Entertainment Computing 26 (2018) 1–9

Contents lists available at ScienceDirect

Entertainment Computing

journal homepage: www.elsevier.com/locate/entcom

Deploying learning materials to game content for serious education game development: A case study

Harits Ar Rosyid[a,*], Matt Palmerlee[b], Ke Chen[a]

[a] School of Computer Science, University of Manchester, UK
[b] Engineering at Mastered Software, Los Angeles, United States

ARTICLE INFO

Keywords:
Serious educational game
Serious game development
Learning material deployment
Game content generation
Chem Dungeon
User survey

ABSTRACT

The ultimate goals of serious educational games (SEG) are to facilitate learning and maximizing enjoyment during playing SEGs. In SEG development, there are normally two spaces to be taken into account: knowledge space regarding learning materials and content space regarding games to be used to convey learning materials. How to deploy the learning materials seamlessly and effectively into game content becomes one of the most challenging problems in SEG development. Unlike previous work where experts in education have to be involved heavily, we proposed a novel approach that works toward minimizing the efforts of education experts in mapping learning materials to content space. For a proof-of-concept, we apply the proposed approach in developing a SEG game, named *Chem Dungeon*, as a case study in order to demonstrate the effectiveness of our proposed approach. This SEG game has been tested with a number of users, and the user survey suggests our method works reasonably well.

1. Introduction

Serious Educational Game (SEG) refers to an alternative learning methodology that applies game technology to primarily promoting players' learning along with gaining positive cognitive and affective experience during such a learning process [1]. Elements of challenge and learning within such a game construct activities for motivation and amusement [2]. SEG is also named in different terminologies such as game-based learning or educational games. In this paper, we treat all those terminologies interchangeably and refers the SEG development to the procedure that builds up a game for a learning purpose.

There are useful approaches to game development for a learning purpose, such as [3,4]. Most of those approaches emphasize that the design of a serious game is mainly from learning materials of a domain knowledge. Hence, those development frameworks have to rely on a close relationship between learning materials and game design (proprietary educational game). Moreover, the proposed development frameworks require rigorous procedures that may involve interviews with target users (including teachers and students) and various experts (e.g., game development, education, psychology and so on), lengthy development stages and testing units. Such development frameworks inevitably incur the high cost because the development process is laborious and time-consuming and hence limit the growth of educational games.

In general, SEG development has to involve two key components: *knowledge* and *game content* spaces [5,6]. The knowledge space is formed to encode learning materials concerning the subject knowledge to be learned by players, while the game content space is formed with playable game elements that convey the knowledge chunks implicitly. This is generally required by any serious games as argued in [7,8] where serious game is defined as a computer program that combines *serious* (for knowledge learning) and *game* (for entertainment) purposes. Thus, how to map the knowledge space to content space becomes one of the most important problems in SEG development. To our knowledge, however, the mapping is a bottle-neck in SEG development as this has to be handcrafted by game developers closely working with education experts in most of existing SEGs.

Unlike most of the existing approaches, we propose an alternative SEG development framework in this paper to address the mapping issue by embedding annotated knowledge chunks into categorized game content/elements seamlessly during SEG development. In one hand, there are abundant education resources (e.g., syllabus or knowledge handbook) that contain the structure of the underlying knowledge chunks as well as sufficient instruction [4] for learning them. Our framework would exploit such information so that knowledge chunks and their connections can be easily annotated by game developers or automatically acquired by using information retrieval techniques. On the other hand, the "purpose-shifting" is a terminology for SEG

[*] Corresponding author.
E-mail addresses: harits.arrosyid@manchester.ac.uk (H.A. Rosyid), matt@masteredsoftware.com (M. Palmerlee), ke.chen@manchester.ac.uk (K. Chen).

https://doi.org/10.1016/j.entcom.2018.01.001
Received 15 March 2017; Received in revised form 31 December 2017; Accepted 1 January 2018
Available online 05 January 2018

development [9,10] which diverts the purpose of an existing commercial game for knowledge learning. This approach exploits the properties of existing commercial games which fit a learning process, e.g., a player has to *learn* game rules, objectives and strategy to succeed a game. Such a typical *learning* process is also applicable in traditional education systems. As an alternative game development methodology, Procedural Content Generation (PCG) technique can generate game content automatically via algorithms using a random or pseudo-random process that produces an unpredictable range of possible gameplays, for instance, [11]. This will significantly lower the cost of game development. Moreover, the latest PCG work [11] suggests that a proper use of the categorized game content may facilitate eliciting positive gameplay experience. Motivated by the previous works, our framework would suggest making use of PCG and existing entertainment games in SEG development (see Section 3.2 for details). In particular, we believe that embedding annotated knowledge chunks into categorised game content/elements makes the mapping easier to accomplish.

We summarise the main contributions of the work presented in this paper as follows: (a) we propose an alternative framework for effective and efficient SEG development; (b) under our proposed framework, we develop a proof-of-concept SEG, Chem Dungeon, to demonstrate the usefulness of our proposed framework; and (c) we test this SEG with human players via user survey and statistical analysis.

The rest of the paper is organized as follows. Section 2 reviews the related works. Section 3 presents our SEG framework and Section 4 describes our proof-of-concept SEG, Chem Dungeon. Section 5 and 6 reports user test analysis results and discussion, respectively. Finally, the last section concludes the research.

2. Related work

In this section, we outline connections and main differences to relevant SEG development approaches.

As argued by Damir et al. [5] based on their interviews with education experts, game developers and players who involve themselves in SEG, it is crucial to have a seamless connection between knowledge and game content spaces in SEG development. Moreover, they further emphasize that two spaces must be controllable [5] to allow for gaining the controllability in tailoring game elements that are likely affecting different kinds of the player's experience such as learning, enjoyment, motivation, engagement and so on. In addition, it is suggested by Hussaan et al. [6] that there are three components in SEG. Apart from learning and game resources, domain concept should be introduced to specify the relationships between learning materials. Specifically, it facilitates using learning resources to formulate strategies in carrying out learning based on game resources. Nevertheless, this approach [6] emphasizes that all of those components have to be taken care by education experts via interactions with students or game players.

Gamification [12] is a typical SEG development approach that explicitly takes knowledge and game content spaces into account in development. The basic idea underlying gamification is directly embedding game elements (e.g., avatar, badges, levels and scores) into the learning process. Doing so make students more actively engaged in the learning process when they are situated in a game-like presentation of the learning materials. In [12], education experts and game developers handcraft the combination of the two spaces, which is laborious and time-consuming. Similarly, Belloti et al. proposed an approach for adaptive SG via building up the proper connection between knowledge and game content spaces [13]. Their approach breaks down a serious game into subsequent tasks by considering diversified connections between learning materials and game elements. Then, adaptation is carried out by offering a proper task sequence to an individual player to maximize their positive learning and positive affective experience [13]. However, the game design (in particular, the mapping between two spaces) relies heavily on education experts. Hence, the development cost is often very high. Technically, such an approach is also subject to limitation. The mapping task becomes difficult if one of the content space is large and complex. Hence, we do not think this approach is extensible in SEG development.

