1-13-2017

Player–Game Interaction and Cognitive Gameplay: A Taxonomic Framework for the Core Mechanic of Videogames

Kamran Sedig
Department of Computer Science, Western University, London, ON N6A 3K7, Canada

Paul Parsons
Department of Computer Graphics Technology, Purdue University

Sedig, Kamran and Parsons, Paul, "Player–Game Interaction and Cognitive Gameplay: A Taxonomic Framework for the Core Mechanic of Videogames" (2017). *Computer Graphics Technology Faculty Publications.* Paper 3.
https://docs.lib.purdue.edu/cgtpubs/3

informatics

MDPI

Article

Player–Game Interaction and Cognitive Gameplay: A Taxonomic Framework for the Core Mechanic of Videogames

Kamran Sedig [1,*], Paul Parsons [2] and Robert Haworth [1]

[1] Department of Computer Science, Western University, London, ON N6A 3K7, Canada; rhaworth@uwo.ca
[2] Department of Computer Graphics Technology, Purdue University, West Lafayette, IN 47907, USA; parsonsp@purdue.edu
* Correspondence: sedig@uwo.ca; Tel.: +1-519-661-2111 (ext. 86612)

Academic Editor: Antony Bryant
Received: 19 December 2016; Accepted: 9 January 2017; Published: 13 January 2017

Abstract: Cognitive gameplay—the cognitive dimension of a player's experience—emerges from the interaction between a player and a game. While its design requires careful consideration, cognitive gameplay can be designed only indirectly via the design of game components. In this paper, we focus on one such component—the core mechanic—which binds a player and game together through the performance of essential interactions. Little extant research has been aimed at developing frameworks to support the design of interactions within the core mechanic with cognitive gameplay in mind. We present a taxonomic framework named INFORM (Interaction desigN For the cORe Mechanic) to address this gap. INFORM employs twelve micro-level elements that collectively give structure to any individual interaction within the core mechanic. We characterize these elements in the context of videogames, and discuss their potential influences on cognitive gameplay. We situate these elements within a broader framework that synthesizes concepts relevant to game design. INFORM is a descriptive framework, and provides a common vocabulary and a set of concepts that designers can use to think systematically about issues related to micro-level interaction design and cognitive gameplay.

Keywords: player–game interaction; core mechanic; interaction design of games; taxonomic framework; cognitive gameplay; cognition; videogames; game design; representation

---

1. Introduction

Considerable research shows that videogames can support, augment, and/or enhance cognition [1–7]. When designing the visual interfaces of such games, at least two key issues must be considered carefully: representation and interaction [8]. Representation design is concerned with how game information (e.g., the player avatar, objects, structures, paths, mazes, etc.) is encoded and displayed visually, and interaction design is concerned with what players can and should do with the represented information, what actions should be made available to them to work and think with the representations, and what their subsequent reactions should be. Thus, the focus of interaction design is the discourse that takes place between the player and the visual representations—i.e., player–game interaction. Through interaction with representations, players engage with game information and perform cognitive activities such as problem solving, planning, decision-making, and learning [9].

The overall experience that emerges from a player interacting with a game is commonly referred to as *gameplay* [10]. Gameplay has many facets, one of which is *cognitive gameplay*—a term that refers more specifically to the cognitive processes that emerge during player–game interaction (see [7]). Here we construe cognitive processes generally, and include those related to both low-level cognition (e.g., attention and memory) and high-level cognition (e.g., problem solving and planning). Because cognitive gameplay is an emergent phenomenon, it can be designed only indirectly via the design of game components. Many game components influence gameplay (e.g., information content, player–game interactions, mechanics, graphics, goals, narrative, and rules), and various design frameworks have been developed that consider some of these components (e.g., [11]). However, little research has been aimed at developing frameworks that can support systematic design of cognitive gameplay.

One critical issue to consider in the design of cognitive gameplay is a game's core mechanic, which refers to the set of essential interactions that are repeatedly performed, in a cyclical fashion, while the game is being played [12]. For example, in the game *Tetris*, the essential interactions are rotation and movement of a shape, and these are repeated continually while playing the game. Consequently, these two interactions form the core mechanic of *Tetris*, and it is primarily through these interactions that gameplay emerges [10,13].

Considerable research shows that interaction design decisions can significantly influence cognition [9,14,15]. Interaction design issues can be discussed at many levels, ranging from high-level philosophical and pedagogical issues to low-level operational and implementation issues. Frameworks and guidelines have been developed to help designers think about cognitive gameplay and high-level interaction design issues, such as those related to experiential learning [16] and constructivism [17]. With respect to low-level interaction design and cognitive gameplay, however, there is a significant gap in the extant literature. To help fill this gap, we propose a taxonomic framework named INFORM (Interaction desigN For the cORe Mechanic), in which low-level interaction design issues are explicated systematically. INFORM extends and adapts elements of an existing framework, EDIFICE-IVT, which, among other things, deals with micro-level interaction design for visualization tools [18]. EDIFICE-IVT is a preliminary framework that was developed for analyzing interaction at micro and macro levels. In EDIFICE-IVT, a set of twelve structural elements of micro-level interaction are characterized in a general manner. In the INFORM framework, we employ these twelve elements, adapt their characterizations to suit the context of videogames, and extend discussions of their potential cognitive influences. We also situate them within a broader set of concepts relevant to game design, such as cognitive gameplay and the core mechanic. Thus, the twelve elements from EDIFICE-IVT function as a skeleton structure, and are significantly extended and re-contextualized in INFORM to make them suitable for use in videogame contexts. As a taxonomic framework, INFORM is descriptive; accordingly, it is not intended to prescribe what designers should do in specific contexts or how specific design decisions should be made. Rather, it provides a common vocabulary and a set of concepts that designers can use to think about and discuss issues related to micro-level interaction design and cognitive gameplay. Additionally, although we discuss implications of micro-level interaction design on cognitive gameplay, the aim here is not to predict cognitive effects of design decisions, nor is it to prescribe design actions to achieve intended cognitive outcomes; such concerns are beyond the current scope of the INFORM framework.

We anticipate that INFORM will enable increased systematicity with respect to three main activities: (1) analysis of micro-level interaction elements within the core mechanic of existing games; (2) design of interactions within the core mechanic of new games; (3) evaluation of the effects of micro-level interaction design decisions on cognitive gameplay. By facilitating the analysis of interactions at a micro-level, videogame researchers can create different versions of games in which interactions within the core mechanic are isolated and varied, enabling the effects on cognitive gameplay to be empirically evaluated. At this stage of the research, only a few variations have been studied in the context of videogames (e.g., see [7,19]). Thus, while we know these elements exist ontologically, affect cognitive gameplay, and are important to investigate, further research is needed to better understand their influence on cognitive gameplay in a systematic, principled manner. We present the INFORM framework with the aim of facilitating this line of research.

The structure of this paper is as follows. In the second section, we examine necessary background concepts and terminology. In the third section, we identify and characterize the twelve elements of the INFORM framework. In the fourth section, we provide an integrated example that demonstrates the application of the framework in a design scenario. Finally, in the fifth section, we provide a summary and discuss some areas of future research.

2. Background

In this section, we present necessary conceptual and terminological background.

2.1. Videogames

Game researchers and developers have discussed, debated, and attempted to identify the essential characteristics of games for many years now (e.g., [20–22]). Although there is no commonly agreed upon definition, in this paper, we use the following definition of Salen and Zimmerman [10]: "A game is a system in which players engage in artificial conflict, defined by rules, that results in a quantifiable outcome" (p. 80). Videogames are specific types of games: they are systems; they engage players in non-real or artificial conflict that is defined by rules; and they have quantifiable outcomes. In addition, they operate on interactive, electronic, computational devices or platforms. The hardware or software used for this platform is not an essential part of the definition; for our purposes, a videogame could be implemented on a personal computer, tablet computer, game console, or mobile device.

2.2. Cognitive Gameplay

The term 'gameplay' can refer to the interaction that occurs between a player and a game [23]. It can also refer to a player's experience that arises from this interaction. To indicate that gameplay refers to an experience, some researchers use terms such as gameplay experience [24] or game experience [25]. Accordingly, in this paper, 'gameplay' is an emergent phenomenon and refers to that which a player experiences when interacting with a game.

As gameplay is a composite construct, it can be decomposed into a number of different dimensions [25]. Many of these dimensions focus on the emotional or aesthetic aspects of experience, such as immersion, tension, or flow [24]. There is also the cognitive dimension of the player's experience [26–28]. We refer to this dimension as *cognitive gameplay*. This dimension includes high-level cognitive activities which emerge from playing a game (e.g., problem solving, planning, and learning) as well as lower-level cognitive processes (e.g., memory encoding and retrieval, pre-attentive processing, and so on). In other words, cognitive gameplay "signifies the emergent cognitive processes within the overall gameplay experience" [7] (p. 249).

Cognitive gameplay is influenced by characteristics of the player (e.g., mental models, thinking styles) and characteristics of the game (e.g., information representation, interaction, rules). Being an emergent phenomenon, cognitive gameplay cannot be designed directly, and has to be designed indirectly via the design of game components. Two key components of games are information content and interaction. Variation in the design of these two components results in different forms of cognitive gameplay [6,7,19]. These two components are discussed next.

2.3. Information, Content, and Representation

Games are often classified into two groups: those for entertainment purposes, and those for non-entertainment purposes (e.g., serious games, educational games; see [29,30]). The tendency for such classification highlights the implicit assumption that games designed for cognitive activities are inherently not entertaining. As a result, a common design approach is to attempt to combine the entertainment aspect of a game in the context of its cognitive activity (e.g., problem solving or learning) [31]. Moreover, such activities are often content-focused—that is, the primary focus is on the information content that should be delivered to the player—thus placing content at the heart of design (e.g., [32]). This tendency can be seen in many games that have desired cognitive goals (e.g., [11,26–28,33,34]).

This common approach leads to an inflation of the role of content in the design of cognitive gameplay. Much research suggests that other factors are at least as important when it comes to engaging the player's cognition. For example, the manner in which information is represented, rather than the information content per se, significantly influences cognitive processes (e.g., see [35–37]). In fact, from the perspective of the player, the *representation is the content*—that is, since the only access the player has to content is through representations, there is a unity of meaning between the information content and its representation [38].

