UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

META PLATFORMS, INC.,
Petitioner

v.

MULLEN INDUSTRIES LLC,
Patent Owner.

Case IPR2025-00745
U.S. Patent No. 9,662,582 B2
Issue Date: May 30, 2017

Title: SYSTEMS AND METHODS FOR LOCATION BASED GAMES AND EMPLOYMENT OF THE SAME ON LOCATION ENABLED DEVICES

PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 9,662,582 B2

Table of Contents

Page

Table of Contents
(continued)

Page

Table of Contents
(continued)

Page

List of Exhibits

Exhibit No.	Description of Document
1001	U.S. Patent No. 9,662,582 B2 to Jeffrey David Mullen (filed Aug. 20, 2007, issued May 30, 2017) ("**'582**" or "**'582 patent**")
1002	Declaration of Jeremy Cooperstock, Ph.D. ("**Cooperstock**")
1003	U.S. Patent App. Pub. No. 2003/0177187 A1 to David A. Levine et al. (filed Feb. 20, 2003; published Sept. 18, 2003) ("**Levine**")
1004	'582 Patent File History
1005	Proof of Service of Complaint
1006	*Mullen Industries LLC v. Meta Platforms, Inc.*, Case No. 1:24-cv-354-DAE, Dkt. 64, Order (1) Adopting Report And Recommendation And (2) Granting In Part And Denying In Part Defendant's Partial Motion To Dismiss (W.D. Tex. Jan. 29, 2025)

I. MANDATORY NOTICES UNDER §42.8(A)(1)

A. Real Party-In-Interest under §42.8.(b)(1)

Meta Platforms, Inc. is the real party-in-interest to this IPR petition.

B. Related Matters under §42.8(b)(2)

The '582 patent was the subject of pending litigation involving Petitioner: *Mullen Industries LLC v. Meta Platforms, Inc.*, Case No. 1:24-cv-00354-DAE (W.D. Tex.). Petitioner was served on April 5, 2024. (EX1005, p.003.) The '582 patent was dismissed with prejudice from the pending litigation against Petitioner on January 29, 2025. (EX1006, pp.009-013.)

Petitioner is filing IPR petitions against twelve related patents asserted by Patent Owner in the pending litigation involving Petitioner:

U.S. Patent No.	IPR Case
8,585,476 B2	IPR2025-00737
9,744,448 B2	IPR2025-00738
10,179,277 B2	IPR2025-00739
10,828,559 B2	IPR2025-00740
10,967,270 B2	IPR2025-00702
11,033,821 B2	IPR2025-00703
11,376,493 B2	IPR2025-00741
11,904,243 B2	IPR2025-00742
11,947,716 B2	IPR2025-00743
12,019,791 B2	IPR2025-00744
9,662,582 B2	IPR2025-00745

U.S. Patent No.	IPR Case
10,974,151 B2	IPR2025-00746

C. Lead and Back-Up Counsel under §42.8(b)(3)

Petitioner provides the following designation of counsel.

LEAD COUNSEL	BACK-UP COUNSEL
Heidi L. Keefe (Reg. No. 40,673) hkeefe@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington, DC 20004 Tel: (650) 843-5001 Fax: (650) 849-7400	Phillip E. Morton (Reg. No. 57,835) pmorton@cooley.com Andrew C. Mace (Reg. No. 63,342) amace@cooley.com Mark R. Weinstein (Admission *pro hac vice* to be requested) mweinstein@cooley.com Lowell D. Mead (Admission *pro hac vice* to be requested) lmead@cooley.com Patrick Lauppe (Admission *pro hac vice* to be requested) plauppe@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington D.C. 20004

D. Service Information

This Petition is being served by Federal Express to the attorney of record for the ’582 patent, 32733 - JEFFREY D. MULLEN, 2212 Hassinger Lane, Glenshaw,

PA 15116. Petitioner consents to electronic service at the addresses provided above for lead and back-up counsel.

II. FEE PAYMENT

Petitioner requests review of four claims, with a $51,875 payment.

III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND CONSIDERATIONS UNDER §§ 314(A) AND 325(D)

A. Grounds for Standing

Petitioner certifies that the '582 patent is available for IPR and that Petitioner is not barred or otherwise estopped.