Unlike the above approaches, our proposed SEG framework would exploit the instructional resources and makes use of appropriate PCG techniques towards minimizing the cost. Thus, our proposed framework is expected to connect knowledge and game content spaces seamlessly in SEG development.

3. Methodology

In this section, we propose an alternative framework for SEG development especially for addressing the mapping issue pertaining to two spaces. To accommodate that, the framework exploits learning resources and making use of the latest PCG techniques.

The advantage of structuring serious game content in two spaces of learning materials and game content provides a higher degree of control for the game generation. In the existing SEG approaches, however, education experts have to be the prominent force in the process of deploying learning materials into a SEG. Thus, an expert is expected to deeply understand characteristics of learning materials and game content according to their expertise in order to link the two spaces. However, it becomes infeasible and unscalable in the presence of complex learning or game content space. Hence, game developers are demanded to utilize the natural and inherent game elements to deal with the knowledge deployment issue. This is feasible since sophisticated education resources are accessible easily and the PCG techniques allow for flexibly controlling game elements to embed knowledge chunks. Thus, we believe that making use of learning resources and making use of the latest PCG techniques could slash the expense of SEG development. Furthermore, given the semantic descriptions of those content spaces, the developer can formulate different aspects between them, which sparks a proper deployment.

To address the issues mentioned above, we propose an alternative framework for SEG development as illustrated in Fig. 1. First, learning materials and game elements are in separate spaces. In one hand, annotation takes place to describe education materials naturally from the meta-data retrievable from reliable resources. Then, we need to establish the strategy for presenting them to players, based on their retrieved properties or using the corresponding learning resources. On the other hand, categorisation of game content space consists of a couple of steps. It starts with a difficulty categorisation which groups game content according to the level of challenge. Subsequently, within each of the pre-defined content categories, e.g., difficulty levels, and given a

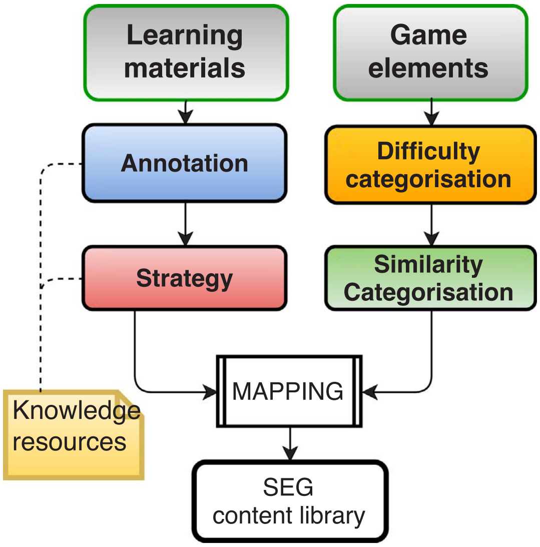

Fig. 1. An alternative SEG development framework.

number of education materials, clustering analysis is applied to group similar game content. Hence, the aspects underlying the descriptive learning materials and game elements can guide a developer to use their logic in formulating the mapping between learning materials and game content. The outcome is a SEG content library comprised of playable games for learning.

3.1. Knowledge space

Knowledge space of a SEG refers to all the relevant materials consisting of items to be learned by a player. Our framework considers the subjects in the low-level learning category. Commonly, learners acquire this category of knowledge through recalling or repetitions. For instance, alphabet learning (i.e. visual appearances, pronunciations and constructions) by repeatedly looking at, listening to and trying to write them. Another example from a more advanced domain is vocabulary which maps words from another language to their meanings in the student's mother tongue. These learning materials should be descriptive; thus, they enable (semi) automatic serious game development.

Bellotti [14] demonstrates an annotation technique for serious games' tasks. However, the author employs experts to annotate subjective attributes. Again, there is no assurance whether this approach can handle the growing size of learning materials. Especially for serious games, where a large number of learning materials have to be recalled, such as biology terms, geographical items or languages.

On the other hand, we argue that the ideal properties for learning materials originate from descriptions provided by a reliable education resource, teaching handout for instance, and the representation of the knowledge (e.g., text, image, audio or video). Hence, we operate the **annotation** based on the objective descriptions of the learning materials with reduced involvement from experts. Accordingly, the developer selects the relevant properties/attributes. Given the available documented resources, an information retrieval technique – beyond our scope – automates the annotation process. Then, another program measures attributes of the knowledge representation as part of the annotation process. For instance, the number of words of the text-based learning material or the length of an educational video. Consequently, the education content space contains comprehensive detail to initiate the **strategy** for delivering the learning materials. If no relationship exists between education materials, an automated method (e.g., sorting) establishes the strategy based on the attribute values. Otherwise, a syllabus or a teaching handout provides the strategy explicitly; thus, players will recall the knowledge accordingly.

3.2. Game content space

Game content space of an educational game refers to all the playable games generated by an entertainment game engine to facilitate the learning defined via knowledge space for a player. Game content is known as elements and objects of a game that the player interacts during a game session. For instance, the *type of enemy* controls the way an enemy behaves in the game environment. In that, a non-player character (NPC) with artificial intelligence can search and traverse the nearest route to fight the avatar. Another example is the *type of weapon* in a First Person Shooter (FPS) game that determines the destructive power to a target. Commonly, a level designer manually designs a limited number of game levels, each of them has a unique configuration of these game content. However, this approach is impractical when the game consists of abundant elements and parameter values. Fortunately, we can randomly or pseudo-randomly generating these elements via Procedural Content Generation (PCG). It generates various game content based on practices or methods (e.g., [11,15,16]) that ensure a game content is playable for the players. For instance, in FPS, NPCs are better to be spawned further from the player's starting point. Distributing them in various places of the game set will create a balanced gaming experience as well. In such a way, the player can sufficiently prepare him/herself to challenge the enemies.

Our method applies PCG in that it provides details of the game content in the parameters configurations. Given the large space for the generated content, manually identifying the category for the content space becomes impractical. Therefore, we adopt a set of steps applied in entertainment game's PCG [11], including: **difficulty categorisation** and **similarity categorisation**. "Difficulty" categorisation provides games for players with different abilities for playing the game [17]. Meanwhile, "similarity" categorisation benefits the space of the game content which provides abundant choices of games which support repetitive sessions of learning.

Robert and Chen suggest that generating categorised game content for players can facilitate positive affective experience [11] via a proof-of-concept first-person shooter game. Prior, developers annotated some game examples and let the categorisation model learned from it. For specifying difficulty levels, a developer can also adopt a rule-based approach by taking into account a small number of game controlling parameters. At this point, developers have to decide the threshold values of those parameters to split content space into proper regions of different difficulty levels. Consequently, content categorisation naturally takes place with the specified difficulty levels. Nevertheless, compared to the aim in [11], a different purpose of clustering analysis occurs here. Given the value of k as a total number of chunks of knowledge, the analysis identifies k groups of similar game content for each education material. As such should prevent boredom developing when multiple repetitions of a game session required.