During gameplay, a player engages with information through representations in the visual interface of the game. It is important to note that information content can also be represented and communicated to the player through auditory, tactile, and other modalities. However, such considerations are outside the scope of this paper. Since the content is represented visually, we henceforth refer to these as visual representations (or simply, representations). Representations (e.g., diagrams, tables, symbols, images, and so on) can encode content intended to engage players in cognitive activities. Representations can also encode other game-related information, such as the current game state and possible actions that a player can perform. For example, consider the game *KAtomic*. In a screenshot of the game (Figure 1), we can identify various representations and the type of information that they encode. Some representations encode game content, such as individual atoms, walls, and the molecule that the player needs to create. Some representations also encode educational content, such as the chemical composition and form of specific molecules. There are also representations that encode possible actions, such as the arrows on which the player can click to move an atom in a specific direction. Some of the representations act as containers, to visually group other representations in the interface. When all of these are considered, one can see how the entire interface of this game consists of representations of the underlying content. As mentioned above, research has clearly shown that different representations of the same information can have significantly different effects on cognitive processes. Although full discussion of this issue is outside the scope of this paper, it is very important for researchers and designers to understand the role of representations in influencing cognitive gameplay.

Figure 1. A screenshot of the game *KAtomic*, used to demonstrate how the entire interface of a game consists of representations of underlying content. *KAtomic* is copyright 1998–2012 KAtomic Team, released under the GNU General Public License (GPL) version 2.

2.4. Cognition and Interaction

Research shows that cognition is distributed across the brain and its external environment [39,40]. Cognition is influenced not only by social, cultural, and contextual factors, but also by objects and by the ways in which people interact with them. For instance, Kirsh and Maglio [41], studying how people played the game *Tetris*, discovered that cognitive processes during gameplay were extended into the external environment through the performance of epistemic actions—actions performed to facilitate mental operations rather than to achieve physical or pragmatic goals. Participants would often rotate and translate *Tetris* shapes within the game not only to achieve the pragmatic goal of placing the shape in a desired location, but also to facilitate mental computation. Moreover, the study determined that it was quicker, easier, and less costly, in terms of attention and memory for participants, to operate on the shapes in the game than to operate on them in the head alone. In other words, the study suggested that the manner in which the participants interacted with objects within the game had a significant impact on their cognitive processes. That is, *player–game interactions had a significant effect on cognitive gameplay*.

All games have components that serve a representational function (e.g., tiles that represent numbers or other concepts, avatars that represent the player, and arrows that represent shooting). Videogames allow the player to engage with a game through interacting with these representations in a dynamic fashion [6,9,42]. In a broad sense, interaction refers to an active, reciprocal relationship between a player and a game [9], with this interaction taking place at different levels of granularity: from low-level interface events, to interaction techniques, to interaction patterns, to specific tasks and high-level activities. At the lowest level, interaction refers to physical actions and events, such as touches, clicks, and drags. At a higher level, interaction can refer to more abstract, general patterns, such as a player rearranging tiles, moving through a game space, or assigning behavior and/or properties to game entities. At the pattern level of interactions, we can think about them independent of the platform and technology on which they are implemented [9]. At still a higher level, interaction gives rise to the emergence of cognitive activities, such as problem solving. Finally, at the highest level, interaction often refers to philosophical design approaches, such as those related to distributed, situated, or embodied cognition; behaviorist or constructionist forms of interaction; or cognitive apprenticeship. In this paper, we are mainly concerned with interaction patterns at the level of action–reaction pairs—that is, the player performing an action and the interface of the game reacting. It is this reciprocal action that binds the player and the game together. Interaction design at this level—the design of actions and reactions—ultimately influences cognitive gameplay (For an in-depth discussion of interaction levels and patterns, the interested reader can refer to [9]).

Viewing individual interactions at the level of patterns allows for a common characterization of the core mechanic; a common method of analyzing the structural elements of individual interactions within the core mechanic; and a common vocabulary for conceptualizing and discussing the elements that give rise to the core mechanic of a game. Without such an approach to interaction, it is extremely difficult to have a general, comprehensive framework that can support systematic thinking about the core mechanic in the context of cognitive gameplay.

2.5. Core Mechanic: The Cognitive Nucleus

All games have mechanics [43,44]. There is no clear agreement among practitioners and researchers, however, as to what constitutes the mechanics of a game [12]. The term ‘game mechanics’ is used as a broad construct that includes such things as rules, methods, feedback, interactions, player behaviors, game objects, and algorithms (e.g., see [43,45–48]). Although what constitutes the mechanics of a game is not clear, there is typically a *core mechanic* that can be identified in any game. This core mechanic has been defined as the “patterns of repeated behavior” [10] and the “essential interactions which a player repeats during play” [13]. In other words, the core mechanic of a game is the continual pattern of interaction with a game. As was stated before, player–game interaction can be analyzed at many levels of granularity. Since we are interested in the role and structure of interaction in the context of the core mechanic of games, we focus on the level of interaction patterns discussed in the previous section—i.e., neither the physical-, event-level interactions, nor high-level cognitive activities. At this level, the player performs an action and the game responds in some way. This pattern of action–reaction is then repeated continually to form a cycle, and this cycle is what constitutes the core mechanic. For example, consider the game *Breakout*. In this game, the player moves a paddle to either the left or the right to hit or intercept (action) a ball (representation), and the ball bounces back (reaction) from a wall (representation) in the game space. This action–reaction pattern is repeated continually and makes up the core mechanic of the *Breakout* game. As another example, consider the *Super Mario Brothers* game. In this game, the player moves the character Mario around on the screen. Mario can walk, run, and jump to different heights. The player can also perform an occasional special move to more easily defeat enemies or overcome obstacles, such as launching a small fireball. Thus, the core mechanic of this game is composed of the actions of walking, running, jumping, and occasionally performing a special attack and the reactions of these actions within the game space.

The core mechanic of any game, then, consists of a set of player–game interactions that repeatedly occur. Hence, it is mainly through the core mechanic that the player acts upon the game, is engaged in cognitive processes, and cognitive activities emerge. As such, the core mechanic is the *cognitive nucleus* of gameplay. For systematic analysis, design, and evaluation of cognitive gameplay, then, it is important to examine what the structural elements of player–game interactions are.

3. INFORM: Interaction Design for the Core Mechanic

As mentioned previously, we have adapted and extended elements from the preliminary EDIFICE-IVT framework in the development of INFORM. According to EDIFICE-IVT, any single interaction can be analyzed into multiple elements that collectively give it structure [18]. Moreover, there are different types of each element, and varying these types can influence cognitive processes. As an individual interaction has both an action and a reaction component, micro-level interaction elements can be categorized into action elements and reaction elements. The twelve elements are divided into six action elements (agency, flow, focus, granularity, presence, and timing), and six reaction elements (activation, context, flow, spread, state, and transition). Table 1 lists these twelve elements.

Table 1. Elements of Interaction in the INFORM Framework.

Action	Reaction
agency	activation
flow	context
focus	flow
granularity	spread
presence	state
timing	transition

While playing a game, the following process typically occurs: a player performs an action on representations in the visual interface; representations change as a reaction to the action (although not the focus of this paper, changes may also occur in the game 'internals' as a result of the action—e.g., the values of internal data structures may change); during action and reaction, the player is perceiving the interface and performing mental operations. This sequence repeats itself over and over, forming the core mechanic of the game. Figure 2 depicts this process.

In the remainder of this section, we characterize and discuss each element and examine some possible types. Examples of existing games will be given where appropriate. The following four points should be considered while examining this section: (1) the terms used here may not be found in the game literature since a framework such as INFORM does not currently exist; as described previously, we have borrowed terms that were devised for the EDIFICE-IVT framework; (2) the discussed studies do not necessarily use the same terms as we are using here, even though they may be examining the same phenomenon; (3) as this is a young area of research, not every element has been studied in the context of cognitive gameplay; where appropriate, we discuss relevant research in other areas, which we expect can be used to guide future studies; (4) finally, it is important to keep in mind that it is ultimately the combination of these elements, and their aggregate mutual influences within the core mechanic, that leads to the emergence of cognitive gameplay.

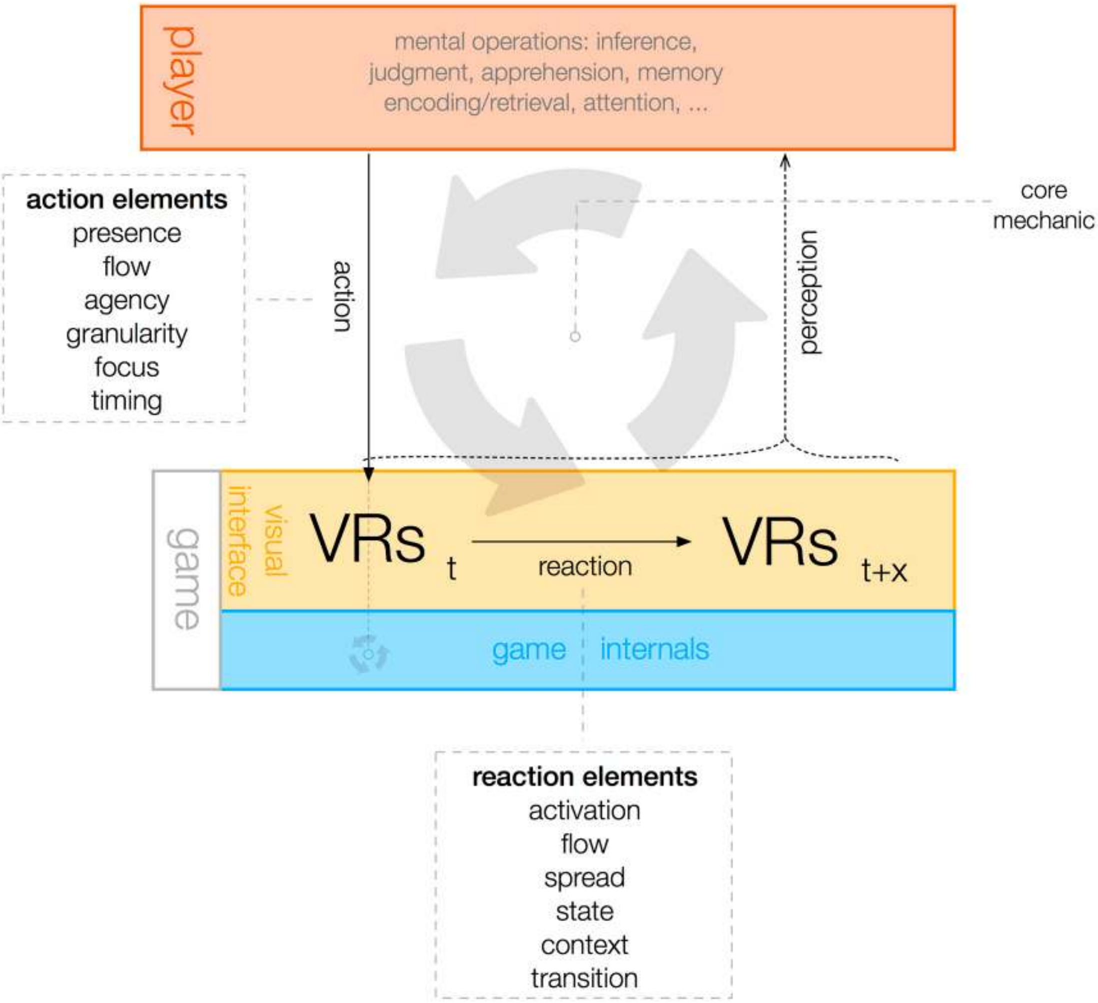


Figure 2. Depiction of the process of playing a game: a player performs an action on representations in the visual interface; representations change as a reaction; the player perceives the interface and performs mental operations throughout. VR stands for visual representation.

