THE COLLABORATIVE EXPERIENCES AND CREATIVITY: AN ANALYSIS ON THE GAMES CREATED IN THE KODU BR COMMUNITY

Paulo Roberto de Azevedo Souza

Program of Post Graduation in Informatics (PPGI-NCE)

Federal University of Rio de Janeiro (UFRJ)

Rio de Janeiro, Brazil

prasouza@ufrj.br

Claudia da Costa Guimarães Santana

Program of Post Graduation in Education

Federal University Fluminense (UFF)

Nitéroi, Brazil

ccgs2011tr@gmail.com

Abstract

This paper describes collaborative experiences in class room through the use of educative software for the creation of games, the Kodu Game Lab. These experiences were made in a vocational school of informatics in the state of Rio de Janeiro, Brazil with teenagers between 14 and 18 years old. This report approaches the introduction of the programming directed to objects, creation of learning objects based on the constructionist paradigm, with the purpose of developing creative learning strategies within these students. The data shows the need to stimulate the creation of more communities that aim the dissemination of collaborative learning for constructing of creative competencies in educational experiences.

Keywords*: Educational Games, Visual Programming Language, Object-Oriented Programming, Creativity*.

Introduction

The Kodu Game Lab, target of the present work, is a free software developed by the research laboratory FUSE (Future Social Experiences) Labs, maintained by Microsoft. This software is used in the creation of games, in a three-dimensional and multimedia environment and holds as principle the stimulation, in classroom, the playful learning of computer programming shaped within the paradigms of object orientation. The games are a strong media and the "make your own games" ideology provided by the Kodu Game Lab software is very attractive for a generation of apprentices in search of constant practical applications in their studies [18].

The school must face informatics technology and communication not only as a pedagogical instrument to ease learning, but mainly as a source of based knowledge construction in a critic perspective of this process. In this sense, it is imperative the pursuit of pedagogical alternatives that improve the indices of learning and the interest of students.

Papert [11] styled "constructionist" the approach through which the apprentice constructs, through the mediation of a computer, his own knowledge. He used this term to demonstrate another level of knowledge construction: The construction of knowledge that happens when a student creates the object of his interest, as a form of art, a report of experiences or a computer program. When the student interacts with the computer passing information to the machine a cycle is established – description – execution – reflection – debugging – description, which is the mover of the process of knowledge construction. For example, to program the computer to solve a problem the student must be capable of putting through the idea of how to solve a problem in the form of a sequence of commands written in the programming language. This means the description of the problem's solution in the programming language.

The process of finding and correcting the error consists in an opportunity for the student to learn about a specific concept involved in the solution of the problem or about problem solving strategies. The student is also able to use his program to relate to his thoughts in a meta-cognitive level. He can analyze his program in levels of effectiveness of ideas, strategies and the style of problem solution. In this case the student starts thinking about his own ideas. The pedagogical conclusion that can be reached based on this means offering to the students and amplifying the experiences in which his creation activity can be developed. Creativity is related to what is offered to the students and the manner they re-elaborate creatively their experiences.

According to Vygotsky [15] "the brain is not only an organ that conserves and replays our past experiences, but also the one that combines and reworks, in a creative manner, elements of the previous experience, erecting new situations and new behavior". Before some challenges and competitions that will be described in the following sections of this article, the students were required to work collaboratively to reach their goals. Vygotsky points to the idea of thinking about the development of our students in a prospective form, in other words, that which he can become and fulfill with the aid of more experienced partners or cultural artifacts that hold the intention of guiding this student beyond what he presents at the current moment as a possibility of function.

Many programs lean on collaboration and experience exchange between students, in this sense games produced in the Kodu Game Lab environment can be shared with other apprentices through conventional storing medias (flash-drives, CDs, DVDs) and depending on internet access, the

student can publish his game directly into the FUSE Labs library, named Planet Kodu, characterizing these games as learning objects to be used and reused by other students.

1. Development of creation activities in the Kodu Game Lab

The way in which a school enables knowledge, many times not relating it to concrete situations, hampers the student's comprehension, leading them to all manners of resistances, mainly of the logical-mathematical kind. In this sense, it0's imperative the search for pedagogical alternatives capable of improving the student's levels of learning. According to Castanha e Castro [19], "we must hear their voices and realize what they call an interesting lesson is associated to your effective participation in the learning process."

