Everything's a Ludeme Well, Almost Everything

Cameron Browne

Everything's a Ludeme
Well, Almost Everything

Cameron Browne
Associate Professor, Maastricht University

While the notion of the *ludeme* as a fundamental element of play appears to be widely accepted, its exact interpretation can vary with each individual practitioner. This paper explores the key properties of ludemes and existing models within which they are understood, and describes how a computational approach might help clarify what is and what is not a ludeme and ultimately lead to a more precise definition of this evasive term.

Keywords Ludeme; Ludii; Game Element; Game Design; Computational Model of Games

The term *ludeme* means different things to different people. While most current game researchers and designers appear to know the term, their interpretation and usage of it can vary significantly. The closest that we have to a definition is that proposed by David Parlett in his thoughtful 2006 paper, in which he observes that ludemes are:

> *… the conceptual elements of the game, most typically equivalent to its "rules" of play…*
>
> *… an element of play, comparable to, but distinct from, a game component or instrument of play…*
>
> *… ludemes only if they are contrastive*. (Parlett, 2006)

Parlett distinguishes between game components or instruments of play – such as the standard chequered chessboard (Figure 1, left) – and the conceptual elements or rules of play that constitute ludemes – such as the moves of a standard Chess knight (Figure 1, right). Importantly, he observes that ludemes must be *contrastive,* that is, changing a ludeme within a game should produce a change in its play. The chessboard is therefore not a ludeme by this definition as Chess can be played on any 8×8 square grid without affecting the game itself, whereas the rules dictating knight movement do constitute a ludeme as modifying them will have an observable effect on play.

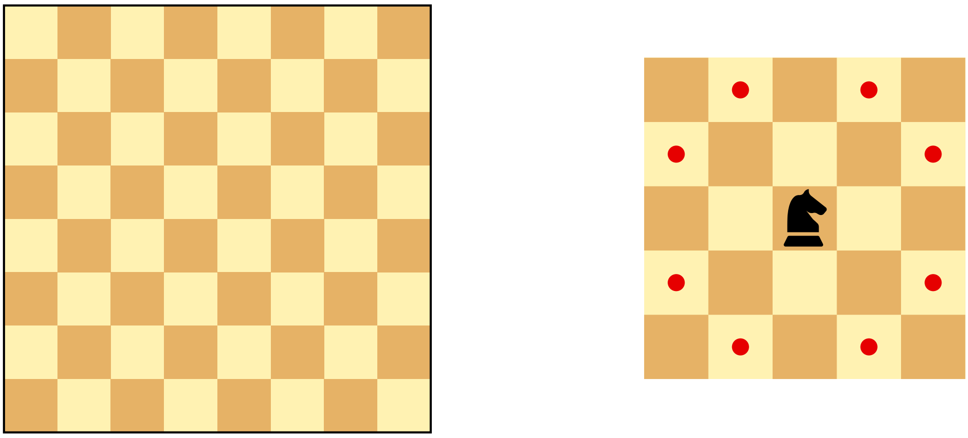

Figure 1. A standard chessboard (left) and the moves of a standard Chess knight (right).

This description provides an excellent starting point if not a precise definition. In my own role as an (amateur) game designer and (professional) computer programmer, I typically describe ludemes in rather nebulous terms as:

- *Units of game-related information*.
- *The "DNA" or building blocks of games*.
- *The conceptual units used by game designers to describe games*.

While these points summarise what I believe is the essence of ludemes, they do not take us any closer to a clear definition. This paper is an attempt to define the term *ludeme* more formally from a computational perspective, based on my experience over many years of modelling games as structures of ludemes in two significant software systems called Ludi and Ludii.[1] Figure 2 shows how these systems fit in the timeline of the term's usage.

Properties of Ludemes

It is useful at this point to define some commonly accepted properties of ludemes. The first three properties in the following list appear to be widely accepted by those who use the term, while the fourth property is contentious.