3.1. Elements of Action

3.1.1. Agency

This element is concerned with the metaphoric way through which a player expresses an action. In other words, this element deals with how a player articulates an action directed towards representations in a game. There are at least two types of agency: verbal and manual. In verbal agency, the player expresses an action using her ‘mouth’, as though she speaks to a game, such as by typing a command into a console. In manual agency, the player expresses an action using her ‘hands’, as though she is reaching into the interface to grasp and manipulate representations in a game, such as using a mouse cursor to rotate an object.

For an example of the different types of agency, consider the puzzle game *Tower of Hanoi* [49,50]. In this game, there are three pegs and several disks of different sizes. The disks can slide onto any peg. To start the game, the disks are stacked on top of each other in ascending order of size on one peg.

The goal of the game is to move the entire stack to another peg by following these rules: only one disk may be moved at a time from one peg to another; only the upper disk from a peg can be moved, though it can be placed on top of another disk; and, finally, no disk may be placed on top of a smaller disk. The core mechanic of this game includes one interaction: moving the topmost disk from one peg to another. In a videogame based on *Tower of Hanoi*, agency could be implemented as follows. With verbal agency, the player could type commands that allow her to move pegs. If the game had three pegs, they could be labeled A, B and C. The player could then type "move A to B" and press enter, and the game would move the top disk from peg A onto peg B. With manual agency, the player could use the mouse to move pegs, such as clicking on a disk, dragging it to the desired target peg, and releasing the mouse button to drop the disk onto the peg.

Svendsen [51] conducted a study to investigate how interaction design influences thinking and problem solving. Two versions of a game based on the *Tower of Hanoi* were created. The versions differed in terms of types of agency—similar to our description in the previous paragraph. Participants who played the version with verbal agency had to think harder and made fewer mistakes than participants who played the version with manual agency. Hence, the type of agency may give rise to different depths or degrees of reflectiveness in cognitive gameplay.

3.1.2. Flow (Action)

This element is concerned with how an action is parsed in time. There are two types of flow: discrete and continuous. An action with discrete flow occurs instantaneously in time and/or is punctuated over time. An action with continuous flow occurs over a span of time in a fluid manner.

As a simple example of implementation of flow in the design of action, consider again *Tower of Hanoi* with manual agency. With discrete flow, the player could click on a disk to select it, and then click on the target peg to move it to that location. With continuous flow, the player could click and drag a disk from one peg to another. A study about the potential effects of flow on cognitive gameplay, for both action flow and reaction flow, is provided later in this paper in the sub-section on reaction flow.

3.1.3. Focus

This element is concerned with the focal point of action with respect to the intended target of action—that is, the target representation to which a player attends in order to act upon a representation of interest. There are two types of focus: direct and indirect. With direct focus, the player acts on the representation of interest. With indirect focus, the player acts on an intermediary representation to effect change in the representation of interest. Hence, in direct focus, the player acts directly on the target representation, while in indirect focus, the player acts indirectly on the target representation via an intermediary representation.

For an example of implementation of different types of focus, consider a game based on the *Chinese Tangram* puzzles [52]. A tangram puzzle includes a 2D outline or silhouette of a shape, and seven 2D polygons of various shapes and sizes. The objective is to arrange the polygons so that all of them fit inside the outline without overlapping each other. Each puzzle differs in the outline given, but the same seven polygons are always used. The core mechanic of this game includes two interactions: rotating a polygon, and moving a polygon into, out of, or within the silhouette. In a videogame based on the *Chinese Tangram* puzzles, focus could be implemented as follows. With direct focus, the player chooses the type of operation to perform (i.e., move or rotate) and then acts on the polygon to transform it accordingly. For instance, to move the polygon, the player chooses the "move" operation and then drags the polygon to its desired location. With indirect focus, the player again chooses the operation, selects the polygon to transform, and then an intermediary representation of the transformation would appear, such as a representation of the arc of rotation or a translation vector. Then the player adjusts the parameters of the intermediary representation (e.g., adjust the magnitude and direction of the translation vector) and then commits the action by clicking on a button labeled "go" (see Figure 3). For instance, if the player wants to move a polygon, she would choose the "move" operation, select the desired polygon to move and the translation representation would appear, adjust the position of the ends of the representation, and then commit the action. The polygon would then move according to the distance and direction indicated by the translation representation.

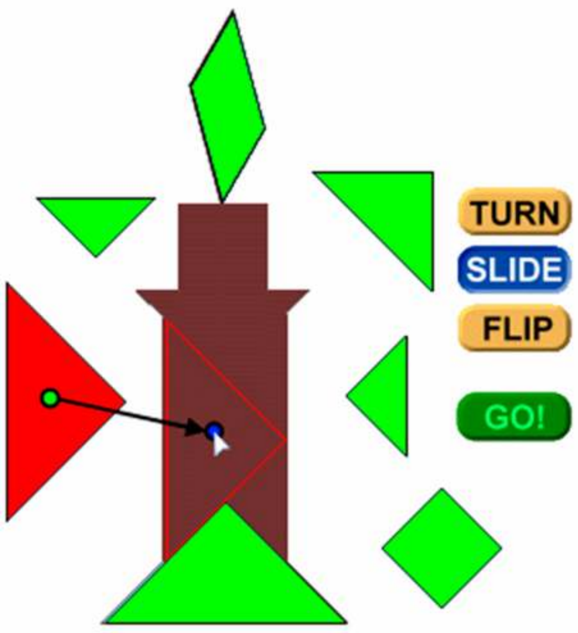


Figure 3. A game based on *Chinese Tangram* puzzles, in which the player is moving a polygon by acting on an intermediary representation (a representation of the translation vector).

Sedig and colleagues [19] conducted a study to investigate the effects of different interface styles on cognition and learning of transformation geometry. Two versions of a game based on the *Chinese Tangram* puzzles were developed, which differed in terms of types of focus, similar to our description in the previous paragraph. The results of the study showed that different types of focus affected learning significantly, measured by differences in the results of a pre-test and post-test on transformation geometry. Even though there was no significant difference in their pre-test results, the participants who played the version with indirect focus performed significantly better on the post-test than the participants who played the version with direct focus. The authors concluded that the focal point of the players' action affected their attention. With direct focus, the participants' focal point of attention had been the polygon shapes, paying little attention to the transformation operations. With indirect focus, their focal point of attention had been the transformation geometry representations, hence paying more attention to how to adjust those than paying attention to the polygons. Hence, this structural element of interaction—focus—and the different types that it takes may affect perceptual and mental attention, and consequently cognitive gameplay.

3.1.4. Granularity

This element is concerned with the steps that the player needs to perform in order to compose an action. There are two types of granularity: atomic and composite. An action which has atomic granularity cannot be decomposed into steps. In other words, an atomic action is itself the only step. An action which has composite granularity can be broken down into more than one step.

As an example, consider the game *Angry Birds* for the iPad. In this game, the player launches a bird from a large slingshot towards a structure so as to destroy the structure. The core mechanic of this game includes one interaction: launching a bird in a parabola. The player taps and drags the bird so as to adjust the launch parameters (e.g., angle of the firing arc, and force applied to the bird), and then she releases her finger from the screen so as to launch the bird. In this case, the main action has atomic granularity. Adjusting the launch parameters and firing the bird are combined into a single step, and that is the only step the player performs in order to act. However, the game could be redesigned for granularity to be composite. To compose an action, the player would go through the following steps. First, she would drag a slider that changed the amount of force applied to the bird. Second, she would drag the bird to adjust the angle for the firing arc. These two steps could be continually performed, but not simultaneously. Unlike in the atomic granularity case, the bird would not be launched when the player released her finger, and thus she could look at the parameters to determine if they were acceptable. When the player is ready to launch the bird (i.e., commit the action), she would tap a button labeled "launch".

The potential effect of granularity on cognitive gameplay can be seen in the previous example with *Angry Birds*. With atomic granularity, the player does not see or examine the finer details of the action, but instead thinks of it as a single whole; in *Angry Birds*, it is difficult for the player to think about the firing angle and force as separate parameters since they are both encapsulated into the same step. With composite granularity, the player can examine and think about the finer details of the action; in the modified *Angry Birds*, the player manipulates the firing angle and force separately, thinks about them as separate parameters, and has the opportunity to carefully fine-tune and reflect upon them before committing the action. Therefore, this element may play a role in how the player engages with and reflects upon the detail, depth, and composition of information, hence affecting cognitive gameplay.

3.1.5. Presence

This element is concerned with whether the game advertises the existence of an action to the player. It has two types: explicit and implicit. In explicit presence, the availability or existence of an action is explicitly advertised to the player. In implicit presence, even though an action is present, its availability is not advertised to the player, and it is assumed that the player knows that it exists. Presence, as a structural element of interaction, is similar to the concept of cognitive affordance that deals with making the semantics of an artifact detectable, observable, and/or understandable to a user [53].

As an example, consider the videogame *KAtomic* shown in the introduction. The goal of this game is to move chemical elements through a maze so as to create the appropriate chemical compound. The core mechanic of this game includes one interaction: moving a chemical element in a specific direction. When the player clicks on a chemical element to move it, a set of arrows appears around the chemical element to indicate the direction in which it could move. These arrows advertise the possibilities for action. Similarly, when the player moves the mouse over a chemical element, it flashes again to advertise the possibility of action. Both of these are examples of explicit presence; the action of moving the chemical element is explicitly advertised. However, imagine if this game were implemented on a tablet or cellphone with a touch screen. It could be changed so that a chemical element does not flash and no arrows appear. Instead, the player simply swipes in a desired direction over a chemical element and this performs the action (i.e., the element moves in the designated direction). In this case, presence is implicit; the player is not informed that such an action can be performed and instead must learn this some other way or must already know about it.

