

Keeping Bots out of Online Games

Philippe Golle
Palo Alto Research Center
3333 Coyote Hill Rd
Palo Alto, CA 94304 USA
pgolle@parc.com

Nicolas Ducheneaut
Palo Alto Research Center
3333 Coyote Hill Rd
Palo Alto, CA 94304 USA
nicolas@parc.com

ABSTRACT

We study the problem of restricting participation in online games to human players, so they can enjoy the game without interference from automated playing agents known as bots. We propose a range of techniques, both software and hardware based, to distinguish bots from human players in a wide variety of online games, from poker to “shoot’em ups.”

1. INTRODUCTION

Multiplayer computer games have become an increasingly important economic, social, and cultural phenomenon: nowadays millions of players routinely gather online to play their game of choice. Online game genres are extremely diverse: First-Person Shooters (FPS), Massively Multiplayer Online Role-Playing Games (MMORPGs), and online card games (e.g. poker) are among the most popular. However all share a similar characteristic: they pit *human* players against one another, unlike earlier forms of computer gaming where most opponents were computer-controlled. The inherent limitations of human physical and mental capabilities, the diverse ways these limitations manifest themselves in individuals, and the ability to transcend these limitations are all essential to the enjoyment of a multiplayer game.

Yet some players do not accept the rules of the game and refuse to play only to the best of their abilities. Instead, they use various forms of automation to gain an unfair advantage. The use of these so-called “bots” (or automated player agents) has become more prevalent recently because of the increasingly porous barriers between real and virtual economies [5]. For instance, writer Julian Dibell recently told Wired he earned \$3,917 in one month by buying and reselling cybergoods from Ultima Online on eBay [3]. The process of collecting artifacts and/or in-game currency in MMORPGs can easily be automated – a practice referred to as “farming” that can lead to substantial monetary gains. In online card rooms, bots can be used to play games entirely based on their statistical properties, thereby earning money against imperfect human players [2]. Finally in FPS, bots

can be used to increase a player’s performance, for instance by artificially increasing targeting accuracy (CounterStrike’s “aimbot” is one of the most blatant examples, see [6]).

The problem of bot participation is not a superficial symptom of the limitations of existing games. Rather, it is an inherent consequence of the fact that repetition is a fundamental component of games [10]. Mastering a “pattern” is an important source of enjoyment, and games with infinite variations are perceived as “chaotic” or “noisy” and poorly received by players. Thus bots cannot be eliminated only by designing richer, more diverse games (too much repetition attracts bots, too little drives humans away). Instead, techniques are needed to distinguish bots from human players so that bots can be eliminated from games.

The problem of distinguishing human players from bots in online games may not at first sound very difficult. After all, there is a clear gap between the abilities of humans and computers (e.g. bots can not carry coherent sustained conversations with humans). For our purposes however, identifying this gap is not enough. We must exploit it in games to distinguish bots from human players in a way that is *cost-effective, immune to cheating and that preserves the human players’ enjoyment of the game*. These three requirements are not easily satisfied together and the best solution often depends on the game.

We propose two broad approaches to keep bots out of online games. The first consists of seamlessly integrating software-based tests to tell humans and computers apart, known as CAPTCHAs [1, 4], into online games. Our second contribution is to propose hardware instantiations of CAPTCHA tests. Hardware CAPTCHAs are extremely versatile and suitable for use in a wide variety of online games, from card and board games to “shoot’em ups.”

2. MODEL

We assume throughout this paper that bots are undesirable, but a disclaimer is in order: not all uses of bots are malicious or detrimental to the quality of the game. In fact, the use of bots is sometimes an integral part of the games. For instance Second Life (<http://www.secondlife.com>), a virtual world focused on content creation, provides extensive scripting and automation capabilities so players can easily create new content. Naturally, our techniques should only be applied to eliminate problematic instances of bot uses. The decision of which bot uses are allowed and which are

forbidden should be made by the game designers and/or the players. Our contribution is to offer tools to enforce these decisions. More precisely, our techniques help enforce one of two properties: human presence or human play.

Human presence. A technique enforces “human presence” in a game if some input from a human is required to play the game. The requirement is only that a human contributes *some of* the interaction with the game, not necessarily all of it. In particular, the human may be assisted by a bot that controls the rest of the interaction with the game. In online poker for example, human presence means that a bot can only play under the supervision of a human. The cards may be chosen by the bot according to probabilities computed by the bot, but the bot cannot play without a human “baby-sitter”. While human presence may seem like a weak property, it is actually surprisingly useful. Consider that the scope of bot involvement in online poker games, FPS or MMORPGs would presumably decrease considerably if every single bot had to be supervised by a human player.

Human play. Human play is a stronger property that requires that *all* interaction with the game software come from a human, without any involvement from a bot.

