Measuring Orthogonal Mechanics in Linguistic Annotation Games

FEDERICO BONETTI, University of Trento, Italy and Fondazione Bruno Kessler, Italy
SARA TONELLI, Fondazione Bruno Kessler, Italy

Gamification has been recently growing in popularity among researchers investigating Information and Communication Technologies. Scholars have been trying to take advantage of this approach in the field of natural language processing (NLP), developing Games With A Purpose (GWAPs) for corpus annotation that have obtained encouraging results both in annotation quality and overall cost. However, GWAPs implement gamification in different ways and to different degrees. We propose a new framework to investigate the mechanics employed in the gamification process and their magnitude in terms of complexity. This framework is based on an analysis of some of the most important contributions in the field of NLP-related gamified applications and GWAP theory. Its primary purpose is to provide a first step towards classifying mechanics that mimic mainstream video games and may require skills that are not relevant to the annotation task, defined as orthogonal mechanics. In order to test our framework, we develop and evaluate Spacewords, a linguistic space game for synonymy annotation.

CCS Concepts: • **Applied computing** → *Computer games*; • **Human-centered computing** → **Human computer interaction (HCI)**.

Additional Key Words and Phrases: games with a purpose, game mechanics, orthogonal mechanics, disjoint design, linguistic annotation

ACM Reference Format:
Federico Bonetti and Sara Tonelli. 2021. Measuring Orthogonal Mechanics in Linguistic Annotation Games. *Proc. ACM Hum.-Comput. Interact.* 5, CHI PLAY, Article 265 (September 2021), 16 pages. https://doi.org/10.1145/3474692

1 INTRODUCTION

Since recent advances in natural language processing (NLP) have benefited from deep learning techniques, which usually require large corpora for training, linguistic annotation of large amounts of data is of paramount importance not only for comprehensive linguistic studies but also to ensure good performance of such classification models [26]. However, expert annotation is expensive both in terms of time and of cost, and crowdsourcing through platforms such as Amazon Mechanical Turk does not always guarantee high-quality annotation. Therefore, games with a purpose (GWAPs) [34] have become more and more used for linguistic annotation, obtaining encouraging results both in terms of annotation quality and overall cost [24, 32]. Some examples of NLP-related tasks that have been dealt with using GWAPs are image labelling [35], coreference resolution [24], sequence labelling [21], image-sense linking [15] and identification of fallacious argumentation [8], among others. An important limitation of the above approaches is that there is no general theory on

Authors' addresses: Federico Bonetti, federico.bonetti@unitn.it, University of Trento, Trento, Italy, P.O. Box 38122 , Fondazione Bruno Kessler, Trento, Italy; Sara Tonelli, satonelli@fbk.eu, Fondazione Bruno Kessler, Trento, Italy, P.O. Box 38123.

2573-0142/2021/9-ART265
https://doi.org/10.1145/3474692

how to create a successful gamified application for linguistic annotation, although many different approaches exist. We distinguish between approaches that try to isomorph the annotation task to some game mechanics and approaches that use mini-games as motivational phases between task sessions. The former is further divided by Prestopnik and Crowston [25] into *task gamification* and *game taskification*. The latter was introduced by Kicikoglu et al. [17] and is referred to as *annotation-motivation paradigm*. Our focus is on task gamification. With the present work we aim to 1) systematize the existing theory about design concepts and strategies pertaining to human computation games for linguistic annotation and 2) propose a novel approach to evaluate the complexity of their mechanics. Being able to rank gamified applications according to the separation between the essential task and the actual interface is of crucial importance in order to strike a balance between enjoyment, accessibility and task accuracy. Knowing which mechanics pose a threat to which annotation actions would be useful to researchers and practitioners when designing games that draw from an established reservoir of widespread game mechanics, such as aiming, driving, jumping and so on.

Following the Mechanics Dynamics Aesthetics (MDA) theory [12], we can study and understand games according to three levels: *mechanics*, *dynamics* and *aesthetics*. While the developer's work is mainly involved in the mechanics component, the user is closer to the aesthetics component, which determines the emotional perception of a user with respect to a given game. Between is the dynamics component, which determines how the mechanics work and interact during game play. This breakdown is useful to bridge the gap, as the authors claim, between a gaming artifact and development or research.

In this paper we build upon the mechanics component, which contains the behaviors that a programmer has designed and that can be potentially translated into the dynamics component during game play. We define mechanics as collections of parameters and rules that establish relations between those parameters, similarly to [28]. With a deep understanding of the mechanics of a given GWAP, we should be able to give an account of what has been added to a task in terms of features and complexity. Along this line, one major contribution of this work lies in the presentation of a novel metric that captures the degree of gamification that has been applied to the mechanics of a certain task and that may facilitate comparisons between different gamified applications. To calculate this score, a good understanding of the building blocks of linguistic annotation tasks and linguistic annotation GWAPs is needed. We integrate Siu et al. [29]'s framework with *disjoint design* by Krause et al. [18] and the concept of *orthogonal mechanics* by Tuite [31] in order to have a more detailed representation of the mechanics underlying gamified applications for linguistic annotation.

Our goal is to understand the distance in complexity and outcome uncertainty between a standard linguistic annotation task, which can be performed with a traditional annotation tool such as BRAT [30] or CAT [1], and a gamified application that employs orthogonal game mechanics. In this way we are able to compute an *orthogonality score*, which may facilitate comparisons between different gamified applications.

The article is structured as follows: first, in Section 2, we give an outline of the most important works in the field of GWAPs for linguistic annotation, accompanied by three existing theoretical contributions. In Section 3 a first categorization of interfaces for linguistic annotation is provided based on their gamification type. In Section 4 we give a formal definition of orthogonality and propose thresholds to determine its intrusiveness. In Section 5, we evaluate a game where orthogonality components are manipulated in order to investigate their impact on annotation quality. Finally, in Section 6 we discuss the main findings and limitations of this work.

2 BACKGROUND

In this section, we describe briefly the contributions that were most relevant to develop our framework, dividing them into existing GWAPs for linguistic annotation (Section 2.1) and existing frameworks to understand human computation game components (Section 2.2).