3.3. Mapping between knowledge and game content spaces

Mapping between knowledge and game content spaces is the essential step that deploys each learning material into game content based on their underlying characteristics. Often, serious game designers view knowledge units as the learning tasks in an educational game. In our framework, we follow the same perspective to allow straightforward mapping. Thus, the player has to address the learning task in a specific game mechanic, such as collection, match-making, destruction or text narration. In the developer's window, the selection of one or more game mechanics from the existing game content is practical to handle. This will become the "container" of the learning mission. For instance, the original Pac Man game requires the avatar to collect all the dots in the game. Meanwhile, the education version replaces these dots with answers to a specific learning-task mission, such as math subjects in Number Muncher educational game [18].

One must ensure that this assignment promotes learning to the subject of interest. Commonly, the game mechanics that directly lead to the game's goal are the candidates. We expect them to imply learning tasks as the prime mission of the game. Thus, players will respect their learning experience to spotlight during game sessions. However, we do not rule out other existing game mechanics to become the container of the learning task as long as they can promote knowledge acquisition and possess non-contrasting perspective with the learning goal. So, the developer must have an adequate knowledge of the game mechanics and s/he must be able to identify their importance in the game session.

Referring to the underlying game mechanic(s), the mapping between learning units with game content may employ an arbitrary or sampling-based mapping. However, it may promote ineffective learning for different players. In fact, learning in an educational game involves various factors [19]. For instance, arbitrary mapping potentially assigns an uncomplicated recall materials with "difficult" games. Hence, a novice player is struggling to play such games trying to overcome the challenges. This situation could hinder a player's aim to recall the learning material. In other words, using arbitrary or sample-based mapping can produce imbalanced outcomes for the players.

Therefore, the mapping should follow specific conditions that produce acceptable deployment of learning materials; thus, it

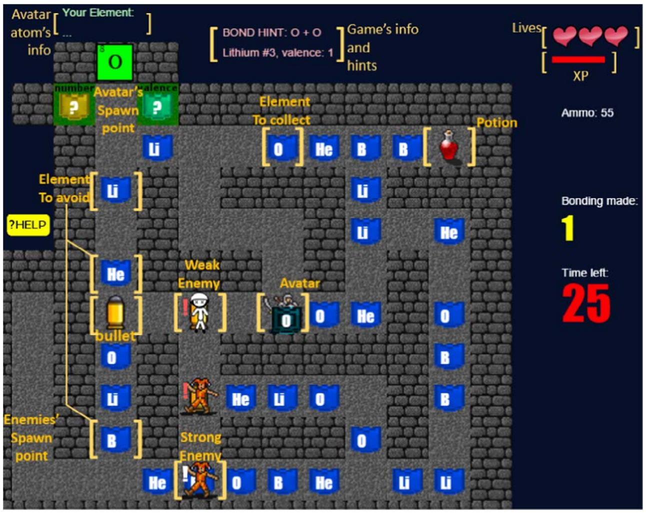

Fig. 2. The Chem Dungeon layout.

accommodates relatively positive experiences for various types of player. For now, our strategy employs the developer's intelligence to exploit the in-depth characteristics of each content space. According to the content structure, the mapping procedure must embed an education material into a unique cluster of game content from each difficulty level. Therefore, it can prevent boredom growing when the player needs to re-playing SEG with the same learning task. Additionally, we recommend the mapping process serves the following steps. Let the education materials be a series of learning tasks. One can identify the situations that elicit different outcomes when learning adjacent, significantly different (e.g., first and last chunk) or correlated knowledge chunks. Identifying those situations is somewhat abstract; however, a developer can put that in practice. Initially, the developer must estimate the specifications of a game cluster that supports an identified condition. Hence, additional rules can drive a more acceptable mapping concerning the player's experiences. Using the rule set, we can deploy learning materials and game elements automatically even when both have large spaces.

4. Case Study: Chem Dungeon for recalling chemical compounds

Using the method presented in Section 3, it allows a developer to transform an existing entertainment game into an educational game by embedding learning materials. Conceptually, the method should be applicable for combining various learning subjects and games. Therefore, the next subsections describe an implementation of our method based on an existing game, Chem Fight, including the solutions tackling the practical challenges.

4.1. Chem Fight

One of the authors (MP) developed the Chem Fight open sourced under MIT licensing,[1] a turn-based game that confronts a single-player versus a Non-Player Character in a chemical compound battle. Whereas, attributes of known 20 atoms from the Periodic Table (PT) and the atom bonding rules construct the gameplay.

Both players have some lives (red heart icon), energy (blue flash icon) and Atom Bucks (yellow dollar sign). The following paragraph explains the game mechanics with clarifications.[2]

The game consists of few rounds until one of the players loses all their lives. Each round contains a purchasing mode, one turn for the player to defend and another for attacking the NPC. The purchasing mode allows each player to buy atoms from the periodic table. An atom has a price specified by the atomic number (e.g., Helium [He] with atomic number 2 costs two Atom Bucks). On the first turn, one player attacks with a single atom. The opposing player (defender) only see the valence electron of the attacking atom. Thus, it earns a chance to appoint a number of atoms for defence. If the attacking atom creates a chemical bond with one or more of the defender's chosen atoms (a successful defence), the defending player receives rewards composed of a number of Atom Bucks and Energy Units. Otherwise, if there is no known possible compound between the attacking element and any of the defending elements, the attack is successful and the defending player receives a penalty for those unbonded defending elements. In fact, such a rule should discourage players from just defending with every element they own each time. Meanwhile, regaining the unused defending elements costs a decrease in energy. However, if the player has insufficient energy, their health decreases in proportion to the deficit. Once each turn ends, players earn a number of Atom Bucks to allow them to spend on additional elements.

4.2. Chem Dungeon: Game mechanics

This section demonstrates the game-play of the developed SEG as observed in Fig. 2. We named the serious game: Chem Dungeon, and deploy it in a web-based personal computer game. It is a single-player game against computer enemies because the original game content was intended for a single-player game. The game field consists of pathways and walls that form a maze with intersections and cul-de-sac. An exit gate, initially closed, is hidden at the bottom-right of the maze. Actors in the game consist of an avatar and some opponents, each with a spawn point. The avatar carries an atom within its shield where the corresponding information is located nearby its spawn point (top-left corner). There is a button to open the periodic table and a *Help* button to pause the game and show mission objectives. Meanwhile, information regarding a compound-forming result or an atom properties is at the top-centre of the game arena. The right side of the game (from top to bottom) contains lives (heart icon), experience progress (XP) in a red[3] bar, ammunition (numeric), the remaining time (90–0 s) and the total correct compounds collected. Inside the maze, bullets (yellow object), atom objects (blue object) and life potions (red object) are collectable for the avatar. Each bullet collected adds some ammunition for the avatar. A bottle of potion can restore the avatar's life to full.