From the experience of a pilot project by Kodu in Australia, researchers point to good results in what concerns the interest and development of logical thinking in children ranging from 7 to 12 years old. The Kodu presents a constructive concept, utilizing in its programming a Visual Programming Language (VPL), allowing for an alternative for the development of logical-mathematical thinking through creative activity.

1.1 Kodu's Environment

Through introductory videos, available in the Kodu official homepage or through the assistance of a teacher, the beginner student will be guided to choose a "World". The student is provided with a wide array of options divided in categories, such as "Samples"; "Lessons", with tutorials in how to create games; "Downloads", where the student can download "worlds" available in the community; and lastly, the "My Worlds" section, where the student save his game and begins to construct his own library of "worlds", also providing them to the community.

The "Worlds" are 3D virtual realms that enable the realization of a series of educational activities such as modeling of complex scenarios through inert objects like trees, boulders, coins, stars, castles, and factories [8].

1.2 The Characters

After the student has familiarized himself with the Kodu environment and with the "worlds", comes the moment to insert characters, the main agents within the "worlds". The characters are, without a doubt, the grand sensation of a game for they function as objects that can move and interact with other objects.

In the Kodu environment, the student can construct the programming of the characters without having any previous knowledge of object oriented programming. The programmed actions follow a syntaxes: "When…do…", in other words, it's possible to choose what a character will do when an event happen. There is a list of movements and possible actions the characters can perform in the "world"; all that is required of the student is to choose one of the commands and create the desired combination. However, the programming of characters requires a logical sequence for it to run and it is this logical sequence that determine the properties and functions of the character within the game. In this sense, the creative activity is encapsulated in the Kodu environment, in other words, **it is the creative activity developing logical thinking**.

Fig. 1. **Operations programmed to a character**

The characters are the agents through which the instructor can begin the teaching of computer programming, according to the paradigms of object oriented programming. In the Kodu Game Lab environment, students can program different characters to be created with similar behaviors from the command “Creatable”.

“In the main language of object oriented programming the sub-class inherit all of the attributes of a super-class and can overlap all methods of the super-class” [3]. A class is defined as a model for creation of many similar objects [1].

As soon as the first contacts with the programming environment, the student is made aware of the workings of the relationship between class inheritance, according to the UML (Unified Modeling Language) graphical annotations, as represented in the images 2 and 3.

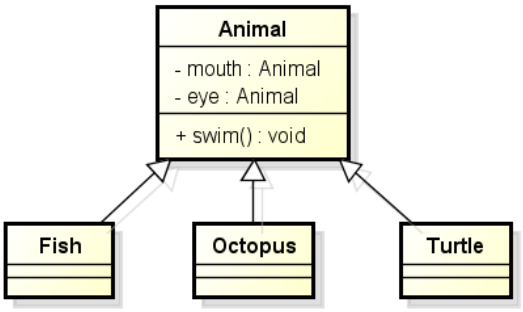


Fig. 2. **Generalization relationship, according to the graphical notation of UML**

Fig. 3. **Generalization relationship in Kodu Game Lab**

Other fundamental contents are also worked for the understanding of the introduction to object oriented programming, concepts of how these properties and operations function within a class. According to Weilkiens and Oestereich [15] “the properties describe the structural characteristics of a class and the operations describe the behavior of the class from the properties and characteristics that were defined for it”. In simpler terms, the properties define what a class is in a system from the attributes that form it, and the operation are what a class can do and execute from its properties.

When a student starts to construct his own models, he learns with them, for if something isn't working in the model he is stimulated to think about what needs to be done for it to work, thus fully contemplating the constructionist teaching [11].

In Papert's notion of constructionism [11], there are two ideas that contribute to this kind of knowledge construction. First the apprentice will construct something, in short it's the learning through doing, a hands-on attitude. Second, the fact the apprentice is constructing something to his own interest and for which he is motivated.

Valente [14] states a computer takes on the role of an educational instrument when it's used through games and simulation software. In this case the pedagogy used is the self-directed exploration, and once the student is immerged in an environment of gaming and simulation, the possibilities of solving a problem are greater.