B. Identification of Challenge and Statement of Precise Relief Requested

Petitioner requests IPR institution based on:

Ground	Claims	Basis for Challenge under §103
1	1, 2, 11, and 13	Obvious over **Levine**

Submitted with this Petition is the Declaration of Jeremy Cooperstock, Ph.D. (**EX1002**) ("Cooperstock"), a qualified technical expert. (EX1002, ¶¶1-18, App. A.)

C. Considerations Under §§ 314(a) and 325(d)

§314(a): The *General Plastic* factors are not relevant as this is the first and only IPR petition filed by Petitioner with respect to the '582 patent.

Nor do the *Fintiv* factors support discretionary denial under §314(a). First, as noted above, there is no parallel litigation involving Petitioner and the '582 patent

because the '582 patent was dismissed with prejudice from the pending litigation against Petitioner on January 29, 2025. (EX1006, p. 009-013.) Second, in addition to the dismissal of the '582 patent, the pending litigation is in a relatively early stage, with little substantive discovery having taken place to date and claim construction proceedings ongoing. Trial is scheduled to start October 5, 2026—over eighteen months from now—and a motion to transfer the case by Petitioner to the Northern District of California is currently pending, which if granted would likely result in an even later trial date. Petitioner also intends to move to stay the litigation pending resolution of the IPRs. Additionally, to allay concerns over duplication of efforts, Petitioner represents that, in the event of institution, it will not pursue in litigation any invalidity defense that was or could have been raised in IPR.

§325(d): This Petition does not raise a § 325(d) issue because the Levine prior art reference was not cited or otherwise identified during prosecution.

Petitioner reserves the right to address and respond to any assertions that Patent Owner may raise regarding discretionary factors.

IV. OVERVIEW OF THE PATENT

A. Level of Ordinary Skill

A person of ordinary skill would have possessed a bachelor's degree in electrical engineering, computer science, or similar field, with two years combined experience in designing and/or developing interactive location-based computer

systems/software, such as video games or other simulations incorporating location information. A person could also have qualified as a person of ordinary skill in the art with some combination of (1) more formal education (such as a master's of science degree) and less technical experience, or (2) less formal education and more technical or professional experience. (Cooperstock, ¶¶21-25.)

B. Specification Overview

The '582 patent purports to provide "an actual, reality-based video game in which a user's physical (actual) location on a playfield, reflects a virtual game character's virtual location in a video game environment." ('582, 2:28-33.) A user may play the game on devices such as a "wireless telephone" or other devices such as "laptops, Personal Digital Assistants (e.g., PDAs), Blackberries, Personal Electronic Device (PED), iPODs, or any other portable electronic device." ('582, 7:21-27.) Each player may choose from one of a variety of "characters" to play. ('582, 11:51-57.) In addition to human players who control characters, the game may include "software-controlled gaming characters (e.g., video game characters with artificial intelligence)." ('582, 14:48-51.)

V. CLAIM CONSTRUCTION

Petitioner does not believe express claim construction is necessary at this time. The prior art cited herein, as demonstrated below, renders the challenged

claims obvious under any reasonable construction. Petitioner therefore respectfully submits that, for purposes of this IPR, express constructions are not required.

VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Overview of Ground

This petition relies upon Levine (**EX1003**), which discloses and/or renders obvious every element of the challenged claims. Levine teaches a location-based online multi-player game with the same relevant features as the '582 patent, described in terms that are strikingly similar to the '582 patent's description. Much like the '582 patent, Levine describes "a distributed, platform-sensitive, location-based, contextual system, method and computer program product for bridging activities in real and virtual environments within the context of multi-user gaming" and other applications. (Levine, ¶0163.)[1] Levine's online game system "accounts for both the physical and virtual location and context of the participating devices and people" to serve "the purpose of coordinating activities in the real (i.e., physical) and virtual worlds." (Levine, ¶0158.)

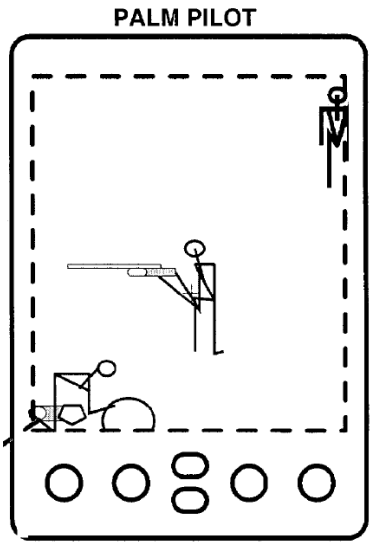
Like the '582 patent, Levine describes that a user may play its online game with a wireless telephone (a mobile phone) or various other type of devices. (Levine, ¶0187 ("client devices **112** would include, for example, a mobile phone **112***a*, a video game console (with Internet connection) **112***b*, a personal digital assistant

[1] All underlining added to quoted material unless noted.