---

[1] The Ludii player is available for download at: http://ludii.games

1970	Pierre Berloquin uses *ludeme* in an interview in *Le Monde* (Depaulis, 2019). This is the first known use of the term.
1976	Richard Dawkins coins the term *meme* in *The Selfish Gene* (Dawkins, 1976).
1977	Alain Borvo uses *ludeme* in his book *L'aluette, ou le jeu de vache* (Borvo, 1977). This is the first known use of the term in print.
1990	David Parlett uses ludeme in *The Oxford Guide to Card Games* (Parlett, 1990).
2005–9	The Ludi software system introduces the *ludemic model* for game description (Browne, 2009).
2005	Video game designers reinvent the term *ludeme* (Cousins, Koster, 2005).
2006	David Parlett explores ludemes in detail in "What's a Ludeme?" (Parlett, 2006).
2018–	The Ludii software system further develops the ludemic model for game description (Browne, 2018).

Figure 2. Timeline of events in the development of the term *ludeme*.

1. Ludemes are Discrete

Each ludeme should be understandable as a single discrete unit of game-related information. For example, the moves of the standard Chess knight shown in Figure 1 are understandable as a single discrete concept, even though it encapsulates several other relevant concepts;

- the shape of the path,
- the adjacency of cells on the board,
- moves can begin in all four (orthogonal) directions,
- moves are reflected in each direction,
- the piece can move over other pieces,
- the piece can land on an enemy piece to capture it, and so on.

It is the combination of these concepts that forms the ludeme we might call "knight moves". While such combinations can be treated as a single unit, this does not imply that they cannot be decomposed into simpler sub-units, just as genes are considered units of hereditary information even though each one can be decomposed into large numbers of simpler DNA nucleotides or "*units within units*" (Dawkins, 1977, p. 195).

2. Ludemes are Transferable

A necessary property of ludemes is that they must be transferable. The essential concepts of a game can be readily described by one person to another – verbally, through example, through writing and illustration, and so on – a fact which underlies our modern understanding of the dispersal of games throughout human history (Murray, 1952) and their historical role as social lubricants between individuals and cultures (Crist, 2016).

An important benefit of this transferability is that ludemes can be readily substituted and recombined in new ways to facilitate the design of new games and variants. Ludemes can be easily transferred digitally both within games and between games, which is fundamental to the ludemic model of game design and the success of computational approaches such as evolutionary algorithms for automated game generation (Browne, 2009).

3. Ludemes are Contrastive

Ludemes must be contrastive such that changing a ludeme within a game changes the game itself, otherwise that item of information would not fulfil the ludeme's role as a relevant element of play. This raises the question: should changing a ludeme change the function of its game in *every* possible case, or is it sufficient to demonstrate a single plausible use case? it would make sense to take the latter (weaker) view; that a ludeme is deemed to be contrastive if it can be shown to make a functional difference to play in any plausible case.

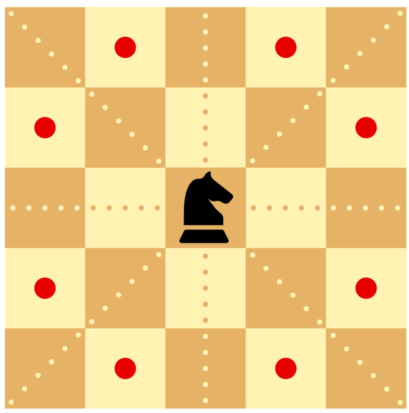

Figure 3. The standard Chess knight moves avoid orthogonal and diagonal lines.

For example, consider the knight move example once again. Note that the moves avoid the orthogonal and diagonal lines radiating from the pieces, which allows the relationship between the source and destination cells of a move to be described in several ways:

1. An "L" shaped walk (forwards, forwards, left/right, forwards) in any of the four

orthogonal directions.

2. Closest non-adjacent cell of a different colour.
3. Closest cell not in an orthogonal or diagonal line.
4. Closest cell of a different colour not in an orthogonal line.

These four descriptions satisfy the discrete and transferable properties of ludemes, but are they contrastive? Not if we consider them only in the context of the square grid, in which case all four rules produce the same eight move destinations (Figure 4, top left).

