Adopting Games Development and Visual Curriculum Design (VCD) Framework For Connected eLearning

Hock Chuan Lim
FEIS, University of Wollongong, Dubai
Knowledge Village, P. O. Box 20813, Dubai
E-mail: hclim@uowdubai.ac.ae

ABSTRACT

Games development offers potentials for both learning or teaching computer science (CS) and programming. It is especially useful as learners are able to see the application of concepts and programming techniques. A games development course itself also adds interest and motivations to the learning process while reducing the learning curve and enhancing knowledge gathering from learners' perspective. From educators' perspective, games development offers versatility and varieties to modernize traditional CS curriculum.

While games development presents multiple benefits, learning or teaching the subject is challenging. This is due to an increasing trend of large and complex modern games development tools as well as tools usage that require competencies in using interactive icons and visual scripting systems. The proliferation of information about these tools over the Internet further adds to the challenge of learning and understanding. All these issues increase the challenges to impart required foundational knowledge and skills. This paper addresses these new challenges by introducing the "Visual Curriculum Design" (VCD) framework. Based on the core components, usage of VCD framework is demonstrated and implications for connected eLearning highlighted.

KEYWORDS

Connected eLearning, Visual Framework, Interactive Curriculum Design, Games Development, Computer Science (CS).

1 INTRODUCTION

As games continue to be pervasive and as gamification processes permeate into nearly every field, the desire to know how to better deploy games development as an educational tool has never been so great. The presence of large, complex games development tools used in the industry further adds to the challenges of imparting foundational skills and eLearning of this domain. These new tools are not only complex, but they also present new areas of focus required by someone trying to gain a foundational understanding of games development, such as the need for swift icon interactions and the need to be apt with visual scripting and visual node interactions.

For all teaching and learning, we desire that for the foundational knowledge that we impart to a new learner, that it be connected and reinforced. Connected learning is "... learning that is socially embedded, interest-driven, and oriented toward educational, economic, or political opportunity. Connected learning is realized when a young person is able to pursue a personal interest or passion with the support of friends and caring adults, and is in turn able to link this learning and interest to academic achievement, career success or civic engagement..."[1] We desire that our new learners have a way to obtain enough foundational skills to start their journey towards specialization and expert researches. The current trend of games development tools require that we re-look at the way we design and teach games development.

In this paper, we specifically address issues related to new learners of games development. Here, we attach the notion that a new learner is one who is beginning his formal learning of games development either via eLearning resources or via educational institutions. We do not address the needs of intermediate or advanced learners. These needs will be address in other separate forum and paper.

This paper is organised as follows: we first

present four scenarios that unveil the new challenges of teaching and achieving connected eLearning. Next, we introduce a new framework, the Visual Curriculum Design (VCD) framework to help educators to design and ensure connected eLearning outcomes. We also highlight notions of connectedness for eLearning and demonstrate the application of the VCD framework.

2 GAMES DEVELOPMENT TOOLS

Learning games development for new learners require that these new learners have a basic understanding of the existing games development tools. This is essential to allow new learners to be able to follow the works coming from the games industries and from these works, for new learners to further advance their studies.
Here, we look at some of the core tools currently active in the games development industry. These tools have been selected based on the key attraction that they are offered free for most education and academic teaching purposes. We arbitrarily organize them under four scenarios based on complexities of learning curve and presence; to demonstrate the trend of large, complex software applications that have an increased demand for visual interactions and scripting.

2.1 Scenario 1: Autodesk Maya

"Comprehensive 3D animation software... Maya® 3D animation, modelling, simulation, and rendering software offers artists a comprehensive creative toolset. These tools provide a starting point to realize your vision in modelling, animation, lighting, and VFX. "[2] Autodesk® Maya® in their product website provides comprehensive coverage of knowledge base to allow a new learner to learn. In selected areas (for example, "Getting Started"), the learning is based on an effective step by step provision of task-oriented activities. While all these steps are useful to guide a new learner, they only provide very basic introductory information and the rest is up to the user to advance his skills and understanding through other channels.
The current version is Maya 2016 released in 4 September 2015. Like in many current tools for use by the gaming industry, Autodesk Maya provides constant updates and bugs fixes. Frequent and periodic features releases are essential to meet industry needs and to ensure leadership in the playing field. This presents one of the many challenges and raises the cost of entry for learning from a new learner's perspective.
In addition, a first look, from a new learner's viewpoint, of this tool suggests a need to understand the "interactive" approach of Autodesk Maya design philosophy for its products. A common observation is that users will apply various combinations of keys and icon selection to achieve various tasks and activities related to modelling and animations. This is one of the few basic concepts that a new user has to get used to.