112*c*, a personal area network with retinal projection displays and/or ear piece **112***d*; a laptop **112***e*, and a desktop computer **112***f*").)

Like the '582 patent, Levine describes that each player may play a character in the game, which is represented by an "avatar." (Levine, ¶¶0224, 0244, 0324-0327.) Like the '582 patent, Levine also describes characters in the game that are controlled by the system as opposed to being controlled by a human player, such as "non-player character" (NPC) entities which are controlled with "artificial intelligence." (Levine, ¶¶0394 (describing "Active Objects (NPCs)—non-player controlled objects"), 0629 (describing "Non-Player Characters (NPCs) that may have an independent life of their own; that walk and talk, or run and hide, or perform other changes of state actively of their own accord" with "[s]ome sort of Artificial Intelligence (AI)"), 0186 (describing "elements of the application not directly controlled by users, such as artificial intelligence or aspects of a simulation that run on their own internal logic and react to other aspects of the simulation").)

Levine provides extensive description and discussion of its location-based online gaming system throughout its disclosure, including architectural and system-level components and features as well as example illustrations of games. Among the example illustrations are a location-based game where players interact with a "monster" and a location-based game with a "sniper" that is illustrated in Figures 45-47 (Figure 47 excerpted at right, illustrating sniper gameplay on Palm Pilot device). (Levine, ¶¶0659-0673 ("monster" game), 0556-0573 & Figs. 45-47 ("sniper" game).)



Levine also describes features matching the claimed '582 patent features that a virtual object can be "picked up" by a character. (Levine, ¶¶0211 ("When a player picks up a sword, the database 104 must record this fact and store it . . ."), 0574 (discussing character "picking up a gold coin").) Finally, Levine discloses the "impenetrable object" claimed in the '582 patent, describing objects such as a wall, rubble, trees, and weapons that a player may "collide" with in the game. (Levine, ¶¶0386, 0681, 0568.) Each character may also have "health points" in the game, which may be impacted by actions such as a character "striking a sword blow" or other impacts that may determine a "damage" metric. (Levine, ¶¶0574, 0255, 0633.) Such features were well-known in prior art games and disclosed by Levine as well.

B. Prior Art Status of Relied-Upon References

Pre-AIA law applies to the challenged claims. Levine (EX1003) qualifies as prior art under § 102(e) because it is a published U.S. patent application that was filed before the priority date.[2]

C. Ground 1: Claims 1, 2, 12, and 13 are Obvious Over Levine

1. Independent Claim 1: "A non-transitory computer-readable medium having program logic provided thereon for providing a location-based game comprising:" (Claim 1[pre])

The preamble of claim 1 recites "[**a**] **non-transitory computer-readable medium having program logic provided thereon for providing a location-based game comprising**." Levine discloses a system for a multiplayer online game that

[2] The '582 patent claims priority to a provisional application (no. 60/499,810) filed September 2, 2003. For purposes of this Petition only, Petitioner has assumed a priority date of September 2, 2003. Petitioner at this time expresses no opinion on whether the '582 patent is actually entitled to the benefit of the provisional application filing date. This issue is not relevant here because all of the prior art references relied upon pre-date September 2, 2003. In the event entitlement to the provisional filing date or another earlier priority date may later become relevant, Petitioner reserves the right to respond at that time.

renders obvious claim 1 as referenced in the summary of Levine above and discussed further below.

To the extent the preamble is a limitation, Levine discloses and renders obvious the preamble. Levine discloses and renders obvious "**a location-based game**." (Cooperstock, ¶58.) Levine describes an online game system for playing multiplayer games:

> The present invention is directed to a system, method and computer program product for a computing grid for massively Multiplayer on-line games and simulations that substantially obviates one or more of the problems and disadvantages of the related art.