2.1 Applications

To date, there have been several attempts to gamify a wide range of linguistic annotation tasks (see Table 1). These include, among others, *Phrase Detectives* [24], *Wormingo* [17] and *PlayCoref* [11] for anaphoric annotation and coreference chain detection, *The Knowledge Towers* [32] and *Puzzle Racer* [15] for concept-image linking, *Infection* [32], *OnToGalaxy* [18] and *JeuxDeMots* [14] for semantic linking, *Argotario* [8] for fallacious argumentation identification, *Zombilingo* [7] for dependecy syntax annotation, *Sentimentator* [23] for sentiment annotation, *WordClicker* [21] for part-of-speech tagging, *RoboCorp* [6] for named entity recognition, *Dr. Detective* [5] for medical knowledge extraction, *Wordrobe* [33], *Ambiguss* [19] and *Ka-Boom!* [15] for word sense disambiguation, *High School Superhero* [2, 3] for abusive language annotation. Researchers stress the fact that GWAPs should be designed in such a way that they integrate the task without sacrificing their 'gamefulness', otherwise the tasks may be perceived as work [32]. Some of these games try to exploit *disjoint design* [18], i.e. a technique by which the goal of the user and the goal of the task are kept separate. The mechanics that allow for this type of design to take place on an interactional level are called *orthogonal game mechanics* [31]. For instance, in *OnToGalaxy* users control a spaceship and have to shoot other spaceships with a certain label that does not satisfy the condition expressed in the instructions. Since this separation, or orthogonality, could potentially harm the quality of the outcome, tasks have to be thought very carefully. Indeed, a goal that is phrased as 'shoot the spaceships with a name that does not satisfy this condition' may very well drive the user's actions differently than a task that says 'click on the label that satisfies the following condition', if only because of the sense of challenge or excitement that arises. On the other hand, challenge and a gameful environment might be exactly what drives the users' actions in the right direction, to the point of improving the annotation quality over standard crowdsourcing methods [32]. This separation is useful for hiding the task and making the whole experience feel less like work and more like play. However, hiding a task does not necessarily mean that the users must not be made aware of its presence. In fact, saying clearly that a game is useful for research purposes can be a motivator for users [31].

In Table 1 we summarise the most widely-used games developed for linguistic annotation, specifying which ones rely on orthogonal game mechanics.

2.2 Theoretical Background

2.2.1 Human computation game mechanics. Siu et al. [29] are the first, to our knowledge, to provide a formal representation of the fundamental mechanics of human computation games, or games with a purpose. They single out four basic components in the game play loop: a) *player*, b) *action mechanics*, c) *verification mechanics*, d) *feedback mechanics*.

The *action mechanics* regard everything the players do when they interact with the artifact to perform a task which, in our case, would be the linguistic annotation of text. The *verification mechanics* take the input and try to match it against a gold standard that, for linguistic annotation, is usually a set of manually labelled sentences. These mechanics are of primary importance since they make sure the users are not submitting low-quality data. The *feedback mechanics* close the gameplay loop by providing feedback to the user. In-game feedback is calculated by taking into

Table 1. Linguistic annotation GWAPs and orthogonal mechanics

GWAP	Mechanics	Linguistic task
Phrase Detectives [24]	Non-orthogonal	Anaphoric annotation
Wormingo [17]	Non-orthogonal	Anaphoric annotation
PlayCoref [11]	Non-orthogonal	Coreference chain detection
Zombilingo [7]	Non-orthogonal	Dependency syntax annotation
Sentimentator [23]	Non-orthogonal	Sentiment annotation
Argotario [8]	Non-orthogonal	Fallacious argumentation identification
Wordrobe [33]	Non-orthogonal	Word sense disambiguation
Ambiguss [19]	Non-orthogonal	Word sense disambiguation
JeuxDeMots [14]	Non-orthogonal	Semantic linking
WordClicker [21]	Non-orthogonal	Part-of-speech tagging
RoboCorp [6]	Non-orthogonal	Named entity recognition
Dr. Detective [5]	Non-orthogonal	Medical knowledge extraction
OnToGalaxy [18]	Orthogonal	Semantic linking
Infection [32]	Orthogonal	Semantic linking
Ka-Boom! [15]	Orthogonal	Word sense disambiguation
Puzzle Racer [15]	Orthogonal	Sense-image linking
The Knowledge Towers [32]	Orthogonal	Sense-image linking
High School Superhero [2, 3]	Orthogonal	Abusive language annotation

account how the *verification mechanics* judged the performance of the players. In this article, we will use the word *feedback* to refer to both in-game and external motivators.

2.2.2 Disjoint design. Krause et al. [18] introduce new terminology to refer to a technique by which a task can be transformed into another, while preserving the same low-level interaction structure and roughly the same outcome. In this way, the user can carry out a task that is phrased and realized as game mechanics rather than one that is phrased and realized as normal software-mediated work. In particular, the work proposed by the authors is a space-shooter game based on semantic linking as its underlying task. The player is given a concept and has to select (save) spaceship, whose label is relevant to the concept, and discard (shoot) the others. While this is an interesting strategy, as it ideally allows one to develop complex and entertaining GWAPs, it has not been formalized yet. Disjoint design may also include aesthetic embellishments and background stories that justifiy or contextualize the task. When disjoint design integrates the game mechanics and the task seamlessly, we speak of intrinsic integration, a design pattern that has been analysed by Habgood et al. [9] in the context of educational games. They developed Zombie Division, a game where the meaning of the game mechanics (dividing zombies that correspond to numbers with specific 'divisor' weapons) is intrinsically integrated with the meaning of the task (dividing numbers to learn division). Although this is an extremely interesting design strategy, it may be difficult for designers to find a game that can be intrinsically integrated with a specific and/or complex lingusitic annotation task and may resort to generic games with orthogonal mechanics instead.

2.2.3 Orthogonal game mechanics. Tuite [31] and Sarkar & Cooper [27] define orthogonal game mechanics as those mechanics that require from the player abilities that do not pertain strictly to the underlying task and do not serve the purpose. This type of mechanics is functional to disjoint design, explained in 2.2.2, and provides a way to look at it from a more practical and objective

way. Games with a purpose should in principle take advantage of game mechanics that are not orthogonal, in order to maximize the task performance, as noted in Madge et al. [20]. In other words, a direct overlap between task mechanics and gameplay may be preferred, while requiring to aim, jump, or carry out other game-specific actions, can compromise the task accuracy. However, since orthogonal game mechanics can sometimes pose an interesting challenge for players, in that they can potentially make the game more playful and interesting, in this article we investigate and formalize its nature by introducing a measure for orthogonality that we call *orthogonality score*.