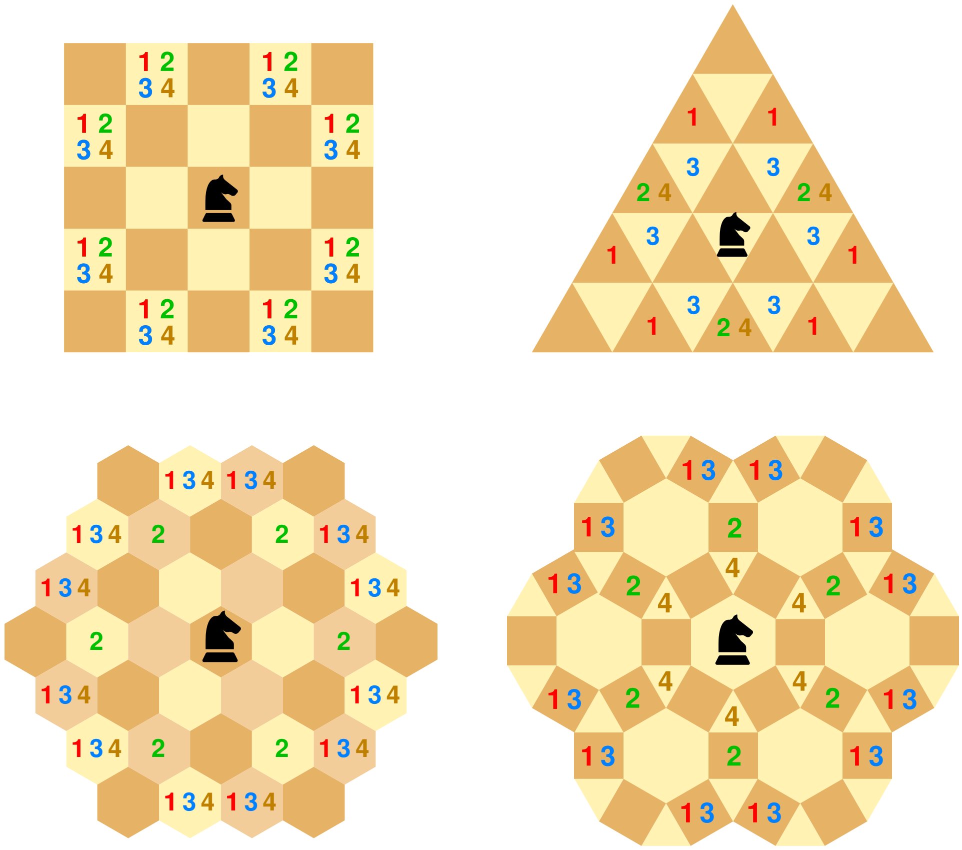

Figure 4. Knight move rules transposed to various grids: square, triangular, hexagonal and 3.4.6.4.

However, if we transpose these four rules to other grids, such as the triangular, hexagonal and semi-regular 3.4.6.4[2] grids as shown in the rest of Figure 4, then it can be seen that each of the four rules produces different subsets of moves on the various grids and would have significant impact on play. These four rules are contrastive and should be con-

[2] The semi-regular 3.4.6.4 or rhombitrihexahedral tiling as found on the Kensington board.

sidered distinct ludemes.

Note, conversely, that changing the geometry of the board can have a significant effect on the game, and so board geometry can itself be considered a ludeme. This is at odds with Parlett's description of ludemes as elements of play distinct from game components or instruments of play, but it would seem that a consistent definition of *ludeme* should include all factors that can affect the function of a game including the geometry of the board and components. The chessboard is not a ludeme because it is chequered – which has no functional effect on play – but because it is a square grid.

The knight move example shown above is an obvious example. We understand the notion of "knight move" as a single complete concept but it is made up of several more finely grained sub-concepts to do with adjacency, direction, rotation, reflection, translation, blockage, capture, and so on.

4. Ludemes can be Compound

While each ludeme is a discrete unit of information, it can still constitute a *compound* ludeme composed of a structure of simpler sub-concepts. This is the only contentious inclusion in this short list of properties – some believe that ludemes should be *atomic* units that cannot be decomposed any further – but I believe that their compound nature should be obvious with some reflection.

Three more examples further illustrate this point. Consider the following rules:

1. *Hop over any adjacent piece* (e.g. Chinese Checkers).
2. *Hop over any adjacent piece to flip it* (e.g. Tortuga).
3. *Hop forwards over any adjacent enemy piece to capture it* (e.g. Draughts).

