Many tasks are trivial for humans but continue to challenge even the most sophisticated computer programs. Traditional computational approaches to solving such problems focus on improving artificial-intelligence algorithms. Here, we advocate a different approach: the constructive channeling of human brainpower through computer games. Toward this goal, we present general design principles for the development and evaluation of a class of games we call “games with a purpose,” or GWAPs, in which people, as a side effect of playing, perform tasks computers are unable to perform.

The Entertainment Software Association (www.theesa.com/facts/gamer_data.php) has reported that more than 200 million hours are spent each day playing computer and video games in the U.S. Indeed, by age 21, the average American has spent more than 10,000 hours playing such games—equivalent to five years of working a full-time job 40 hours per week.

Data generated as a side effect of game play also solves computational problems and trains AI algorithms.

BY LUIS VON AHN AND LAURA DABBISH

Designing Games With A Purpose
People play not because they are personally interested in solving an instance of a computational problem but because they wish to be entertained.
sense facts. However, the Open Mind approach involves two drawbacks: reliance on the willingness of unpaid volunteers to donate their time and no guarantee that the information they enter is correct. GWAPs differ from Open Mind in that they are designed to be enjoyable while ensuring that the data they collect is free from error.

Interactive machine learning. Another area leveraging human abilities to train computers is “interactive machine learning” in which a user provides examples to a machine-learning system and is given real-time feedback as to how well an algorithm is learning. Based on the feedback, the user is able to determine what new examples should be given to the program. Some instances of this approach have utilized human perceptual skills to train computer-vision algorithms to recognize specific objects.

Making work fun. Over the past 30 years, human-computer-interaction researchers have recognized and written about the importance of enjoyment and fun in user interfaces. For example, systems (such as the StyleCam) aim to use gamelike interaction to increase enjoyment and engagement with the software. Many researchers have suggested that incorporating gamelike elements into user interfaces could increase user motivation and the playfulness of work activities. Some projects have taken this notion further, turning the user interface itself into a game. For instance, PSDoom provides a first-person-shooter-style interface for system-administrator-related tasks. The idea of turning work tasks into games is increasingly being applied in children’s learning activities. Researchers note, as we do here, that it is important to not simply slap a gamelike interface onto work activities but to integrate the required activities into the game itself; there must be tight interplay between the game interaction and the work to be accomplished.

Desire to Be Entertained
The GWAP approach is characterized by three motivating factors: an increasing proportion of the world’s population has access to the Internet; certain tasks are impossible for computers but easy for humans; and people spend lots of time playing games on computers. In contrast to other work that has attempted to use distributed collections of individuals to perform tasks, the paradigm we describe here does not rely on altruism or financial incentives to entice people to perform certain actions; rather, they rely on the human desire to be entertained. A GWAP, then, is a game in which the players perform a useful computation as a side effect of enjoyable game play. Every GWAP should be associated with a computational problem and therefore generate an input-output behavior.

A game can be fully specified through a goal players try to achieve (the winning condition) and a set of rules that determines what players can and cannot do during the game. A GWAP’s rules should encourage players to correctly perform the necessary steps to solve the computational problem and, if possible, involve a probabilistic guarantee that the game’s output is correct, even if the players do not want it to be correct.

The key property of games is that people want to play them. We therefore sidestep any philosophical discussions about “fun” and “enjoyable,” defining a game as “successful” if enough human-hours are spent playing it.

We advocate a transformative process whereby a problem is turned into a GWAP. Given a problem that is easy for humans but difficult or impossible for computers, the process of turning the problem into a GWAP consists of first creating a game so that its structure (such as rules and winning condition) encourages computation and correctness of the output. Having created many GWAPs, including the ESP Game, Peekaboom, Phetch, and Verbosity, we explore three game-structure templates that generalize successful instances of human computation games: output-agreement games, inversion-problem games, and input-agreement games.

Output-agreement games. Output-agreement games (see Figure 1) are a generalization of the ESP Game (see the sidebar “The ESP Game and Verbosity” on page 65) to its fundamental input-output behavior:

Initial setup. Two strangers are randomly chosen by the game itself from among all potential players;

Rules. In each round, both are given the same input and must produce outputs based on the input. Game instructions indicate that players should try to produce the same output as their partners. Players cannot see one another’s outputs or communicate with one another; and

Winning condition. Both players must produce the same output; they do not have to produce it at the same time but must produce it at some point while the input is displayed onscreen. When the input is an image and the outputs are keyword descriptions of the image, this template becomes the ESP Game (see Figure 2).

