Making Serious Games with Reusable Software Components

Open Universiteit
www.ou.nl

Making serious games with reusable software components

Wim van der Vegt[1], Enkhbold Nyamsuren[1], and Wim Westera[1]

[1] Open University of the Netherlands, Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands

{wim.vandervegt; enkhbold.nyamsuren, wim.westera}@ou.nl

Abstract.
This paper explains the RAGE project, which proposes a component-based software architecture to accommodate and amplify serious game development. The RAGE project (rageproject.eu) is a serious gaming flagship project funded by the Horizon 2020 Programme of the European Commission. Compliancy with the component-based architecture preserves the portability of software to different platforms and programming languages and its easy integration in wide variety of game engines. RAGE has developed up to 40 cutting edge reusable software components (all free, open source software) and has made these available on its market place portal at gamecomponents.eu.

Keywords: serious games, software components, game development, reuse, cross-platform, portability, game engines.

1   Introduction

Serious gaming is a priority area of the European Commission, having recognised the potential of games for e.g. teaching and training, social inclusion, heath, the digital transformation and other societal purposes. Although scholars and teachers have shown great interest in serious games for quite some years, the uptake of games in schools and business has been quite limited [1,2]. While the leisure game industry has become a well-established industry dominated by major hardware vendors, publishers, and a fine-grained network of development studios, distributors and retailers, the serious game industry still displays many features of an emerging, immature branch of business, which is characterised by weak interconnectedness, limited knowledge exchange, and absence of harmonising standards [3]. Notably, progress is hampered by the wide variety of programming languages, game development systems and delivery platforms that game studios have in use, all of which go with specific technical constraints and incompatibilities that pose severe barriers to growth. Moreover, the small game studios often do not have access to emerging media technologies that could be incorporated in serious game projects, such as novel adaptation algorithms, artificial intelligence kernels, or natural language processing methods, while the alternative of in-company development

of such technologies is not feasible, either because of required investments or because of lacking know-how.

This paper provides a brief summary of the work in RAGE in order to provide a starting point for the technical workshop arranged by RAGE at the Joint Conference of Serious Games 2018.

2 The RAGE component-based software architecture (RCSA)

The RAGE component-based software architecture (RCSA) was devised to accommodate the development of software components that can be easily reused and integrated in serious game projects across a wide variety of prevailing technology platforms. The RCSA [4,5] distinguishes between server-side components and client-side components. While remote communications of server-side components with centralised applications can be easily achieved with web services using the HTTP-protocol (e.g., REST), client-side components need to be integrated into client-machine applications (viz. game engines), which is often problematic. With respect to client-side components, the RCSA is based on a limited set of established coding practices and software patterns (API, Bridge, Publish/Subscribe, Web Services), generally aiming at the abstraction of operations in order to accommodate reuse in different technical environments. Components based on the RCSA avoid dependencies on external software frameworks and minimise code that may hinder integration with game engine code. They do not access or do not make assumptions about the underlying operating system, do not directly interfere with the game code, and consequently, do not access the game's user interface, thus leaving all functional and creative decisions about screen layout style, look and feel in the hands of the game designers and developers. Technical proofs of concept of the RCSA portability have been established for C#, C++, Java and JavaScript/TypeScript, which are among the predominant programming languages used in game development [6]. Also, RCSA-compliant components have been successfully integrated in various game engines, including e.g. Unity3D, MonoGame, Cocos2D, Xamarin and Emergo, and have been deployed at the most popular desktop and mobile delivery platforms [5].

3 Game software components at gamecomponents.eu

An initial set of game components can be accessed through the RAGE marketplace portal at gamecomponents.eu (Figure 1). The exposed components offer a variety of cutting-edge functionalities of potential relevance for serious games, ranging from adaptation and personalisation, to language-based sentiment analysis, facial emotion recognition, sensor-based arousal detection, social gamification, affective computing and other topics. All components are compliant with the RCSA, which guarantees the seamless integration into a variety of game engines or other software environments. The collection of components includes both server-side components and client-side components, as well as hybrids. To maximize the reuse of software, all components use