2.3 Terminology

In the remainder of this article we will make use of the following terminology:

- *Mechanics*: a collection of parameters and rules that establish relations between the parameters (borrowed from [28]'s game mechanics); A component of the MDA framework (mechanics, dynamics, aesthetics) by [12], directly manipulated by the designer.
- *Action mechanics* [29]: the mechanics make it possible for the user to interact with the program and carry out the annotation task. For example selecting, writing, but also jumping, shooting, and so on.
- *Verification mechanics* [29]: the mechanics that ensure users provide high-quality data, for example agreement between players and specific validation mechanics such as evaluating annotations made by other users.
- *Feedback mechanics* [29]: the mechanics that manage in-game rewards, either numerical (such as scores) or cosmetic (such as collectibles), given to the user.
- *Disjoint design* [18]: design strategy that separates the goal of the user from the goal of the application.
- *Orthogonal game mechanics* [31]: the game mechanics that require additional abilities to those already required by the task.

3 ANALYSIS OF GWAP MECHANICS

In this section we present a breakdown of mechanics both in gamified and non-gamified linguistic annotation programs. In so doing we will be able to understand exactly which components of a program realize orthogonality and which ones are affected by it. We propose to divide linguistic annotation strategies into three categories, based on the decoupling from the fundamental task mechanics.

Within Type 0, which is a typical example of common crowdsourcing interfaces like Amazon Mechanical Turk[1], or standard linguistic annotation tools, there are no gameful parameters. What the user performs, mostly through a simple application that can be online or locally installed, is a range of limited actions based on the task of adding one or more information layers to a given text. The string to be annotated can range from a long span to a single token or even sub-token (for example in the annotation of morphemes, or affixes). Within this type, the user interacts with the *task mechanics*, such as selecting a text string and assigning a pre-defined label from a list, or adding information as free text. For example, annotations for named entity recognition tasks may be collected by asking the user to select a span denoting a name and assign a label among Organization, Person, Location, and so on. A more complex task of paraphrase generation would require annotators to select a clause or sentence, and rewrite the same text using a different wording. In this context, we define the task mechanics as the collection of parameters and rules that determine the logic of the simplest possible implementation of a given task.

[1] https://www.mturk.com/

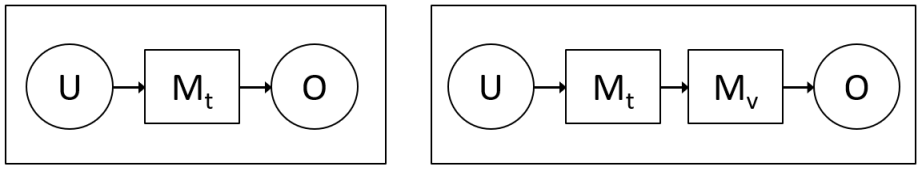

Fig. 1. Type 0. The user U interacts with the task mechanics M_t (with the possibility of verifying the output through the verification mechanics M_v, on the right) in order to accomplish a task objective O (the required annotation output).

The representation of this process is reported in Figure 1 (left) and can be summarised as follows: the user interacts with some *task mechanics* which serve the *task objective* (the annotation output). The feedback in this case is outside the loop, since financial compensation does not intrinsically depend on how the user is interacting with the program elements. It is worth noting that with feedback, which we mention in the next schemes, we refer to those game elements that foster satisfaction and not, for example, physical rewards like financial compensation. A variant where the verification mechanics are used is also possible (Fig. 1 right). That would be a task that verifies the data against some form of gold or silver standard.

Within Type 1 (Figure 2), added mechanics may be present (such as scores, leaderboards, cosmetic elements, and rewards) but they do not interfere with the range of actions required to carry out a task or their modality (although they potentially increase or decrease the user's commitment). In this this type, the user interacts with the task mechanics but receives immediate or delayed feedback thanks to the feedback mechanics. By feedback we mean any game element or event that rewards the player inside the game world. Common instances of feedback are badges, achievements, scores; additional feedback is represented by so-called *juicy feedback* [10] and is composed of sounds and graphic components that foster enjoyment and satisfaction. In addition to these types of feedback, cognitive feedback [4] may also be present which signals players how well they are doing with praise ('well done!') or negative feedback ('woops!'). This is done, for example, in Phrase Detectives.

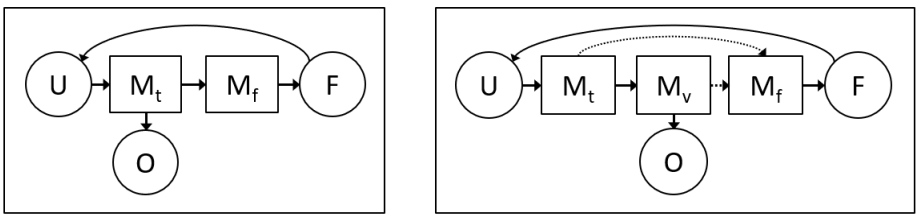

Fig. 2. Type 1. The user U interacts with the task mechanics M_t (i.e. moving the mouse and clicking on buttons). The input can be processed by the verification mechanics M_v, which ensure the data provided by the user is acceptable, for example via inter-annotator agreement scores. The verification mechanics can be absent (left) if rewarding the player in real time on the base of annotation accuracy is not a requirement. The feedback mechanics M_f calculate and show the feedback F accordingly. Dashed elements are optional. For example, verification mechanics can input to the task objective alone, in which case the task mechanics would be directly linked to feedback.

An example of linguistic annotation programs belonging to Type 1 is Phrase Detectives [24], a game that aims at collecting annotations of anaphoric information, for example which textual element a pronoun refers to. In the game, users are both annotators and validators, which means they

can both annotate a coreference occurrence and validate other users' annotations. Obtaining reliable results from non-experts is no easy task but the strategy of combining annotation with validation seems to be effective in this case. The gameplay unfolds as follows: there are 2 phases, 'Name-the-Culprit' and 'Detective Conference'. The first one corresponds to the annotation task (the user must indicate whether the highlighted word, usually a pronoun, has a reference in the text); the second one corresponds to the validation task. In total, by 2012, 407 documents were fully annotated. In this game feedback is added as an incentive to good performance in the annotation task, as displayed in Figure 2. In this type of applications, the *feedback mechanics* are responsible for providing the user with score tables, cosmetic rewards (such as graphical avatar enhancements), badges, achievements, and the like. All of these elements, which have meaning only inside the GWAP and not in the real world, fall under the category of feedback. In Phrase Detectives, the task mechanics consist of providing annotations by clicking the tokens that constitute a coreference chain; the *verification mechanics* ensure the input provided is adequate by submitting the annotations to other users to check for agreement; the feedback mechanics calculate the score of a user accordingly. Finally, feedback is given in the form of score. These games belong to the Type 1 variant that implements verification mechanics (Fig. 2 right). A variant where the verification mechanics are not used could in principle be developed. So long as the feedback mechanics are still there, however, the application would still fall under this category.