The objectives of the game are collecting compound-forming atom objects and entering the exit gate within the 90-s time limit. Initially, the avatar starts from its spawn point while the enemies are spawned in the diagonal paths of the maze (bottom-left to top-right). The avatar can walk in 4-degrees of freedom: left, down, right, up controlled by keyboard keys *a*, *s*, *d*, *w*, respectively. When exploring the maze, the avatar should avoid collision with an enemy or an atom object. Otherwise, it loses a life when colliding with an incorrect atom object or a "normal" enemy. Luckily, shooting an atom object opens a path due to the shot atom is reallocated to another empty tile. Meanwhile, shooting an enemy transforms it to a weak mode (white-coloured character). A weak enemy re-spawns back to its home when crashing with the avatar, thus, opening another clear route. Accordingly, the avatar can seek and assemble the correct atom object which creates a compound. At this point, an educative message pops up which contains information concerning the chemical compound. Indeed, this game state should encourage players to read and retain knowledge in their memory. When the avatar has collected the correct atom object ten times, the exit gate reveals to open. Finally, by entering the exit gate, the avatar gets a

[1] Accessible online: http://js13kgames.com/games/chem-fight, and the source code is available online: https://github.com/mpalmerlee/ChemFight.

[2] Available online: https://github.com/mpalmerlee/ChemFight.

[3] For interpretation of color in Fig. 2, the reader is referred to the web version of this article.

Victory. Otherwise, losing all lives or running out of time issues a *Defeat*.

There are some helpful hints for players to play the game. Each game aims to form one compound (repeatedly). Novice players can adopt a trial-and-error strategy and are fully aware not to lose all their lives. Therefore, the player ought to actively read the text message at the top-centre position of the game which holds the latest result for the compound-forming attempt. Meanwhile, if only one life remains, a player can regain full lives by collecting a potion. Or, similarly, by accumulating experience (XP) bars through accurate shots and hit weak enemies. Once the XP reaches a full bar, one additional life replaces it. However, such an endeavour should consider the remaining bullets/ammunition and the 90-s time limit. These restrictions impede players abusing such tactical practices merely for entertainment while disregarding the goal of playing the game: memorising compounds' atoms.

All these game mechanics existed in the original rogue-like game. Only the chosen game mechanics were converted to represent the question (i.e. avatar's atomic shield) and choices of the learning task (i.e. coin collection). Additional elements were also added (i.e. learning-related information) to reinforce the knowledge acquisition task.

The next subsection outlines the construction of Chem Dungeon via our framework.

4.3. Chem Dungeon: Game development

Inspired by Chem Fight, we apply our proposed approach in Section 3 to develop a new SEG: Chem Dungeon. As a matter of fact, we use the library of education materials and the basic rule (pairing atoms to create a compound) as the core of Chem Dungeon. Moreover, an existing rogue-like game [4] is employed to represent the game content. Given both spaces are available, the following subsections describe the process details.

4.3.1. Learning Materials: Chemical compounds

The educational game has a purpose in promoting the memorization of chemical compounds for players. For this case study, there are 100 compounds composed of at least two atoms. The textual representation informs a compound's symbol, name and the atoms. For instance, *2 Hydrogen and 1 Oxygen construct an H2O* which is known as the *water* compound. The single atom appears as a game object with a text-based atomic symbol, e.g., *O, Ca, Cl.* Meanwhile, if the compound comprises of numerous atoms of the same type, it appears as a concatenation of strings between the total atom and symbol, such as *2O, 2H, 2Cl, 6B.*

According to Fig. 1, there are two general steps to proceed. First, given the periodic table data, attributes appointment operates according to the forming atoms and compound representation. Attributes of the forming atoms (atom-1 and atom-2) include *atom-1-number* (discrete),*atom-2-number* (discrete), *total-types-of-atom* (discrete) and *total-atom* (discrete). Attri-butes associated with compound and atom representations include: *total-character-symbol-1* (discrete) and *total-character-symbol-2* (discrete). Subsequently, a computer program retrieves necessary data from the periodic table and measures the total characters for the involving atoms, then, annotates the attributes automatically. For instance CO_2 contains one Carbon and two Oxygen atoms. The annotated values of this compound are *atom-1-number* = *6* (C), *atom-2-number* = *8* (O), *total-types-of-atom* CO_2 is 2 *(1 C + 1 O)*, the *total-atom* is 3 *(1 C + 2 O)*, *total-character-symbol-1* and *total-character-symbol-2* are both 1.

Second, with the fact that no correlations exist between compounds, the strategy of remembering them takes into account the properties. In fact, recalling them should be driven by the complexity of each compound. In other words, the more complex the representation of a compound, the more *difficult* it is to memorise. Accordingly, the strategy in our case associates with structuring education materials in a specific order based on the priority of attributes for sorting. Therefore, based on recall priority, compounds are ordered based on *total-types-of-atom*, *total-atom*, *atom-1-number* and *atom-2-number*, *total-character-symbol-1* and *total-character-symbol-2*, respectively. As a result, the easiest compound to remember is H2 (composed of two Hydrogen atoms) and the hardest to recall is CaB6 (formed from one Calcium atom with six Boron atoms). Hence, the sorted compounds are then represented by the CompoundID attribute which has numeric values from 1 to 100.

Table 1
Difficulty categorisation rule set.

Difficulty	Enemy-type	Total-enemy	Total-bullets	Maze
Easy	0	< 4	Any	
Medium	0	> 3		
	1	< 3		
Hard	1	< 2		

4.3.2. Game content space: rogue-like maze

The game content space was constructed from an existing rogue-like and maze game to confirm that it segregates from the learning materials. Henceforth, the categorization and mapping processes become revealing for our demonstration. As an overview, generating game elements using parameter values applies here which consist of *maze-id* (categorical), *enemy-type* (0: random-move enemy, and 1: smart enemy), *total-enemy* (1–5), *total-bullets* (1–5). By default, the game content space counts 48,600 different parameter configurations.

In **difficulty categorisation**, three levels of challenges separate the game content. To our best knowledge, the parameters *enemy-type* and *total-enemy* distinguish the difficulty quite noticeably within the rule set in Table 1. As a result, 22,365 of game content is categorised as **Easy**, 15,660 is **Medium**-level game content and 10,575 of content has a **Hard** difficulty level. Fig. 3 illustrates three different levels of difficulty. The image on the left is identified as an **Easy** game. Due to this fact, there is merely a single obstacle from one enemy which moves randomly, but the avatar can wander around the maze freely without very much concern being hit by the sole enemy. The image in the middle and on the right are **Medium** and **Hard** difficulty levels, respectively. In a medium-difficulty game, the avatar can still move freely although there are four enemies moving randomly. Thus, the game provides additional challenges for the player to avoid collision with these enemies. Meanwhile, the game content with five Smart enemies demands a high level of tactical practice in decision-making because these enemies are capable of traversing the shortest path to the avatar.

Our goal in the *similarity categorisation* is to provide a selection of similar game content for each learning material. To accommodate that, a clustering analysis builds ($k = 100$) clusters of similar game content inside a difficulty group. Given that *maze-id* parameter does not describe a maze explicitly, five numeric parameters provide details of the corresponding maze. The details are measures of maze's *total-path*, *total-corners*, *total-intersections*, *total-deadend* and *complexity*. Aside from this, we are aware of some issues: (1) the large size of game content space, and (2) the dynamic size of the content space due to the previously played games. Accordingly, our choice falls to Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) which is fast and flexible even with very large samples (details available in [20]).