(Levine, ¶0021.) Levine describes a system architecture for hosting games, such as a Game Server, a Grid, and a database that maintains game-related information:

> A Gateway Server (hereafter usually referred to as "Gateway"), a Hosting Environment (a "Game Server" in the case of a gaming application, an "Application Server" in more generic contexts, a "Collaborative Engineering Environment Server" in other contexts, or a "Context Server" if the application were to be thought of as a "context"), and a Daemon Controller (all discussed in detail below) are examples of processes, each of which may be multi-threaded, and each of which runs on a physical host. These processes, which collectively comprise a single application (e.g., a game) or multiple applications, may run on a single host, or may be distributed across multiple hosts.

(Levine, ¶0176.)

> The Grid is a collection of hosts that decouples semantic and syntactic context in a packet that is exchanged between clients (and that relates to the game itself) from information that is in some sense "essential" to the Grid itself. In other words, the Grid can mediate the state of the object(s) without knowing what the states actually means. The Grid thus becomes a host for the context of the application (i.e., game) while being agnostic about the context itself.

(Levine, ¶0179.)

> Thus, at the center of every persistent-state, massively multi-player game lies its database **104**. The database **104** manages the persistence of object state across the game world: from login to login, session to session, Avatar to Avatar, property to property, it keeps a record of all significant state changes.

(Levine, ¶0211.)

The analysis for purposes of analyzing a "**location-based game**" considers the features and functions that a person of ordinary skill in the art would have understood and found obvious to be present in a game in Levine's system, in view of Levine's teachings. (Cooperstock, ¶59.) Levine's disclosure is not limited to any one or more specific games. Instead, Levine broadly teaches a variety of features, functions, and components to be used in implementing a multiplayer online location-based game system. Levine describes, for example, **(1)** design features generally applicable to the system's online games, **(2)** a "Gaming Example" with a "monster" character that players can "kill" (Levine, ¶¶0656-0666), **(3)** "Alternate

Embodiments" of the "monster" game including functions such as running away from the "monster" (Levine, ¶¶0668-0673), **(4)** a "sniper" game that is provided for "illustration" and reflected in Figures 45-47 (Levine, ¶¶0556-0558, 0569-0573, Figs. 45-47), and **(5)** additional examples of features, characters, and objects that may be present in a game, such as a "Dragon or a Troll," a "dog" that "brushes against [a] flower," actions of "picking up a gold coin, striking a sword blow, losing stamina," and others (*e.g.*, Levine, ¶¶0394, 0574, 0628, 0652). In view of these teachings, a person of ordinary skill in the art would have understood that each disclosed game and feature is a non-limiting illustration of how Levine's teachings may be applied. A person of ordinary skill in the art would have considered ways to apply Levine's teachings when implementing a game. For example, a person of ordinary skill in the art would have understood that features present in the "monster" game may be implemented in the "sniper" game and vice versa. This interchangeability of gameplay features would be particularly appreciable given the similarities between the "monster" and "sniper" games, including the location-based gameplay of characters approaching and retreating from each other and the roles of the characters being adversarial with "shooting" capability (for example, a "monster" versus other players that can "shoot" and "kill" the monster, and a "victim" versus a "sniper" having "a rifle scope" consistent with the capability to shoot the rifle). (*E.g.*, Levine, ¶¶0676 ("shooting a 'monster'"), 0666 (mobile user character can "kill the 'monster'

character"), 0571-0573 ("sniper" and "victim" characters, where victim may be "in range" of the sniper because "the sniper may have a rifle scope"), Figs. 45-47 (showing sniper character pointing a rifle), ¶0252 ("A PROPERTY_VECTOR state could represent the direction in which a game character's gun is pointed.").) "Combining two embodiments disclosed adjacent to each other in a prior art patent does not require a leap of inventiveness." *Boston Scientific Scimed, Inc. v. Cordis Corp.*, 554 F.3d 982, 991 (Fed. Cir. 2009).

For example, each of the "monster" and "sniper" games discloses a **location-based** game. (Cooperstock, ¶60.) In the "monster" game, a mobile user playing the game is physically located on Wall Street in New York City and in the game a "monster" is created at a corresponding nearby "location" in the game:

> In step **5806**, the PC user would register the new "monster" character with Grid system **100**. That is, the communications flow described with reference to FIG. 2 would allow the server **102** to centrally store the attributes of the new character in application database **104**.
>
> In step **5808**, server **102** would cause the new "monster" character to be delivered to all other users playing the same instance of the interactive multi-user gaming application as the PC user. Such deliver[y] would be affected by translator **108**, under the control of server 102, via transportation network **103**. Further, the server would place the new "monster" character in a PC user-dictated location within the synthetic environment, say for example, the Wall Street area of New York City. . . .