Within Type 2, orthogonal mechanics are used that modify the essential actions required to perform the task, either by manipulating their underlying parameters and rules or by adding new actions. Within this type, the user interacts with the *action mechanics* component, and we define this component as the sum of the task mechanics and the orthogonal mechanics. We conceive orthogonal mechanics as the collection of parameters and rules that have been added to the existing task mechanics during the gamification process. Then, differently from the definition given in [29], introduced in Section 4.1, we conceive the action mechanics as the union of the task mechanics and the orthogonal mechanics.

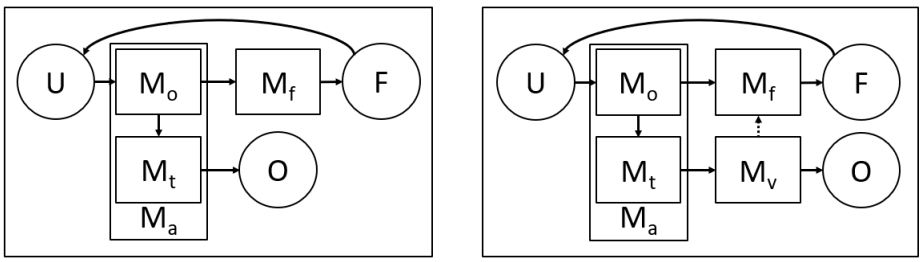

Fig. 3. Type 2. The user U interacts with the orthogonal mechanics M_o, which output to two mechanic sets: the task mechanics M_t, by manipulating the fundamental parameters; and the feedback mechanics M_f, which now show animations and sounds, in addition to scores. The feedback mechanics also receive data from the verification mechanics M_v in order to reward good annotators with higher scores. Behavior and task mechanics are both part of the action mechanics M_a. Similarly to the previous examples, verification mechanics can be absent (left). Verification mechanics can optionally avoid retrieving information to the feedback mechanics and only provide filters for a better output.

Within this type, the actions of the user always produce an enhanced perceptible feedback, which ultimately contains the essence of a video game environment and could be called *juiciness* [10, 16]. The feedback mechanics now take two inputs: one from the orthogonal mechanics and one from the verification mechanics. An instance of this type of gamification is OnToGalaxy.

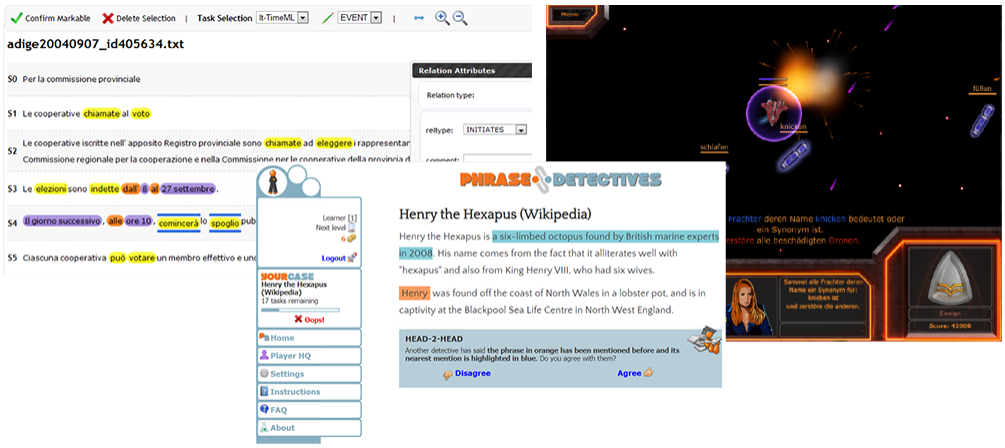

Fig. 4. A screenshot from CAT (Type 0, [1]); Phrase Detectives (Type 1, [24]); OnToGalaxy (Type 2, [18]). CAT is a traditional annotation tool. Phrase Detectives is a gamified interface for anaphora resolution with motivators such as points and a narrative. OnToGalaxy is a space shooter game for semantic linking where orthogonal mechanics are used.

In this game, the orthogonal mechanics consist of controlling a spaceship and shooting hostile entities while rescuing correctly labeled spaceships. These mechanics, accessible to the player at the interaction level, translate to the fundamental task mechanics, which consist of selecting appropriate entries (i.e. words) for a given sense. Specifically, users must select the spaceships whose label is semantically adequate based on the instructions (for example, selecting touchable objects) to populate a semantic network. They do so by destroying the spaceships that do not carry an adequate label. The feedback mechanics reward the player both with scores and graphical/audio effects; the verification mechanics make sure the user can be trusted by assigning a trust score. This score is based on test relationships where the correct answer is already known [18].

A summary of the three different types is reported in Table 2.

Table 2. Overview of the categories of annotation platforms based on presence and type of gamification strategies.

Type	Gamified	Artifact	User role	Orthogonal mechanics
0	No	Crowdsourcing interface	Worker	No
1	Yes	Phrase Detectives	Player	No
2	Yes	OnToGalaxy	Player	Yes

4 ORTHOGONALITY SCORE

Our approach assumes that the more orthogonality there is, the farther away the application will be from the fundamental annotation interaction (i.e. simply selecting or writing text labels). As [13] put it, “the different isomorphic representations of a problem affect the complexity of the task and the behavioral outcomes". Our assumption is that an orthogonal program is the isomorph of a fundamental task. Understanding the interaction decoupling between the two is useful to rank the application according to the amount of uncertainty it puts between regular interaction and

gamified one. Furthermore, it could be useful in research to determine whether the gamification process of a certain task is to be held responsible for accuracy improvement or worsening. In our view, orthogonality measures neither good design nor enjoyability. On the one hand, adding orthogonality allows the creation of more complex and unpredictable experiences, which may foster more enjoyment; on the other hand, creating entertaining programs with the lowest complexity possible could be beneficial to the annotation task.