2.2 Scenario 2: Unreal Engine 4

"Made by game developers for game developers, from 2D mobile games to console blockbusters and VR..."[3] Unreal Engine 4 is the latest of Unreal Engine. It was launched in March 2014 and current stable version (as of this writing) is version 4.9. As of 11 November 2015, epic released version 4.10. Visit the Unreal Engine website and you will be introduced to Unreal's ecosystems for learning. What is clear is that a large pool of knowledge, tools, communities-of-practitioners all await to share nuggets of information with a new learner.
It is also true that an uninitiated or unguided learner can feel intimidated by the wealth of information and complexities of tasks that is set before him - be it using the Blueprint visual scripting system or via C++ programming approach. As seen in most introductory courses, a new learner is exposed first to visual scripting with Unreal Engine as this removes the need for the user to have a prior understanding of C++ programming. This in turn introduces a new challenge - that of visual scripting and solving problems using visual nodes and nodes interaction.
While the new learner does not need to know

C++ programming, the need to learn both, domain knowledge of games development and understanding visual scripting is not to be seen as a trivial task. New learners may not be able to gather knowledge within a specified time frame as needed. These new challengers if not addressed lead to fragmented learning and do not created connected and reinforced learning as desired.

2.3 Scenario 3: Unity 5

"Unity is a flexible and powerful development platform for creating multi platform 3D and 2D games and interactive experiences. It's a complete ecosystem for anyone who aims to build a business on creating high-end content and connecting to their most loyal and enthusiastic players and customers."[4] Like both Maya and Unreal Engine, Unity is a industry-leading tool for games development. The core concepts of Unity include customizable layouts, multiple views, transforms tools and cameras. A new user that enters the Unity website will encounter similar knowledge resources - tutorials, documentations in the learning section as well as community section. In addition to these on-line resources, Unity also has other resources such as written books and tutorials.
Unlike Unreal Engine, Unity allows three scripting languages (C#, JavaScript and Boo). While Unreal Engine 4 has a blueprint visual scripting system, Unity 5 requires scripting that takes time and effort to master. Scripting is an essential ingredient in all Unity games. Simplest game will need scripts for all aspects such as response to user inputs and for management of events. Scripts are also needed for effects, behaviours of objects and/or implement custom AI system. Hence, a new user will require basic competencies with scripting languages as well as able to establish Unity procedures and workflow. This increases the demands for multiple related skills.

2.4 Scenario 4: Others

Other tools include image processing, such as Adobe PhotoShop, Blender and other forms of game engines / frameworks such as CryEngine, Construct 2, Game Maker, Cocos2d-x and the list includes many other software applications that are either non-free software or are not industry-specific applications. These applications although they may not be industry-specific software or are not industry-lead application, they are in their own sense, large and complex entities and require a new user to invest time and resources to master.

2.5 Scenarios Implications

From these scenarios, it is clear that a new learner faces a myriad of tools and a large pool of on-line and printed resources as well as communities-of-practitioners (CoP) to enable and enhance his learning. These resources while useful also present many obstacles. Under these newer challenges and environments, we need a comprehensive approach to curriculum design in order to ensure a connected and reinforced eLearning.

