Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

GECKA3D: A 3D Game Engine for Commonsense Knowledge Acquisition

Erik Cambria[*]
School of Computer Engineering
Nanyang Technological University
cambria@ntu.edu.sg

Tam V. Nguyen[*]
ARTIC Research Center
Singapore Polytechnic
nguyen_van_tam@sp.edu.sg

Brian Cheng
ARTIC Research Center
Singapore Polytechnic
brian_cheng@sp.edu.sg

Kenneth Kwok
Institute of High Performance Computing
Agency for Science, Technology and Research
kenkwok@ihpc.a-star.edu.sg

Jose Sepulveda
ARTIC Research Center
Singapore Polytechnic
sepulveda_jose@sp.edu.sg

Abstract

Commonsense knowledge representation and reasoning is key for tasks such as artificial intelligence and natural language understanding. Since commonsense consists of information that humans take for granted, gathering it is an extremely difficult task. In this paper, we introduce a novel 3D game engine for commonsense knowledge acquisition (GECKA3D) which aims to collect commonsense from game designers through the development of serious games. GECKA3D integrates the potential of serious games and games with a purpose. This provides a platform for the acquisition of re-usable and multi-purpose knowledge, and also enables the development of games that can provide entertainment value and teach players something meaningful about the actual world they live in.

Introduction

Many existing multimedia annotation and artificial intelligence systems benefit from large-scale datasets with accurate annotations. For example, 40 participants were gathered to rate the attractiveness scores for 1240 subjects in Beauty Sense project (Nguyen et al. 2012; 2013a). In another work, 69 participants were requested to provide touch saliency for at least 400 images (Ni et al. 2014). However, constructing such datasets is generally tedious and labor-intensive, as it involves massive manual-cropping and hand-labeling. Therefore, games with a purpose (GWAPs) are a simple yet powerful means to collect useful information from players in a way that is entertaining for them (von Ahn 2006).

A GWAP embeds data collection implicitly into the game playing process. It is developed from the human computation technique which relies on the collaboration of game players to solve problems. Each player performs a small part of a massive computation, similar to a processor in a distributed system. Unlike computer processors, however, humans require a certain incentive to participate in such collective computation. GWAP aims to implicitly embedding the computational tasks in an enjoyable and attractive games while guaranteeing correct solutions to the problems.

Over the past few years, many GWAPs have been proposed to exploit the brainpower made available by multitudes of casual gamers to perform tasks that, despite being relatively easy for humans to complete, are difficult for machines. GWAPs are best known for image annotation. In the 'ESP' game (von Ahn and Dabbish 2004), players guess content objects or properties of random images by typing what they see when it appears on the screen. Other image annotation games include: Matchin (Hacker and von Ahn 2009), a two-player game in which each player is shown two images and asked to click on the image their partner would prefer, and Phetch (von Ahn et al. 2006), a game that collects explanatory descriptions of images in order to improve Web accessibility for the visually impaired. Recently, Purposive Hidden-Object-Game (Feng et al. 2012) was proposed. It was designed to imperceptibly embed localizing objects into an enjoyable game process thus attracting many people to make voluntary contribution to annotating images.

Besides images, GWAPs have been used for video annotation. For example, OntoTube (Siorpaes and Hepp 2008), Yahoo's Videotaggame (van Zwol et al. 2008), and Waisd (Addis et al. 2010), are all games in which two players have to quickly agree on a set of tags for the same streaming YouTube video. GWAPs have also been exploited to automatically tag music tracks with semantic labels. HerdIt (Barrington et al. 2009), for example, asks players to accomplish various tasks and answer quizzes related to the song they are listening to. In Tagatune (Law et al. 2007), two players listen to an audio file and describe to the other what each is hearing to then decide whether the same soundtrack is being played to both. One of the most interesting tasks GWAPs can be used for is commonsense knowledge acquisition from members of the general public (Chklovski 2003; Speer 2007; Cambria, Xia, and Hussain 2012).

[*]indicates the co-first authorship

One example, Verbosity (von Ahn, Kedia, and Blum 2006), is a real time quiz game for collecting commonsense facts. In the game, two players take different roles at different times: one functions as a narrator, who has to describe a word using templates, while the other has to guess the word in the shortest time possible. Another example is FACTory Game (Lenat and Guha 1989), a GWAP developed by Cycorp which randomly chooses some facts and presents them to players in order for them to guess whether a statement is true or false. A variant of the FACTory game is the Concept Game on Facebook (Herdagdelen and Baroni 2010), which collects commonsense knowledge by proposing random assertions to users and asks them whether the given assertion is meaningful or not.