As in the case of cognitive affordance, different types of presence offer tradeoffs in terms of cognitive gameplay. With explicit presence, there may be too many interface elements to notify the player of action possibilities; this may become perceptually overwhelming. Making too many things explicit may lead to confusion [53,54]. However, implicit presence may place extra cognitive load on the player by requiring her to remember the existence of action possibilities. The player may also have to search for actions or information regarding how to act, spending time on tasks unrelated to the game. Either type of presence can affect cognitive processes. Studies are needed to carefully investigate how these different types, when integrated in the core mechanic of a game, influence cognitive gameplay.

3.1.6. Timing

This element is concerned with the amount of time available to the player to compose and/or commit an action. There are at least two types of timing: player-paced and game-paced. When the timing is player-paced, the player is not constrained by any time limitations for composing and committing an action. Using this type of timing, the player has as much time as needed to think about and examine a situation before committing an action. When the timing is game-paced, the player has a set amount of time to compose and commit an action before the game imposes a penalty on the player (e.g., her action is canceled, she is prevented from acting on the same representation again, she must restart the current level).

For example, consider a game based on the *Tower of Hanoi* [49,50] described above. The timing element of the main interaction (i.e., moving a disk from one peg to another) can be implemented to be either player-paced or game-paced. If timing is player-paced, the player has unlimited time for each action. If timing is game-paced instead, in the corner of the screen would be a timer that is continually counting down. Whenever the player moves a disk, the clock would reset to its initial value (e.g., 30 s). As such, the player only has 30 seconds to move one disk, but gains additional time when a disk is moved. If the clock runs out, the game ends and the player must restart.

Van Harreveld, Wagenmakers, and van der Maas [55] investigated the effect of time constraints on problem-solving skill and thought processes of *Chess* players. Within a standard game of chess, a difference in skill noticeably affects the performance of players. However, as the time constraints become increasingly strict, skill eventually ceases to matter; players of any skill level play similarly. In other words, when players had enough time to think to the best of their ability, they performed better. When players were given too little time to think, they became more prone to making mistakes, regardless of skill level. Thus, the ability to think clearly and deeply (modulated by the players' skill level) depended on how much time was available. Hence, this structural element of interaction—timing—and the different types that it takes seem to affect clarity of thought and depth of processing and analysis (i.e., the type of timing seems to affect cognitive gameplay). Further studies should carefully investigate how these different types of timing can be used singly or in concert in the core mechanic of games and what their effects are on cognitive gameplay.

3.2. Elements of Reaction

Collectively, the six elements of reaction can also be referred to as feedback. Although feedback has been discussed by multiple researchers and game developers, the term is often used in a manner that does not distinguish between levels of interaction. Feedback at the level of an individual interaction can be better understood by characterizing the structural elements that make up the reaction component of the interaction. In this paper, reaction refers to effects of an action that are visibly perceptible in the interface, and not those that may occur internally in the game and are hidden from view of the player (e.g., changes in internal data representations). Furthermore, an action often results in the interface going through fluctuations before the reaction process is completed and the interface reaches equilibrium. Therefore, some of the reaction elements discussed here are concerned with the reaction during fluctuation while others are concerned with the reaction as the interface reaches an equilibrium and the reaction process is complete.

3.2.1. Activation

This element is concerned with the commencement of a reaction after the player has committed an action. There are at least three operational types of activation: immediate, delayed, and on-demand. In immediate activation, the reaction occurs instantaneously after the action is committed. In delayed activation, an action is committed and then a span of time passes before its reaction occurs. In on-demand activation, the reaction only occurs once the player requests it.

For an example, consider the game *Temple Swap* (see [7]). In this game, the player is given a set of tiles with different symbols. The main interaction available is to swap the position of two adjacent tiles. The tiles can form rows or columns, and the goal of each level is to create rows and columns such that each tile is in at least one row or column with a common symbol. The core mechanic of this game contains one interaction: swapping two adjacent tiles. The types of activation in this game could be implemented as follows. With the immediate type of activation, the tiles would immediately move once the player has selected two tiles to swap. With the delayed type of activation, the player could select two tiles to swap but they would not move until after the player has selected the next pair. With the on-demand type of activation, the player can select tiles to swap but no swapping would occur until after she has clicked on a button labeled "go". At that point, all the paired tiles would swap in order, and the player would then see the sequence of swaps that she had performed.

Sedig and Haworth [7] conducted a study to investigate the effect of different types of activation on cognitive gameplay. Two versions of *Temple Swap* were created, using immediate and on-demand activation as described above. The results of the study indicated that on-demand activation promoted more effortful and long-term thinking, while immediate activation prevented this or encouraged less effortful and shorter-term thinking. In the version with immediate activation, participants did not try to plan carefully but merely moved tiles around and eventually discovered a solution. In the version with on-demand activation, participants were able to plan carefully and felt that the game required them to think this way. From this study, it seems that the type of activation can influence the degree of mental effort exerted, and whether the player thinks in a more reflective or reactionary manner. Hence, this element of interaction should be designed carefully as it can play an important role in the core mechanic of games.

3.2.2. Context

This element is concerned with the general context in which representations exist as the interface reaches equilibrium. There are two types of context: changed and unchanged. Before an action is committed, a representation exists within some context. During the reaction process, that context can change or it can remain the same. With 'changed' context, representations will be in a different context once the reaction process finishes. With 'unchanged' context, representations remain in the same context after the completion of the reaction.

For example, consider a game in which the player must navigate a 3D space to reach various goal points in a particular order. In this game, the core mechanic includes one interaction: navigating through the in-game environment. As the player navigates the environment of the game, there is no change in the context: the representations in the interface still show the world from the orientation and position of the player within it (i.e., a first-person perspective). However, the context could change in the following manner. When the player reaches a goal point, the interface changes so that it presents the player with an overhead view of the whole space, highlights the position of the player, and indicates the position of the next goal. In such a scenario, the player may need to press a button to return to the previous context of moving in the world from a first-person perspective. In this example, the context element of reaction is 'changed' when the player reaches a goal point and 'unchanged' when the player moves throughout the rest of the world.

The potential influences of different types of context on cognitive gameplay are not known. While it is possible that 'changed' context may interrupt the player and cause disruption, such as disorientation during navigation, this has not been studied empirically. However, research on the role of context has consistently shown that it has significant effects on cognitive performance and memory tasks (e.g., [56–58]). Both recall and recognition tasks degrade when there are changes in context, even when embedded information stays the same. Several possible explanations include context dependence, need for mental model realigning, need for increased mental processing for foregrounding of information, and need for new segmentation of experience in a new context. On the other hand, a change of context may require new representations, and considerable research in the cognitive sciences shows that different representations of the same information can help users understand and mentally model the information better [59–62]. Research is needed to investigate how different types of this element should be implemented in videogames to lead to desired cognitive gameplay.

3.2.3. Flow (Reaction)

This element is concerned with how a reaction is parsed in time. There are two types of flow: discrete and continuous. A reaction with discrete flow occurs instantaneously in time and/or is punctuated. A reaction with continuous flow occurs over a span of time in a fluid manner.

As an example, consider again the game *Angry Birds*. After the player adjusts the launch parameters and launches the bird, it moves gradually across the screen until it collides with a structure or the ground. In this case, the reaction has continuous flow. However, the flow of the reaction could have been designed to be discrete. For instance, after the bird was launched, it could immediately appear at the endpoint of the arc and show any resulting destruction. In this case, the reaction has discrete flow since the reaction (i.e., the movement of a bird along the firing arc) occurs instantaneously in time.

Liang and colleagues [63] conducted a study to investigate the effects of different types of both action flow and reaction flow on cognition. Even though this study is not in the context of a game, it deals with an exploratory computer activity that accurately captures the concept of flow and how it could be applied to games. The study involved a software tool that used representations to aid users in exploring the structural properties and relationships of certain 3D geometric solids. Four versions of this tool were created, one for each combination of the types of action and reaction flow. The results of the study suggested that the type of flow affected cognitive processes and overall learning, as measured by the difference in scores between a pre-test and post-test on transformation and properties of the solids. The participants who used the version with discrete action and reaction flow performed the best on the post-test. Based on participant behavior, it seemed that those who used the continuous action and reaction flow version did not reflect much on the content that they were exploring, as the tool made the performance of action and interpretation of reaction less mentally demanding. In contrast, participants who used the discrete action and reaction flow version spent more time reflecting and planning their actions. The authors concluded that the ease and intuitiveness of continuous flow may not always be desirable, as it can be counter-productive when reflective cognition and investment of mental effort are needed. Hence, the different types of this element and the manner of their design can play a significant role in cognitive gameplay and how much effort players put into the processing of game information.

3.2.4. Spread

This element is concerned with the spread of the effect that an action causes. An action can cause a change to occur in a particular representation of interest, or the effect can propagate to other representations in the game. Hence, there are two types of spread: self-contained and propagated.

As an example, consider the videogame *Bejeweled*. In this game, the player is given a grid full of different colored gems. The player needs to swap adjacent gems so as to create a match of three or more gems with the same color. Hence, the core mechanic contains one interaction: swapping two adjacent gems. When a match occurs, all of the matched gems are removed from the grid, all gems above the removed ones in the grid fall down into the empty space, and new gems are added to the grid to fill in any remaining empty space in the grid. When the player swaps two gems and nothing else occurs (i.e., a match is not created), then spread is the self-contained type. However, when a match occurs and other gems are removed from the grid, then the type of spread is propagated. The most extreme example of propagated spread in *Bejeweled* is when a cascade of matches occurs. One match removes several gems, and others fall into the empty space, but new matches are made; this causes more gems to be removed, others to fall in, more matches made, and so on. It is quite possible to create a long cascade of matches, all triggered by a single action.

The potential influences of different types of spread on cognitive gameplay are unknown. Even though this element exists ontologically, as demonstrated from the above example, we have not been able to find any empirical studies that investigate how this structural element affects cognitive processes of players. Consequently, there is much room for research to investigate how different types of spread and their combinations can affect cognitive gameplay.