1.3 Reports of experiences with the Kodu Game Lab in the Vocational Center of Três Rios – Brazil.

According to a research carried out by the Department of Education and Early Childhood Development – Melbourne / Australia (2010), Kodu offers an educational support for the development of critical thinking and problem solving skills. Developing also the collaboration and involvement of students in activities proposed by the teacher, who in this case will act as a moderator and partner in interactive experiences. To demonstrate the differences in development of logical thinking in 3D virtual worlds when contrasted with the traditional method, we've chosen an example of logical thinking exercise (Chart 01), commonly used in vocational courses of Web Informatics FAETEC (in this case Escola CVT – Três Rios).

Trip Nº	Original riverside	Trip	Arriving riverside
(start)	farmer wolf sheep lettuce		
1	wolf lettuce	farmer sheep ->	
2	wolf lettuce	<- farmer	sheep
3	lettuce	farmer wolf ->	sheep
4	lettuce	<- farmer sheep	wolf
5	sheep	farmer lettuce ->	wolf
6	sheep	<- farmer	wolf lettuce
7		farmer sheep ->	wolf lettuce
(end)			farmer wolf sheep lettuce

Chart 1: Logical Thinking Exercise [20]

The first step is to carry the sheep to the other riverside, for any other step will result in either the sheep or the lettuce being eaten. When the farmer returns to the original riverside, he'll have the option of taking the wolf or the lettuce. If he carries over the wolf, he must return with the sheep in order to bring the lettuce, resulting in the death of the sheep to the wolf. If he brings back the lettuce, he'll have to return and carry the wolf, which results in the lettuce being devoured by the sheep. Therein lies the dilemma. We can solve it by carrying over the wolf or the lettuce to the other side and returning with the sheep. Now he can take the lettuce or the wolf back, leaving the sheep and, finally, returning to pick up the sheep completing the task.

Watch the full programming of the farmer character in the game's development environment.

Fig. 4. **Programming of the farmer character. Source: KoduFarmer [21]**

When the student is immersed in an environment of gamming and simulation, the possibilities of problem solving of this kind are greater, since the language, visual and interactive experience of games are part of his everyday life. In this case, the student can create a scene, testing and discovering many propositions or conditions within a tridimensional environment, in a collaborative manner with his classmates and with the Kodu community. We see then that the student wouldn't be limited to the abstraction of questions through trial and error, trait of instructionist softwares, but would learn in the long term through his actions the reasoning behind a condition not generating the desired results.

In the second half of 2012, a pilot Kodu project was started in the Vocational Center of Technology mentioned previously with the intent of verifying the changes in the lessons of programming logics, which before the high deficit their students were bringing to the Elementary School needed to be reprogrammed. The project aspired to also diminish the course evasion, once programming logics is still one of the subjects that deviates the most students from the vocational school of informatics.

In the construction of the first version of this game, many symbols were used, signs, subtitles and signalizations. In this phase it was observed that the Kodu Game Lab is more than an integrated environment for the development of games, for it offers an educational support for the development of critical thinking and problem solving skills. Students were also capable of:

- Creating and narrating stories.
- Studying the effects of colors in the construction of tridimensional environments (semiotics).
- Learning that computer programming is a creative process.
- Developing critical thinking, where problems are solved in small steps.
- Understanding that the development of games is a collaborative process.