For our case, configuring BIRCH with $k = 100$ and setting the branching factor $B = 2$ constructs a binary tree of game content space. Subsequently, the BIRCH operates to search for an optimum threshold value T which identifies 100 clusters with the highest silhouette score as an evaluation measure. The result of BIRCH on game content space under normalised values attests to Low, Medium and High difficulty groups using a threshold $T_l = T_m = T_h = 0.02$ to reach the highest silhouette of 0.23, 0.2 and 0.23, respectively.

[4] Available from http://www.kiwijs.org/.

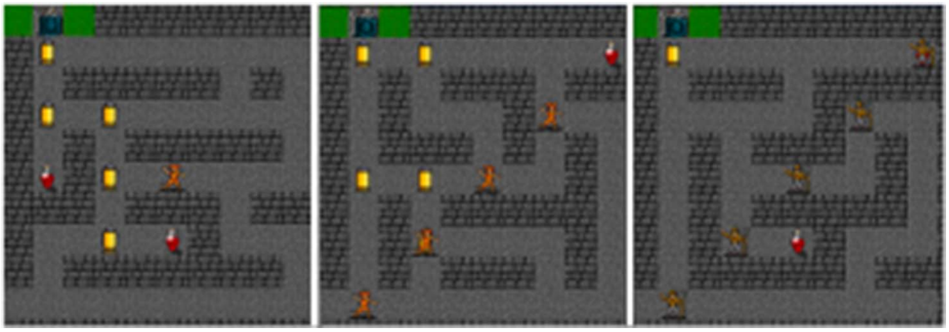

Fig. 3. Exemplary games of different difficulty levels (left-right): Easy, Medium, and Hard.

Overall, 300 clusters are identified and equally divided into three difficulty levels which are ready for deployment with the educational materials.

4.3.3. Mapping: a rule-based approach

Before carrying out the mapping, we manually identified, analysed and assigned candidates of game mechanics that fit our SEG construction. We took such steps by assuming a learning task as a multiple-choice question. Consider two groups of atoms (e.g., X and Y) that form a chemical compound. The question is one of the atom group (X) which appears on the avatar's atomic shield. And group Y is among the choices. We spotted some game mechanics that may represent the question-choices pair. Then, we sort them based on the interest level in the existing game content.

1. Battle an enemy.
2. Coins collection.
3. Bullets collection for adding ammunition.
4. Potions collection to refresh the lives.

Based on our knowledge, fighting enemies to *combine their atoms* has a conflict of purposes between the game mission and the learning process. In one hand, the fighting scenario performs *destructive* actions to be undertaken. In contrast, the avatar collects the correct atom(s) that *construct* a chemical compound. To some extent, the first game mechanic fails to meet our goal. Meanwhile, the remaining three game mechanics have purposes that potentially promote rote learning. However, the collecting-bullet interaction is considered less important because it is the prerequisite for shooting obstacles. Similarly, the potion collection is not convincing as well. Because there is an alternative to resuscitating lives by accumulating XP level to the maximum value. This game attribute is achievable via shooting obstacles, killing enemies or collecting coins. Alternatives to reach one goal are often introduced in various games to engage strategic actions. Thus, we prefer to keep potion-collecting as is. But other researchers or developers may alter its functionality as a hint towards the correct coin-atom to be formulated with the avatar's atom. In our case, the fittest game mechanic to endorse rote learning is the *coins collection*. In the SEG, we transform coins as atom objects that interact with the avatar's atomic shield. Moreover, the abundance of atom objects can serve the repetitions required for reinforcing the rote learning. Meanwhile, the enemies and incorrect atom objects are the obstacles of the SEG.

Previous steps categorised the game content and learning materials for the mapping process. The game content clusters carry details including total game content, the linear-sum of each parameter, the sum-of-squares of each parameter and the centroid of each cluster. These statistics can serve as the game content's description.

Our deployment strategy operates a rule-based method. Given the specifications found in learning material and content space, the following crucial rule applies: a compound deploys into three unique content clusters, all from different difficulty levels. Indeed, this rule ascertains no duplicates of game content for multiple compounds. However, additional criteria ensure an appropriate mapping based on our notion of possible learning conditions between *simple* versus *complex* compounds. We assume that the likelihood of failures to recall complex compounds may be higher than the easier ones. Thus, a higher number of games to support such repeated attempts may transpire for learning complex compounds. As a consequence, a slight difference in the game elements for recurrence of memorization may accustom the player to those games without the fear of boredom growing. Therefore, the player may have a *wider space* for focusing on the learning goal. Given these expectations, the cluster details resemble those aforementioned conditions including the quantity of game content (represents the number of repetitions) and the sum of standard deviations of game content features (represents the variety of games) under non-normalized parameter values. The following pseudo-code shows the deployment rules in practice.

```
1. ASSIGN education materials with string CompoundID $E_j$, where $j$: 0,1,...,($n$−1) (n is the learning materials size).
2. Within each cluster ($j$: 0,1,...,($n$−1)) of each difficulty level ($i$: 0,1,...($m$−1) total difficulty levels): COUNT total games ($N_j^i$), SUM the standard deviations of parameters ($S_j^i$) and ASSIGN the game content with string ID $G_j^i$.
3. SORT clusters within each difficulty level based on the value of $N$ (ascending) and $S$ (descending), respectively.
4. Create PAIRS of [$E_j$,$G_j^i$], where $j$: 0...($n$−1) and $i$: 0...($m$−1), enabling an education material gets a cluster of game content from each difficulty level.
```

Mapping priority starts with the number of games in a cluster and is followed by the standard deviation of the cluster. Fig. 4 depicts result details regarding the number of games for each compound and Fig. 5 shows deployment details with respect to variations of the game content. SQL-based tables store the mapping result and details of both content spaces.

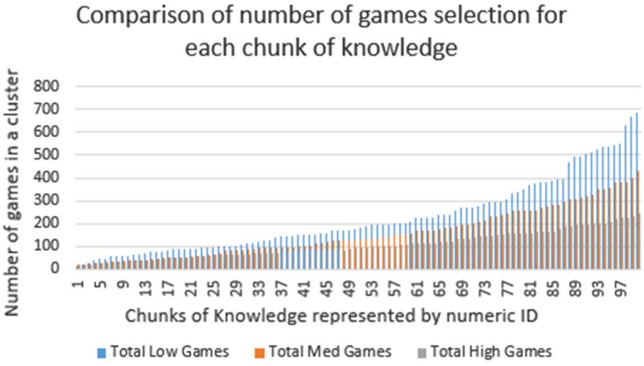

Fig. 4. Mapping result in terms of number of games in a cluster.