3.2.5. State

This element is concerned with the conditions of the interface (i.e., the interface's representations) once the reaction process is complete and the interface reaches equilibrium. There are three types of state (that is, the states that representations affected by an action can assume): created, deleted, or altered. In a created state, new representations are created and appear in the interface. In a deleted state, one or more representations are deleted from the interface. Finally, in an altered state, one or more properties of some representations (e.g., their value, position, size, orientation, etc.) are modified.

As an example, consider a game in which the player must guide a robot through a grid-based maze full of different types of objects. The core mechanic for this game contains one interaction: moving the robot through the grid. This can be done by indicating the direction in which the robot is to move, how many grid cells it should move, and then pressing a "go" button. The reaction component of this interaction is that the robot moves accordingly and may interact with other objects in the maze. For instance, as the robot is moving, it can hit a boulder and push it in the same direction of movement as the robot; this change in the position of the boulder is an implementation of the altered type of state. Similarly, the robot could pass over a switch, triggering the disappearance of walls from the grid (i.e., deleted state). Alternatively, the robot can pass over another kind of switch, causing the spawning of new walls that are placed in the path of the robot (i.e., created state).

As in the case of the previous interaction element—spread—even though state exists ontologically, as the above example shows, we have not been able to find any empirical studies that investigate how this structural element affects cognitive processes of players. Empirical studies need to be conducted to investigate how different types of state and their combinations can affect cognitive gameplay.

3.2.6. Transition

This element is concerned with how change is presented on a 2D display. Representations in games that are dynamic and/or interactive are spatio-temporal entities. As such, when an action causes change in them, the change can take place along both the temporal and spatial dimensions [64]. Consequently, presenting change effectively can be a difficult design challenge, as the result is often distortion of one of the dimensions. Based on the two dimensions for presenting change, there are two general types of transition: stacked and distributed [64].

When transition is stacked, changes in a representation are sequentially 'stacked' one on top of another in time. Although the representation is visually changing over some duration of time, only the current state at one point in time is perceivable; past stages of change disappear and future stages are not shown. This type of reaction behaves like a movie, where changes to a scene are stacked in time and one scene replaces another. If transition is distributed, multiple stages of change in a representation are simultaneously perceivable by being spatially distributed. Of all the changes through which an entity (e.g., player or game object) may pass, several are chosen as snapshots. Those snapshots are then displayed as new representations in the game, such that the player can view them in parallel without previous stages disappearing in time. This type of transition is similar to a storyboard, where transitional scenes are shown as separate images. Hence, stacked transition constrains the visual change to one location while distributed transition communicates changes over a region of space.

Potential cognitive effects of transition can be gleaned from a study conducted by Sedig, Rowhani, and Liang [65]. In this study, the effect of types of transition on conceptual navigation was investigated. Although this study was not in the context of a game, the activity being investigated (i.e., navigation) is common in games; hence, this study is applicable for understanding the types of transition and how to design the transition element in games. For the study, a software tool was developed in which participants explored relationships between 3D geometric solids. Participants were expected to develop cognitive maps of the objects and the paths of change from one form of an object to another. Three versions of the tool were developed in which transition was implemented as stacked only, distributed only, and combined. In the combined version, the user interface contained the representations from both the stacked and distributed interfaces. These representations were linked together; acting on one representation caused both representations to react. The results of the study indicated that participants improved their ability to navigate the conceptual space in different ways (i.e., they developed cognitive maps differently). Participants developed landmark knowledge (the main objects) in the stacked transition version, route knowledge (the paths of change or transition) in the distributed transition version, and survey knowledge (the overall landscape of objects and paths) in the combined version. Therefore, each type of transition (and the combination of these types) has its own inherent strengths and weaknesses. For instance, spatially distributed change requires the player to make constant back-and-forth eye movements, while this is avoided when change is temporally stacked in one location [64]. However, stacking temporally forces a player to compare different states of an object from memory, since she cannot see multiple states of an object simultaneously; this issue is avoided in the distributed type of transition.

As the example study shows, this structural element of interaction—transition—and the different types that it takes affect perceptual, memory, and mapping processes of cognition—hence, cognitive gameplay—and must be analyzed and designed carefully. Figure 4 shows the two types of transition that were used in the study—distributed (left) and stacked (right).

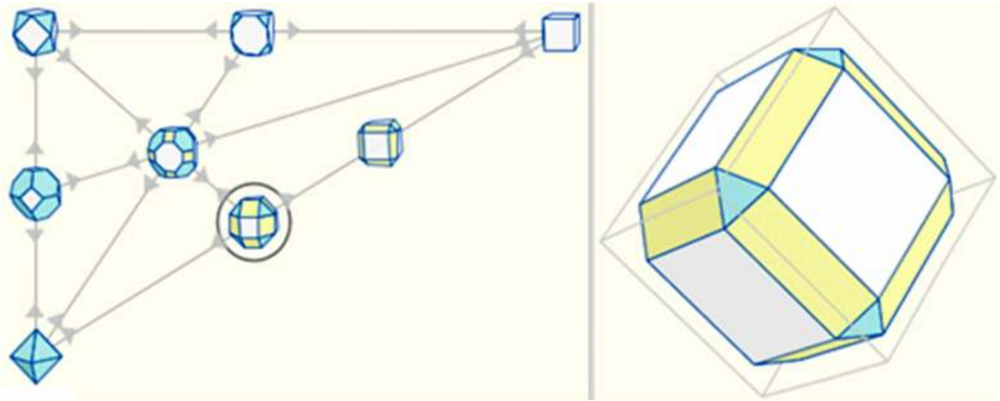


Figure 4. An interface in which transition is distributed (**left**) and stacked (**right**). From the tool used in [65].

4. Application of INFORM in a Scenario

Here, we present a scenario to demonstrate how INFORM can be used in the design and evaluation of the core mechanic of a videogame. Since INFORM is a descriptive *taxonomic framework*, and is not intended to offer prescriptive guidance for game designers, this scenario will demonstrate how INFORM facilitates systematic thinking about cognitive gameplay.

As a brief example, consider a researcher wanting to investigate how different design variations affect planning activities. Using INFORM, this researcher can identify the types of specific structural elements that he or she thinks will affect planning, and then study the identified types systematically. For instance, assume that *activation* is one of the structural elements identified. To determine what types of this element have an effect on planning, if any, the researcher could conduct an experiment by randomly assigning participants to play one of three versions of the same game: the version that implements the *immediate* type of activation, the *delayed* type, or the *on-demand* type. Then, depending on the results, the researcher could focus on other structural elements and conduct further studies to narrow down what combinations seem to work best for desired planning strategies.

An experiment such as this was conducted in [7]. In this experiment, participants playing the version with the on-demand type of activation reported engaging in deeper and more effortful planning than the ones playing the version with the immediate type. Once the effect of one element is determined, the effects of other structural elements can also be systematically studied.

The remainder of this section will describe a more detailed example. Consider a researcher studying differences in how players engage in spatial reasoning and mental manipulation of three-dimensional shapes. To start, the researcher designs a videogame with a simple core mechanic. Game components that may impact the desired cognitive gameplay can be minimized, so that effects from the structure of interaction can be more easily identified. In this hypothetical game, the player must recreate a series of patterns in a step-by-step fashion. For every level of the game, the player is given a visual pattern and a cube. The pattern is broken down into square sections, matching the size and shape of the faces on the cube. Each face has a different image that corresponds to specific sections of the pattern. In other words, the pattern is a composition of the faces on the cube. The player is also given a blank working space in which to create a copy of the given pattern. To create a copy, the player rotates the cube and stamps the topmost face into a section of the working space. This is repeated until each section of the working space is equivalent to the corresponding section of the pattern. Thus, the core mechanic is composed of two interactions: *rotating* the cube, and *stamping* one face of the cube into the working space. To win this game, the player must rotate the cube such that the face showing the next section of the pattern is the topmost one, stamp that face into the necessary sections of the workspace, and repeat this until the pattern is successfully copied. The player is also shown the number of rotations that were performed to copy the pattern. To play efficiently, the player must determine how to rotate the cube as few times as possible. Figure 5 depicts the components of this hypothetical game.

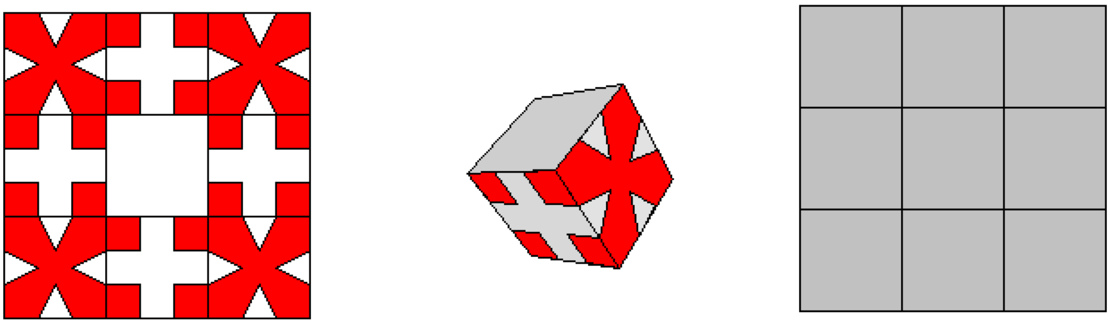


Figure 5. A screenshot of this hypothetical videogame showing the pattern to recreate (**left**); the cube with which the player interacts (**middle**); and the working space in which a copy of the pattern is created (**right**).

This game is designed to encourage the player to reflectively engage in a series of spatial reasoning tasks. To achieve the intended outcome (i.e., winning the game), the player must develop her spatial reasoning abilities to meet increasingly difficult challenges. To study how the core mechanic could be implemented to most effectively support spatial reasoning, the researcher could use INFORM to guide the design of different variations of this videogame. An experiment could be conducted for each structural element characterized in INFORM, providing the researcher with a systematic approach to design and evaluate this videogame. To illustrate how INFORM could be used in this manner, the rest of this section will provide examples of how each type for all twelve elements of the rotating interaction could be implemented for this hypothetical videogame. Depending on the game, there are many possible ways that a type can be implemented; therefore, due to spatial constraints, we provide only one implementation to demonstrate how thinking about the elements and their types assists with systematic design. For the sake of brevity, only the rotating interaction is analyzed here. This method of analysis, however, should clearly indicate how a similar analysis could be done for the stamping interaction.

