
 - po raz czwarty: system uznaje, że użytkownik opanował część materiału, pytanie nie będzie zadawane ponownie.

Platforma zawiera wewnętrzną sieć społecznościową, pozwalając użytkownikom na wysyłanie i przyjmowanie zaproszeń do grona znajomych, udostępnianie im kursów, do których sami mają dostęp, oraz śledzenie postępów tych znajomych.

Aby zapewnić jej niezależność od platformy systemowej, platforma Flippi została zaimplementowana jako aplikacja webowa i może być obsługiwana za pomocą dowolnej przeglądarki internetowej obsługującej HTML5, CSS3 i JavaScript. Prócz wymienionych technologii przy budowie Flippi wykorzystano języki PHP i Sass, frameworki jQuery i CodeIgniter, a także system zarządzania bazą danych MySQL. W latach 2014–2015 platforma była dostępna bezpłatnie pod adresem: <http://florzak.unixstorm.org>, obecnie trwa jej przebudowa.

Tym, co wyróżnia Flippi wśród istniejących bezpłatnych systemów wspomagających naukę z użyciem fiszek, takich jak Anki (Elmes, 2006) czy Mnemosyne (2006), są zawarte w niej elementy gamifikacji. Jako najważniejsze z nich należy traktować:

- punkty – uzyskiwane za opanowanie fiszek (przy czym liczba przyznanych punktów zależy od regularności w przestrzeganiu terminów powtórek) oraz za osiągnięcia (ukończone wyzwania);
- wyzwania – dotyczące różnorodnej aktywności na platformie, np. dodawania kursów czy poszerzania grona znajomych;
- informacja zwrotna – obejmująca różne aspekty aktywności na platformie (np. ukończone wyzwania czy aktywność w sieci społecznościowej);
- poziomy rozwoju – osiągane wraz ze zgromadzeniem odpowiedniej liczby punktów;
- śledzenie postępu – graficznie ukazujące, jak wiele dzieli użytkownika od wejścia na wyższy poziom;
- awatary – żartobliwe przedstawienia zwierzątek wraz z krótkimi satyrycznymi historyjkami specyficznymi dla każdego z nich; nie są wybierane przez użytkownika, ale zmieniają się wraz z osiąganiem przez niego kolejnych poziomów rozwoju (od lamy – stanowiącej nawiązanie do słowa „lamer”, będącego w slangu informatycznym określeniem słabego programisty – do, mającego oczywiste konotacje, lwa), czyli pełnią we Flippi funkcję zbliżoną do odznaczeń;
- ranking – ukazujący listę użytkowników uporządkowaną według ich aktualnej w danej chwili liczby punktów.

4. Założenia, plan i przebieg eksperymentu

Chociaż platformę Flippi zaprojektowano do wspomagania samodzielnej nauki programowania, po kilku modyfikacjach (zdefiniowanie kursów domyślnych, do których automatycznie przypisywani byli nowo zarejestrowani użytkownicy, zapewnienie prowadzącemu możliwości wglądu w aktywność i postępy poszczególnych studentów, korekta punktacji) stała się ona gotowa do wykorzystania w realiach kursu uczelnianego.

Aby ocenić przydatność Flippi w tej nowej roli, postanowiono dokonać jej próbnego wdrożenia w formule eksperymentu. Udział w nim – na zasadzie dobrowolności (choć zapowiedziano doliczenie dodatkowych punktów do oceny końcowej studentów uczestniczących w przedsięwzięciu) – zaproponowano 13 października 2014 obecnym na wykładzie studentom kierunku „Zarządzanie i inżynieria produkcji” na Wydziale Nauk Ekonomicznych i Zarządzania Uniwersytetu Szczecińskiego.

W planie eksperymentu znalazły się:

1. Test wstępny z teorii (obejmujący materiał z pierwszego wykładu z przedmiotu „Podstawy programowania”).
2. Test wstępny z praktyki (obejmujący materiał z trzech pierwszych zajęć laboratoryjnych z przedmiotu „Podstawy programowania”).
3. Aktywne korzystanie przez studentów z platformy Flippi (pierwotnie przyjęto, że w okresie do 1 listopada 2014).
4. Ankietę na temat oceny platformy Flippi i jej przydatności w nauce podstaw programowania, przeprowadzoną wśród studentów uczestniczących w eksperymencie.
5. Test końcowy z teorii (zgodny zakresem pytań z testem wstępnym).
6. Test końcowy z praktyki (zgodny zakresem pytań z testem wstępnym).