2. Evaluation of the use of Kodu Game Lab

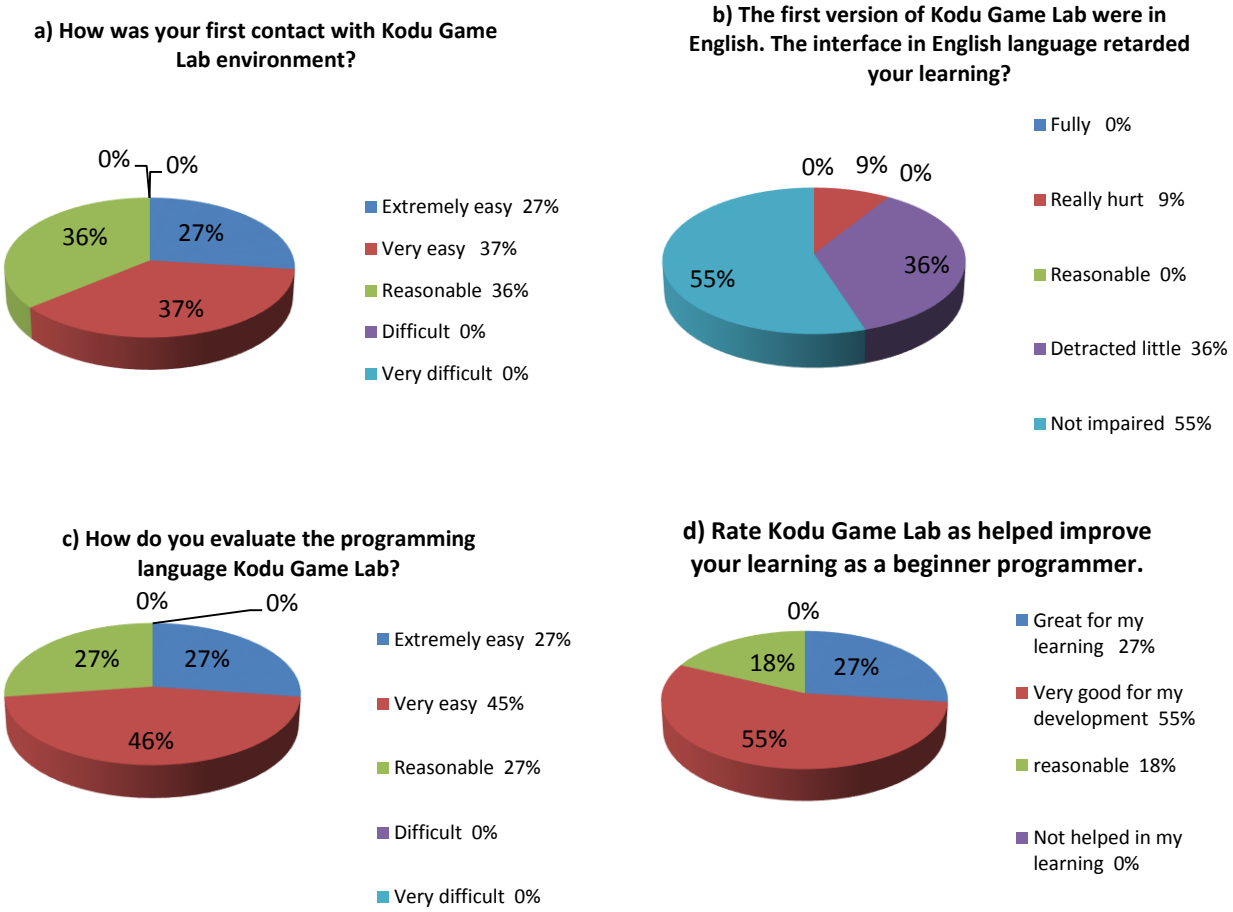
The Kodu Game Lab was used in the initial series of vocational courses in informatics, with 11 youngsters ranging from 14 and 18 years old, as support tool for the learning of computer programming.

We observe that Kodu Game Lab not only facilitates the understanding of concepts of object oriented programming but also develops other cognitive skills in the students. The process in game creation requires: choosing characters, themes, elaboration of scripts, concern with design and other elements.

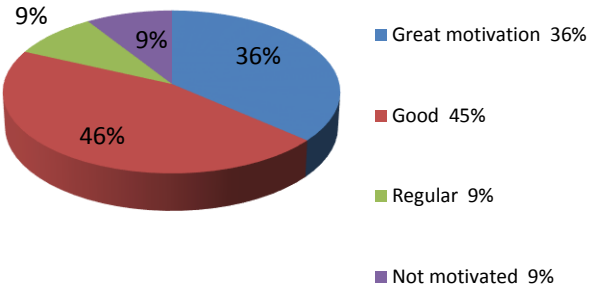
We also detected an acceleration in the acquisition of basic concepts of Unified Modeling Language, as is depicted in the pic. 2, where normally it's concepts are detailed in theoretical manner and with a few exercises through CASE (Computer-Aided Software Engineering) tools.

Obeying an intensity scale, we have next an applied evaluation of 11 students linked directly in this activity.