2.6 Problem Solving - Visual Way

How do we address the challenges of learning and teaching games development? and more importantly, while we have a deep desire for increased and connected eLearning, how do we know that "means lead to the ends"? The key lies with the use of Visual Curriculum Design (VCD) Framework

3 VCD FRAMEWORK

To address the need for a comprehensive approach to curriculum design, we propose the use of Visual Curriculum Design (VCD) Framework. A VCD framework comprises a matrix of core and coverage components arranged in a grid. Here we place the coverage component as columns and the core components as rows in a grid. The framework includes:

- Technical component
- Domain component
- Filter component and

- Coverage component;

3.1 Technical Component

This component focuses on providing a comprehensive coverage of the technical areas of a particular tool. The depth of the technical section will depend on the level of expertise required of a course and the course's context. For example, for teaching of Unity 5 C# scripting, the technical section will include core aspects of C# programming, familiarity with monodevelop interface, compiling and debugging. Take another example, the case of teaching of Unreal Engine 4 Blueprint visual scripting system, the technical components will include Blueprint concepts, node interactions, logic, compilation and debugging. Essentially, this technical component ensures a comprehensive and in-depth presentation of information to a new learner of a games development tool as required and within specific context.

3.2 Domain Component

Domain component addresses the games development requirements. Here, it is possible for the domain component to focus on other aspects that may be essential prerequisites of games development, for example, essential mathematics, or computer graphics. The domain component for games development will include aspects of game art asset preparations (3D modelling and animations), game design, HUD design, software development processes, prototyping, game playtesting, game AI and game physics. Other domain components can be included based on the context and duration of the course. For example if it is year long course, then domain component may include other areas such as graphics effects and/or particle systems.

The choice of networked/massively multiplayer games versus non-networked games will depend on the course requirements. These requirements can be combined as a single domain component or be separated into two distinct components. The key idea behind a domain component is that it is there to guide the formulation of domain-specific information and to ensure that these essential information is provided in correct context.

3.3 Filter Component

The two components (technical and domain) require linkage. The filter component serves the purpose to gel these two components. The filter component matches the technical and the domain components with the delivery methods and evaluation tasks. Combined, this permits a visual display of delivery, presentation of materials and assignment of tasks. These information are collectively arranged inside the cell, in the VCD framework. The visual aspect of the VCD comes from the coding of delivery methods, assignment tasks and application of technical and domain mapping.

An example of application of technical and domain mapping in the context of learning Maya could be an animation project or an in-game keyframed animation. The primary focus of a filter component is to ensure appropriate mapping of technical and domain components and to ensure that these components result in the right combination of delivery methods, assignment tasks and projects to ensure a connected and reinforced eLearning. The delivery methods, in the context of modern games development tools include video tutorials, task preparations, lectures, tests and presentations.

Connected and reinforced eLearning is achieved via the ecosystem effects of different delivery methods. By coding the different delivery methods, assignment tasks and projects, a visual cue is formed and this allows the educator to model and adjust the curriculum as required or as the course progresses. The visual cue also provide a clue as to the connectedness of eLearning and with appropriate evaluation tools, an educator will also have a clue to the degree of reinforced eLearning of the technical and domain dimensions.

The filter component also has an important role to help gel learning and eLearning. eLearning refers the gain from an understanding of e-Business systems where these systems play

an important and critical role for many monetization schemes. Learning refers to the understanding gained from sharing and helps to ensure that both technical and domain components create a suitable experience for a new learner.

3.4 Coverage Component

The coverage component sits along the columns of the framework grid. The coverage component is also the duration, range or spread. Here the coverage component can be 13 weeks for a single semester or can be dual semester with each semester having 13 weeks. The coverage component provides the control. The cells at the intersection of each coverage component with either a technical or domain component define what delivery methods, assignment tasks, projects or presentation that will take place. The coverage component can be as refined as the activities per day or per week. Ideally, the coverage component should be per week day as this guides the provision of eLearning contents and allows a new learner to know when the assignment tasks, projects or presentation is due and to plan accordingly.

Table 1. This table shows an outline of the VCD Framework. Color coding provides a visual cue

Coverage Component		
Technical	**Week01**	**Week02**
item1	Lect	Labs
item2	Video	Lect
item3	Tuts	Proj

4 DEMONSTRATIONS

A course curriculum for 3D modelling and animation using Maya 2015 was designed based on the VCD framework. The course took place over a coverage duration of a 13-week period and students were new learners and have no prior knowledge of 3D modelling and animation or Maya 2015. Here we show the outcomes of selected students' works that demonstrate connected and reinforced learning.