Each of these rules is discrete, transferable and contrastive and constitutes a ludeme. But it is clear that there are multiple contrastive sub-elements within them occurring at different levels of granularity, as underlined below. Modifying any of these contrastive sub-elements will change the behaviour of that ludeme:

1. *Hop over any adjacent piece*.
2. *Hop over any adjacent piece <u>to **flip** it</u>*.
3. *Hop <u>forwards</u> over any adjacent <u>enemy</u> piece <u>to **capture** it</u>*.

Further, the "*<u>to **flip** it</u>"* and "*<u>to **capture** it</u>*" sub-elements both contain a further level of nested contrastive sub-sub-element as shown in bold. We can legitimately have ludemes within ludemes within ludemes. It is useful to distinguish between *atomic ludemes* which cannot be further subdivided and *compound ludemes* (or *ludemeplexes*) which constitute ludeme structures that can be subdivided into simpler sub-ludemes, if those sub-ludemes are embedded in a contrastive way.

Games as Ludemes

This raises the question: *Can an entire game be considered a ludeme?* To many, the answer is a clear "no", but I argue that the answer is a clear "yes". Many games constitute discrete, transferable, contrastive yet compound units of information.

For example, the traditional Greek game of Tavli consists of the three following sub-games (Portes, Plakoto and Fevga/Moultezim) played in cycle until a certain number of points is achieved. The same equipment is used for each sub-game, and only the rules change slightly between them, but each has its own distinct character that contrasts with the other two. Most readers will know Portes as the name "Backgammon", typically considered its own complete game, yet here used a transferable contrastive sub-component within a larger metagame.

There are many other examples of complete games as sub-games. One well known example is the use of the game Roshambo (i.e. Rock-Paper-Scissors) as a pre-game decider to make some necessary choice such as who moves first in the actual game to be played.

An important consideration on this point is whether the concept of the game can exist independently of its component sub-games. For example, consider the game Chameleon which is played identically to the classic game Hex – in which players place a piece of their colour each turn and win by connecting their board sides with a connected chain of their pieces – but in this case players can place a piece of either colour each turn and win by connecting their board sides with a chain of either colour (shown in Figure 5).

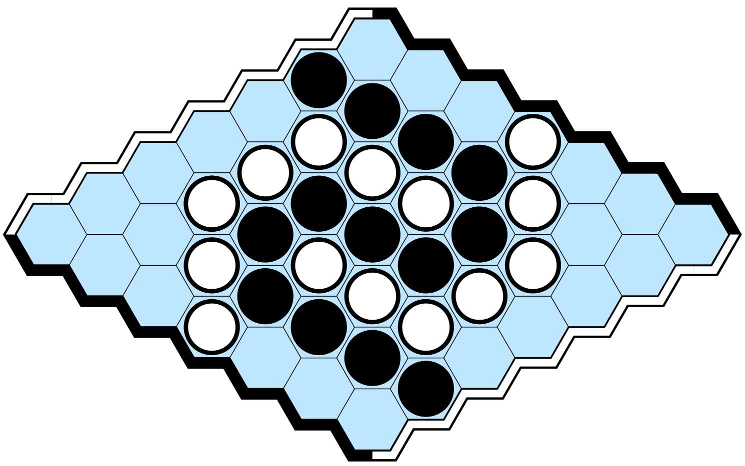

Figure 5. A game of Chameleon won by White, who has connected their sides with a chain of black pieces.

This simple change in the rules subverts the connective strategies of Hex brilliantly to produce an interesting game with its own character. It is certainly contrastive and the rule differences from Hex can be described as a discrete (if compound) unit. However, they cannot be sensibly transferred to another game *without also bringing the rules of Hex* with them as a defining framework; the "piece of either colour" and "chain of either colour" qualifications assume that the framing game in which they are embedded meaningfully defines these concepts, and these concepts exactly describe the fundamental elements of Hex.

The game of Chameleon should therefore not be considered as a complete ludeme in itself but as the product of Hex with the "Chameleon ludeme" applied. By contrast, the sub-games of Tavli are interchangeable with others – we could replace any of the sub-games with some other game that can be played with the same equipment – and the game could continue; the sub-games of Tavli are complete ludemes in themselves.

Existing Models of Ludemes

Before outlining the computational model of ludemes in detail, it is useful to consider the similarities and differences in what appear to be the dominant existing models of understanding ludemes.