Since the two players cannot communicate and know nothing about each other, the easiest way for both to produce the same output is by entering something related to the common input. Note, however, that the game rules do not directly tell the players to enter a correct output for the given
contributed articles

and Verbosity—they, in their most general form (see Figure 3), can be described through the following rules:

Initial setup. Two strangers are randomly chosen by the game itself from among all potential players;

Rules. In each round, one player is assigned to be the “describer,” and the other player is assigned to be the “guesser.” The describer is given an input. Based on this input, the describer produces outputs that are sent to the guesser. The outputs from the describer should help the guesser produce the original input; and

Winning condition. The guesser produces the input that was originally given to the describer.

Verbosity (see the sidebar) is an inversion-problem game where the input is a word and the outputs are commonsense facts related to that word. Following the input word “milk,” the game might output such facts as “it is white” and “people usually eat cereal with it.”

Although the design of Verbosity involves other game elements not described here, the basic idea is that players need not be asked directly for facts about “milk.” The game is designed such that facts are collected as a side effect of playing. Players told to “please enter facts about milk” might not be motivated to do so or enter incorrect information.

In inversion-problem games, partners are successful only when the describer provides enough outputs for the guesser to guess the original input. If the outputs are incorrect or incomplete, the guesser will not be able to produce the original input. Therefore, the game structure encourages players to enter correct information. At the same time, having one player guess the input while the other describes it is an enjoyable social interaction, similar to the popular children’s game “20 Questions.”

Additional elements can be added to inversion-problem games to increase player enjoyment, including transparency and alternation:

Transparency. In post-game questionnaires, players of inversion-problem games have expressed a strong desire to see their partner’s guesses. We therefore experimented with adding a level of transparency between players so the actions of one would be visible to the other. In games like Verbosity and Peekaboom this transparency is achieved by displaying partner guesses to the describers and allowing them to indicate whether each guess is “hot” or “cold.” This design feature increases the social connection between the players without compromising output correctness.

Alternation. Unlike output-agreement games (where both players continually perform the same task), inversion-problem games are asymmetric in that each player in the pair performs a different task. In some games of this type, one of the two roles involves more interaction or is faster-paced and thus more enjoyable than the other role. In such cases, to balance the game and maintain an equal level of player engagement, player roles can switch after each round; the guesser becomes the describer, and the describer becomes the guesser.

Input-agreement games. Representing a generalization of games like Edith Law’s TagATune (see Figure 4), they can be described through the following rules:

Initial setup. Two strangers are randomly chosen by the game itself from among all potential players;

Rules. In each round, both players are given inputs that are known by the game (but not by the players) to be the same or different. The players are instructed to produce outputs describing their input, so their partners are able to assess whether their inputs are the same or different. Players see only each other’s outputs; and

Winning condition. Both players correctly determine whether they have been given the same or different inputs.

In TagATune, the input is a sound clip, and the output is a series of labels or tags for the clip. The two players achieve the winning condition (and obtain points) only if they both correctly determine whether they have the same input song. Because players want to achieve the winning condition, they each want their partner to be able to determine if their inputs are the same. This means it is in their own best interest to enter accurate outputs that appropriately describe their individual inputs.
It is essential that the number of tasks for players to complete within a given time period is calibrated to introduce challenge and that the time limit and time remaining are displayed throughout the game.

To discourage players from randomly guessing whether their inputs are the same, scoring in input-agreement games strongly penalizes incorrect guesses. One way to do this (while maintaining a positive scoring system) is to give an increasing number of points for streaks of correct answers and zero points for incorrect answers.

Increase Player Enjoyment
Perhaps the most important aspect of GWAP is that the output is produced in a way that’s designed to be enjoyable. As noted with respect to the ESP Game, players are not directly instructed to enter keywords for a given image. Rather, they are told to type what they think their partner is typing. The fact that people enjoy the game makes them want to continue playing, in turn producing more useful output.

It is important to note that the three basic templates defined earlier describe the basic structure of a GWAP; additional game mechanisms must be added to them to increase player enjoyment. For example, much of the previous work describing game-design principles cites challenge as a key aspect of any successful game.11,12,14,20 Challenge translates into game features (outlined by Malone’s11,12) like timed response, score keeping, player skill level, high-score lists, and randomness:

Timed response. Setting time limits for game sessions introduces challenge into a game in the form of timed response.11,12 Players are told to complete a designated number of problem instances within an assigned time limit. If they accomplish it, they may be given extra points for their performance. Timed response is effective for introducing challenge because it establishes an explicit goal that is not trivial for players to achieve if the game is calibrated properly.11,11 We know from the literature on motivation in psychology and organizational behavior that goals that are both well-specified and challenging lead to higher levels of effort and task performance than goals that are too easy or vague.19 It is essential that the number of tasks for players to complete within a given time period is calibrated to introduce challenge and that the time limit and time remaining are displayed throughout the game.