- **Agency:** *manual* or *verbal*. With manual agency, the player rotates the cube using the mouse—e.g., by clicking and dragging the cube. Alternatively, the cube could be rotated by clicking on arrows that appear around the cube as the mouse cursor approaches it (see Figure 6). With verbal agency, the player could type commands into a console to rotate the cube—e.g., "rotate left" to rotate the cube to the left once, and "rotate left 3" to rotate the cube to the left three times (see Figure 8).
- **Focus:** *direct* or *indirect*. With direct focus, the cube is the focal point of action—e.g., the player clicks on and drags the cube to rotate it. As the interaction is directed toward the intended target (i.e., the cube), focus is direct. If the focus were indirect, however, another representation would be the focal point of action. A simple example is having a set of arrows in a control panel in the interface. When the player clicks on an arrow, the cube rotates in the corresponding direction. This is indirect focus, as the action is directed toward arrows in the control panel, and not the cube itself. As another example, the player could be provided with an additional, alternative representation of the cube (e.g., a formula that represents its 2D projection). The player would be required to act on the alternative representation to control the rotation of the cube.
- **Flow (action):** *discrete* or *continuous*. Consider the above examples in which the player clicks on an arrow button to rotate the cube. These are examples in which action flow is discrete, as the action occurs instantaneously: the player simply clicks a button. Alternatively, the player could click on and drag a slider to specify the amount of rotation—an example where action flow is continuous since the action happens over a period of time in a fluid manner.
- **Granularity:** *atomic* or *composite*. With atomic granularity, the interaction has only one step—e.g., dragging the cube to rotate it. With composite granularity, the interaction would require more than one step to complete. For example, to rotate the cube, the player would have to first select it, then specify the desired direction of rotation, then select a button to execute the rotation. Although the composite type of granularity may seem unnecessary here, it could be quite relevant if, for example, the player could supply important additional parameters to the action (such as the angle or speed of rotation).
- **Presence:** *implicit* or *explicit*. With implicit presence, the possibility of rotating the cube is not advertised to the player. The player must have existing knowledge that the shape can be rotated and how to go about rotating it. For instance, if the player could rotate the cube by clicking on it and dragging the mouse, but nothing on the cube or anywhere else in the interface suggests that this action is possible, then presence is implicit. With explicit presence, the possibility of rotating the cube would be advertised to the player. One simple example is to have a label below the cube stating: "To rotate the cube, click and drag it". Alternatively, the cube could be wiggling with a small rotation sign attached to it to suggest the possibility of this interaction.
- **Timing:** *player-paced* or *game-paced*. With player-paced timing, the player can take as much time as she wants to rotate the cube. However, if timing is game-paced, a time restriction is placed on the player. For example, there could be a timer that begins at 60 seconds and counts down. When it reaches 0 seconds, the player loses points. Every time the player performs the rotating action, the timer for this interaction can be reset.
- **Activation:** *immediate*, *delayed*, or *on-demand*. With immediate activation, the cube rotates as soon as the player acts—e.g., the player clicks an arrow button and the cube rotates immediately. With delayed activation, the cube would not rotate until a period of time elapses or some other event occurs. For instance, the player clicks on the arrow button to rotate the cube to the left but the cube does not rotate until a subsequent action is performed. With on-demand activation, the player could specify a sequence of multiple rotations, but the cube would not rotate until a separate button is clicked, and the sequence would unfold (see Figure 9).

- **Context:** *changed* or *unchanged*. In the examples thus far, context is implemented with the unchanged type (i.e., the context in which the interaction occurs does not change once the reaction is finished). One possible implementation of the changed type would be the following: the game places a limit on the number of times the player can rotate the cube. Once the player rotates the cube beyond this limit, the game reacts by resetting the level, and this changes the context in which the player is operating. This context change will likely force the player to think about recreating the whole pattern again, and remember the set of steps used before the level was reset.
- **Flow (reaction):** *discrete* or *continuous*. The following is an example of continuous reaction flow: the player clicks on and drags a slider to rotate the cube, and the cube gradually rotates until its orientation matches that specified by the slider's position. If the reaction flow was discrete, however, the cube's orientation would immediately change to match that specified by the slider's position—it would not change gradually over time. In this case, the reaction occurs instantaneously without any fluid motion. Separating action and reaction flow can be conducive to mindful planning in certain situations (see [63]).
- **Spread**: *self-contained* or *propagated*. In the case of self-contained spread, only the focal representation is affected. For instance, if the player drags a cube to rotate it, no other representation in the interface is affected. However, in the example given below in which transition is distributed, the player types a command to rotate the cube and, as a result, multiple representations are created to display the orientation of the cube at different stages in the rotation. In this example, the spread is propagated to other representations in the interface.
- **State:** *created*, *deleted*, and/or *altered*. To demonstrate implementations of the different types of this element, the game will enforce a limit on the number of times the cube can rotate. Once this limit is reached, the player can no longer rotate the cube and must restart the puzzle. Consider the case in which the player rotates the cube by typing a 'rotate' command into a console. Assume that, in addition to the cube rotating, representations elsewhere in the interface encoding the number of performed rotations are also affected. One possibility is that, as the cube is rotated, the color and/or arrangement of other representations change to reflect the number of remaining rotations that are available to the player. In this case, the properties of the representations (i.e., their colors and positions) are altered—an example of the 'altered' type of state. Another possibility is that each time the cube is rotated, a representation is removed from the interface to indicate that one less rotation is available to the player—an example of the 'deleted' type of state. For instance, there could be a row of small cubes, and each rotation results in one of these cubes being removed. A third possibility is that representations of the number of rotations are added to the interface after the performance of each rotation—an example of the 'created' type of state. For example, there may be an empty grid in which a small copy of the cube is placed after each rotation to signify that an interaction has taken place.
- **Transition**: *stacked* or *distributed*. In the above example, transition is stacked—i.e., the cube rotates such that orientations are stacked on top of one another, and previous orientations are not displayed (see Figure 7). Alternatively, the transition could be distributed—e.g., the player types a command to rotate the cube three times to the right, and several representations appear to encode the intermediate orientations. All representations remain on the screen, so that the player can see the different orientations which the cube had while it was rotating (see Figure 10).

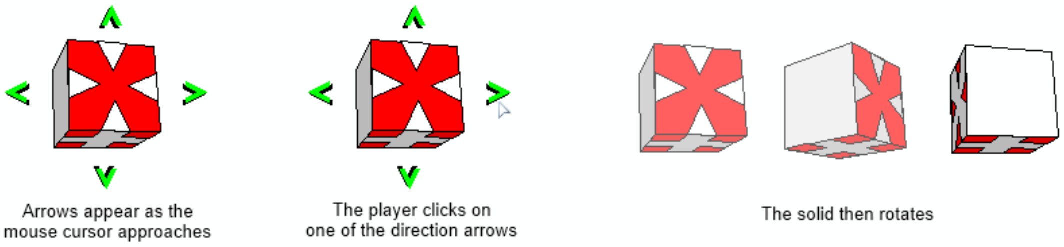
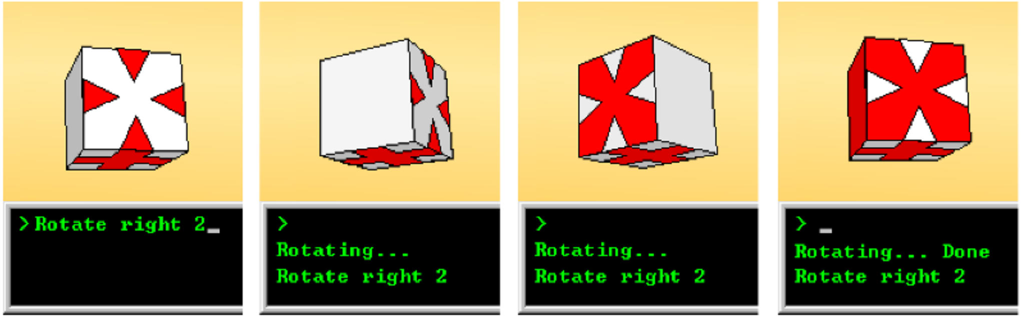


Figure 6. An example of rotating the cube when the type of agency is manual.



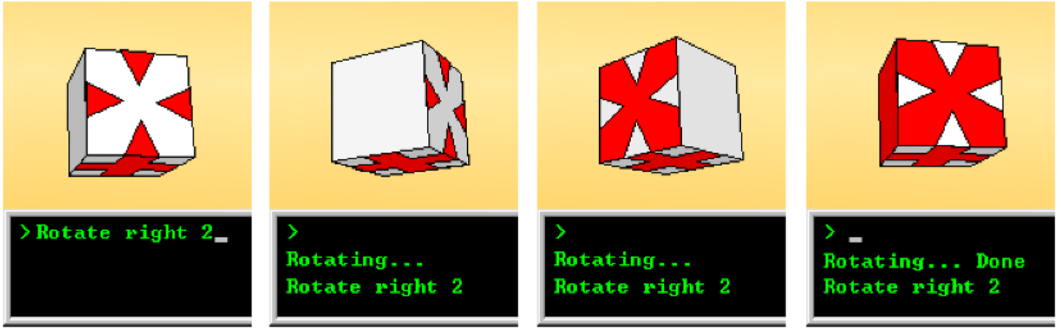
```
>Rotate right 2_
```

```
>
Rotating...
Rotate right 2
```

```
>
Rotating...
Rotate right 2
```

```
> _
Rotating... Done
Rotate right 2
```

Figure 7. An example of the cube rotating with the stacked type of transition.



```
>Rotate right 2_
```

```
>
Rotating...
Rotate right 2
```

```
>
Rotating...
Rotate right 2
```

```
> _
Rotating... Done
Rotate right 2
```

Figure 8. An example of rotating the cube when the type of agency is verbal.