Adversarial model. We assume that the game software is tamper proof and that bots can only interact with the game via the input and output interfaces defined by the game. These assumptions are justified for online multiplayer games since the game software resides on secure game servers under the control of a game operator. We make the standard assumption that the adversary is rational, and derives no utility from the *intrinsic* use of a bot (no defense appears possible against an adversary who has more intrinsic utility for using a bot than for winning the game).

3. CAPTCHA TESTS

There exists a variety of automated tests for telling humans and computers apart in an online environment. These tests are known collectively as CAPTCHA tests [1], for “Completely Automated Public Turing Test to Tell Computers and Humans Apart.” A CAPTCHA test is a program that can generate and grade tests that most humans can pass but that current computer programs cannot pass. The most widely used CAPTCHA tests rely on humans’ ability to recognize randomly distorted text or images [4]. Even the best computer programs fail these tasks with high probability.

CAPTCHA tests can be used during a game to verify *human presence*. A wide range of testing strategies is possible. A single CAPTCHA challenge may be presented when a player joins a game, or else repeated CAPTCHA challenges may be presented to the player at random intervals over the course of a game. The game may allow the player to sometimes not reply, or provide an incorrect answer. A player that fails one or several challenges may be suspended from the game, either for a limited time or permanently (in which case all progress made in the game is lost). CAPTCHA tests deter bot participation most effectively in long stateful games where players have a lot to lose if they fail a test. Games played in short, stateless rounds may require more frequent tests.

We call this use of CAPTCHAs “out-of-band” because the CAPTCHA test is not integrated within the world of the game, but occurs outside of it. The main advantage of out-of-band CAPTCHAs is that they are extremely flexible and can be used with just about any game. However, out-of-band CAPTCHA tests suffer from the following limitations:

CAPTCHA tests are *disruptive*: they draw the player’s attention away from the game. The interruption is only momentary (it takes only a few seconds for a human to solve a CAPTCHA) but breaks the suspension of disbelief that is so important to gaming. CAPTCHAs can also adversely affect the pace of a game: frequent requests to solve CAPTCHAs in the middle of a game would be intolerable to human players, since pacing and a sense of “flow” is essential to the enjoyment of a game [10]. We address this issue in Section 3.1.

CAPTCHA tests can be *outsourced*: a CAPTCHA must be solved by a human, but that human need not be the person playing the game. For example, a bot playing a game without human assistance may forward the CAPTCHA tests it receives to low-cost workers that specialize in solving them (outsourcing boring game tasks to low-cost workers is already a reality: see [7]). Traditional digital CAPTCHAs may thus not be very effective to prevent bot participation in online games. We address this issue in Section 4.2.

3.1 Embedding CAPTCHAs Into Games

With a little design effort, CAPTCHA tests can be embedded almost seamlessly into games. We illustrate this approach with the example of in-game resource collection or creation in MMORPGs. It would be consistent with the game’s environment to require players to obtain “licenses” before allowing them to perform certain actions. Licences would take the form of an “exam” that is a CAPTCHA in disguise. For example, a licence to craft items may require players to prove that they can “read engineering blueprints” (that is, solve a text-based CAPTCHA); a license to hunt may require players to identify an animal in a picture, etc.

Like out-of-band CAPTCHAs, licenses allow for a broad range of testing frequency and strategy. CAPTCHAs that are tightly integrated into, and consistent with, the game’s environment do not negatively affect suspension of disbelief and are also less likely to be outsourced.

Applicability. Embedding CAPTCHAs into games is a suitable approach to ensure human presence in games that are slow-paced, have high-entropy environments and rules that are not too specific (most MMORPGs satisfy these requirements). A slow-paced game ensures that players have enough time to solve embedded CAPTCHAs. A high-entropy environment, such as a rich graphical world, allow embedded CAPTCHAs to blend naturally into the environment. Finally, the absence of very specific rules makes it possible to embed CAPTCHAs that are not noticeably out of place.

4. PHYSICAL CAPTCHAS

We propose hardware-based “physical” CAPTCHA tests that distinguish human players from bots based on humans’ ability to interact with the physical world. Human players can easily perform physical actions such as pressing a button, moving a joystick or tapping on a touch-sensitive screen,

but it is difficult and expensive to design hardware that automates these physical tasks. Physical CAPTCHAs offer several advantages over the digital CAPTCHAs of the previous section: they are harder to outsource, they can be less intrusive and can offer the stronger property of *human play*.

A physical CAPTCHA is a device, such as a keyboard or a joystick, that accepts physical inputs (e.g. motion, pressure) and produces digital outputs. The property desired of a physical CAPTCHA device is that the only way to learn the digital output corresponding to a given input at a given time is to physically produce that input at that time. To guarantee this property, a physical CAPTCHA device must be *tamper-proof* and *authenticate* its output.