4.1 Orthogonal Mechanics and Orthogonality

From our analysis of the literature, orthogonality seems to emerge in two flavors: one that acts on mechanics that employ continuous numeric variables (i.e. pointer speed and item speed) and one that acts on mechanics that employ boolean variables (i.e. being allowed to select or not and item availabilty on the screen). Games like OnToGalaxy, for example, presuppose aiming at labels that move around at a certain speed. Furthermore, it may be possible to hit those labels only if bullets are available. In Infection, a game for validating semantic links, zombies or people must be shot down who utter a word that is unrelated to a given word, before they reach a certain point in the map. The principle is the same: labels move around at a certain speed and the possibility to shoot them is subject to projectile availability. The definition we give of orthogonality is thus based on two (of potentially more) fundamental ways in which orthogonal mechanics can appear: *manipulation* and *restriction*. For example, if the items move, their position is being manipulated. If the items disappear after a certain amount of time, so as to increase the sense of urgency and challenge, their availability to the action of the users is restricted. We speak of manipulation when orthogonality influences the *position* or *shape* parameters (where is the target?). We speak of restriction when orthogonality influences the *state* parameter (is the target active or visible? Are the resources to perform the annotation sufficient?). Some games lean more towards one of the two. To summarize, orthogonality takes place where annotation is challenged by mechanics that make it more unpredictable than in standard tools. Arguably, additional coordination skills are required when manipulation takes place, while more reasoning and planning, or more time, may have to be required when restriction is introduced. It is therefore important to assess the degree of orthogonality that exists in a given game. We introduce a threshold based on resemblance with the base task and focus in particular on the shooting mechanic. It can be argued that when the user's selection speed is intact, which means that it is a lot higher than the item speed, orthogonality is low on the manipulation side. When users can click and select items as many times as they want, orthogonality is low on the restriction side. Conversely, manipulation grows when the user's selection speed is in some way compromised or challenged (for example, it equals item speed). Restriction grows when the selection availability is challenged by a resource amount (for example, the number of bullets equals the number of enemies multiplied by their life points). There can be other mechanics interested by manipulation and restriction, but we focused on these ones for the sake of explanation and for our artifact evaluation.

It is worth noting that in a scenario with no bullets, selection speed depends directly on the user's selection speed, thus relying on their bare pointing skills.

5 EVALUATION OF SPACEWORDS

In order to be able to test the variables mentioned above, we developed Spacewords, a browser space shooter game for word similarity annotation, inspired by OnToGalaxy. Before starting a game session, a common word such as 'house', 'beautiful' and 'work' is displayed to the player. The game consists of moving a spaceship and shooting the enemies when they carry a label that is a synonym of the given word. For example, if 'house' is given, 'home', 'residence' and 'habitation' should be targeted, while ignoring enemies carrying labels such as 'desk', 'car' or 'mountain'. The repository

Table 3. Orthogonality threshold examples and scores.

Relation between parameters	Orthogonality
Manipulation	
Projectile speed ≫ Item speed	Low - 0
Projectile speed ≈ Item speed	High - 1
Restriction	
Number of projectiles ≫ Number of target items * item health points	Low - 0
Number of projectiles ≈ Number of target items * item health points	High - 1

of terms and synonyms given in input to the game is built starting from online dictionaries. The final goal of the task behind this game could be building or validating WordNet-like [22] language resources, where taxonomical information is organised around groups of synonyms.

The game is in 2D and was developed with Unity[2] and free resources found on itch.io[3], a well known website for independent game and asset sharing. Our evaluation addresses the following research question: *How do different orthogonality scores impact annotation accuracy?*

5.1 Participants and Procedure

Participants (43 people in total) were recruited in our affiliation facilities. They were asked to play a browser game where they had to recognise semantically similar words, carried around by spaceships in the forms of labels, and shoot them while sparing the others. In the end, we removed 3 players who scored lower than 0.5 in more than 2 conditions, which is the value that is obtained when a player does nothing. The final number of valid participants was N=40, with 11 assigned to the first order and 9 assigned to the third order. The other two orders contained 10 participants each. At the beginning of the game, a brief demographic questionnaire was administered. 70% of players were aged 25-34; 14% were 18-24; 16% were 35-44. Females, males and other accounted for 33%, 64% and 3% respectively. 22% held a high school diploma, 8% had a bachelor's degree, 53% had a master's degree and 17% had a PhD. 30% were not gamers, while 28% reported playing every day and 28% once a month. The remaining 15% played between once and 3 times a week. Finally, among the people who reported playing sometimes or often, 46% played on the computer, 36% on consoles, 12% on smartphones or tablets and 6% answered 'other'.

Participants were administered 4 conditions in a 2x2 within-subjects factorial design. The order of conditions was counterbalanced using a balanced latin square. The two independent variables were a *manipulation* variable (bullet speed) and a *restriction* variable (amount of bullets). Since the groups of synonym words were extracted from existing dictionaries, we could use them as a gold standard to evaluate the quality of the annotated terms. We therefore consider three dependent variables: Precision (percentage of hit targets that were synonym words), Recall (percentage of synonym items displayed to the user that were correctly hit) and F-score (the harmonic mean between the two). Every participant played four levels corresponding to the four conditions, each starting with a training session to get familiar with the controls. When the bullet amount was low, it was still the minimum amount (three bullets) needed to carry out the task with maximum accuracy if no mistakes were made (enemy health was one life point). When the bullet speed was low, it was still the same speed as the spaceships, while the spaceship controlled by the player was

[2]http://unity.com
[3]http://www.itch.io, resources made by user MattWalkden

significantly faster than the enemy spaceships across all conditions. Indeed, players could reduce the impact of bullet speed by moving towards enemy spaceships, but they also had to stay as far as possible from them as collision resulted in damage to the player.

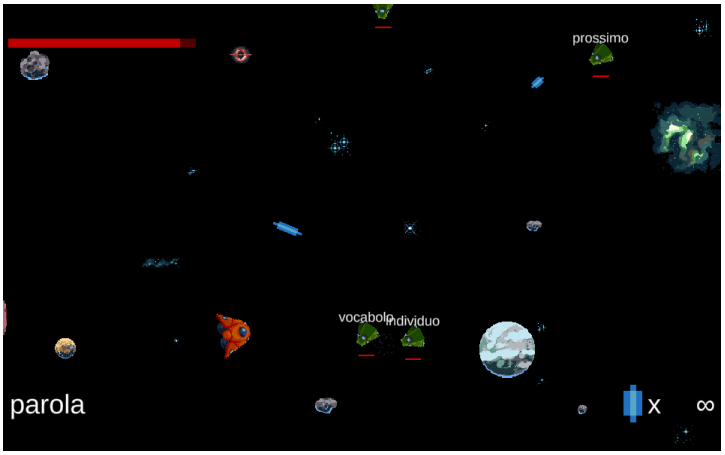

Fig. 5. A screenshot from our game, Spacewords. The enemies (green) carry either a wrong or a correct label.