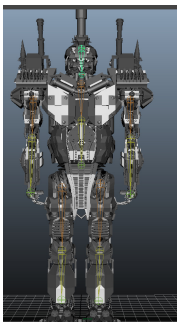

Figure 1. Figure shows student A's work - Robot.

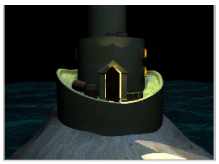

Figure 2. Figure shows student B's work - Lighthouse.

5 DISCUSSION

How successful is the VCD framework? The introduction and use of VCD is timely as with many of the newer frameworks introduced, especially where learners are exposed to "newer" tools [5]. It certainly was useful to help design a games development curriculum within a short planning time frame and it provided a visual cue to allow for re-adjustment and re-organization as needed.

Learning and eLearning information flow are better connected with the application of VCD framework. The combination of the three different components ensure that a new learner is not only learning, but also linked to the many eLearning information flow and information highway.

One of the limitations is that the VCD framework does not allow interactive issues to be fully addressed. For instance, an item that is covered in the technical section may require early coverage of certain domain topics. The spitting of the components means that it is not easy to address these interactive effects. One way to overcome this limitation is to provide duplication of information. Students were informed in advance of these revisions and were told to review them outside of the coverage duration. This allows the course to run without the need to increase the coverage duration.

ISBN: 978-1-941968-19-2 ©2015 SDIWC

Another issue is the need to cater to students who join the course at a later phase. This is not strictly a limitation of the VCD per se, but rather one of implementation. In this case, if the student is significantly late, a separate course design and be prepared and the VCD framework can be applied interactively to adjust and move the filter components as needed. Most likely, the coverage component will be shorter than what is given in the normal case and an exception curriculum can be crafted. Hence, the use of VCD framework provides for interactive design of curriculum that is aligned with recent developments to provide more "student-centered open learning environments" [6] and is able to serve as a visual cue to gauge connected and reinforced eLearning.

6 CONCLUDING REMARKS

This paper introduces the VCD framework that addresses the new challenges of teaching and learning games development. Based on the framework and course taught, students' outputs were shown, demonstrating that the framework allows visual modelling and interactive curriculum design. Further, due to its visual nature, it serves as a simple indication of connected and reinforced Learning as well as allowing users to be linked to essential eLearning information flow. This research effort is the first of its series towards understanding the ecosystem of learning and teaching games development. Future work will leverage the VCD framework and integrate this undertaking with Social Network Analysis, connected learning and games development.

REFERENCES

[1] M. Ito, K. Gitierrez, S. Livingstone, B. Penuel, J. Rhodes, K. Salen, J. Schor, J. Sefton-Green, and S. C. Watkins, "Connected learning: an agenda for research and design," Digital Media and Learning Research Hub, Irvine, CA, USA. ISBN 9780988725508, 2013

[2] Autodesk Team, "Comprehensive 3D animation software," 3D Animation And Modelling Software. [Online]. Available at: http://www.autodesk.com/products/maya/overview. [Accessed: 2015].

[3] T. Sweeney, "Welcome to Unreal Engine 4," March 2014. [Online]. Available at: https://www.unrealengine.com/blog/welcome-to-unreal-engine-4. [Accessed: 2015].

[4] Unity Team, "More beauty platforms players," Unity 5. [Online]. Available at: http://unity3d.com/5. [Accessed: 2015].

[5] A. Buhu and L. Buhu, "Possibilities of application of open source tools Xerte for studying technical fabrics", International scientific conference on eLearning and software for education, vol. 4, pp. 295-298, 2014.

[6] M. J. Hannafin, J. R. Hill, S. M. Land, and E. Lee, Student-Centered, Open Learning Environments: Research, Theory, and Practice, in J. M. Spector, M. D. Merrill, J. Elen, & M. J. Bishop, Eds. Handbook of research on educational communications and technology, New York: Springer, 2014, pp. 641-651.