Authentication. The output of the physical CAPTCHA must be authenticated, either to a remote game server or to the game software running on the user’s PC or game console, to prove that it comes from the device at a given time.

Tamper-proof. A tamper-proof CAPTCHA device immediately loses its ability to authenticate outputs if an attempt is made to tamper with it (e.g. the authentication key is wiped out from memory). This ensures that inputs can only be submitted to the physical CAPTCHA device via its physical input interface. Any attempt to bypass that interface (opening the device, rewiring its microcontrollers) causes the loss of the authentication key.

4.1 CAPTCHA Input Devices

In this section, we consider turning existing game input devices into CAPTCHA devices. Joysticks and other small game input devices can be rendered tamper-proof at little cost, and adding authentication capability is also inexpensive. Larger devices, such as keyboards, would be comparatively more expensive to make tamper-proofs (we propose an alternative solution for keyboard-based games in the next section). We call a tamper-proof joystick that authenticates its communication with the game software a “CAPTCHA joystick”. Games played with a CAPTCHA joystick ensure that there is no other way for a player to send inputs to the game than by physical action on the joystick (pressing, pushing, etc.) A human can exert this physical action much faster and much more reliably and easily than a machine.

CAPTCHA input devices ensure not only *human presence*, but also the stronger property of *human play*, since every interaction with the game software is mediated via the CAPTCHA input device. CAPTCHA input devices are well adapted to a wide variety of games and FPS in particular: they guarantee that “aimbots” and other artificial performance-enhancing techniques cannot be used. We believe that we will see more and more authenticated, tamper-proof CAPTCHA joysticks in the future: FPS games are progressively becoming akin to spectator sports [8], with regular tournaments offering large prize pools - which greatly increases the importance of ensuring that humans, not bots, are playing.

Limitations. CAPTCHA input devices combine two functionalities: 1) they record the player’s input stream and relay it to the game software and 2) they ensure that the input stream comes from a human. While very well adapted

to a number of games, this combination can also sometimes be unwieldy. For example, board or card games are more naturally played with a keyboard than with a joystick, but designing a tamper-proof, authenticated keyboard would be expensive and difficult, given the size of a keyboard. Consider also that there is a tremendous amount of “legacy” game hardware in use, and that users might balk at the cost of upgrading to a new CAPTCHA joystick. These examples show the need, in current circumstances, for separating the task of recording a player’s input from the task of verifying that the player is human. We therefore propose a cheaper, more versatile alternative to CAPTCHA input devices: the CAPTCHA token.

4.2 CAPTCHA Token

The only function of a “CAPTCHA token” is to test that a player is human. A CAPTCHA token does not serve as an input device to a game. CAPTCHA tokens are generic, versatile and cheap, and can be used in combination with just about any game. However, unlike CAPTCHA joysticks, they ensure only *human presence*, not *human play*.

CAPTCHA tokens can take different form factors. We describe an implementation that takes the form factor of a small calculator, equipped with a keypad and a screen (see Figure 1). The CAPTCHA token also has a clock (which need not be very accurate) and a CPU. We assume that the CAPTCHA token is tamper-proof, such that the only way to enter data is by physically depressing the keys on the keypad. We further assume that it would be difficult and expensive to design a mechanical machine that taps the keypad of the token in the right location. We estimate that CAPTCHA tokens would be inexpensive to manufacture in bulk (a few dollars each).



Figure 1: A prototype CAPTCHA token

The token and the game server share a long (infinite) sequence of secret numbers s_1, s_2, \dots (each a few digits long). At time t (e.g. measured in minutes), the token authenticates itself to the game server with the value s_t . We assume that knowledge of the values s_1, \dots, s_t does not help an adversary learn anything about the value s_{t+1} . Standard cryptographic techniques allow such sequences to be generated and stored very efficiently: we may define $s_i = E_k(i)$ where E is a symmetric cipher (such as AES) and k a key known to the CAPTCHA token and the game server.

The CAPTCHA token can be in one of two states: active or inactive. When active, the token displays on its screen the value s_t corresponding to the current time t (expressed, say, in minutes). This value is updated every time the timer t is incremented. When inactive, the token does not display s_t but instead displays a random value c , which we call a challenge. To activate an inactive token, the user must type

the challenge c on the keypad. If the correct value is entered, the token becomes active for a short period of time (say, one minute). After that, the token generates a new challenge c' at random, displays it on its screen, and automatically returns to the inactive mode. The new challenge must be typed to return the token to the active mode.

Use in Games. When a player signs up with an online game provider, the provider sends the player a CAPTCHA token. While playing the game, the player is occasionally asked for the current secret s_t . The player activates the token (which requires manually typing the challenge into the token's keypad), reads the value s_t off the screen of the token, then sends this value to the game server (via the network connection of the game).