Questions are very important in collecting commonsense. Kuo and Hsu (Kuo and Hsu 2011) introduced the problem of resource-bounded crowdsourcing of commonsense knowledge and proposed an approach to find a productive question set automatically. Another problem with existing GWAPs is that the information gathered from them is often unrecyclable; acquired data is often applicable only to the specific stimuli encountered during gameplay. Moreover, such games often have a fairly low 'sticky factor', and are often unable to engage gamers for more than a couple of minutes. The fun factor sometimes is not considered during the design of such a GWAP system.

In this paper, we propose a 3D game engine for commonsense knowledge acquisition (GECKA3D), that aims to overcome the main drawbacks of traditional data-collecting games by empowering users to create their own GWAPs and by mining knowledge that is highly reusable and multipurpose. In addition, we also consider adding game mechanism to enhance the fun factor to engage the players. In particular, the GECKA3D framework has two modes, editor and player. The editor mode allows game designers to create compelling serious games for their peers to play. In doing so, they gather commonsense knowledge useful for intelligent applications in any field requiring in-depth knowledge of the real world, including reasoning, perception and social systems simulation. Meanwhile, the player mode is used by players to perform the task. The framework was built on top of the cocos2d game engine[1], a suite of open-source, cross-platform, and game-development tools used by thousands of developers all over the world.

Proposed Framework

An important difference between traditional artificial intelligence systems and human intelligence is the human ability to harness commonsense knowledge gathered from a lifetime of learning and experience to make informed decisions. This allows humans to adapt easily to novel situations where artificial intelligence fails catastrophically due to a lack of situation-specific rules and generalization capabilities. Commonsense knowledge also provides background information enabling humans to successfully operate in social situations where such knowledge is typically assumed (Poria et al. 2013).

[1]http://www.cocos2d-x.org

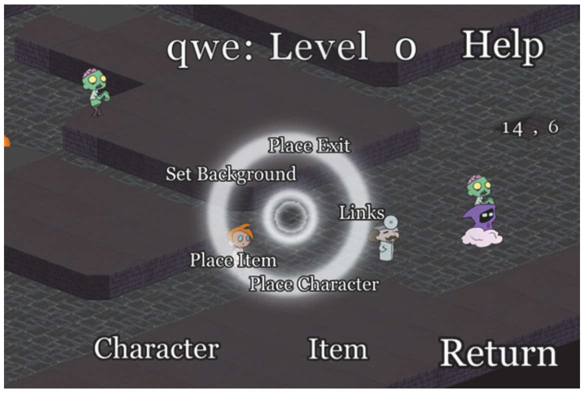


Figure 1: Editor mode. Game designers can drag and drop objects and characters from the library and specify how these interact with each other.

To this end, GECKA3D allows users to design compelling serious games via editor mode. As opposed to commonsense concept extraction techniques (Rajagopal et al. 2013) or traditional GWAPs, GECKA3D does not limit users to specific, often tedious, tasks, but rather gives them the freedom to choose both the kind and the granularity of knowledge to be encoded, through a user-friendly and intuitive interface. Not just a system for the creation of microgames, GECKA3D is a serious game engine that aims to give designers the means to create long adventure games to be played by others. GECKA3D offers functionalities typical of role-play games (RPGs), e.g., a question/answer dialogue box enabling communication and the exchange of objects (optionally tied to correct answers) between players and virtual world inhabitants, a library for enriching scenes with useful and yet visually-appealing objects, backgrounds, characters.

Initial Knowledge Construction

The core elements in the proposed GECKA3D framework are objects and actions. Their semantics can be specified through prerequisite-object-action-goal (POAG) quartets. Prerequisites indicate what must be present or have been done before applying the object or action. Objects indicate states or objects of the world (including emotional states, e.g., "if I give food to hungry people, their happiness is likely to rise").

