Published in EG 2014 - Education Papers (editors: J.-J. Bourdin, J. Jorge, and E. Anderson); Proceedings of Eurographics 2014, the 35th Annual Conference of the European Association for Computer Graphics; The Definitive version is available at http://diglib.eg.org/; DOI: 10.2312/eged.20141029

The Four I's Recipe for Cooking Up Computer Graphics Exercises and Assessments

Christopher E. Peters[1] and Eike Falk Anderson[2]

[1]Royal Institute of Technology, Stockholm, Sweden
[2]The National Centre for Computer Animation, Bournemouth University, United Kingdom

Abstract

The design of meaningful student activities, such as lab exercises and assignments, is a core element of computer graphics pedagogy. Here, we briefly describe our efforts towards making the process of defining and structuring computer graphics activities more explicit. We focus on four main activity categories that are building blocks for practical course design: Independent, Iterative, Incremental and Integrative. These "Four I's" of computer graphics activity provide the fundamental ingredients for explicitly defining the design of activity-oriented computer graphics courses with the potential to deliver significant artefacts that may, for example, constitute a portfolio of work for assessment or presentation to employers. The categorisations are intended as the first steps towards more clearly structuring and communicating exercise specifications in collaborative course development settings.

1. Structuring Student Activities

The design of engaging and significant activities is a core element of computer graphics pedagogy, particularly when graduate destinations include highly competitive industries focussing on excellent practical skills, such as visual effects and computer games [LH11]. This has been recognised in the literature: Cunningham [Cun99], for example, stresses the importance of meaningful activities which need to be carefully designed. While there exists literature on the design of activities for various computing disciplines, especially network computing and computer security, with the exception of problem sets published in the CGEMS repository [OAL*08], only limited literature exists on the design process for activities such as lab exercises or assessments in the computer graphics domain. Activity designs and their relationships often seem implicit or abstract, and it is difficult to understand how activities evolve through time or may combine to form more substantial student deliverables.

Here, we provide the first steps towards explicitly defining the design of student activities (see Figure 1) in a computer graphics course [AP09] by presenting some fundamental ingredients of course design. We hope that the ingredients may also have application for analysing the structure of pre-existing courses. We have applied these activity categories in our attempts to better specify course and assessment designs for several years and at several institutions, using the four activity types (Independent, Iterative, Incremental and Integrative) as a basis for guiding the structuring of various computer graphics courses. Next, we discuss these activity types and provide some examples of their use.

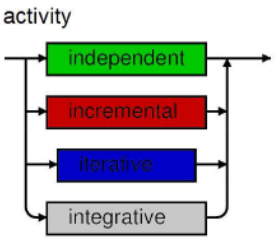

Figure 1: *An activity is a practical element of a course, for example, a lab exercise, a problem set or an assignment.*

1.1. Independent Activities

Independent activities are those activities that may not be fully integrated or relate strongly to other activities, but that provide needed flexibility in relation to course design (see Figure 2). They have short-term outcomes and are indepen-

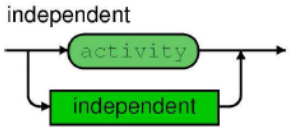

Figure 2: *Independent activities are those that are disjunct from one another and that can exist on their own, without influencing any other activity.*

dent from each other, i.e. they are unrelated and usually result in different learning outcomes. This type of activity is suitable for stand-alone assignments or the exploration of thematic excursions that are not core to the curriculum.

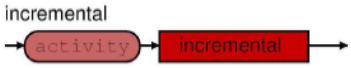

Figure 3: *Incremental activities are those that include elements of progressive refinement of work produced during one or more previous activities.*

1.2. Incremental Activities

This type of activity adds to previous activities, i.e. the starting point for one activity is based on the results of a previously completed activity (see Figure 3). A series of incremental activities can either form part of a larger learning outcome or a set of related learning outcomes from each of its component activities. In a course with regular lab sessions, these activities generally follow one another in subsequent lab sessions, directly building on the activity that the students had been set before.

Figure 4: *Iterative activities are those where an activity revisits topics covered in an earlier activity. It may reconsider a task from a slightly different perspective, e.g. to create the same output by using a different programming language, library or platform; or to consider the design of a graphics program first from the perspective of the programmer and then from the perspective of the end user.*

1.3. Iterative Activities

These activities (see Figure 4) reiterate and elaborate previously covered concepts, allowing difficult concepts to be revisited and previous material to be considered in greater detail, in which case the activity's learning outcomes will be similar or identical to those associated with the original activity. We have found this to be particularly useful for the beginning of a course or after breaks when students may require some refresher tasks to help re-familiarise themselves with subjects learned previously. In addition, these activities may be useful when a course involves assessment by examination and concepts from the beginning of the course are being revisited as part of exam preparation activities.

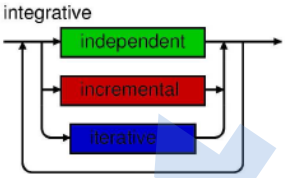

Figure 5: *Integrative Activities are those that allow the creation of larger pieces of work combining the results of other activities. An example may be a library of maths functions created as an independent activity that is then integrated into programs developed in other activities.*

1.4. Integrative Activities

Integrative activities (see Figure 5) consolidate the results of a set of previously completed independent, iterative, incremental or integrative activities. This synthesis of the results of activities does not have to be confined to the integration of activities within the same theme or course, but can span several courses, bridging the thematic boundaries of different concepts. An example for an integrative activity within a single course could be a final assignment that combines the outcomes of previous assignments. An example for an integrative activity that spans several courses could be a larger assignment for which different learning outcomes are assessed in different courses, or an assignment in one course into which the outcomes from assignments in a different course are integrated. Essentially, Integrative Activities allow the creation of more significant outcomes and deliverables and are usually end-points in activity sets that produce exemplars or showcases of student abilities.