the Apache 2.0 license (white label software), which allows for reuse by third parties both for commercial and non-commercial purposes, either under open source or closed source conditions.
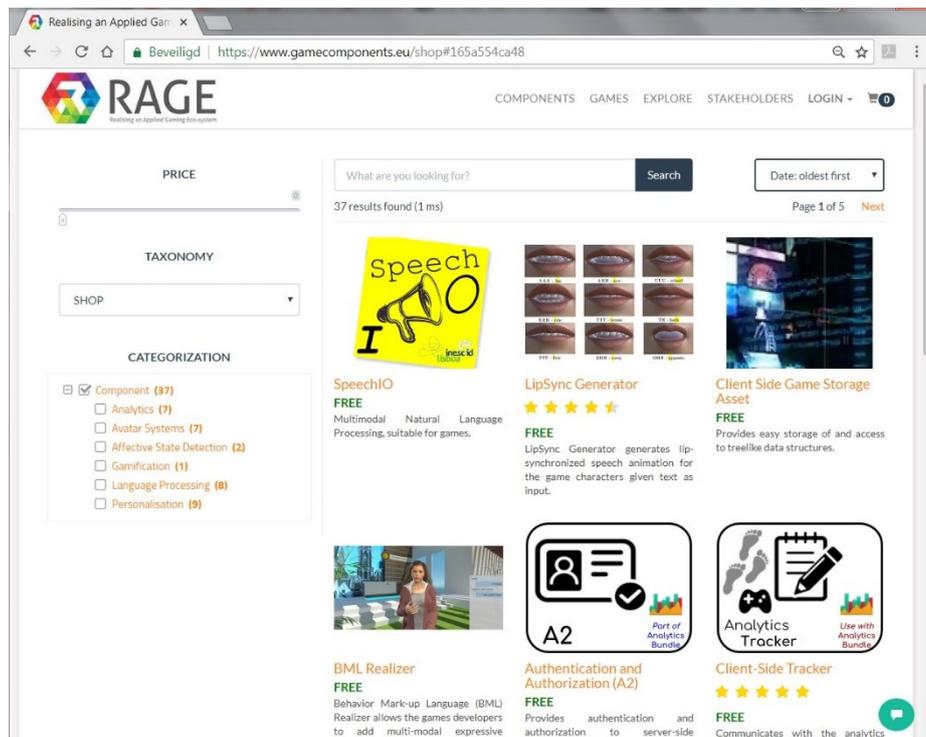

Fig. 1. Screenshot of the gamecomponents.eu marketplace.

All software exposed at the portal are enriched with user guides, manuals, instructional materials, integration templates, demonstrators, and proof cases. For the purpose of practical validation a set of seven serious games based on the various software components was developed, which were then tested and evaluated in real end-user pilots. The games focus on various social and entrepreneurial skills and address diverse educational contexts. Overall, over 1500 participants were involved in the game pilots. The gamecomponents.eu marketplace portal differs from existing portals by its platform independency: the compliancy with the RCSA preserves the easy integration of components across a variety of different software environments. In contrast, existing platforms such as the Unity Asset Store and the Unreal Marketplace, are inherently platform bound, and thereby lead to vendor lock-in. Moreover, existing marketplaces are mostly dominated by media objects (e.g. terrains, audio, buildings, weapons, user-interface objects, and templates) rather than software, which reduce the efforts required for content creation, but still preserve the programming load.

4   The component integration workshop

The workshop arranged at the Joint Conference on Serious Games 2018 comprises a hands-on session about the technical integration of RCSA components in game projects. In a step-by-step process, participants will be guided to quickly unpack, install and integrate software components in their game project, which not just speeds up the game development process, but also allows the inclusion of cutting-edge functionalities that would not be easily accessible otherwise. Component-based approaches and the reuse of software will support developers at creating better games easier, faster, and more cost-effectively.

5   Acknowledgment

6   References

1. Carl Abt: Serious games. Viking Press, New York (1970).
2. T.M. Connolly, E.A. Boyle, E. MacArthur, T. Hainey and J.M. Boyle: A systematic literature review of empirical evidence on computer games and serious games. In: Computers & Education 59 (2), 661–686. DOI: 10.1016/j.compedu.2012.03.004 (2013).
3. Stewart, J., Bleumers, L., Van Looy, J., Mariën, I., All, A., Schurmans, D., Willaert, K., De Grove, F., Jacobs, A., and Misuraca, G.: The Potential of Digital Games for Empowerment and Social Inclusion of Groups at Risk of Social and Economic Exclusion: Evidence and Opportunity for Policy. Centeno, C. (Ed.), Joint Research Centre, European Commission. (2013).
4. G.W. van der Vegt, W. Westera, E. Nyamsuren, A. Georgiev and I. Martinez Ortiz: RAGE architecture for reusable serious gaming technology components. In: International Journal of Computer Games Technology. Article ID 5680526. DOI: 10.1155/2016/5680526. (2016).
5. W. van der Vegt, E. Nyamsuren and W. Westera: RAGE Reusable Game Software Components and Their Integration into Serious Game Engines. In: Proceedings of the 15th International Conference on Software Reuse (ICSR 2016). Springer International Publishing, Basel, 165-180 (2016).
6. RedMonk: The RedMonk programming languages rankings: January 2015, http://redmonk.com/sogrady/2015/01/14/language-rankings-1-15/, last accessed 2018/09/05.