1. Memetic Model

The notion of the *ludeme* resonates strongly with Richard Dawkins's notion of the *meme* as "*a unit of cultural transmission, or a unit of imitation*" (Dawkins, 1977, p. 192). He describes the following properties of memes which all apply to ludemes:

- *"unit of cultural transmission"*
- *"propagate... via... imitation"*
- *"can be sub-divided into components... separate memes"*

Ludemes are indeed often casually described as "game memes". For example, the game Chameleon described above was actually invented twice – totally independently – within the space of weeks in November 2003 by amateur board game designers Randy Cox in the USA and the late Bill Taylor in New Zealand. Different names were proposed for the two new games but their rules were exact in every detail despite their being invented without any knowledge of the other. One commentator at the time described this remarkable occurrence as the result of "game memes in the air".

It surprised me to learn that the invention of the term *ludeme* predated the invention of the term *meme* by several years (Depaulis, 2019). This demonstrates that *ludeme* did not just come about as the "game version" of the meme after the latter term's popular uptake, but suggests that the two terms capture some fundamental and meaningful mechanisms of the realms within which they apply.

2. Emic Model

Another model with which ludemes may be compared is the linguistic model of *emic units* that define "*an invariant form obtained from the reduction of a class of variant forms to a limited number of abstract units*" (Nöth, 1995). Such emic units as the *phoneme* (which describes the smallest unit of sound in speech) and the *morpheme* (which describes the smallest meaningful unit in a language) have been used for over a century. However, the more recent introduction of the *grapheme* (the smallest meaningful unit in a writing system) in 1986 has proven problematic.

Meletis points out that while the grapheme is a central concept in grapholinguistics, the lack of consensus on how it should be defined has split that research community and limited work in this area (Meletis, 2019). The *referential view* defines graphemes in terms of their equivalent phonemes while the *analogical view* defines graphemes as the smallest unit of writing that distinguishes meaning.

It is enticing to cast ludemes in the emic model so that the tools and theoretical frameworks developed for linguistics and related fields over the centuries could be applied. However, a common factor of all emic units – apart from the fact that they are transferable and contrastive – is that they refer to the smallest measurable unit in their domain that cannot be segmented any further, which is at odds with the important property of ludemes as possibly compound structures of simpler sub-ludemes. While finding similarities between ludemes and emic units is an interesting intellectual exercise and may yield some useful insights, I do not believe that attempting to cast ludemes as emic units would prove fruitful.

3. Video Game Model

The term *ludeme* emerged in 2005 in the video game design community apparently without knowledge of its prior use in the context of tabletop games over the preceding decades. Cousins (2005) described video games as hierarchies of *atoms* each defining the smallest loop of interaction between the player and the game, which Koster (2005) then equated with the term *ludeme*. Bojin summarises this "atomist" view as follows:

> "*Ludemes then are not the smallest measurable interactive 'unit' in a game, but are instead the smallest loop of engagement in a game that encompasses player choice*

within a game's context and its resulting feedback within that context." (Bojin, 2010).

This view resonates with the ideas discussed so far – the moves in a board game are indeed player choices that result in feedback within the game's context – but this definition seems deliberately open-ended for the broader context of video games. We instead want a more precise and concrete definition of what is and what is not a ludeme.

Ludemic Model of Games

The *ludemic model of games* is a computational method for describing games by their component elements, each of which is implemented in a corresponding piece of computer code (Browne, 2009). This model was first developed for the Ludi software system and implemented in the C++ programming language in 2005, then an improved version was developed for its successor the Ludii software system in 2018 in the java Programming language (Browne, 2018).

For example, the game Tic-Tac-Toe is described in the Ludii game description language as shown below. The game is decomposed into its essential parts – players, equipment and rules – each of which is further decomposed into simpler sub-parts as needed to form a nested structure. Each part and sub-part corresponds to a ludeme.

```
(game "Tic-Tac-Toe"
    (players 2)
    (equipment { (board (square 3)) (piece "Disc" P1) })
    (rules
        (play (move Add (to (sites Empty))))
        (end (if (is Line 3) (result Mover Win)))
    )
)
```

Game descriptions are composed of the following three types of elements assembled in a nested structure:

1. **Class Names** Denoted by bracketed lowercase keywords, e.g. `(players …)`.
   Each class name corresponds to a Java class in the Ludii code base, which will typically have multiple parameters within its brackets.