Najważniejszy rezultat eksperymentu stanowić miały wyniki ankiety (pozyskane na etapie 4), w których spodziewano się znaleźć odpowiedź, czy wspomaganie nauczania programowania z użyciem Flippi jest z punktu widzenia studentów uzasadnione, a jeśli tak – w jaki sposób z niej korzystać i ewentualnie jakich zmian w niej dokonać.

Testy wprowadzono do eksperymentu w przekonaniu, że będą one stanowiły element motywujący do poważnego potraktowania nauki w jego ramach. Dodatkowo ich wyniki miały pozwolić porównać efekty uczenia się z pomocą Flippi w zakresie teorii i praktyki programowania.

Niestety, pierwotnie założone cele eksperymentu nie zostały osiągnięte, o czym przesądziła zbyt mała liczba studentów, którzy zdecydowali się wziąć w nim udział (nawet mimo dwukrotnego przedłużania terminu jego zakończenia, nie przyniosło to bowiem spodziewanego przyrostu liczby użytkowników):

- tylko 7 osób rozwiązało testy wstępne (26% powiadomionych);
- tyle samo osób rozpoczęło naukę na platformie Flippi, jednak tylko 3 spośród tych 7 osób (43%) korzystało z platformy tak regularnie, by osiągnąć znaczące postępy;
- tylko 5 osób rozwiązało testy końcowe, a 4 wypełniły ankietę.

Uzyskane wyniki testów wstępnych, choć pochodzą od bardzo małej liczby studentów, wskazują na:

- dobre przyswojenie przez nich treści wykładów (we wstępnym teście teoretycznym uzyskali oni średnio 7,24 na 10 możliwych punktów);
- bardzo słabe opanowanie przez nich umiejętności ćwiczonych na zajęciach laboratoryjnych (we wstępnym teście praktycznym uzyskali średnio 1,67 na 10 możliwych punktów).

Wśród osób, które rozwiązały zarówno testy wstępne, jak i końcowe, odnotowano poprawę wyników średnio o:

- 1,38 punktu (19%) w przypadku zagadnień teoretycznych (co wskazuje na bardzo dobre opanowanie materiału);
- 1 punkt (60%) w przypadku zagadnień praktycznych (co, niestety, nadal oznacza bardzo słabe opanowanie materiału).

Powyższe wyniki podano, oczywiście, informacyjnie, gdyż ze względu na małą liczbę uczestników testu nie można wyciągnąć wniosków co do efektywności nauki teorii i praktyki programowania z wykorzystaniem Flippi.

Ankieta była anonimowa i składała się z pięciu pytań, na które ankietowani udzielali odpowiedzi w pięciostopniowej skali Likerta, oraz z dwóch pytań otwartych. Spośród czterech ankietowanych:

- dwie osoby przyznały zdecydowanie, że „Flippi może poprawić skuteczność uczenia się przedmiotu takiego jak «Podstawy programowania»”; pozostałe udzieliły odpowiedzi neutralnej;
- trzy osoby (w tym jedna zdecydowanie) potwierdziły, że „Flippi bardziej nadaje się do wspomaganie uczenia się teorii niż praktyki programowania”; jedna udzieliła odpowiedzi neutralnej;
- trzy osoby przyznały (w tym dwie zdecydowanie), że „Elementy gamifikacji we Flippi (awatary, punkty, rankingi) zwiększają motywację do nauki”; jedna udzieliła odpowiedzi neutralnej;
- jedna osoba zgodziła się z tym, że „Efektywna nauka z pomocą Flippi wymaga więcej czasu”, niż student może „na to przeznaczyć”; jedna się nie zgodziła, a dwie udzieliły odpowiedzi neutralnej;

- dwie osoby (w tym jedna zdecydowanie) zadeklarowały: „Zamierzam korzystać z programu Flippi, przygotowując się do zaliczenia”; pozostałe udzieliły odpowiedzi neutralnej.

Zdanie: „Najbardziej podobało mi się we Flippi...”, uzupełniano następująco: „powtarzanie tego, czego nie umiałam, aż do skutku”, „ranking”, „możliwość korzystania na telefonie w każdym miejscu, nawet w tramwaju”. Żadna z ankietowanych osób nie dokończyła zdania: „Najbardziej nie podobało mi się we Flippi...”.