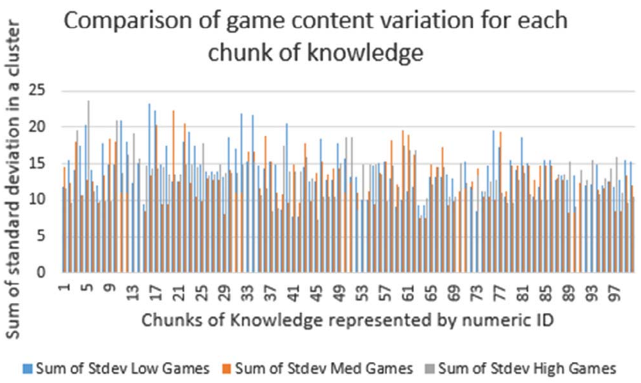

Fig. 5. Mapping result in terms of the sum of standard deviation of game content parameters in a cluster.

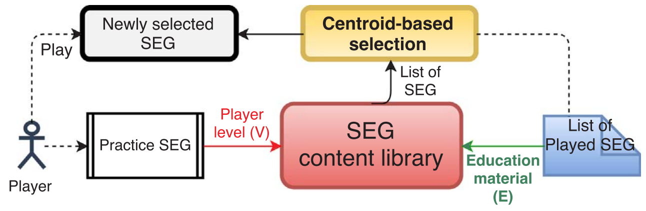

Fig. 6. Procedure in an SEG game session.

4.4. SEG game engine

Our framework developed a new game for players. Hence, a game engine should properly situate different players accordingly. Fig. 6 shows several stages in the SEG game engine. Initially, a new player should accustom him/herself with the game-play in the *practice game session* which contains the educational game with dummy learning materials. Meanwhile, an existing player may enter the practice game session for updating his/her Player Level. This session estimates the mastery level (denoted as V) of the player with the game based on his/her *score achievement*. Whereas, the mastery level V corresponds to the difficulty level of the game content.

In principle, the score originates from the player's game actions which consist of positive (a^+) and negative (a^-) actions. Logically, positive game actions increase *score* such as through successful navigation or accurate shots while negative game actions reduce *score*, for instance, a failed navigation or failed battle. In addition, various weights (if known by the developer) on particular actions may yield a more accurate scoring. Equation $score = \sum_i^k \alpha_i a_i^+ - \sum_i^l \beta_i a_i^-$ provides the basic formula for scoring, where a_i^+ be the ith positive game action and a_i^- be the ith negative game action. A value of k counts the number of positive game actions while l measures the total negative game actions. Values of α_i and β_i set the ith weights for positive and negative game actions, respectively. Then, the threshold values of *score* categorise a player into a particular level V.

Initially, a new player starts playing the game with the first learning task (E_0). Meanwhile, an existing player may progress the educational game according to his/her game session record (List of Played SEG). Based on V and E, the generation engine searches through the *content library* for a specific learning task, and the corresponding game content cluster, as game content candidates. For a new player, the candidates are *all* games in the selected cluster. Then, the played game content is excluded from being a candidate. Subsequently, a *centroid-based selection* chooses the closest game to the centroid x_m, measured by (1), of the pool as the newly selected game. Whereas, x_i be the i_{th} game content in the pool and n be the number of game content candidates.

$$x_m = \frac{\sum_i^n x_i}{n} \tag{1}$$

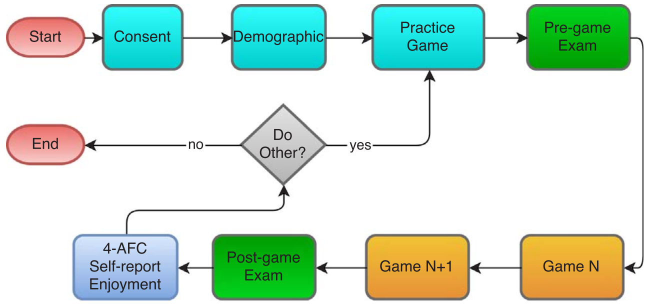

Fig. 7. Procedure for user survey.

Finally, the game engine generates the *newly selected game* composed of parameters incorporating the *CompoundID* (based on the value of E) and the value of V which associates the game content features.

5. User test

Developing an educational game using the method presented in this paper can produce a 'new' game, due to the mix of learning materials and game content. A survey containing the SEG allows players to play the game and report their experiences. The survey opens only for players at least 18 years old and computer literate.

Fig. 7 depicts the procedures for the survey commencing with a Consent form, Demographic questionnaire, Practice session and Pre-game Exam (randomly chosen learning materials). Afterwards, players play a pair of games, each with a single education material contained in the Pre-game Exam and a difficulty level for the game content according to his/her level measured from the Practice game session. Following each pair of games, players report their *fun* (enjoyment) from the latest pair of games, then, they complete a Post-game Exam. And each game session produces a log of gaming activities for further analysis. The consent form confirms a player's participation in the survey. Meanwhile, the Demographic form collects participant data, including age, location, player-id, email address and a unique code for players to re-enter the survey. A 4-AFC questionnaire expects a player to compare his/her enjoyment of both games [21,22]. Question-wording for the reported enjoyment appears as follows: (a) *Game N + 1 is more FUN than Game N*, (b) *Game N is more FUN than Game N + 1*, (c) *Both Games are FUN* and (d) *NONE of the Games are FUN*. Meanwhile, Pre and Post-game Exams employ Multiple Choice Question (MCQ) design [23,24].

Subsequently, a player may revisit the training session if s/he requires improving his/her gaming ability before continuing to the next section of the survey. Alternatively, s/he may opt to directly play a new pair of games initialised by completing another pre-game exam, or s/he ends the survey.

5.1. Data analysis

We administered the survey in three months and 50 players participating were adults and computer literate. The youngest player taking part was an 18-year old university student. Meanwhile, the oldest was a middle-aged participant (39 years old). On average, the age of the participants was around 27 years old. In the game session, participants were encouraged to play several pairs of games; hence, 540 reports obtained. Ten games were played on average. Four players played only a pair of games while 85 percent played between 4 and 14 games. One player played and reported 15 pairs of games.

Statistically, 352 reports confirmed the games were entertaining while 188 games reportedly not enjoyable. Table 2 summarises three z-tests evaluating H_0 against H_a. The null hypothesis H_0: $\pi = 0.5$, where π indicates the proportion of FUN reports. Given the 0.01 significance level, two z-tests reject the null hypothesis while 99% confident the proportion of FUN reports (0.652) is greater than 0.5 proportion.

Table 2
Z-test on proportion of gained enjoyment.

	Z-test, H_0 against:		
Indicators	H_a: $\pi \neq 0.5$	H_a: $\pi > 0.5$	H_a: $\pi < 0.5$
p-value	0.00000	0.00000	1
99% Conf. intervals	0.59–0.74	0.6–1.0	N/A
H_0 status	Rejected	Rejected	Rejection failed

We observed the game log and found that there are slight differences between various gaming activities that separate the reported Fun and Not Fun. We suspect that players interpret differently to such a subjective experience. One player feels 'entertained' if the game content fits his skill. Meanwhile, another player experiences an enjoyment when the game content is more difficult to conquer. This is a factor among many others that different players could have various perceptual/cognitive experience in response to the same stimuli. Moreover, the affective experience may change overtime or known as *concept drift* [11]. Providing thorough questionnaires that accommodate various aspects of enjoyment [25] can produce a consistent report.