Goals in turn specify the specific scene goals that are facilitated by that particular POAG quartet. In order to provide designers with initial knowledge, we first build the corpus of Object-Action groups. We consider verbs as actions whereas nouns can be regarded as objects. In the initial step of corpus retrieval, we utilized the combinations of verbs and nouns as the search terms fed into Microsoft Bing search engine[2] in the form of 'how to + a <noun>'. The number of searched results can be considered as the joint likelihood of the combination of action and object.

We also integrate boost library[3] for semantic parsing. In

[2]http://www.bing.com

[3]http://www.boost.org

the post-processing step, we completed cleaning of verb and noun corpuses with normalization, e.g., forcing all characters to lowercase, and removing html tags. As a result, we collect the initial corpus with 1500 nouns and 636 verbs in English.

Editor Mode

Unlike other serious games or data collection tools (Russell et al. 2008; Feng et al. 2012; Nguyen et al. 2013b), GECKA3D provides authoring tools (or so called editor mode) for designers to create the training scenarios. The editor mode is built in 2.5D which is similar as 2D graphical projections and simulates the appearance of being three-dimensional (3D). In other words, the gameplay is in a 3D video game that is restricted to a 2D plane or has a virtual camera with a fixed angle.

This is inspired by games such as SimCity[4]. Figure 1 illustrates the editor mode in being used by a designer. The designers are able to use the editor to create the training scenario in the form of tile based levels. The GECKA3D framework allows designers to define actions, objects, and goals. In case an action or an object is not available in the library, GECKA3D allows game designers to define their own custom items by building shapes from a set of predefined geometric forms or applying transformations to existing items. This enables the creation of new objects for which there is no available icon by combining available graphics and predefined shapes, and the use of transformations to create various object states, such as a 'broken cup'. The ability of users to create their own custom items and actions is key to maintaining an undisrupted game flow.

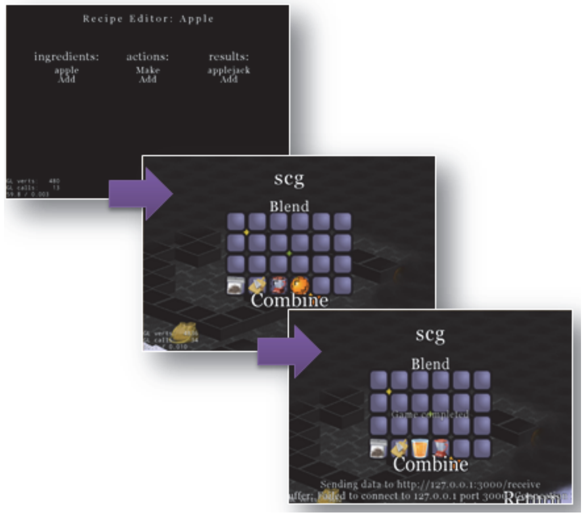
Each game has enter/exit portal sprites. The designer has to define the starting location of the in-game character. It is easy for designer to know which portal when defining entrance. This feature allows the character to traverse in different levels. As mentioned in (von Ahn 2006), people play games not because they are personally interested in solving an instance of a computational problem but because they wish to be entertained. Therefore, in order to gain the sticky factor of the game, GECKA3D embeds some game mechanism. For example, it allows designers to add monsters/zombies to chase the in-game player. Therefore, the player has to make decision quickly in order to avoid the pursuers.

Object-Action Combinations

Game designers drag and drop objects and characters from action/object libraries into scenes. In addition, the designers can add item combinations and the corresponding outcomes, for example, orange + blender = orange juice. In particular, designers can create, load and delete such item combinations. They can specify POAG quartet by grouping objects with the actions performed over it (Fig. 2). POAG quartets give us pieces of commonsense information like "the result of blending an orange with a blender, is orange juice".

Towards the goal of improving gameplay, and since we are mainly interested in typical commonsense knowledge,

[4]https://en.wikipedia.org/wiki/SimCity



Sending data to http://127.0.0.1:3000/receive
uffer: failed to connect to 127.0.0.1 port 3000

Figure 2: The designers specify the item combination rule. They also define the outcome of the combination. Note that items can be objects or actions.

POAG quartets associated with a specific object type are shared among all the instances of such an object ('inheritance'). Whenever a game designer associates a POAG to an object in the scene, that POAG instantly becomes shared among all the other objects of the same type, no matter if these are located in different scenes. New instances inherit this POAG as well. Game designers, however, can create exceptions of any object type through the creation of new custom objects. A 'moldy bread' custom object, for example, normally inherits all the POAGs of 'bread' but these can be changed, modified, or removed at the time of object instantiation without affecting other 'bread' type objects.