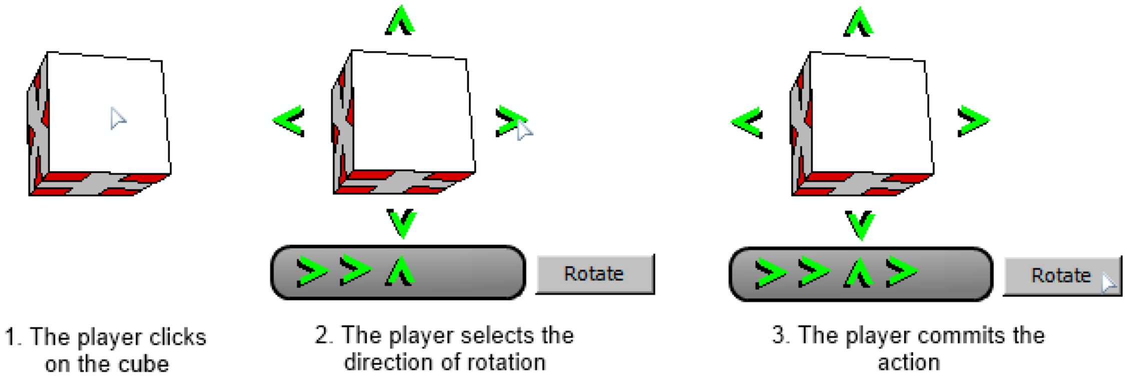
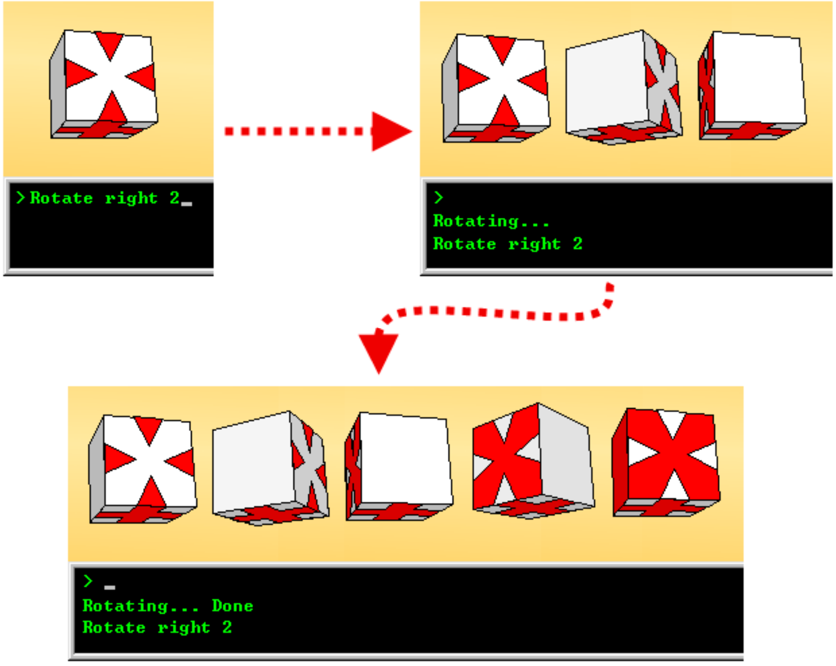


Figure 9. An example of multiple rotations of the cube with the on-demand type of activation.



```
>Rotate right 2_
```

```
>
Rotating...
Rotate right 2
```

```
> _
Rotating... Done
Rotate right 2
```

Figure 10. An example of the cube rotating with the distributed type of transition.

Although the above examples have focused on a single element, their interrelationships should also be considered. For example, the rotation interaction could be designed for a touch screen such that the player presses on the cube and, as she drags her finger along the screen, the cube gradually rotates. The types for the action elements of this interaction are agency–manual; flow–continuous; focus–direct; granularity–atomic; presence–implicit; timing–player-paced. The types for the reaction elements of this interaction are activation–immediate; context–unchanged; flow—continuous; spread–self-contained; state–altered; transition–stacked.

A researcher could take this interaction and try to change the type of one element to measure its effects on cognitive gameplay. For example, the new interaction could be that the player taps an arrow placed beside the cube and the cube rotates once in that direction—i.e., the player is no longer able to drag the cube. Since this new interaction uses a different type for the focus element (indirect), effects of its implementation could be studied. However, this new interaction also uses a different type for action flow (discrete) and presence (explicit), and the researcher must be cautious about which elements are being isolated. Without a framework, such as INFORM, it can be very difficult to think systematically about such issues.

5. Summary

Cognitive gameplay—the cognitive dimension of a player's experience—emerges from the interaction between a player and game. While its design requires careful consideration, cognitive gameplay can be designed only indirectly—via the design of game components such as mechanics, narrative, rules, information content, and graphics. In this paper, we focus specifically on the core mechanic—i.e., the essential interactions that bind a player and game together, the cyclical performance of which results in gameplay. With respect to the core mechanic and cognitive gameplay, extant research provides support for thinking about high-level pedagogical and philosophical interaction design issues. With respect to low-level interaction design issues, however, there is a significant gap in the literature. To help fill this gap, we have proposed INFORM (Interaction designN For the cORe Mechanic), a taxonomic framework that supports systematic thinking about micro-level interaction design and cognitive gameplay.

In the INFORM framework, we employ twelve previously established micro-level elements that constitute the structure of interaction, adapt their characterizations to suit the context of videogames, and extend discussions of their potential cognitive influences. We situate these elements within a broader framework that synthesizes concepts relevant to game design, such as cognitive gameplay and the core mechanic. INFORM is a descriptive framework and is not intended to give prescriptive design guidance. Rather, it provides a common vocabulary and a set of concepts that researchers and designers can use to think about and discuss issues related to micro-level interaction design and cognitive gameplay. Table 2 lists the six action and six reaction elements, their concerns, and some of their types.

Table 2. Summary of player–game Interaction Elements in the INFORM Framework.

	Element	Concern	Types
action	agency	metaphoric way through which action is expressed	verbal, manual
	flow	parsing of action in time	discrete, continuous
	focus	focal point of action	direct, indirect
	granularity	steps required to compose an action	atomic, composite
	presence	existence and advertisement of action	explicit, implicit
	timing	time available to player to compose and/or commit action	player-paced, game-paced
reaction	activation	commencement of reaction	immediate, delayed, on-demand
	context	context in which representations exist once reaction is complete	changed, unchanged
	flow	parsing of reaction in time	discrete, continuous
	spread	spread of effect that action causes	self-contained, propagated
	state	condition of representations once reaction process is complete	created, deleted, altered
	transition	presentation of change	stacked, distributed

We anticipate that INFORM can be used to facilitate increased clarity and rigor with respect to three main activities: (1) analysis of micro-level interaction elements within the core mechanic of existing games; (2) design of interactions within the core mechanic of new games; (3) evaluation of the effects of micro-level interaction design decisions on cognitive gameplay. By facilitating the analysis of interactions at a micro-level, designers can create different versions of games in which interactions within the core mechanic are isolated and varied, enabling the effects on cognitive gameplay to be empirically evaluated.

INFORM not only enables systematicity in design, but also helps to stimulate creativity in the design process. For instance, designers can vary the structure of the core mechanic for different situations within a game, different levels within a game, or even different versions of a game, each of which has different implications for cognitive gameplay. Although not all twelve elements are important in every situation, even if only half have a significant influence on cognitive processes the possible combinations for each interaction are 2^6, or 64. Without a taxonomic framework, such as INFORM, it is likely impossible to consider the many design possibilities systematically.

At this stage of the research, only a few elements and their variant types have been studied in the context of cognitive gameplay. Thus, while we know that these elements are important to investigate, further research is needed to better understand their effects. Another line of future research can be in integrating INFORM into a larger framework that considers other important aspects of game design, such as the emotional components of gameplay (e.g., motivation, immersion, flow), which can be examined in relation to the operational features of the core mechanic. It is our hope that INFORM will stimulate much research in these areas.

Acknowledgments: This research has been supported financially by the Natural Sciences and Engineering Research Council of Canada. We would also like to express our gratitude to the reviewers of this paper for their constructive and helpful comments. This paper would not be in its current shape without their insightful suggestions.

Author Contributions: K.S. and R.H. conceived the central topic; K.S. and P.P. elaborated the topic in terms of interaction and cognition; K.S. and R.H. elaborated the topic in terms of cognition and videogames; P.P. and R.H. searched for and designed examples; K.S., P.P., and R.H. wrote, edited, and repeatedly revised the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Barab, S.; Thomas, M.; Dodge, T.; Carteaux, R. Making learning fun: Quest Atlantis, a game without guns. *Educ. Technol. Res. Dev.* **2005**, *53*, 86–107. [CrossRef]
2. Gros, B. Digital games in education. *J. Res. Technol. Educ.* **2007**, *40*, 23–38. [CrossRef]
3. Ke, F.; Grabowski, B. Gameplaying for maths learning: Cooperative or not? *Br. J. Educ. Technol.* **2007**, *38*, 249–259. [CrossRef]
4. Linehan, C.; Lawson, S.; Doughty, M.; Kirman, B.; Haferkamp, N.; Krämer, N.C.; Schembri, M.; Nigrelli, M.L. Teaching group decision making skills to emergency managers via digital games. In *Media in the Ubiquitous Era: Ambient, Social and Gaming Media*; Lugmayr, A., Fanssila, H., Naranen, P., Sotamaa, O., Vanhala, J., Eds.; IGI Global: Hershey, PA, USA, 2012; Volume 3, pp. 111–129.
5. Liu, M.; Yuen, T.; Horton, L.; Lee, J.; Toprac, P.; Bogard, T. Designing technology-enriched cognitive tools to support young learners' problem solving. *Int. J. Cogn. Technol.* **2013**, *18*, 14–21.
6. Sedig, K. From play to thoughtful learning: A design strategy to engage children with mathematical representations. *J. Comput. Math. Sci. Teach.* **2008**, *27*, 65–101.
7. Sedig, K.; Haworth, R. Interaction design and cognitive gameplay: Role of activiation time. In *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '14), Toronto, ON, Canada, 19–22 October 2014*; ACM Press: New York, NY, USA, 2014; pp. 247–256.
8. Haworth, R.; Tagh Bostani, S.S.; Sedig, K. Visualizing decision trees in games to support children's analytic reasoning: Any negative effects on gameplay? *Int. J. Comput. Games Technol.* **2010**, *2010*, 1–11. [CrossRef]
9. Sedig, K.; Parsons, P. Interaction design for complex cognitive activities with visual representations: A pattern-based approach. *AIS Trans. Hum. Comput. Interact.* **2013**, *5*, 84–133.
10. Salen, K.; Zimmerman, E. *Rules of Play: Game Design Fundamentals*; MIT Press: Cambridge, MA, USA, 2004.
11. Bedwell, W.L.; Pavlas, D.; Heyne, K.; Lazzara, E.H.; Salas, E. Toward a taxonomy linking game attributes to learning: An empirical study. *Simul. Gaming* **2012**, *43*, 729–760. [CrossRef]
12. Sicart, M. Defining game mechanics. *Game Stud.* **2008**, *8*, 1–14.
13. Campbell, T.; Ngo, B.; Fogarty, J. Game design principles in everyday fitness applications. In *Proceedings of the ACM 2008 Conference on Computer Supported Cooperative Work (CSCW '08), San Diego, CA, USA, 8–12 November 2008*; ACM Press: New York, NY, USA, 2008; p. 249.
14. Hartson, R. Cognitive, physical, sensory, and functional affordances in interaction design. *Behav. Inf. Technol.* **2003**, *22*, 315–338. [CrossRef]
15. Kirsh, D. Interaction, external representation and sense making. In Proceedings of the 31st Annual Conference of the Cognitive Science Society, Amsterdam, The Netherlands, 29 July–1 August 2009; pp. 1103–1108.
16. Kiili, K. Digital game-based learning: Towards an experiential gaming model. *Internet High. Educ.* **2005**, *8*, 13–24. [CrossRef]
17. Annetta, L.A. The "I's" have it: A framework for serious educational game design. *Rev. Gen. Psychol.* **2010**, *14*, 105–112. [CrossRef]
18. Sedig, K.; Parsons, P.; Dittmer, M.; Haworth, R. Human-centered interactivity of visualization tools: Micro- and macro-level considerations. In *Handbook of Human-Centric Visualization*; Huang, W., Ed.; Springer: New York, NY, USA, 2014; pp. 717–743.