2. **Attributes** Denoted by uppercase keywords, e.g. `Line`.
   Each attribute describes a single constant value.
3. **Variables** Denoted by numbers, strings or true/false values, e.g. `5` or `“Disc”`.
   Each variable represents a single user-set value.

The Ludii language allows the user to “define” custom behaviours and give them meaningful names, such as the following “CaptureTo” expression which specifies that if the “to” location of a move is occupied by an enemy piece then that piece is captured by removal:

```
(define “CaptureTo"
    (to  if:(is Enemy (who at:(to)))
            (apply (remove (to))))
)
```

We can then define the movement rules for the standard queen, bishop and rook pieces of Chess quite simply as follows. Note that the capture behaviour of these pieces – with is common to all of them – is extracted out into the “CaptureTo” define:

```
(piece "Queen"  Each (move Slide            “CaptureTo"))
(piece "Bishop" Each (move Slide Diagonal   "CaptureTo"))
(piece "Rook"   Each (move Slide Orthogonal "CaptureTo"))
```

We can also extract out the common structure of these three piece rules by writing another define called “Slider” which takes the piece name and direction as parameters (the #1 and #2 symbols indicate where the arguments passed in are to be expanded), as follows:

```
(define "Slider"
    (piece #1 Each (move Slide #2 “CaptureTo”))
)
```

This allows the movement rules for these three piece types to be defined succinctly:

```
(“Slider" "Queen"  Adjacent)
("Slider" "Bishop" Diagonal)
("Slider" "Rook"   Orthogonal)
```

The “Queen” pieces slides along any Adjacent line and captures an enemy piece it lands on by replacement; The “Bishop” pieces slides along any Diagonal line and captures an enemy piece it lands on by replacement; and the “Rook” pieces slides along any Orthogonal line and captures an enemy piece it lands on by replacement, as expected. These abbreviated descriptions are simpler and easier to use, but more importantly they highlight *contrastive* aspects of the behaviour being modelled, by extracting out the non-contrastive aspects into separate defines. In this case, it is clear that the direction of travel is a contrastive element between these three piece types.

Such expressions within game descriptions constitute discrete units if they are *complete*, i.e. all necessary parameters are provided and each opening bracket has a corresponding closing bracket. The digital representation makes transferability of such expressions a non-issue; any expression in a game description can be replaced by any other if the defining grammar allows such a replacement. Such expressions satisfy all the properties of ludemes.

Implementation Details

The Ludii game descriptions are based on a grammar that is automatically generated from the Java classes in the Ludii code base using a *class grammar* approach (Browne, 2016). This formal grammar is an Extended Backus-Naur Form (EBNF)-style grammar consisting of a set of production rules, each corresponding to exactly one Java class implementation. For example, the following Java class in the Ludii library:

```
public class Is extends Ludeme
{
    public static BooleanFunction construct
    (
        final IsLineType isType,
        final IntFunction length
```

```
    )
    {
        // Java code
    }
}
```

generates the following production rule:

```
<is> ::= (is Line <int>)
```

which allows the following expression to occur within game descriptions:

```
(is Line 3)
```

Locating the Ludemes

So where are the ludemes in this model? Do they exist at the *semantic level* (in the Java code), at the *syntactic level* (in the grammar rules) or at the *symbolic level* (as symbols and expressions in resulting game descriptions)? We can make the following observations:

1. *Attributes constitute atomic ludemes*.
2. *Expressions and defines constitute compound ludemes*.

Both of these cases constitute discrete (but possibly compound) units of information that are transferable and contrastive within and between games. In other words, almost everything in a Ludii game description is a ludeme.

The only exceptions are the user-set variables in the form of numbers, strings and boolean true/false values. These are not ludemes in themselves as they are only meaningful within the context of the surrounding expression. For example, `(is Line 3)` and `(is Line 4)` are clearly contrastive ludemes but their component sub-elements `3` and `4` in are not (in isolation). And while it may be tempting to classify simple expressions such as `(is Line 3)` as atomic ludemes since they do not involve a nested structure, those with one or more parameter should still be considered compound ludemes as their function depends on the combination of concepts represented by their parameters with the ludeme class itself. Expressions should only be considered atomic if they have no parameters and no nested sub-ludemes, e.g. `(to)` in the examples above.