The POAG specification is among the most effective means to collect commonsense knowledge, given that it is performed quite often by the game designer during the creation of scenes. From a simple POAG definition we may obtain a large amount of knowledge, including interaction semantics between different objects, prerequisites of actions, and the goals commonly associated with such actions. These pieces of commonsense knowledge, are very clearly-structured, and thus easy to assimilate into the knowledge base, due to the fixed framework for defining interaction semantics.

Player Mode

Besides allowing for the acquisition of knowledge from game designers, GECKA3D enables players of the finished games to be educated in useful ways, all while being entertained. In particular, these use the player mode to control the in-game character for fulfilling the predefined goals. The gameplay is also built in 2.5D as similar as the editor mode.

As aforementioned, the player has to make decisions quickly to avoid the zombies or monsters. The in-game rooms and corridors are randomly generated to make sure the players have no bias about the spatial layout. Inspired by

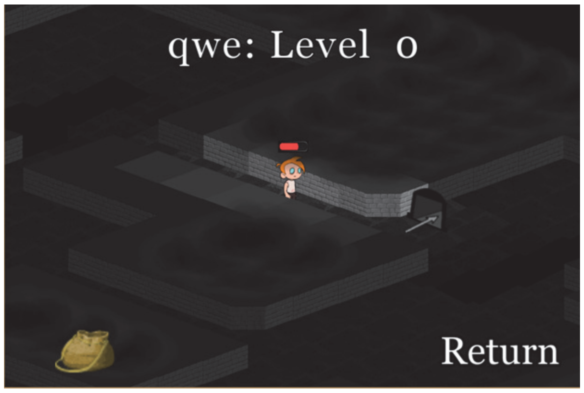


Figure 3: Status of an in-game character in the scene. The in-game character has to fulfill the goal of the game, meanwhile he has to avoid the attack of the embedded zombies.

other games such as Diablo[5], only a small portion of view close to in-game character is revealed. The more the character moves, the clearer vision the player achieves. We also utilize the path finding algorithm for player movement and explorer. When the user clicks to one point in the map, the system automatically finds the shortest path from the user's current position and the target point. The in-game zombies are inserted to put pressure on the player. Note that the zombies chase the player in a turn based manner.

Server

Server is yet another important component of the framework. It is used to store all of the information of the gameplay. The core functionality include game data uploading from remote clients to server. Note that our editor mode and player mode are deployed in many platforms thanks to the cross-platform characteristic of the chosen game engine. In particular, we use mongoDB and nodejs for server behavior. Http requests are integrated into mobile communication. Data can be transferred from phone to arbitrary server on internet. All the history of game making from the designers and gameplay from the players is stored for further analysis.

Evaluation

In order to perform a preliminary evaluation of the type and quality of knowledge that GECKA3D can potentially gather, we tested it on 20 university students and staff, who were given the game on tablet and were asked to design a few game scenes over the span of a few hours.Game designers' actions were collected and encoded according to a specific XML format that encodes the semantics associated with such actions (Fig. 4).

Specific procedures translate these XML files into pieces of commonsense knowledge to be fed to the SenticNet framework (Cambria and Hussain 2015). Such pieces of commonsense knowledge were manually evaluated (reasonable vs. unreasonable) by 5 annotators, resulting in an accuracy of 85.7%. They also were asked to try the predefined

[5]https://en.wikipedia.org/wiki/Diablo_(video_game)