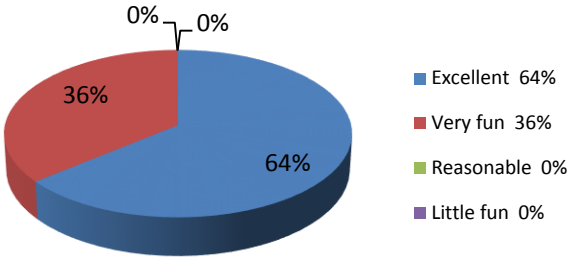
2.1 Survey about Kodu Game Lab



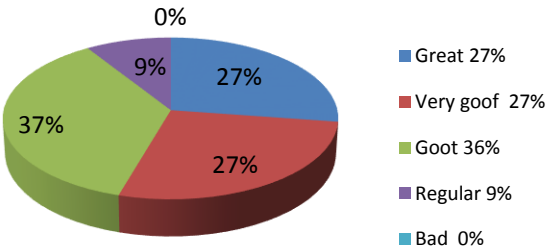
e) Rate if the environment Kodu Game Lab for being three dimensional and multimedia motivated your interest in learning other softwares related to design and desktop publishing. Examples: Corel Draw, CAD, 3DMax.



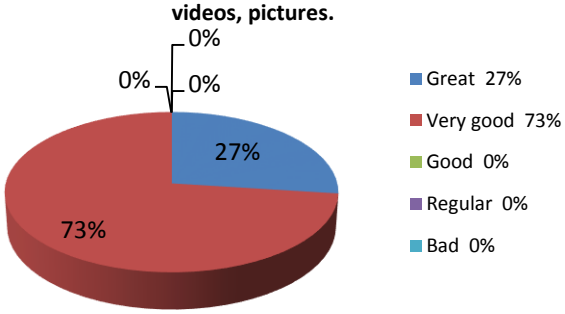
f) Did you find fun and playful learning of programming using Kodu Game Lab?



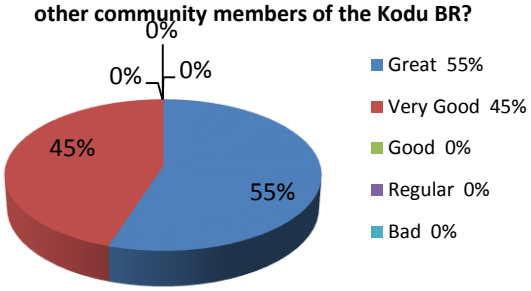
g) Evaluate your development on other topics covered in building games using the Kodu Game Lab as: creation and storytelling, studying the effects of colors, choice of songs for the games.



h) How do you evaluate the materials that were available in the community? Examples: Lectures, videos, pictures.



i) How do you evaluate the interactivity with other community members of the Kodu BR?



Considering the limits of this work and making a brief analysis of the graphics presented, we see that the students showed ease in the use and the interface of the Kodu tool. The ease of learning of the Kodu language present in 100% of the students points to it being easy in handling. However, despite its simplicity, it demands from the students the use of complex psychological functions such as: abstraction, creation, and perception. 82% of the students also stated the tool aided in the learning of Beginner Programming, which goes to show Kodu is a moderating tool in creative activities. On the other hand, as much as the software helps in the programming activity, it does not define the student's direction to other areas as electronic editorial and computer graphics.

3. Final Considerations

The popularization of the Kodu Game Lab is growing thanks to the many elements patched in every software update. According to Microsoft Imagine Cup [5] millions of uploads were made to the Kodu Challenge competition.

Souza and Dias [11] state that before the high capacity of processing of present-day computers, educational game producers have gone on to invest in games compatible with already established consoles in the game universe, such as the Playstation or Xbox, prof of which is the existing version of Kodu Game Lab for the Xbox.

We believe that it's no longer possible to ignore the utilization of educational games in classrooms and understand that depending on the use made of the digital medias, these can contribute to the development of intellectually mature actions, inherently creative and flexible processes.

We must reflect upon more effective methodologies in the process of teaching and learning. Valente [14], warns that many times in the use of games the apprentice might not be aware of the concepts and the use of strategy in the correct manner. It's necessary that the teacher file the observations made by the apprentice during the game, and discuss them out of the situation, recreating them, presenting conflicts and challenges with the goal of providing conditions for the apprentice to comprehend what he is doing.

We believe that pedagogical alternatives, such as the Kodu Game Lab and other softwares that work with simulations, computational modeling in the education can improve qualitatively our indices of learning and personal development.