5. Podsumowanie

Jakkolwiek – ze względu na zbyt małe zaangażowanie studentów – eksperyment nie spełnił oczekiwań badawczych, interesujących wniosków może dostarczyć analiza domniemywanych przyczyn jego niepowodzenia. Wymienić tu należy czynniki takie jak:

- niewłaściwy moment rozpoczęcia:
 - początek semestru, a nie koniec (gdy studenci przygotowują się do zaliczeń);
 - na drugim, a nie na pierwszym wykładzie (na którym studenci oczekują informacji organizacyjnych);
- niewłaściwy czas przeprowadzenia:
 - zbyt krótki okres (oryginalnie około dwóch tygodni);
 - koniec października (przerwa w wykładach wynikająca ze świąt 1 i 11 listopada utrudniła motywowanie studentów do udziału w przedłużonym eksperymencie);
- niewłaściwy kontekst:
 - wykład, a nie zajęcia laboratoryjne (co więcej, osoba je prowadząca nie została nawet poinformowana o eksperymencie);
 - studentom powiedziano, że chodzi o eksperyment, a nie o standardowy element nauki przedmiotu;
- niewłaściwa grupa docelowa: z poczynionych obserwacji wynika, że studenci kierunku „Zarządzanie i inżynieria produkcji” w większości nie uważają programowania za umiejętność istotną dla ich przyszłej kariery, a co za tym idzie, nie są zainteresowani jej

opanowaniem w stopniu wykraczającym poza wymogi uzyskania zaliczenia przedmiotu¹;

- niewłaściwy dobór materiału:
 - zbyt duża ilość (trzy zestawy liczące łącznie ponad 50 fiszek);
 - zbytnia trudność (w ciągu pierwszego miesiąca nauki większość studentów nie opanowuje praktycznej umiejętności programowania nawet w stopniu podstawowym);
- niewłaściwe zasady punktacji:
 - niejasne dla uczestników zasady punktacji na platformie i zasady nagradzania uczestnictwa w eksperymencie w kontekście uzyskania zaliczenia przedmiotu;
 - zbyt rygorystyczne zasady punktacji (pierwsze punkty za opanowanie fiszki użytkownik mógł dostać dopiero po około tygodniu regularnego korzystania z narzędzia – brak szybkiej informacji zwrotnej).

Na bazie powyższej listy można zaproponować następujące sugestie dotyczące wprowadzania analogicznych do Flippi pomocy w tok pracy dydaktycznej:

- Należy robić to już na pierwszych zajęciach, a jeśli jest to niemożliwe, rozważyć wstrzymanie się do momentu, w którym sami studenci – z powodu zbliżającego się terminu zaliczenia przedmiotu czy kolokwium – poszukiwać będą narzędzi wspomagających naukę.
- Studenci muszą mieć dłuższy czas na zapoznanie się z narzędziem – należy tu uwzględnić także możliwy brak ich aktywności w okresach wolnych od zajęć dydaktycznych.
- Narzędzie wspomagające naukę w pewnym aspekcie powinno być wdrażane we właściwym kontekście (którym np. dla umiejętności praktycznych są zajęcia ćwiczeniowe lub laboratoryjne, a nie wykład) i z udziałem osoby prowadzącej daną formę zajęć.

.....

¹ Zapewne lepszą grupę odbiorców stanowiliby studenci funkcjonującego na tym samym wydziale kierunku „Informatyka i ekonometria”. Jednak w ich planie studiów analogiczny przedmiot występuje dopiero w semestrze letnim, a omawiany eksperyment przeprowadzony został w semestrze zimowym m.in. po to, by w letnim można już było wykorzystać jego wyniki.

- Dobór materiału powinien odpowiadać potencjałowi opanowywania wiedzy i umiejętności przez studentów.
- Zasady punktacji powinny być proste i przejrzyste, a punkty – spełniać funkcję szybkiej informacji zwrotnej.

Podziękowania

Autor składa podziękowanie twórcy oprogramowania Flippi, inż. Rafałowi Florczakowi, za pomoc w przeprowadzeniu eksperymentu, w szczególności za wiele poprawek programistycznych, które dostosowały platformę do jego wymogów.