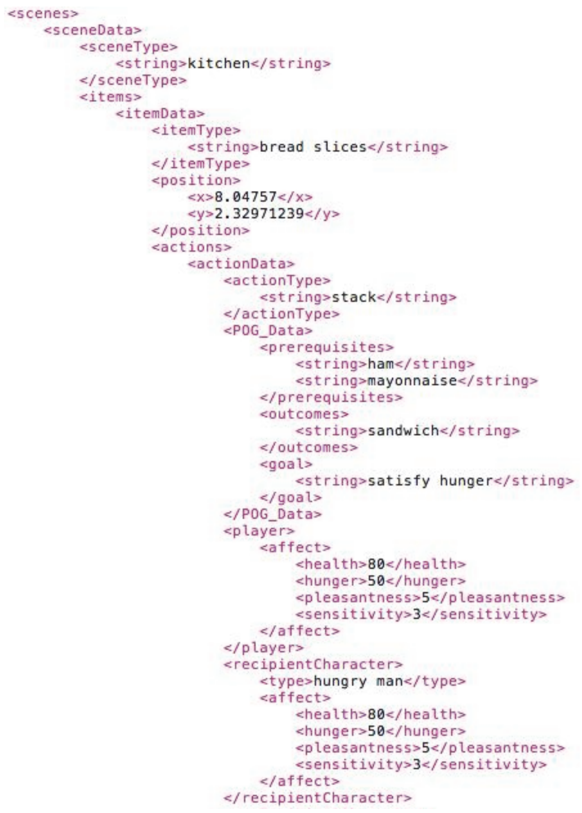

```
<scenes>
    <sceneData>
        <sceneType>
            <string>kitchen</string>
        </sceneType>
        <items>
            <itemData>
                <itemType>
                    <string>bread slices</string>
                </itemType>
                <position>
                    <x>8.04757</x>
                    <y>2.32971239</y>
                </position>
                <actions>
                    <actionData>
                        <actionType>
                            <string>stack</string>
                        </actionType>
                        <POG_Data>
                            <prerequisites>
                                <string>ham</string>
                                <string>mayonnaise</string>
                            </prerequisites>
                            <outcomes>
                                <string>sandwich</string>
                            </outcomes>
                            <goal>
                                <string>satisfy hunger</string>
                            </goal>
                        </POG_Data>
                        <player>
                            <affect>
                                <health>80</health>
                                <hunger>50</hunger>
                                <pleasantness>5</pleasantness>
                                <sensitivity>3</sensitivity>
                            </affect>
                        </player>
                        <recipientCharacter>
                            <type>hungry man</type>
                            <affect>
                                <health>80</health>
                                <hunger>50</hunger>
                                <pleasantness>5</pleasantness>
                                <sensitivity>3</sensitivity>
                            </affect>
                        </recipientCharacter>
```

Figure 4: A sample XML output deriving from the creation of a scene in GECKA3D. Actions are collected and encoded according to their semantics.

games on the player mode. 18 out of 20 participants highly appreciate the playability and interestingness of the provided player mode which is crucial for attracting enough players to contribute to the embedded human computing. The other two only care about finishing a task as in traditional GWAPs. More tests are due to verify how such new pieces of commonsense knowledge actually improve the reasoning capabilities of SenticNet on specific tasks such as sentiment analysis (Poria et al. 2012b; 2012a).

Conclusion and Future Work

In this paper, we have introduced the GECKA3D framework which is used for commonsense knowledge acquisition from game designers through the development of serious games. In fact, this not only provides a platform for the acquisition of commonsense knowledge, but also enables the development of games that transfer back the meaningful lessons to gamers. In the future, we aim to update the gameplay, incentives, and interfaces based on analysis of the data resulting from the game, leading to further understanding of the game itself as well as the nexus between game enjoyment and knowledge generation, e.g., finding out what makes providing information fun or what kinds of information are more 'fun' than others.

Item	Action	Prereq.	Outcome	Goal
blender	blend	orange	orange juice	quench thirst
bread	cut	knife	bread slices	–
bread slices	stack	cheese, ham	sandwich	satisfy hunger
coffee beans	hit	pestle	coffee powder	–
coffee maker	fill	coffee powder, boiled water	coffee	keep awake
can	open	can opener	food	satisfy hunger
bag	fill	sand	sandbag	flood control

Table 1: List of most common POAG quartets collected during the pilot testing at Singapore Polytechnic.

Acknowledgments

This work was supported by Singapore Ministry of Education under research Grants MOE2012-TIF-2-G-015 and MOE2014-TIF-1-G-007.

References

Addis, M.; Boch, L.; Allasia, W.; Gallo, F.; Bailer, W.; and Wright, R. 2010. Digital preservation and access in the PrestoPRIME project. In *DPIF Symposium*.

Barrington, L.; O'Malley, D.; Turnbull, D.; and Lanckriet, G. 2009. User-centered design of a social game to tag music. In *ACM SIGKDD*, 7–10.