19. Sedig, K.; Klawe, M.; Westrom, M. Role of interface manipulation style and scaffolding on cognition and concept learning in learnware. *ACM Trans. Comput. Interact.* **2001**, *8*, 34–59. [CrossRef]
20. Costikyan, G. I have no words but i must design: Toward a critical vocabulary for games. In *Computer Games and Digital Cultures Conference Proceedings*; Tampere University Press: Tampere, Finland, 2002; pp. 9–33.
21. Crawford, C. *The Art of Computer Game Design*; Osborne/McGraw-Hill: Berkeley, CA, USA, 1984.
22. Suits, B. *The Grasshopper: Games, Life and Utopia*, 3rd ed.; Broadview Press: Peterborough, ON, USA, 2014.
23. Ang, C.S. Rules, gameplay, and narratives in video games. *Simul. Gaming* **2006**, *37*, 306–325. [CrossRef]
24. Ermi, L.; Mäyrä, F. Fundamental Components of the Gameplay Experience: Analysing Immersion. In Proceedings of the 2005 International Conference on Changings Views Worlds in Play, Vancouver, BC, Canada, 16–20 June 2005; pp. 15–27.
25. Poels, K.; de Kort, Y.; IJsselsteijn, W. "It is always a lot of fun!": Exploring dimensions of digital game experience using focus group methodology. In Processings of the 2007 Conference on Future Play, Toronto, Canada, 14–17 November 2007; pp. 83–89.
26. Callele, D.; Neufeld, E.; Schneider, K. A proposal for cognitive gameplay requirements. In Proceedings of the IEEE 2010 Fifth International Workshop on Requirements Engineering Visualization, Sydney, Australia, 28 September 2010; pp. 43–52.
27. Connolly, T.M.; Boyle, E.A.; Macarthur, E.; Hainey, T.; Boyle, J.M. A systematic literature review of empirical evidence on computer games and serious games. *Comput. Educ.* **2012**, *59*, 661–686. [CrossRef]
28. Cox, A.; Cairns, P.; Shah, P.; Carroll, M. Not doing but thinking. In *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems (CHI '12), Austin, TX, USA, 5–10 May 2012*; ACM Press: New York, NY, USA, 2012; p. 79.
29. Lee, K.M.; Peng, W. *What Do We Know About Social and Psychological Effects of Computer Games? A Comprehensive Review of the Current Literature*; Lawrence Erlbaum Associates Publishers: Mahwah, NJ, USA, 2006.
30. Spires, H.A. 21st century skills and serious games: Preparing the N generation. In *Serious Educational Games: From Theory to Practice*; Annetta, L.A., Ed.; Sense Publishers: Rotterdam, The Netherlands, 2008; pp. 13–23.
31. Ritterfeld, U.; Weber, R. *Video Games for Entertainment and Education*; Lawrence Erlbaum Associates Publishers: Mahwah, NJ, USA, 2006.
32. Moreno-Ger, P.; Martínez-Ortiz, I.; Sierra, J.L.; Fernández-Manjón, B. A content-centric development process model. *Computer* **2008**, *41*, 24–30. [CrossRef]
33. Dondlinger, M.J. Educational video game design: A review of the literature. *J. Appl. Educ. Technol.* **2007**, *4*, 21–31.
34. Fisch, S.M. Making educational computer games "educational". In *Proceeding of the 2005 Conference on Interaction Design and Children (IDC '05), Boulder, CO, USA, 8–10 June 2005*; ACM Press: New York, NY, USA, 2005; pp. 56–61.
35. Cox, R.; Brna, P. Supporting the use of external representations in problem solving: The need for flexible learning environments. *J. Artif. Intell. Educ.* **1995**, *6*, 239–302.
36. Larkin, J.; Simon, H. Why a diagram is (Sometimes) worth ten thousand words. *Cogn. Sci.* **1987**, *11*, 65–100. [CrossRef]
37. Zhang, J.; Norman, D. Representations in distributed cognitive tasks. *Cogn. Sci.* **1994**, *18*, 87–122. [CrossRef]
38. Cole, M.; Derry, J. We have met technology and it is us. In *Intelligence and Technology: The Impact of Tools on the Nature and Development of Human Abilities*; Sternberg, R.J., Preiss, D.D., Eds.; Lawrence Erlbaum Associates: Mahwah, NJ, USA, 2005.
39. Clark, A.; Chalmers, D. The extended mind. *Analysis* **1998**, *58*, 7–19. [CrossRef]
40. Hutchins, E. *Cognition in the Wild*; MIT Press: Cambridge, MA, USA, 1995.
41. Kirsh, D.; Maglio, P. On distinguishing epistemic from pragmatic action. *Cogn. Sci.* **1994**, *18*, 513–549. [CrossRef]
42. Sedig, K.; Sumner, M. Characterizing interaction with visual mathematical representations. *Int. J. Comput. Math. Learn.* **2006**, *11*, 1–55. [CrossRef]
43. Adams, E.; Dormans, J. *Game Mechanics; Advanced Game Design*; New Riders: Berkeley, CA, USA, 2012.
44. McGuire, M.; Jenkins, O.C. *Creating Games: Mechanics, Content, and Technology*; CRC Press: Boca Raton, FL, USA, 2008.
45. Adams, E. *Fundamentals of Game Design*, 3rd ed.; Pearson Education: London, UK, 2014.

46. Hunicke, R.; LeBlanc, M.; Zubek, R. MDA: A formal approach to game design and game research. In Proceedings of the AAAI Workshop on Challenges Game AI, San Jose, CA, USA, 25–26 July 2004.
47. Lundgren, S.; Björk, S. Game mechanics: Describing computer-augmented games in terms of interaction. In Proceedings of the First International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE 2003), Darmstadt, Hessen, Germany, 24–26 March 2003; Fraunhofer IRB Verlag: Stuttgart, Germany; pp. 45–56.
48. Iii, R.; Ogden, S. *Game Design: Theory & Practice*; Wordware Publishing: Plano, TX, USA, 2005.
49. Petkovi, M. *Famous Puzzles of Great Mathematicians*; American Mathematical Society: Providence, RI, USA, 2009.
50. Sniedovich, M. OR/MS games: 2. Towers of Hanoi. *INFORMS Trans. Educ.* **2002**, *3*, 45–62. [CrossRef]
51. Svendsen, G.B. The influence of interface style on problem solving. *Int. J. Man Mach. Stud.* **1991**, *35*, 379–397. [CrossRef]
52. Slocum, J. *The Tao of Tangram: History, Problems, Solutions*; Barnes & Noble: New York, NY, USA, 2007.
53. Norman, D. *The Design of Everyday Things: Revised and Expanded Edition*; Basic Books: New York, NY, USA, 2013.
54. Johnson, J. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*; Morgan Kaufmann: Burlington, MA, USA, 2014.
55. Van Harreveld, F.; Wagenmakers, E.J.; van der Maas, H.L.J. The effects of time pressure on chess skill: An investigation into fast and slow processes underlying expert performance. *Psychol. Res.* **2007**, *71*, 591–597. [CrossRef] [PubMed]
56. Eich, E. Context, memory, and integrated item/context imagery. *J. Exp. Psychol. Learn. Mem. Cogn.* **1985**, *11*, 764. [CrossRef]
57. Radvansky, G.A.; Krawietz, S.A.; Tamplin, A.K. Walking through doorways causes forgetting: Further explorations. *Q. J. Exp. Psychol.* **2011**, *64*, 1632–1645. [CrossRef] [PubMed]
58. Sahakyan, L.; Kelley, C.M. A contextual change account of the directed forgetting effect. *J. Exp. Psychol. Learn. Mem. Cogn.* **2002**, *28*, 1064–1072. [CrossRef] [PubMed]
59. Ainsworth, S. DeFT: A conceptual framework for considering learning with multiple representations. *Learn. Instr.* **2006**, *16*, 183–198. [CrossRef]
60. Kirsh, D. Thinking with external representations. *AI Soc.* **2010**, *25*, 441–454. [CrossRef]
61. Anderson, T.R.; Schönborn, K.J.; du Plessis, L.; Gupthar, A.S.; Hull, T.L. Identifying and Developing Students' Ability to Reason with Concepts and Representations in Biology. Series: Models and Modeling in Science Education. In *Multiple Representations in Biological Education*; Treagust, D.F., Tsui, C.Y., Eds.; Springer: Dordrecht, The Netherlands, 2013; Volume 7, pp. 19–38.
62. Scaife, M.; Rogers, Y. External cognition: How do graphical representations work? *Int. J. Hum. Comput. Stud.* **1996**, *45*, 185–213. [CrossRef]
63. Liang, H.-N.; Parsons, P.; Wu, H.-C.; Sedig, K. An exploratory study of interactivity in visualization tools: "Flow" of interaction. *J. Interact. Learn. Res.* **2010**, *21*, 5–45.
64. Tufte, E.R. *Visual Explanations: Images and Quantities, Evidence and Narrative*, 1st ed.; Graphics Press: Cheshire, CT, USA, 1997.
65. Sedig, K.; Rowhani, S.; Liang, H.-N. Designing interfaces that support formation of cognitive maps of transitional processes: An empirical study. *Interact. Comput.* **2005**, *17*, 419–452. [CrossRef]