All players annotated exactly the same words, with 12 related and 12 unrelated examples (disturb words) for each condition. To prevent some players from annotating more words than other players did, and avoid some players practicing too much on the same condition, we removed the possibility to die and lose the game, and told participants we would consider their remaining health as their score to maintain an acceptable level of challenge. Every condition lasted the same amount of time, with 24 enemy spaceships per condition staying 10 seconds on the screen unless they were destroyed. Usually, in this type of games, the ground truth or gold standard is only limited to a small fraction of the data, which is useful for the verification mechanics, while in this case we already knew in advance all the true positive and true negative labels. We did not provide users with information about the correctness of their judgments. Our game belongs to Type 2 from our classification in Table 2.

5.2 Results

We compare the average Precision, Recall and F-score obtained for each condition and report them in Figure 6. Results show that the condition ensuring the best annotation quality is the one with manipulation and restriction both set to 0 (0.886, fast x infinite), which is the condition that imitates a standard annotation interface most closely, with infinite bullets and very high selection speed (bullet speed was 10 times faster in the fast conditions than in the slow conditions). The worst condition with respect to the F-scores was the one with manipulation and restriction both set to 1 (0.802, slow x limited), although not by much if compared to the fast x limited condition (0.835). Fast x limited is in the middle with 0.856. What is interesting to note, however, is that there are noticeable differences between the values of precision and recall. Increasing manipulation or restriction yielded mixed results with respect to these two measures.

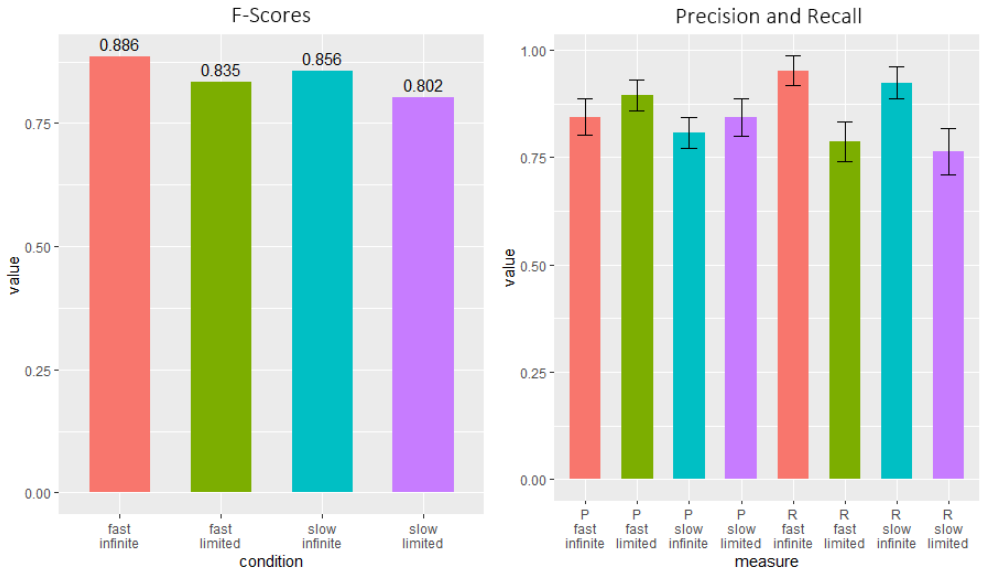

Fig. 6. F-Scores of each condition (left); Macro-averaged Precision (P) and Recall (R) of the players for each condition (right).

It seems that with limited bullets, not only are fewer enemies (items) hit, as one would expect (thus decreasing recall noticeably), but the ratio between relevant items and irrelevant items stays or becomes unbalanced in favor of the relevant items, which leads to a higher precision, as if participants paid more attention when they knew their bullets were limited. This suggests that whenever false positives are a problem in a dataset, limiting the bullets (which is however quite common in video games) might even be advantageous. It is possible that having unlimited selectors (bullets) in a complex context such as a space shooter can cause a higher error probability, or maybe players become just more eager to shoot, which is after all a core game mechanic. It is worth noting that in this setting, restriction set to 1 was already quite extreme: after running out of bullets there was no way to recharge them before the following condition. On average, out of the 3 allowed projectiles, 0.73 and 0.79 were projectiles wasted in the limited and slow x limited conditions respectively. Allowing one more projectile may then be beneficial to recall in the two conditions with limited shots. In total 40% of all perfect precision scores (i.e. 0 false positives) with at least 50% of positives (6 out of 12) come from the fast x limited condition (with a mean of true positives $M = 10.3$, $SD = 1.48$). We provide an overview of the different orthogonality scores in terms of micro-averaged Precision, Recall and F-score in Table 4.

A score as low as 0 does not however mean that there is no orthogonality whatsoever. Our score only refers to the specific variables that we manipulated, increasing and decreasing orthogonality with respect to that already present in our specific game implementation.

6 DISCUSSION

In this work we consider orthogonal game mechanics in games with a purpose for linguistic annotation, an aspect that is still poorly studied and has unclear impact on the interaction between users and linguistic annotation artifacts. We borrow the concept of action mechanics (the mechanics users interact with in any human computation game) from Siu et al. [29] and we conceive them as the union of the fundamental task mechanics and, if present, orthogonal mechanics. By representing

Table 4. Micro-averaged Precision, Recall and F-Score obtained with different combinations of Manipulation and Restriction.

Manipulation	Restriction	Precision	Recall	F-Score
0 - Fast	0 - Infinite	0.83	**0.95**	**0.886**
0 - Fast	1 - Limited	**0.892**	0.785	0.835
1 - Slow	0 - Infinite	0.798	0.923	0.856
1 - Slow	1 - Limited	0.846	0.761	0.802

these two components clearly, it is possible to assess their distance in terms of complexity, that is, the orthogonality that separates them. We measure its magnitude by differentiating between manipulation and restriction strategies.

Orthogonal Mechanics: After carrying out an analysis of the literature about GWAPs for linguistic annotation, what emerges is that the majority does not employ orthogonal mechanics, i.e., mechanics that require skills or expertise that potentially go beyond the annotation task requirements. Therefore, the majority of the games taken into consideration relies on perfect overlap between task mechanics and game mechanics. Although this could be good practice with respect to task performance and annotation accuracy, it could be detrimental to motivation and dissemination. Some rather well-designed attempts at merging full-fledged games and annotation games with perfect overlap exist, but we find it reasonable to expect that attractive games should also try to rely more on orthogonal game mechanics like aiming, jumping, slashing, driving and dodging, which are all hallmarks of successful commercial video games such as Super Mario, Call of Duty, Need for Speed, The Legend of Zelda, to name a few.