Score keeping. One of the most direct methods for motivating players is by assigning points for each instance of successful output produced during the game. For the ESP Game,12 pairs of players are given points for each image for which they successfully agree on a word (which then becomes a label for the image). Using points increases motivation by providing a clear connection among effort in the game, performance (achieving the winning condition), and outcomes (points).11,12 A score summary following each game also provides players with performance feedback,10 facilitating progress assessment on score-related goals (such as beating a previous game score and completing all task instances within the set time limit).

Player skill levels. Player skill levels, or “ranks,” are another way for game developers to incorporate goal-based motivation into GWAP design. For example, the ESP Game and Peekaboom each have five skill levels players are able to achieve based on the number of points they accumulate. Each newcomer to the game initially has no points and is assigned to the lowest level (“newbie”) then has to earn a certain number of points to advance to the next level.

Following each game session, players are shown their current skill level and the number of points needed to reach the next level.10 Data from the ESP Game indicates that presentation of this skill-level information strongly influences player motivation and behavior. Of the 200,000+ players as of July 2008 with an account on the ESP Game, 42% have scores that fall within 5,000 points of the rank cutoffs. Given that these skill-level point intervals cover less than 2% of the space of possible cumulative scores, the data suggests that many players continue playing just to reach a new rank.

High-score lists. Another method for motivating GWAP play is the use of high-score lists showing the login names and score of the subset of players with the highest number of points over a certain period of time. The score needed by players to be listed on a high-score list varies in terms of difficulty relative to the list’s time period, ranging from highest scores achieved in the past game session over the past hour or week all the way to the history
of the game. For example, an hourly high-score list gives players a specific point total to aim for to get onto the list, as well as relatively quick feedback (within the hour) about their progress toward it. A daily high-score list and all-time high-score list define goals of increasing difficulty. These multi-level goals, varying in difficulty, provide strong, positive motivation for extended game play—and related data generation.

**Randomness.** GWAPs should also incorporate randomness. For example, inputs for a particular game session are typically selected at random from the set of all possible inputs, and players are randomly paired to prevent cheating.

Because inputs are randomly selected, their difficulty varies, thus keeping the game interesting and engaging for expert and novice players alike. It also means that every game session involves uncertainty about whether all inputs will be completed within the time limit, adding to the challenge experienced by players.

Random partner assignment also ensures the uniqueness of each game session. Anecdotal evidence from the ESP Game suggests that during each game session players develop a sense of their partners’ relative skill, a perception that affects their joint performance. The feeling of connection that players can get from these games is one of the factors that motivates repeated play.

**Output Accuracy**

Additional mechanisms must be added to GWAPs beyond the basic template structure to ensure output correctness and counter player collusion. For example, players of the ESP Game might try to circumvent the game’s built-in verification mechanism by agreeing prior to the game that for every image they will always type the letter “a”; in this case, they would always match each other, and incorrect data would therefore be entered into the system. We describe generally applicable mechanisms in the following sections that have proved successful in guarding against player collusion and guaranteeing the correctness of the computation across all game templates.

**Random matching.** GWAPs are meant to be played by hundreds, if not thousands, of people at once, most in distributed locations. Players paired or grouped randomly have no way of knowing their partner’s identity so have no easy way to agree ahead of time on any cheating strategy. Thus, under random matching, the probability of two or more cheaters using the same strategy being paired together should be low.

**Player testing.** Games may randomly present players inputs for which all possible correct outputs are already known. For them, if the output produced by a particular player does not match the known correct outputs, the players should be considered suspicious, and none of their results should be trusted. Depending on the number of “test” inputs presented to players, this strategy can guarantee with high probability that the output is correct. To illustrate, assume half of the inputs given to a player are test inputs. The probability is thus that a new output by the player is correct, given of course that the player is correct on all the test inputs at least 50% of the time, a probability that can be increased through repetition.

**Repetition.** A game should be designed so it does not consider an output correct until a certain number of players have entered it. This strategy for determining correctness enables any GWAP to guarantee correct output with arbitrarily high probability. As an example, consider an output-agreement game; if for a given input the game accepts an output as correct only after pairs have entered it, and the game itself knows that each of these pairs entered a correct output with at least 50% probability (as a result of player testing), then the output is correct with probability of at least \(1 - \frac{1}{2^n}\).