References

[1] BOOCH G.; RUMBAUGH J.; JACOBSON I. “UML Guia do Usuário”. Rio de Janeiro: Elseiver, 2006.

[2] BRASIL. Secretaria de Educação Fundamental. “Parâmetros Curriculares Nacionais: Matemática”. Brasília: MEC/SEF, 1998.

[3] FOWLER M. “UML Essencial”. Porto Alegre: Bookman, 2005.

[4] GARDNER, H. “Estruturas da mente: a Teoria das Múltiplas Inteligências”. Porto Alegre: Artes Médicas, 1994.

[5] IMAGINE CUP 2013. “Kodu Challenge,”2013 [Online]. Available: http://www.imaginecup.com/kodu. [Accessed 16 1 2014].

[6] INEP. “Universo Avaliado,” 2013[Online]. Available: http://portal.inep.gov.br/internacional-novo-pisa-resultados. [Accessed 11 3 2014].

[7] JACOBI, P. “Educação Ambiental, Cidadania e Sustentabilidade,” 2003. Cadernos de Pesquisa, n. 118. [Online]. Available: http://www.scielo.br/pdf/cp/n118/16834.pdf. [Accessed: 10 12 2013].

[8] KELLY J. F. “Kodu for kids: The Official Guide to Creating Your Own Video”. USA : Pearson Education, 2013.

[9] KODU PILOT. “The impact of web 2.0 technologies in the classroom,” 2009. Knowledge Bank: Next Generation research report. Kodu excerpt. State of Victoria: Department of Education and Early Childhood Development, Melbourne, Australia. [Online]. Available: http://csamarktng.vo.msecnd.net/kodu/pdf/web20_technologies_in_the_classroom_kodu.pdf. [Acessed 11 2 2014].

[10] PIAGET, J. “Seis Estudos de Psicologia”. Rio de Janeiro: Forense Universitária, 1998.

[11] PAPERT, S. “A Máquina das Crianças: repensando a Escola na era da Informática”. Porto Alegre: Artes Médicas, 1994.

[12] SOUZA, P.R.A; DIAS, L.R.. “Kodu Game Lab: Estimulando o Raciocínio Lógico através de Jogos,” 2012. Anais do 23º Simpósio Brasileiro de Informática na Educação – SBIE [Online]. Available: www.br-ie.org/pub/index.php/sbie/article/download/1733/1494. [Accessed 7 2 2014].

[13] TAPSCOTT, D. “Growing Up Digital: The Rise of the Net Generation”. Mcgraw-Hill, 1997

[14] VALENTE, J.A. “O Computador na Sociedade do Conhecimento”. Campinas: NIED/UNICAMP, 1999.

[15] VYGOTSKY, L. S. “A Formação Social da Mente”. São Paulo: Martins Fontes, 1991.

[16] WEILKIENS, T.; OESTERREICH. B. “UML 2 Certification Guide”. São Francisco: Elseiver, 2007.

[17] WILEY, D. A. “Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy,” 2000. In D. A. Wiley (Ed.), The Instructional Use of Learning Objects. [Online]. Available: http://reusability.org/read/chapters/wiley.doc. [Accessed 20 11 2013].

[18] WINDHAM, C. “The Student's Perspective,” 2005, in D Oblinger & JL Oblinger (eds). Educating the Net Generation, EDUCAUSE, Ch 5, pp. 5.1–5.16. [Online]. Available: http://net.educause.edu/ir/library/pdf/pub7101e.pdf. [Accessed 20 11 2013].

[19] CASTANHA, D.; CASTRO, M. B. de. A necessidade de refletir sobre as necessidades pedagógicas para atender as necessidades da geração Y. Revista de Educação do Cogeime. Ano 19. Nº 36. Jan./Jun, 2010.

[20] PRESSMAN, I.; SINGMASTER, D. "The Jealous Husbands" and "The Missionaries and Cannibals"". The Mathematical Gazette 73 (464): 73–81, 1989. [Online]. Available: http://www.jstor.org/discover/10.2307/3619658?uid=2&uid=4&sid=21104241396623. [Accessed 20 5 2014].

[21] KODU BR. KoduFarmer, 2012. [OnLine]. Available: https://www.dropbox.com/s/y0yjlsslxfbugyq/FamerKoduBR%20v02%2C%20by%20KoduBR.Kodu. [Accessed 20 5 2014].