Cambria, E., and Hussain, A. 2015. *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*. Springer.

Cambria, E.; Xia, Y.; and Hussain, A. 2012. Affective common sense knowledge acquisition for sentiment analysis. In *LREC*, 3580–3585.

Chklovski, T. 2003. Learner: a system for acquiring commonsense knowledge by analogy. In *K-CAP*, 4–12.

Feng, J.; Ni, Y.; Dong, J.; Wang, Z.; and Yan, S. 2012. Purposive hidden-object-game: Embedding human computation in popular game. *IEEE TMM* 14(5):1496–1507.

Hacker, S., and von Ahn, L. 2009. Matchin: Eliciting user preferences with an online game. In *CHI*, 1207–1216.

Herdagdelen, A., and Baroni, M. 2010. The concept game: Better commonsene knowledge extraction by combining text mining and game with a purpose. In *AAAI CSK*.

Kuo, Y., and Hsu, J. Y. 2011. Resource-bounded crowdsourcing of commonsense knowledge. In *IJCAI*, 2470–2475.

Law, E.; von Ahn, L.; Dannenberg, R.; and Crawford, M. 2007. Tagatune: A game for music and sound annotation. In *ICMIR*, 361–364.

Lenat, D., and Guha, R. 1989. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc*.

Nguyen, T. V.; Liu, S.; Ni, B.; Tan, J.; Rui, Y.; and Yan, S. 2012. Sense beauty via face, dressing, and/or voice. In *ACM Multimedia*, 239–248.

Nguyen, T. V.; Liu, S.; Ni, B.; Tan, J.; Rui, Y.; and Yan, S. 2013a. Towards decrypting attractiveness via multi-modality cues. *TOMCCAP* 9(4):28:1–28:20.

Nguyen, T. V.; Xu, M.; Gao, G.; Kankanhalli, M. S.; Tian, Q.; and Yan, S. 2013b. Static saliency vs. dynamic saliency: a comparative study. In *ACM Multimedia*, 987–996.

Ni, B.; Xu, M.; Nguyen, T. V.; Wang, M.; Lang, C.; Huang, Z.; and Yan, S. 2014. Touch saliency: Characteristics and prediction. *IEEE TMM* 16(6):1779–1791.

Poria, S.; Gelbukh, A.; Cambria, E.; Das, D.; and Bandyopadhyay, S. 2012a. Enriching SenticNet polarity scores through semi-supervised fuzzy clustering. In *IEEE ICDM*, 709–716.

Poria, S.; Gelbukh, A.; Cambria, E.; Yang, P.; Hussain, A.; and Durrani, T. 2012b. Merging SenticNet and WordNet-Affect emotion lists for sentiment analysis. In *IEEE ICSP*, volume 2, 1251–1255.

Poria, S.; Gelbukh, A.; Agarwal, B.; Cambria, E.; and Howard, N. 2013. Common sense knowledge based personality recognition from text. In *Advances in Soft Computing and Its Applications*. Springer. 484–496.

Rajagopal, D.; Cambria, E.; Olsher, D.; and Kwok, K. 2013. A graph-based approach to commonsense concept extraction and semantic similarity detection. In *WWW*, 565–570.

Russell, B. C.; Torralba, A.; Murphy, K. P.; and Freeman, W. T. 2008. Labelme: A database and web-based tool for image annotation. *IJCV* 77(1-3):157–173.

Siorpaes, K., and Hepp, M. 2008. Ontogame: Weaving the semantic web by online games. In *ESWC*, 751–766.

Speer, R. 2007. Open Mind Commons: An inquisitive approach to learning common sense. In *Workshop on Common Sense and Interactive Applications*.

van Zwol, R.; Garcia, L.; Ramirez, G.; Sigurbjornsson, B.; and Labad, M. 2008. Video tag game. In *WWW*.

von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *CHI*, 319–326.

von Ahn, L.; Ginosar, S.; Kedia, M.; Liu, R.; and Blum, M. 2006. Improving accessibility of the web with a computer game. In *CHI*, 79–82.

von Ahn, L.; Kedia, M.; and Blum, M. 2006. Verbosity: A game for collecting common sense facts. In *CHI*, 75–78.

von Ahn, L. 2006. Games with a purpose. *IEEE Computer Magazine* 6:92–94.