Orthogonality Score: Our orthogonality score differentiates between two strategies: manipulation and restriction. Parameters that act on continuous numeric variables such as coordinates and areas fall under the manipulation category and may require coordination skills; parameters that act on boolean variables such as visibility or activation fall under the restriction category and may involve strategic planning and reasoning. We consider as manipulative or restrictive those mechanics that produce a change over time and oppose the user's will, in other words, mechanics that make annotation accuracy more uncertain.

Evaluation of Spacewords and design recommendations: To assess the impact of orthogonal mechanics on linguistic annotations, we developed a game inspired by OnToGalaxy. We administered to 40 participants four conditions with different combinations of manipulation and restriction scores. We observed the best accuracy score in the condition with orthogonality set at minimum and the worst condition with orthogonality set at maximum. However, with respect to the two components of the F-score, namely precision and recall, we observed interesting differences as players seem to hit positives more carefully when bullets are limited. This can have several implications for future GWAP design as manipulation and restriction seem to influence gameplay in different ways. Furthermore, having limited bullets is a very common trope in video games and could be therefore recommended in some scenarios. We also observed that allowing 1 more bullet in the limited conditions might be already quite helpful for players.

Limitations and Future Work: Our approach has still several limitations. First, we provide no account for the aesthetics part of the MDA framework, while calculating the aesthetical abstraction (given by graphical or richness and complexity) between a game and a basic annotation task would certainly be interesting and relevant to disjoint design. Second, speaking of our evaluation, one

limitation concerns the very specific type of game we implemented, namely a space shooter. In addition, only 2 variables were manipulated (bullet speed and bullet availability). Further work should explore whether other parameters from the same category (manipulation or restriction) have the same impact on the task.

7 CONCLUSION

In this paper, we have proposed a theoretical tool to study and comprehend linguistic annotation games and the possible impact of their design on interaction. First, we integrated Siu et al.'s framework [29] for human computation games with Krause et al.'s concept of disjoint design [18] and Tuite's defition of orthogonal mechanics [31] to break down and analyze linguistic GWAPs from the point of view of their mechanics. Second, by expanding on the concept of orthogonal game mechanics, we proposed a preliminary metric to evaluate the magnitude of orthogonality that is implemented in a given GWAP, by distinguishing between manipulation and restriction strategies. We stress the importance of quantifying orthogonality, since the skills required to play a GWAP may vary with the amount of orthogonality employed. To test the practical utility of our metrics we developed Spacewords, a simple space shooter where players have to shoot synonym words while ignoring unrelated ones. Our results indicate that indeed the preferable condition for annotation might be the one that imitates the essential task most closely (unlimited bullets with very high speed) but decreasing the available amount of bullets actually plays in favor of a specific measure of accuracy, namely precision. We therefore observed that increasing orthogonality may yield mixed results with respect to specific game design patterns such as ours.

ACKNOWLEDGMENTS

The research activities presented in this work have been partially supported by the EU program REC-RDAP-GBV-AG-2020, grant n. 101005518 – KID_ACTIONS.

REFERENCES

[1] Valentina Bartalesi Lenzi, Giovanni Moretti, and Rachele Sprugnoli. 2012. CAT: the CELCT Annotation Tool. In *Eight International Conference on Language Resources and Evaluation (LREC'12)*. 333–338.

[2] Federico Bonetti and Sara Tonelli. 2020. A 3D Role-Playing Game for Abusive Language Annotation. In *Workshop on Games and Natural Language Processing*. European Language Resources Association, Marseille, France, 39–43. https://aclanthology.org/2020.gamnlp-1.6

[3] Federico Bonetti and Sara Tonelli. 2021. Challenges in Designing Games with a Purpose for Abusive Language Annotation. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*. Association for Computational Linguistics, Online, 60–65. https://aclanthology.org/2021.hcinlp-1.10

[4] Kristy Elizabeth Boyer, Robert Phillips, Michael Wallis, Mladen Vouk, and James Lester. 2008. Balancing Cognitive and Motivational Scaffolding in Tutorial Dialogue. In *Intelligent Tutoring Systems*. Vol. 5091. Springer Berlin Heidelberg, Berlin, Heidelberg, 239–249. https://doi.org/10.1007/978-3-540-69132-7_28 ISSN: 0302-9743, 1611-3349 Series Title: Lecture Notes in Computer Science.

[5] Anca Dumitrache, Lora Aroyo, Chris Welty, Robert-Jan Sips, and Anthony Levas. 2013. "Dr. Detective": combining gamification techniques and crowdsourcing to create a gold standard in medical text. In *Proceedings of the 1st International Workshop on Crowdsourcing the Semantic Web (CrowdSem 2013), 12th International Semantic Web Conference*.

[6] Dagmara Dziedzic and Wojciech Włodarczyk. 2017. Making NLP games with a purpose fun to play using Free to Play mechanics: RoboCorp case study. In *Proceedings of the EACL 2017 Workshop "Using Games and Gamification for Natural Language Processing (Games4NLP)"*. 2.

[7] Karën Fort, Bruno Guillaume, and Hadrien Chastant. 2014. Creating *Zombilingo* , a game with a purpose for dependency syntax annotation. In *Proceedings of the First International Workshop on Gamification for Information Retrieval - GamifIR '14*. ACM Press, Amsterdam, The Netherlands, 2–6. https://doi.org/10.1145/2594776.2594777

[8] Ivan Habernal, Raffael Hannemann, Christian Pollak, Christopher Klamm, Patrick Pauli, and Iryna Gurevych. 2017. Argotario: Computational Argumentation Meets Serious Games. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Copenhagen, Denmark, 7–12. https://doi.org/10.18653/v1/D17-2002

[9] M. P. Jacob Habgood and Shaaron E. Ainsworth. 2011. Motivating Children to Learn Effectively: Exploring the Value of Intrinsic Integration in Educational Games. *Journal of the Learning Sciences* 20, 2 (April 2011), 169–206. https://doi.org/10.1080/10508406.2010.508029

[10] Kieran Hicks, Patrick Dickinson, Juicy Holopainen, and Kathrin Gerling. 2018. Good Game Feel: An Empirically Grounded Framework for Juicy Design. In *Digra '18 - Proceedings of the 218 Digra International Conference*. 17.