**Taboo outputs.** For problems in which many different outputs can be associated with one input (such as labeling images with words), ensuring sufficient coverage of the output space is an important consideration. The use of “taboo,” or off-limits, outputs provides some guarantee that a larger proportion of all possible outputs will be entered by all players. Taboo outputs are known correct outputs displayed onscreen during game sessions that players are not allowed to enter. They can be taken from correct outputs gen-
erated in previous rounds of the game itself. It is important for the game's designer to randomize which taboo outputs are presented in order to account for potential output-priming effects (in which the particular taboo outputs shown to the players influence the guesses they enter) and ensure wide coverage of all potential outputs for a given input.

Other Design Guidelines

The general schemes we’ve presented here for designing GWAPs rely on the participation of two players per game session. Now we show that the games can be modified to accommodate single or more than two players.

Prerecorded games. Paired game play makes GWAPs social, meaning that players are able to validate each other’s computation. However, two-player games present logistical challenges. For instance, there may be times when an odd number of people want to play a particular game, meaning at least one of them cannot play. In addition, when a game is just beginning to gain popularity, it is difficult for game administrators to guarantee that many people will be able to play at the same time. We thus recommend that game developers apply a technique—prerecorded game play—introduced by the ESP Game.22 A dyadic game, normally played by multiples of two players, can be transformed into a single-player game by pairing a single player with a prerecorded set of actions.

In the case of an input-agreement game or output-agreement game (such as the ESP Game), implementing automated players is relatively easy. When two people are playing, the game should simply record every action they make, along with the relative timing of each action. Then, when a single player wishes to play, the system can pair that single player with a prerecorded set of moves.

In inversion-problem games, implementing prerecorded game play is more complex because one of the players (the guesser) must dynamically respond to the other (human) player’s actions. Peekaboom, Phetch, and Verbosity have each implemented a single-player version using techniques customized to each game.23–25

More than two players. The three GWAP templates can be extended to include more than two players; for example, output-agreement games can be extended to incorporate more players by modifying the winning condition such that the first two players who agree on the output are the winners of the round (and granted a higher number of points than the nonwinners). Similarly, the template for inversion-problem games can be extended to incorporate multiple players by substituting an individual guesser with an arbitrary number of players in the role of guesser, all racing to be first to correctly guess the input (winning condition).

These extensions change the nature of the games considerably. Whereas the two-player versions of each template are cooperative in nature (players work together to obtain points), the multiplayer versions are competitive. Cooperative, as well as competitive, games involve advantages and disadvantages. For certain players, competitive games may be more enjoyable than

A Sampling of GWAPs

The ESP Game and Verbosity

Two of the most popular GWAPs—ESP and Verbosity—can be played online at www.gwap.com.

The ESP Game has generated millions of labels for random images located throughout the Web. In it, two players are randomly paired for two-and-a-half minutes as they are shown a series of images to label. The game does not directly ask them to label the images. Rather, both players must try to enter the same word as their partner for each image on the screen; neither player can see the partner’s words. When both players agree on a word, each is given a new image. The goal is to agree with the partner on words for as many images as possible. The words the players agree on for each image are extremely accurate labels that can be used to improve image search throughout the Web. To increase the quality of these labels, as well as to motivate player engagement, the game forbids the use of “taboo words” from being entered. In the screenshot (see Figure a), players cannot use the words “dog” or “pillow” when trying to agree on a word with their partner.

Verbosity is a word-guessing game in which two players alternate roles. The describer is given a secret word the guesser must figure out as quickly as possible. The describer helps the guesser by providing clues about the secret word using sentence templates that must be completed without using the secret word itself. In the example here (see Figure b), the secret word is “sock,” and the sentence template “It is a kind of ______” has been instantiated to the clue “It is a kind of clothing.” The describer sees all of the guesser’s inputs and indicates which ones are “hot” and which are “cold.” The computational purpose of the game is to collect a database of commonsense facts about the secret words (such as “Sock is a kind of clothing”).

![Figure a: Players of the ESP Game try to guess what their partner is typing on each image.](image)

![Figure b: Players of Verbosity enter commonsense facts to help their partner guess a secret word.](image)
contributed articles

their cooperative counterparts. On the other hand, having more players work on the same input is wasteful in terms of “computational efficiency,” an important criterion for evaluating the utility of a given game.

GWAP Evaluation

How might a game’s performance be judged successful? Given that two different GWAPs solve the same problem, which is best? We describe a set of metrics for determining GWAP success, including throughput, lifetime play, and expected contribution.