Regarding the learning performance of the players, each question item in an exam represented a learning material. Thus, pre and post-game exams produced binary values indicating prior knowledge and recalling results, respectively. The difference in scores between pre and post-game exams produced three types of learning performances: unchanged, improvement and decay. However, we only use the unchanged and improved learning performances here. Because the negative score (decay) likely originated from arbitrary answers or random guess [26]. Therefore, we divided 309 reports involving not known prior knowledge into 219 game sessions of "improvement" and 90 sessions of "unchanged". For this case, the same z-tests operate using the same values for the null hypothesis and alternative hypotheses while π indicates the proportion of improved Learning. Table 3 summarises three z-tests results. Given the 0.01 significance level, two z-tests reject the null hypothesis while 99% confident the proportion of Improved Learning reports (0.694) is greater than 0.5 proportion.

Furthermore, we investigate the recorded gaming activities corresponding to *learning* and *not learning* outcome and find gaming activities seem correlated to "learning" outcome. In general, a game session where players recalled most of the education materials has more gaming activities than a game session where players only recalled few or no the education materials. In fact, the total time spent in reading the successfully collected compound corresponding to *learning* actions take around 15 s on average. In contrast, the *not learning* actions always take less than three seconds. Overall, the total actions in *learning* game sessions over the *not learning* in-game activities have been doubled approximately. This is due to the fact that the goal of such an educational game is designed to collect as many correct atoms as possible (i.e. reflects a "learning").

The statistical evidence in Table 2 confirms that the Chem Dungeon game considered as successful from the players' perspective regarding their learning and enjoyment, which is consistent with Pavlas' testimony [19].

On the other hand, we also look into the relationship between learning outcome and affective experience reported by the survey participants. Table 4 summarizes such information collected from all the game sessions. It is evident from Table 4 that our SEG allows more players to gain positive learning outcome and Fun together as there are 154 out of 309 falling into this category. This clearly demonstrates that the use of separate content spaces and a proper mapping proposed in our framework may lead to a SEG that fits all the characteristics described by Abt in 1970s [27]. He mentions that in a serious game, "learning" may be primary but other experiences involved should not be overlooked. Furthermore, serious games involve learning and entertainment dimension as a unity during game sessions [28–31]. Recent research by Pavlas found that enjoyment arising from the playing activities may affect the learning of a player in a serious game [19]. While the learning in serious games is a primary objective that any players have to achieve, our work emphasizes the importance of enjoyment (entertainment). Overall, the experimental results reported above indicate that, to a great extent, our game content and rules may elicit positive affective experience and many players gain such enjoyment when they engage in learning via game playing.

Table 3
Z-test on proportion of improved learning.

	Z-test, H_0 against:		
Indicators	H_a: $\pi \neq 0.5$	H_a: $\pi > 0.5$	H_a: $\pi < 0.5$
p-value	0.00000	0.00000	1
99% Conf. intervals	0.65–0.73	0.66–1.0	N/A
H_0 status	Rejected	Rejected	Rejection failed

Table 4
Survey results: learning outcome vs. affective experience.

	Not learning	Learning
Not fun	42	65
Fun	48	154

6. Discussion

It is worth stating that our current case study based on Chem Fight is subject to limitation. Chem Fight is designed purely for SEG in which the game mechanics strongly correlate the properties of atoms and compounds. Nevertheless, the Chem Dungeon introduces a PCG-based SEG which currently aiming at rote-learning. It presents repetition strategy that can be applied directly to the game content. However, there is an opportunity to apply our scheme for a more complex type of knowledge, such that found in the *understanding* category. Our framework also accepts some learning types under this category that includes grouping, identification, recognition, selection or translation. Based on the fact that they allow learning strategy similar with recalling, i.e. repetition. Meanwhile, learning methods (in the understanding category) such as describe, discuss, explain and report are not suitable for our development framework. They require learning strategies beyond repetition. In that, the game content should provide a proper platform to facilitate such a high order of learning. For instance, a narrative-based game and multi-player games are the potentials because a player can convey or present his/her knowledge to other players. Additionally, the scenario generation (such as via narratives [32,33]) or meta-cognitive learning support in the game [34] maybe helpful embedding a complex knowledge. Hence, the higher the complexity of learning materials to be embedded into a game content space, the smaller applicability of our framework to build the SEG.

Currently, our framework fits suitably with the existing action game genre. Other game genres like action-adventure, adventure, logic (e.g., puzzles or mazes) and trivia are also suitable because their gameplay and game mechanics allow direct mapping with the rote-learning subjects. It requires relatively the same portion of modifications to the target game content. For instance, a platformer game such as Mario Bros. can be modified to ask the player to collect atom items instead of power items. Then, Mario/Luigi must enter a warp-pipe to form chemical compounds accordingly. We can also modify a puzzle game, Tetris for instance, by transforming its puzzle objects into atom objects (boxes) and whenever the adjacent atom boxes form a chemical compound, a corresponding information pops up. Basically, our framework accepts a wide range of game genres, especially, the games with simple and easy-to-play mechanics. A game genre like Real Time Strategy (RTS) may be too complex to be applied in our framework. In that, the developer is taking a critical role in altering the current game mechanics, rules or scenarios to allow mapping process.

Meanwhile, our method is focusing on single-player games. Indeed, the produced SEG was distributed to be played in a single-player mode only. Typical players, especially Achievers and Killers types [35], should prefer learning-playing in this way. We have not tested our framework in a multiplayer game setting. If we deploy the produced SEG as a multiplayer game, we have to provide an additional feature in the SEG to give an impression of a multi-player game. The simplest to add is the player ranking based on their overall learning tasks achievements, scores or other measurement methods. Alternatively, we can also apply a multiplayer game content into our framework. The produced serious game will have to undergo through a higher degree of modifications from the source game content. This is driven by the added dimensionality of interactions involved in the game session, such as communication and competition between players. Notably, social-type and exploration-type players are the hardcore-fan of this typical serious game where they exploit those multiplayer-game features to engage themselves [35]. Thus, the developer has to do some creativity to ensure the optimal utilisation of multi-player facilities for players' learning. For instance, one can modify the scenario that encourages players using the chat room to discuss the solution of a learning task. Sung et al. give the insight to apply collaborative learning in a multi-player SEG through what so-called knowledge engineering process that promotes a higher level of cognition [36].

7. Conclusion

We have successfully developed Chem Dungeon using a strategy that combines education materials and an entertainment game. Retrieving inherent details of the learning materials demonstrates advantages in two regards. First, the learning materials has a natural description held by the attributes enabling a developer to organize them semantically. In the second, computer programs can automatically annotate those attributes with little interference from experts and with a concern merely for the learning materials' size. On the other hand, the procedurally generated game elements in our approach unlock another route towards rapid development of SEG in which categorization becomes automated using a combination of rule-based approach and machine learning. Hence, those detailed descriptions underlying both content spaces facilitate a developer in establishing the mapping rules based on his/her knowledge. Besides, the two-space structure could be a baseline for further research in a procedural serious game generation wherein attributions are concerning learning materials, game elements, and game rules. Moreover, the new educational game we have developed using our method has shown reasonable results in supporting players' learning and entertaining them.