[11] Barbora Hladká, Jiří Mírovský, and Pavel Schlesinger. 2009. Play the language: play coreference. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers on - ACL-IJCNLP '09*. Association for Computational Linguistics, Suntec, Singapore, 209. https://doi.org/10.3115/1667583.1667648

[12] Robin Hunicke, Marc Leblanc, and Robert Zubek. 2004. MDA: A Formal Approach to Game Design and Game Research. *AAAI Workshop - Technical Report* 1 (01 2004).

[13] Peter Jamieson, Jack Hall, and Lindsay Grace. 2012. Research Directions for Pushing Harnessing Human Computation to Mainstream Video Games. In *Meaningful Play 2012*. East Lansing, MI.

[14] Alain Joubert, Mathieu Lafourcade, and Nathalie Le Brun. 2018. The JeuxDeMots Project is 10 Years Old: What We have Learned. In *Proceedings of the 2018 LREC Workshop "Games and Gamification for Natural Language Processing (Games4NLP)"*. Miyazaki, Japan, 22–26. https://anawiki.essex.ac.uk/dali/games4nlp/

[15] David Jurgens and Roberto Navigli. 2014. It's All Fun and Games until Someone Annotates: Video Games with a Purpose for Linguistic Annotation. *Transactions of the Association for Computational Linguistics* 2 (Dec. 2014), 449–464. https://doi.org/10.1162/tacl_a_00195

[16] Jesper Juul. 2010. *A Casual Revolution: Reinventing Video Games and Their Players*. MIT Press (MA). https://books.google.it/books?id=heEsCwAAQBAJ

[17] Doruk Kicikoglu, Richard Bartle, Jon Chamberlain, and Massimo Poesio. 2019. Wormingo: a 'true gamification' approach to anaphoric annotation. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*. ACM, San Luis Obispo California USA, 1–7. https://doi.org/10.1145/3337722.3341868

[18] Markus Krause, Aneta Takhtamysheva, Marion Wittstock, and Rainer Malaka. 2010. Frontiers of a paradigm: exploring human computation with digital games. In *Proceedings of the ACM SIGKDD Workshop on Human Computation - HCOMP '10*. ACM Press, Washington DC, 22–25. https://doi.org/10.1145/1837885.1837893

[19] Mathieu Lafourcade and Nathalie Le Brun. 2017. Ambiguss, a game for building a Sense Annotated Corpus for French. In *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*. https://www.aclweb.org/anthology/W17-6920

[20] Chris Madge, Richard Bartle, Jon Chamberlain, Udo Kruschwitz, and Massimo Poesio. 2019. The Design Of A Clicker Game for Text Labelling. In *2019 IEEE Conference on Games (CoG)*. IEEE, London, United Kingdom, 1–4. https://doi.org/10.1109/CIG.2019.8848068

[21] Chris Madge, Massimo Poesio, Udo Kruschwitz, and Jon Chamberlain. 2018. Testing TileAttack with Three Key Audiences. In *Proceedings of the 2018 LREC Workshop "Games and Gamification for Natural Language Processing (Games4NLP)"*. Miyazaki, Japan, 6–11. https://anawiki.essex.ac.uk/dali/games4nlp/

[22] George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

[23] Emily Öhman and Kaisla Kajava. 2018. Sentimentator: Gamifying Fine-grained Sentiment Annotation. In *Proceedings of the Digital Humanities in the Nordic Countries 4th Conference (DHN 2019)*, Vol. 2084. Helsinki, Finland, 98–110. http://ceur-ws.org/Vol-2084/

[24] Massimo Poesio, Jon Chamberlain, Udo Kruschwitz, Livio Robaldo, and Luca Ducceschi. 2013. Phrase detectives: Utilizing collective intelligence for internet-scale language resource creation. *ACM Transactions on Interactive Intelligent Systems* 3, 1 (April 2013), 1–44. https://doi.org/10.1145/2448116.2448119

[25] Nathan Prestopnik and Kevin Crowston. 2012. Purposeful Gaming Socio-Computational Systems: A Citizen Science Design Case. In *Proceedings of the 17th ACM International Conference on Supporting Group Work* (Sanibel Island, Florida, USA) *(GROUP '12)*. Association for Computing Machinery, New York, NY, USA, 75–84. https://doi.org/10.1145/2389176.2389188

[26] James Pustejovsky and Amber Stubbs. 2012. *Natural Language Annotation for Machine Learning - a Guide to Corpus-Building for Applications*. O'Reilly. http://www.oreilly.de/catalog/9781449306663/index.html

[27] Anurag Sarkar and Seth Cooper. 2019. Using a Disjoint Skill Model for Game and Task Difficulty in Human Computation Games. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*. ACM, Barcelona Spain, 661–669. https://doi.org/10.1145/3341215.3356310

[28] Miguel Sicart. 2008. Defining Game Mechanics. *Game Studies. The International Journal of Computer Game Research* 8 (12 2008), 1–14.

[29] Kristin Siu, Alexander Zook, and Mark O. Riedl. 2017. A Framework for Exploring and Evaluating Mechanics in Human Computation Games. *arXiv:1706.03311 [cs]* (June 2017). http://arxiv.org/abs/1706.03311 arXiv: 1706.03311.

[30] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations at the 13th Conference of the*

European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Avignon, France, 102–107. https://www.aclweb.org/anthology/E12-2021

[31] Kathleen Tuite. 2014. GWAPs: Games with a Problem. In *Proceedings of the 9th International Conference on the Foundations of Digital Games*. http://www.fdg2014.org/proceedings.html

[32] Daniele Vannella, David Jurgens, Daniele Scarfini, Domenico Toscani, and Roberto Navigli. 2014. Validating and Extending Semantic Knowledge Bases using Video Games with a Purpose. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 1294–1304. https://doi.org/10.3115/v1/P14-1122

[33] Noortje J Venhuizen, Kilian Evang, Valerio Basile, and Johan Bos. 2013. Gamification for Word Sense Labeling. In *Proceedings of the International Conference on Computational Semantics (IWCS)*. 397–403.

[34] Luis von Ahn. 2006. Games with a purpose. *Computer* 39, 6 (2006), 92–94. https://doi.org/10.1109/MC.2006.196

[35] Luis von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*. ACM Press, Vienna, Austria, 319–326. https://doi.org/10.1145/985692.985733

Received February 2021; revised June 2021; accepted July 2021