> In step **5810**, a user on a laptop client device **112***e* ("laptop user") would now "see" the new "monster" character on their laptop. More specifically, the laptop user would see the "monster" character on the synthetic representation of Wall Street in New York City. Grid system **100** ensures that the "monster" character is properly rendered for each user utilizing a different type of client device **112**.
>
> In step **5812**, the laptop user sends a message to a user on a mobile phone client device **112***a* ("mobile user"). Such message, for example, would convey that "a new 'monster' character is two blocks from you." This message may be sent because the mobile user is represented in the synthetic environment as being on Wall Street in New York City because in the physical world, they are. . . .
>
> In step **5816**, the mobile user can interact with "monster" character (i.e., manipulate the "monster" character entity). Such interaction would involve, for example, pressing *9999 on their mobile phone client device **112***a* to kill the "monster" character. In step **5818**, the synthetic representation of the "monster" character would disappear from the PC user's, laptops user's and mobile user's client devices. Again, Grid system **100** would ensure that the "monster" character's death would be properly rendered (using the proper signal) for each player's different type of client device.

(Levine, ¶¶0661-0666.)

Levine further illustrates how the "monster" game is a location-based game because players may move locations in the physical world, such as by taking a taxi

to "run away" from the "monster" for example, that result in corresponding changes of location within the game:

> In one embodiment, users of Multi-User Bridging system **100** may further bridge the synthetic environment with the physical environment. More specifically, in step **5816** of flow **5800**, the mobile user may have taken a taxi in order to "run away" from (i.e., interact with) the "monster" character. . . .
>
> In another embodiment of the present invention, as one skilled in the relevant art(s) will appreciate after reading the description herein, if the mobile user's taxi ride takes them outside of the Wall Street area of New York City, then the synthetic representation of the mobile user would disappear from the PC user's and laptops user's client devices.
>
> In another embodiment of the present invention, as one skilled in the relevant art(s) will appreciate after reading the description herein, a user may create an MP3 file that includes audio content (e.g., a recorded voice message) that is played on a registered client device owned by another player when that player enters a specific area of the synthetic or physical environment. For example, the PC user could specify that the "monster" character speaks each time another player enters a specific building located on Wall Street in New York City. That sound would be played, for example, on a player's mobile phone **112***a* when they walk into the physical building, or on a player's PC **112***f* speaker when a player's synthetic representation walks into the specified building.

(Levine, ¶¶0670-0672.)

The "monster" game is thus a "**location-based**" game. (Cooperstock, ¶62.) Gameplay depends on location—for example, as shown above, the mobile user is physically located on Wall Street in New York City, the game determines a location for the "monster" which is two blocks away from the mobile user, and the mobile user can "interact" with the "monster" located nearby and/or "run away" from the "monster" by changing locations in the physical world (for example, taking a taxi).

Levine's exemplary "sniper" game is also a "**location-based**" game. For example, the "sniper" and other characters such as "bicyclist" or "courier" and "victim" interact with each other—including the ability to "see" each other and "collide" with each other—based on the location of each, as determined for example by the "POSITION" of each Avatar within its "Locale" of record. (Levine, ¶¶0569 ("In the example shown in FIG. 45, the area-of-interest of the 'sniper' Server Thing is the region centered about the POSITION of the embodiment-of-record of that Avatar on its Server-of-Record in its Locale."), 0556, 0557 ("**FIG. 46** shows a Game Server of Record **4601** that includes a Locale of Record **4602** with a sniper standing inside the Locale of Record **4602**. Box **4603** represents the sniper's region of presence, and box **4604** represents the sniper's region of interest."), 0571-0573, 0232-0236 ("Each object in every game has an entry in the Thing table. The Thing table controls the behavior of objects across the Grid, and maintains their common basic states: position, orientation, range, presence, region of interest type, whether

they are active or passive in nature. . . . Position—where this object is located in the game world."), 0198-0204 (OAP device in each client device participating in the game determines and provides position information).) (Cooperstock, ¶63.)