Literatura

- Code Avengers (2012). Online: <<http://www.codeavengers.com>>. Data dostępu: 31 grudnia 2014.
- Code Combat (2013). Online: <<http://codecombat.com>>. Data dostępu: 31 grudnia 2014.
- Colao, J.J. (23 kwietnia 2014). With 24 Million Students, Codecademy Is Bigger Than You Thought. Online: <<http://www.forbes.com/sites/jjcolao/2014/04/23/with-24-million-students-codecademy-is-bigger-than-you-thought>>. Data dostępu: 31 grudnia 2014.
- Deterding, S., Dixon, D., Khaled, R., Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining „Gamification”. W: *Proceedings of the 15th International Academic MindTrek Conference Envisioning Future Media Environments* (s. 9–15). New York: Association for Computing Machinery. Online: <<http://www.cs.auckland.ac.nz/courses/compsci747s2c/lectures/paul/definition-deterding.pdf>>.
- Elmes, D. (2006). Anki. Online: <<http://ankisrs.net>>. Data dostępu: 31 grudnia 2014.
- Florczak, R. (2014). *Interaktywny system do nauki języka programowania*. Gorzów Wielkopolski: niepublikowana praca dyplomowa.
- Kapp, K. M. (2012). *The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education*. San Francisco: Pfeiffer.

- Khan Academy (2014). *Hour of Code*. Online: <<http://www.khanacademy.org/computing/hour-of-code>>. Data dostępu: 31 grudnia 2014.
- Knewton (2008). *The Gamification of Education*. Online: <<http://www.knewton.com/gamification-education>>. Data dostępu: 31 grudnia 2014.
- Kosowski, A., Małafiejski, M., Noiński, T., Pomykalski, P. (2005). Zintegrowany system do automatycznej oceny rozwiązań oraz prowadzenia zajęć laboratoryjno-projektowych: Sphere Online Judge. *Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej. Technologie Informacyjne*, 3(2), 523–528.
- Leitner, S. (2011). *Naucz się uczyć. Droga do sukcesu* (tłum. M. Urban). Gdańsk: Wydawnictwo „Cztery Głowy”.
- Leutenegger, S., Edgington, J. (2007). A „Games First” Approach to Teaching Introductory Programming. *ACM SIGCSE Bulletin*, 39(1), 115–118. Online: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.158.6814&rep=rep1&type=pdf>>. Data dostępu: 12 października 2016.
- Malone, T. W., Lepper, M. R. (1987). Making Learning Fun: A Taxonomy of Intrinsic Motivations for Learning. W: R. E. Snow, M. J. Farr (red.), *Aptitude, Learning, and Instruction. Volume 3: Conative and Affective Process Analyses* (s. 223–253). Hillsdale: Erlbaum.
- Marczewski, A. (2012). *Gamification: A Simple Introduction*. Raleigh: Lulu.
- Mazur, M. (1961). *Terminologia techniczna*. Warszawa: Wydawnictwa Naukowo-Techniczne.
- Mnemosyne (2006). Online: <<http://mnemosyne-proj.org>>. Data dostępu: 31 grudnia 2014.
- Nah, F., Zeng, Q., Telaprolu, V. R., Ayyappa, A. P., Eschenbrenner, B. (2014). Gamification of Education: A Review of Literature. W: F. F.-H. Nah (red.), *HCI in Business: First International Conference, HCIB 2014, held as part of HCI International 2014, Heraklion, Crete, Greece, June 22–27, 2014. Proceedings* (s. 401–409). Cham: Springer. Online: <http://link.springer.com/chapter/10.1007%2F978-3-319-07293-7_39#page-1>.
- Neve, P., Livingstone, D., Hunter, G., Edwards, N., Alsop, G. (2014). More than just a game: Improving students’ experience of learning programming through gamification. Online: <<http://www.heacademy.ac.uk/system/files/comp-224-p.pdf>>. Data dostępu: 13 października 2016.
- Paspallis, N. (2014). A Gamification Platform for Inspiring Young Students to Take an Interest in Coding. W: V. Strahonja, N. Vrčėk., D. Plantak