1.5. Activity Combination

Activities may be combined to create more sophisticated outputs (i.e. student deliverables). For example, a sequence of seemingly independent activities may subsequently lead to an integrative activity that combines them into a more substantial project deliverable.

2. Application Examples

The aforementioned activity sets were used in an Advanced Games Programming course during the course of which students were set several small game graphics related exercises, comprised of a mix of independent, incremental and iterative activities. Some of the 3D models used by the students for these exercises had been created as an assignment in a

3D Modelling and Animation course that ran in parallel to the Advanced Games Programming course (see Figure 6), effectively making this an integrative activity spanning two courses. Independent activities in this case were lab exercises for implementing techniques covered in lectures that were considered in isolation, without linkages to other techniques included in the course's syllabus. Examples for the latter two types of activities are shown below.

2.1. Incremental Lab Exercises

This set of exercises involved the creation of a graphics application that was incrementally extended with different computer graphics techniques.

1. In the first lab exercise students were asked to develop a simple OpenGL application that loads different 3D objects and displays these as a hierarchical animated model (see Figure 6).
2. In the second exercise the students extended the application's functionality by implementing different levels of detail for the animated model.
3. In the third exercise students once again extended the original application by adding a bounding volume hierarchy for collision detection to the 3D model.

2.2. Iterative Lab Exercises

An iterative exercise that students in this course completed was attempted at a different stage of the course after students had been exposed to game engine architecture and typical game and graphics middleware systems [AP10]. For this the students were asked to revisit the graphics application created before and to rewrite it using a game rendering engine rather than native OpenGL.

3. Concluding Remarks

We have briefly described four activity categories that form the basic ingredients for more explicitly structuring and communicating student activities. They constitute fundamental building blocks that can be combined to form more elaborate exercises and projects. The categories represent one aspect of our continuing effort towards defining a methodology to support computer graphics exercise designs that may be more clearly specified, managed and communicated in formal and collaborative course development and administration settings, for example, between course instructors, teaching assistants and external examiners.

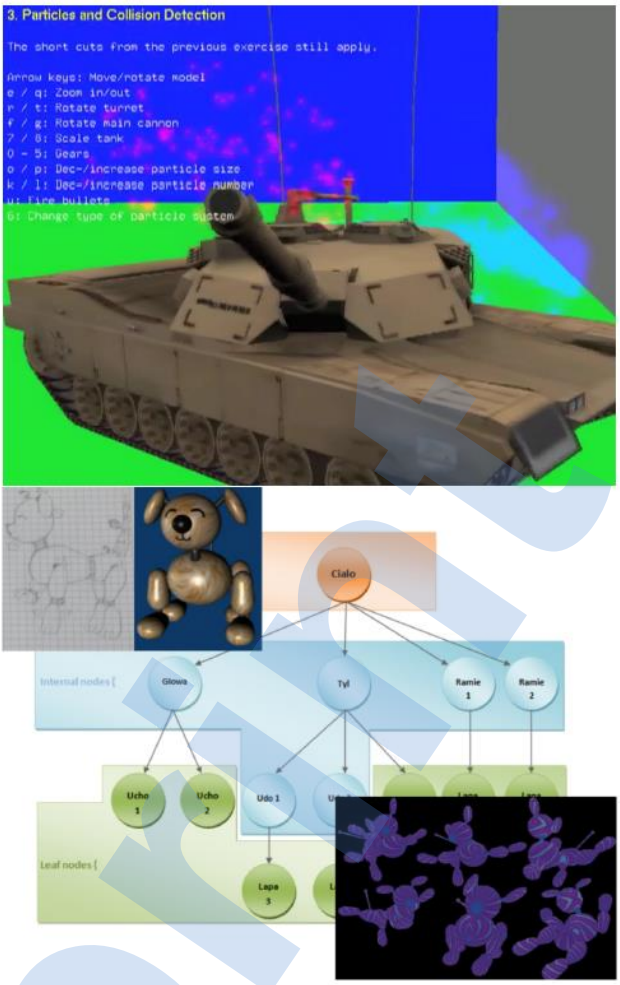

Figure 6: *Example student work from the Advanced Games Programming course. The top image shows an interactive hierarchical tank model by student Andy Abgottspon. The bottom image shows the design, hierarchy and implementation of an interactive virtual dog by student Jadwiga Krawczyk, created in the 3D Modelling and Animation course.*

References

[AP09] ANDERSON E. F., PETERS C. E.: On the provision of a comprehensive computer graphics education in the context of computer games: An activity-led instruction approach. In *Eurographics 2009 - Education Papers* (2009). 1

[AP10] ANDERSON E. F., PETERS C. E.: No more reinventing the virtual wheel: Middleware for use in computer games and interactive computer graphics education. In *Eurographics 2010 - Education Papers* (2010). 3

[Cun99] CUNNINGHAM S.: Re-inventing the introductory computer graphics course: providing tools for a wider audience. In *GVE '99: Proceedings of the Graphics and Visualization Education Workshop* (1999). 1

[LH11] LIVINGSTONE I., HOPE A.: Next Gen: Transforming the UK into the World's Leading Talent Hub for the Video Games and Visual Effects Industries. Report produced for the National Endowment for Science, Technology and the Arts, 2011. Available online at http://www.nesta.org.uk/publications/next-gen. 1

[OAL*08] ORR G., ALLEY T., LAXER C., GEIGEL J., GOLD S.: A knowledge base for the emerging discipline of computer graphics. In *CGEMS: Computer Graphics Educational Materials Source* (2008), The CGEMS Project. 1