Similarly, the class name keywords in the grammar (e.g. `<is>` in the example above) each describe a valid ludeme type but are not in themselves ludemes, as they are not complete or contrastive units without the parameters that define how they are to behave.

Mathematical Model

These observations can be formalised as follows to give a mathematical interpretation of the computational ludemic model described above. Given the following sets:

R	The set of known `<rule>`s in the Ludii grammar.
A	The set of known `Attribute`s in the Ludii grammar.
D	The set of known game descriptions in the Ludii game language.
$d_a, d_b, \ldots$	The set of specific game descriptions (complete expressions).
L	The set of potential ludemes.

and the following relationships:

$l \in L$	Ludeme l is an element of the set of potential ludemes L.
$l \in R$	Ludeme l is a known class (i.e. left hand side of a `<rule>`) in R.
$l \in A$	Ludeme l is a known `Attribute` in A.
$(\exists d_n)\ [l \in d_n]$	Ludeme l is an expression in game description d_n.
$d_x \otimes l = d_y$	Applying ludeme l to game description d_x gives game description d_y.
$f(d_x)$	Function of play that defines the game described by d_x.
$f(d_x) \neq f(d_y)$	Games defined by descriptions d_x and d_y are functionally different.

this allows the complete definition as follows:

$$l \in L \mid \begin{array}{c} (l \in R \ \vee \ l \in A \ \vee (\exists d_n)[l \in d_n]) \\ \wedge \\ (\exists d_x)(\exists d_y)[(d_x \otimes l = d_y) \wedge (f(d_x) \neq f(d_y))] \end{array}$$

This rather cumbersome definition simply states that a given candidate l is a ludeme if and only if l is a class or an attribute in the grammar, or an expression in a known game description, and there exist two game descriptions whose only difference is l and the resulting games are functionally different.

What's Not a Ludeme?

As discussed above, a game-related concept should not be considered as a ludeme if it is dependent on some other additional information and cannot be meaningfully transferred to another game as a discrete unit without that additional information. Examples include user-set variables whose values have no meaning without the context of the surrounding expressions, or rules that are contingent on other aspects of the game and are meaningless or incomplete in isolation.

The Ludii grammar itself often helps distinguish what is and is not a ludeme as the instructions for describing games are clearly separated intoL 1) game logic, and 2) other metadata for describing help information about the game, hints for AI agents, graphics hints for improved visualisation, and so on. For example, when describing Chess, the 8×8 square board is defined in the game logic section but the chequered "Chess" style in which it it rendered is defined in the separate metadata section. The board has a functional impact on the game and is a ludeme, whereas the chequered style is a visual nicety for the (human) player with no functional bearing on the game and should not be considered a ludeme.

```
(game
    // ...
    (board (square 8))
    // ...
)
```

```
(metadata
    (graphics (board Style Chess))
)
```

It is also important to distinguish between ludemes – which are specific to games – and memes – which can apply to any cultural trait. For example, if I teach someone the complete rules of Chess in a single session then I am transmitting to that person the ludeme of "Chess" as a game. But Chess is often used in popular culture as a metaphor for intelligence or rational thought; some sports are described as "Chess at 100 miles an hour" to emphasise their strategic aspect, corporate logos often feature chessboards or pieces to convey the impression of good decision making, a chessboard is often used in avatars to indicate an interest in games, and so on.

A prime example of the use of game as metaphor is the scene in Bergman's classic *The Seventh Seal* (1957) in which The Knight, travelling through a landscape ravaged by the Black Plague, is met by Death and challenges him to a game of Chess in order to delay his impending doom. This is Chess as a battle of wits, as an attempt to impose rules on the inevitable forces of nature; this is Chess as a meme. The use of game as metaphor is taken to its ridiculous extreme in the less classic *Bill and Ted's Bogus Journey* (1991) in which Bill and Ted confound Death by challenging him to games of Battleship, Twister and Electric Football. These games exemplify other forms of uncertainty – luck, dexterity and physical coordination as opposed to the analytical uncertainty of Chess – but are again games as memes.