In our ongoing research, we are further developing the game to enable it to predict players' experiences via gaming data, which would lead to a corresponding adaptation method for personalized learning in the SEG. In the future, we are also going to investigate our proposed approach to new SEG development by combining PCG-based or multi-player entertainment game platforms and different learning materials.

Acknowledgement

The authors would like to thank the action editor and anonymous reviewers for their valuable comments that improve the presentation of this manuscript.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.entcom.2018.01.001.

References

[1] D. Michael, S. Chen, Serious Games: Games that Educate, Train, and Inform, Thomson Course Technology, 2006.
[2] J.P. Gee, Good video games and good learning, Phi Kappa Phi Forum, vol. 85, The Honor Society of Phi Kappa Phi, 2005, p. 33.
[3] S. de Freitas, S. Jarvis, A framework for developing serious games to meet learner needs, in: Interservice/Industry Training, Simulation and Education Conference.
[4] R. Ibrahim, A. Jaafar, Educational games (eg) design framework: Combination of game design, pedagogy and content modeling, International Conference on Electrical Engineering and Informatics, 2009. ICEEI'09, vol. 1, IEEE, 2009, pp. 293–298.
[5] D. Ismailović, J. Haladjian, B. Köhler, D. Pagano, B. Brügge, Adaptive serious game development, 2012 2nd International Workshop on Games and Software Engineering (GAS), IEEE, 2012, pp. 23–26.
[6] A.M. Hussaan, K. Sehaba, A. Mille, Tailoring serious games with adaptive pedagogical scenarios: a serious game for persons with cognitive disabilities, in: ICALT, 2011, pp. 486–490.
[7] D. Djaouti, J. Alvarez, J.-P. Jessel, Classifying serious games: The G/P/S model, in: P. Felicia (Ed.), Handbook of Research on Improving Learning and Motivation through Educational Games, IGI Global, 2011, pp. 118–136 (Ch. 6).
[8] J. Alvarez, D. Djaouti, An introduction to serious game definitions and concepts, Serious Games & Simulation for Risks Management (2011) 11.
[9] J.P. Gee, What video games have to teach us about learning and literacy, Comp. Entertain. (CIE) 1 (1) (2003) 20.
[10] D.W. Shaffer, How Computer Games Help Children Learn, Macmillan, 2006.
[11] J. Roberts, K. Chen, Learning-based procedural content generation, IEEE Trans. Comput. Intell. AI Game 7 (1) (2015) 88–101.
[12] K.M. Kapp, The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education, John Wiley & Sons, 2012.
[13] F. Bellotti, R. Berta, A.D. Gloria, L. Primavera, Adaptive experience engine for serious games, IEEE Trans. Comput. Intell. AI Game 1 (4) (2009) 264–280.
[14] F. Bellotti, R. Berta, A. De Gloria, L. Primavera, A task annotation model for sandbox serious games, IEEE Symposium on Computational Intelligence and Games, 2009. CIG 2009, IEE, 2009, pp. 233–240.
[15] J. Togelius, G.N. Yannakakis, K.O. Stanley, C. Browne, Search-based procedural content generation: a taxonomy and survey, IEEE Trans. Comput. Intell. AI Game 3 (3) (2011) 172–186.
[16] G.N. Yannakakis, J. Togelius, Experience-driven procedural content generation, IEEE Trans. Affect. Comput. 2 (3) (2011) 147–161.
[17] M. Csikszentmihalyi, Flow: The Psychology of Optimal Experience, Harper Perennial, USA, 1990.
[18] M.E.C. Consortium, Number Munchers Game < https://archive.org/details/Number_Munchers_v1.2_1986_MECC > (cited 3 November 2017).
[19] D. Pavlas, A Model of Flow and Play in Game-Based Learning: The Impact of Game Characteristics, Player Traits, and Player States, Ph.D. thesis, University of Central Florida, 2010.
[20] T. Zhang, R. Ramakrishnan, M. Livny, Birch: an efficient data clustering method for very large databases, ACM Sigmod Rec. vol. 25, ACM, 1996, pp. 103–114.
[21] C. Pedersen, J. Togelius, G.N. Yannakakis, Modeling player experience for content creation, IEEE Trans. Comput. Intell. AI Games 2 (1) (2010) 54–67.
[22] F. Jäkel, F.A. Wichmann, Spatial four-alternative forced-choice method is the preferred psychophysical method for naïve observers, J. Vis. 6 (11) (2006) 13.
[23] M. Papastergiou, Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation, Comp. Educ. 52 (1) (2009) 1–12.
[24] M. Manske, C. Conati, Modelling learning in an educational game, Proceedings of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning Through Intelligent and Socially Informed Technology, IOS Press, Amsterdam, The Netherlands, 2005, pp. 411–418.
[25] F.-L. Fu, R.-C. Su, S.-C. Yu, Egameflow: a scale to measure learners enjoyment of e-learning games, Comp. Educ. 52 (1) (2009) 101–112.
[26] L. Bao, Dynamics of Student Modeling: A Theory, Algorithms, and Application to Quantum Mechanics, Ph.D. thesis, University of Maryland at College Park, 1999.
[27] C. Abt, Serious Games, Viking Press, 1970.
[28] L. Michaud, J. Alvarez, Serious Games: Advergaming, Edugaming, Training and More, IDATE Consulting & Research.
[29] K.P. Jantke, Toward a taxonomy of game based learning, 2010 IEEE International Conference on Progress in Informatics and Computing (PIC), vol. 2, IEEE, 2010, pp. 858–862.
[30] M. Zyda, From visual simulation to virtual reality to games, Computer 38 (9) (2005) 25–32.
[31] U. Ritterfeld, M. Cody, P. Vorderer, Serious Games: Mechanisms and Effects, Routledge, 2009.
[32] J. Rowe, B. Mott, S. McQuiggan, J. Robison, S. Lee, J. Lester, Crystal island: a narrative-centered learning environment for eighth grade microbiology, in: Workshop on Intelligent Educational Games at the 14th International Conference on Artificial Intelligence in Education, Brighton, UK, 2009, pp. 11–20.
[33] N. Peirce, O. Conlan, V. Wade, Adaptive educational games: Providing non-invasive personalised learning experiences, in: 2008 Second IEEE International Conference on Digital Games and Intelligent Toys Based Education, 2008, pp. 28–35, doi:http://dx.doi.org/10.1109/DIGITEL.2008.30.
[34] B. Kim, H. Park, Y. Baek, Not just fun, but serious strategies: using meta-cognitive strategies in game-based learning, Comp. Educ. 52 (4) (2009) 800–810.
[35] R. Bartle, Hearts, clubs, diamonds, spades: players who suit muds, J. MUD Res. 1 (1) (1996) 19.
[36] H.-Y. Sung, G.-J. Hwang, A collaborative game-based learning approach to improving students' learning performance in science courses, Comp. Educ. 63 (2013) 43–51.