Comparison with other authentication tokens. The particular implementation of a CAPTCHA token that we have described bears a superficial resemblance to tokens used in two-factor authentication, such as RSA's pinpad [11], but should not be confused with them. These two types of tokens offer completely different functionalities: tokens used for two-factor authentication let users prove knowledge of a secret, whereas CAPTCHA tokens let users prove that they are human (users need not remember any secret).

4.3 Properties of CAPTCHA Tokens

CAPTCHA tokens require an interruption on the part of the player and thus appear most suitable for slow-paced games with very specific rules and low-entropy inputs and outputs, such as card and board games (chess, poker, etc). These are precisely the games that are least amenable to the "CAPTCHA joystick" technique of Section 4.1. Thus CAPTCHA tokens and CAPTCHA joysticks appear to complement one another very well. In simulation/fantasy games such as current MMORPGs, the token could be used as part of the "licensing" process we described in Section 3.1, thereby preserving suspension of disbelief while offering even stronger guarantees that the player is human. Note also that since CAPTCHA tokens are hard to outsource (see below), verifications can be fairly infrequent (once a day or even once a week might suffice), which greatly reduces disruptions of the player's gaming experience.

Outsourcing. CAPTCHA tokens are difficult to outsource because they rely on a physical rather than digital challenge. For one thing, shipping a physical object requires a higher degree of trust than forwarding a digital challenge. Furthermore, it would be difficult to ensure consistent and reliable access to an outsourced token, since no redundancy is possible. The token occupies only one physical location, which may become unavailable for any number of reasons (the workers at that location may be asleep, or busy, etc.) Finally, it would be a logistical challenge to process many tokens in one location: storing hundreds or thousands of tokens in a way that allows for fast access (and without losing or misplacing any token) is hard. In summary, CAPTCHA tokens cannot be outsourced nearly as efficiently as digital CAPTCHAs that can be outsourced at virtually no cost to an anonymous, changing crowd of low-cost workers.

Automating the physical input. With the help of a webcam and optical character recognition (OCR) software, an

adversary can read the current activation challenge c . Automatically typing the value c is harder: it requires customized hardware to tap the keys of the token in the right sequence. Attacks that require hardware are typically much harder to propagate than software attacks, since hardware cannot be downloaded but requires physical shipping.

If however hardware for automatically typing challenges on a token's keypad became sufficiently cheap and available, a simple counter-measure would be to replace the keypad of the token with a touch-sensitive area that could be overlaid on the token's screen (much like the screen of a PDA). Compared to a keypad, a touch-sensitive screen offers a much broader method of data entry. For example, the token could draw a curve on the screen and ask the human to track that curve with a stylus. While humans can easily do so, automated hardware to do the same would be very costly.

5. CONCLUSION

The use of bots to assist or replace human players in multiplayer online games is becoming increasingly problematic. It is difficult to develop techniques to prevent bot use that are simultaneously cost-effective, immune to cheating, applicable to a wide variety of online game genres, and most importantly that preserve the human player's enjoyment of the game. We argue in favor of hardware-based bot detection mechanisms: the CAPTCHA joystick and CAPTCHA token. We believe that, as multiplayer gaming gains economic and social importance, hardware-based bot prevention techniques such as our CAPTCHA token will become increasingly prevalent.

6. REFERENCES

- [1] L. von Ahn, M. Blum, N.J. Hopper and J. Langford. CAPTCHA: Using hard AI problems for security. In *Proc. of Eurocrypt '03*, pp. 294–311.
- [2] Are poker 'bots' raking online pots? <http://www.msnbc.msn.com/id/6002298/>
- [3] Boring game? Outsource it. <http://www.wired.com/news/print/0,1294,64638,00.html>
- [4] The CAPTCHA project. <http://www.captcha.net>
- [5] E. Castranova. On virtual economies. *Game Studies*, 3 (2). (2003)
- [6] Cheating: Multiplayer gaming Achille's heel? <http://www.tomshardware.com/game/20030517/cheating-03.html>
- [7] J. Dibbell. Serfing the Web: Black Snow Interactive and the world's first virtual sweat shop. <http://www.juliandibbell.com/texts/blacksnow.html>
- [8] Gaming as spectator sport. In *Gamespot*, January 19, 2003. <http://www.gamespot.com/gamespot/features/all/gamespotting/011903/4.html>
- [9] R. Koster. Raph Koster's writings on game design. <http://www.legendmud.org/raph/gaming/index.html>
- [10] R. Koster. A theory of fun for game design. Paraglyph Press.
- [11] RSA SecurID pinpad. <http://www.rsasecurity.com/>