Conclusion

Putting this all together, we can provide an informal yet reasonably precise definition of the term *ludeme* as follows:

> *A ludeme is a discrete unit of information relevant to any game, which may be atomic or compound in nature, and which can be readily transferred between games to change the function of the game in at least one plausible case.*

This definition is based on my experiences as a game designer and game programmer, and discussions with many people over many years, and reflects my rather functional approach to understanding games. It resonates strongly with the notion of ludemes as "game memes" and agrees almost exactly with Parlett's 2006 description except for the observation that the equipment and components of a game should also be considered ludemes if they have a functional impact on play. This agreement is encouraging given that the conclusions expressed in this paper were derived from a very different (computational) route.

In conclusion, I would tend towards a rather liberal stance of what should be considered a *ludeme*. In a well-defined game, the relevant concepts should be described clearly and completely and every statement should have a purpose. In a well-defined game, almost everything's a ludeme.

Acknowledgements

This research is part of the European Research Council-funded *Digital Ludeme Project* (ERC Consolidator Grant #771292), led by Cameron Browne at Maastricht University over 2018–2023. Thanks to Cesco Reale for inspiring this paper, and Joao Pedro Neto, Stephen Tavener, William Taylor, Christian Freeling, Phil Bordelon, Russ Williams, Nestor Romeral Andres and many others including the members of the Ludii team (Eric Piette, Matthew Stephenson, Walter Crist and Dennis Soemers) for discussions that helped shape the ideas presented here.

References

BERLOQUIN Pierre (1970) Les Echechs, le Go, l'Ecriture, *Le Monde*, 30 October 1970.

BOJIN Nis (2010) Ludemes and the Linguistic Turn, in *Futureplay '10: Proceedings of the International Academic Conference on the Future of Game Design and Technology, Vancouver*, pp. 25–32.

BORVO Alain (1977) *Anatomie D'un Jeu de Cartes: L'Aluette ou le Jeu de Vache*, Librarie Nantaise Yves Vachon, Nantes.

BROWNE Cameron (2009) *Automatic Generation and Evaluation of Recombination Games*, PhD Thesis, Queensland University of Technology, Brisbane.

BROWNE Cameron (2016) A Class Grammar for General Games, *Advances in Computer Games (ACG 2016)*, Leiden, Springer, LNCS vol. 10068, pp. 167–182.

BROWNE Cameron (2018) The Digital Ludeme Project, *Board Game Studies Colloquium (BGS 2018)*, Athens (presentation).

COUSINS Ben (2005) Low-Level Game Design: Atoms, Measurement and Hierarchies, *Game Developers Conference Europe (GDC Europe)*, London.

CRIST Walter, DE VOOGT Alexalter and DUNN-VATURI Anne-Elizabeth (2016) Facilitating Interaction: Board Games as Social Lubricants in the Ancient Near East, *Oxford Journal of Archaeology*, vol. 35, no. 2, pp. 179–96.

DAWKINS Dawkins Richard (1976) *The Selfish Gene*, Oxford, Oxford University Press, pp.189–201.

DEPAULIS Thierry (2019) On the Origins of the Word 'Ludeme' (French *Ludème*), in Browne *et al*. *Foundations of Digital Archaeoludology*, Report, Schloss Dagstuhl Research Meeting, Saarbrucken, pp. 23–26.

KOSTER Raph (2005) A Grammar of Gameplay (Game Atoms: Can Games be Diagrammed?), *Game Developers Conference (GDC '05),* San Francisco.

KOSTER Raph (2005) *A Theory of Fun*, New York, Paraglyph Press.

MELETIS Dimitrios (2019) The Grapheme as a Universal Basic Unit of Writing, *Writing Systems Research*, vol. 11, no. 1, pp. 26–49.

MURRAY Harold J. R. (1952) *A History of Board Games other than Chess*, Oxford, Clarendon Press.

NÖTH Winfried (1995) *Handbook of Semiotics*. Indiana University Press, Indiana.

PARLETT David (1990) *The Oxford Guide to Card Games*, Oxford, Oxford University Press.

PARLETT David (2006) What's a Ludeme?, *Board Game Studies Colloquium (BGS 2016)*, Ouro Preto (presentation).