Game efficiency and expected contribution. If we treat games as if they were algorithms, efficiency would be a natural metric of evaluation. There are many possible algorithms for any given problem, some more efficient than others. Similarly, many possible GWAPs are available for any given problem. In order to choose the best solution to a problem we need a way to compare the alternatives in terms of efficiency. Efficiency of standard algorithms is measured by counting atomic steps. For instance, QuickSort is said to run in $O(n \log n)$ time, meaning it sorts a list of $n$ elements in roughly $n \log n$ computational steps. In the case of GWAPs, the notion of what constitutes a computational step is less clear. Therefore, we must be able to define efficiency through other means.

First, we define the throughput of a GWAP as the average number of problem instances solved, or input-output mappings performed, per human-hour. For example, the throughput of the ESP Game is roughly 233 labels per human-hour. This is calculated by examining how many individual inputs, or images, are matched with outputs, or labels, over a certain period of time.

Learning curves and variations in player skill must be considered in calculating throughput. Most games involve a certain type of learning, meaning that with repeated game sessions over time, players become more skilled at the game. For the game templates we described earlier, such learning can result in faster game play over time. To account for variance in player skill and changes in player speed over time as a result of learning, we define throughput as the average number of problem instances solved per human-hour. This average is taken over all game sessions through a reasonably lengthy period of time and over all players of the game.

Games with higher throughput should be preferred over those with lower throughput. But throughput is not the end of the story. Because a GWAP is a game, “fun” must also be included. It does not matter how many problem instances are addressed by a given game if nobody wants to play. The real measure of utility for a GWAP is therefore a combination of throughput and enjoyability.

Enjoyability is difficult to quantify and depends on the precise implementation and design of each game. Even seemingly trivial modifications to a game’s user interface or scoring system can significantly affect how enjoyable it is to play. Our approach to quantifying this elusive measure is to calculate and use as a proxy the “average lifetime play” (ALP) for a game. ALP is the overall amount of time the game is played by each player averaged across all people who have played it. For instance, on average, each player of the ESP Game plays for a total of 91 minutes.

“Expected contribution” is our summary measure of GWAP quality. Once a game developer knows on average how many problems are solved per human-hour spent in the game (throughput) and how much time each player can be expected to spend in a game (ALP), these metrics can be combined to assess each player’s expected contribution. Expected contribution indicates the average number of problem instances a single human player can be expected to solve by playing a particular game. Developers can then use these metrics to determine the broadest set of outputs. The game templates we have developed thus far have focused on similar problem-template fit, that is, whether certain templates are more suited for some types of computational problems than others.

The game templates we have developed thus far have focused on similarity as a way to ensure output correctness; players are rewarded for thinking like other players. This approach may not be optimal for certain types of problems; in particular, for tasks that require creativity, diverse viewpoints and perspectives are optimal for generating the broadest set of outputs. Developing new templates for such tasks could be an interesting area to explore.

We would also like to understand what kinds of problems, if any, fall outside the GWAP approach. The games...
we have designed so far have focused on problems that are easily divided into subtasks. The “bite-size” nature of these games adds to their popularity and appeal to casual gamers in particular, since such players typically go for games they can play “just one more time” without having to make too much of a time commitment.

The GWAP approach represents a promising opportunity for everyone to contribute to the progress of AI. By leveraging the human time spent playing games online, GWAP game developers are able to capture large sets of training data that express uniquely human perceptual capabilities. This data can contribute to the goal of developing computer programs and automated systems with advanced perceptual or intelligence skills.

Acknowledgment

We would like to thank Manuel and Lenore Blum, Mike Crawford, Shiry Ginosar, Severin Hacker, Susan Harsh-enko, Mihir Kedia, Edith Law, Bryant Lee, and Roy Liu for their help in this research.

References


This work was partially supported by National Science Foundation grants CCR-0122581 and CCR-0085082 (ALADDIN) and generous gifts from Google, Inc., and the Heinz Endowment. Luis von Ahn was partially supported by a Microsoft Research Graduate Fellowship, a Microsoft Research New Faculty Fellowship, and a MacArthur Fellowship.

Luis von Ahn (lava@andrew.cmu.edu) is an assistant professor in the Computer Science Department at Carnegie Mellon University, Pittsburgh, PA.

Laura Dabbish (dabbish@andrew.cmu.edu) is an assistant professor of information technology and organizations in the Heinz School, with a joint appointment in the Human-Computer Interaction Institute in the School of Computer Science at Carnegie Mellon University, Pittsburgh, PA.

© 2008 ACM 0001-0782/08/0800 $5.00