Game Design Toolkit: Individual Reflection

Lee Ackerman

University of British Columbia

Dr. Lori MacIntosh

ETEC 510-65A (2018)

Software development is a passion. I fondly remember working with early computers such as the TI-99/4A[1]. Learning came from code examples located in the BASIC manual and issues of Compute! magazine[2]. Sometimes the programs would just be printed in Machine Code – a series of lines all written with ones and zeroes. Feature rich development tools with syntax highlighting, debuggers and code assist were not available. For a kid, the results were disappointing as the programs often didn't run and debugging was a manual process of line-by-line comparison between the printed example and what was typed on the screen. Creating programs (and games!) was a challenge of limitations and constraints. Support was limited, the community was small and dispersed, and the underlying technology was in its infancy.

In early 2000, I joined a startup company called Catapulse[3]. Our main product was the "Hosted Development Service" (HDS). The HDS was a software as a service (SaaS) solution, providing clients with tools for collaboratively developing software. Up to this point in time, if a company (or a group) wanted to collaborate (requirements, code, ideas), project set-up and on-going administration was difficult, time-consuming and expensive.

The Game Design Toolkit[4] (**Toolkit**) reflects a stark contrast to this background as it is introduced during a time of abundance; which brings many benefits and some new challenges. This abundance is applicable to tools, information, content, support, and even gaming. Such an environment provides a great foundation for constructionism and social-based learning.

GitHub is a great place to start when considering the benefits of abundance as it provides a free collaborative workspace. The site is responsive, has excellent support (documentation and videos), and has been embraced by a large user community. There is also an abundance of

[1] https://en.wikipedia.org/wiki/Texas_Instruments_TI-99/4A
[2] https://en.wikipedia.org/wiki/Compute!
[3] https://www.gartner.com/doc/318492/purchase-catapulse-simplifies-rationals-path
[4] https://github.com/leemet16/game-design-toolkit/wiki

feature rich development platforms such as Scratch, Kodu, ARIS, GameMaker Studio and others (too many to cover in the Toolkit). Each of these platforms, in turn, provides samples, documentation, and communities. Going further afield, the abundance continues with images, sounds and related tools to support game development. And last but not least, books, tutorials, presentations and videos provide a wealth of support for learning and succeeding with game development.

This abundance also provides a challenge. We all have limitations in how much we can filter, review and consume – and the time we can dedicate to such efforts. Day-to-day I see professionals that struggle to keep up with the pace of change and the many options available to help them succeed. The Toolkit attempts to help students find a path to success as it provides a simplified, narrowed view of available assistance. These efforts are focused on preventing the student from getting overwhelmed. The overall abundance and the challenge of the game development project will require a motivated facilitator and a group of motivated students.

This abundance also supports more sophisticated approaches to design. As we remove the friction associated with connecting with others, it makes it easier to test out ideas, capture feedback, and better meet the needs of our audience. The Toolkit reinforces the iterative, feedback-driven, user-focused approach to design. Unfortunately, in creating the toolkit, the short timeline and context of the creative team limited our opportunities to operate in a similar manner. My hope is that we can view this version of the Toolkit as an MVP, and that feedback and support surfaces as we introduce it to a wider audience. Options such as adding collaborators, taking input via Issues, or Forking the content all increase the likelihood that the Toolkit will live beyond the duration of ETEC510.