- Vukovac, C. Barry, M. Lang, H. Linger, C. Schneider (red.), *Information Systems Development: Transforming Organisations and Society through Information Systems (ISD2014 Proceedings)*. Online: <<http://aisel.aisnet.org/isd2014/proceedings/Education/1>>. Data dostępu: 31 grudnia 2014.
- Robins, A., Rountree, J., Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137-172. Online: <<http://home.cc.gatech.edu/csed/uploads/2/robins03.pdf>>. Data dostępu: 13 października 2016.
- Starzyński, S. (2012). Gryfikacja - modny termin nadużywany przez marketerów. W: A. Świtkowska, D. Sokołowski, M. Małecki, A. Wilma (red.), *Raport Game Industry Trends 2012* (s. 46-47). Warszawa: Nowy Marketing. Online: <<http://socialpress.pl/wp-content/uploads/2012/10/Raport-Game-Industry-Trends-2012.pdf>>. Data dostępu: 13 października 2016.
- Swacha, J. (2010). Nowe rozwiązania w dydaktyce programowania komputerów. W: M. Kowalski, A. Olczak (red.), *Edukacja w przebiegu życia. Od dzieciństwa do starości* (s. 115-126). Kraków: Impuls.
- Swacha, J., Baszuro, P. (2013). Gamification-based e-learning platform for computer programming education. W: N. Reynolds, M. Webb (red.), *Learning While We Are Connected: IFIP 10th World Conference on Computers in Education - WCCE2013, Toruń, Poland; July 1-7, 2013. Volume 1: Research Papers* (s. 122-130). Toruń: Wydawnictwo Naukowe Uniwersytetu Mikołaja Kopernika.
- Tillmann, N., de Halleux, J., Xie, T., Bishop, J. (2014). Code Hunt: Gamifying Teaching and Learning of Computer Science at Scale. W: M. Sahami (red.), *L@S 2014: Proceedings of the first ACM Conference on Learning@Scale: March 4-5, 2014, Atlanta, Georgia, USA* (s. 221-222). New York: Association for Computing Machinery.
- Tkaczyk, P. (2012). *Grywalizacja. Jak zastosować mechanizmy gier w działaniach marketingowych*. Gliwice: Helion.
- Werbach, K. (2014). (Re)Defining Gamification: A Process Approach. W: A. Spagnolli, L. Chittaro, L. Gamberini (red.), *Persuasive Technology. 9th International Conference, PERSUASIVE 2014, Padua, Italy, May 21-23, 2014. Proceedings* (s. 266-272). Cham: Springer. Online: <http://link.springer.com/chapter/10.1007%2F978-3-319-07127-5_23#page-1>. Data dostępu: 13 października 2016.

Woźniak, P. A., Gorzelańczyk, E. J. (1994). Optimization of repetition spacing in the practice of learning. *Acta Neurobiologiae Experimentalis*, 54(1), 59–62. Online: <<http://www.lac.ane.pl/pdf/5409.pdf>>. Data dostępu: 12 października 2016.

Xie, T., de Halleux, J., Tillmann, N., Schulte, W. (2010). Teaching and Training Developer-Testing Techniques and Tool Support. W: C. Videira Lopes (red.), *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications* (s. 175–182). New York: Association for Computing Machinery.

dr hab. Jakub Swacha – profesor nadzwyczajny w Instytucie Informatyki w Zarządzaniu (Uniwersytet Szczeciński, Szczecin). W swoich badaniach od lat zajmuje się problematyką nauczania podstaw programowania komputerów. W ostatnim okresie zaangażował się w poszukiwanie rozwiązań opartych na gamifikacji.

Wykorzystanie systemu planowania powtórek z elementami gamifikacji w nauczaniu podstaw programowania komputerów

Abstrakt: Trudność w nauczaniu programowania komputerów motywuje do podejmowania prób realizowania tego zadania z wykorzystaniem nowatorskich technik i narzędzi. W niniejszym artykule opisano eksperyment polegający na użyciu w tym celu internetowej platformy Flippi, łączącej funkcjonalności systemu planowania powtórek, sieci społecznościowej oraz elementów gamifikacji. Jakkolwiek niespodziewany przebieg eksperymentu nie pozwolił na osiągnięcie pierwotnie założonych celów badawczych, ostatecznie udało się jednak wyciągnąć wnioski, które mogą być użyteczne przy wdrażaniu analogicznych (zawierających elementy gamifikacji) narzędzi do wspomagania nauczania.

Słowa kluczowe: gamifikacja, fiszki, system planowania powtórek, nauczanie programowania komputerów