Turning to the **non-transitory computer-readable medium having program logic provided thereon** for providing the location-based game, Levine broadly teaches that its disclosed invention may be implemented with hardware and/or software implemented on one or more computer systems, such as a computer system running applications as illustrated by computer system **5900**:

> The present invention may be implemented using hardware, software or a combination thereof and may be implemented in one or more computer systems or other processing systems. An example of a computer system **5900** is shown in **FIG. 59**. The computer system **5900** represents any single or multi-processor computer. In conjunction, single-threaded and multi-threaded applications can be used. Unified or distributed memory systems can be used. Computer system **5900**, or portions thereof, may be used to implement the present invention. For example, the system **100** of the present invention may comprise software running on a computer system such as computer system **5900**.

(Levine, ¶0693; *see also* Levine, Fig. 59 (illustrating system **5900**), ¶¶0180-0182 (describing the Grid "system **100**" "for an interactive, multi-player gaming application" that may be implemented in one server or multiple servers), 0657-0666 and Fig. 58 (exemplary game process flow of Grid system **100**).)

Levine further describes that the illustrative computer system **5900** includes **a non-transitory computer-readable medium** that allows **program logic** (logic in computer programs defining the game) to be loaded for execution. Levine describes that the system **5900** contains "main memory **5946**" that is preferably RAM and "secondary memory **5948**" that can be a hard disk drive. Each of the "main memory" (e.g., RAM) and "secondary memory" (e.g., hard drive) discloses the claimed non-transitory computer-readable medium, and these components together also disclose the claimed non-transitory computer-readable medium. Levine describes:

> Computer system **5900** also includes a main memory **5946**, preferably random access memory (RAM), and can also include a secondary memory **5948**. The secondary memory **5948** can include, for example, a hard disk drive **5950** and/or a removable storage drive **5952**, representing a floppy disk drive, a magnetic tape drive, an optical disk drive, etc. The removable storage drive **5952** reads from and/or writes to a removable storage unit **5954** in a well known manner. Removable storage unit **5954** represents a floppy disk, magnetic tape, optical disk, etc., which is read by and written to by removable storage drive **5952**. As will be appreciated, the removable storage unit **5954** includes a computer usable storage medium having stored therein computer software and/or data.
>
> In alternative embodiments, secondary memory **5948** may include other similar means for allowing computer programs or other instructions to be loaded into computer system **5900**. Such means can include, for example, a removable storage unit **5962** and an interface

> **5960**. Examples can include a program cartridge and cartridge interface (such as that found in video game console devices), a removable memory chip (such as an EPROM, or PROM) and associated socket, and other removable storage units **5962** and interfaces **5960** which allow software and data to be transferred from the removable storage unit **5962** to computer system **5900**.

(Levine, ¶¶0698-0699.)

Levine further describes that computer programs stored in the main memory and/or secondary memory can be executed to perform the features of Levine's disclosed invention, which includes its online location-based game system as discussed previously and discussed further for the claim elements below. Levine also describes the computer programs as computer "control logic." (Levine, ¶703.) The computer programs ("control logic") disclose the claimed "**program logic provided thereon for providing a location-based game**."

> Computer programs (also called computer control logic) are stored in main memory **5946** and/or secondary memory **5948**. Computer programs can also be received via communications interface **5964**. Such computer programs, when executed, enable the computer system **5900** to perform the features of the present invention as discussed herein. In particular, the computer programs, when executed, enable the processor **5944** to perform features of the present invention. Accordingly, such computer programs represent controllers of the computer system **5900.**

> The present invention can be implemented as control logic in software, firmware, hardware or any combination thereof. In an embodiment where the invention is implemented using software, the software may be stored in a computer program product and loaded into computer system **5900** using removable storage drive **5952**, hard disk drive **5950**, or interface **5960**. Alternatively, the computer program product may be downloaded to computer system **5900** over communications path **5966**. The control logic (software), when executed by the one or more processors **5944**, causes the processor(s) **5944** to perform functions of the invention as described herein.

(Levine, ¶¶0702-0703.)

Accordingly, Levine discloses and renders obvious "[**a**] **non-transitory computer-readable medium having program logic provided thereon for providing a location-based game**" in the form of a non-transitory computer-readable medium (hard drive and/or RAM) having program logic provided thereon (the logic implemented by computer programming for a location-based game) for providing a location-based game (a location-based game as described by Levine). (Cooperstock, ¶¶64-67.)

(a) "a virtual playfield," (Claim 1[a])

Levine discloses "**a virtual playfield**" as part of each online game. For context, Levine's online game system "accounts for both the physical and **virtual** location and context of the participating devices and people" and operates "for the

purpose of coordinating activities in the real (i.e., physical) and **virtual** worlds." (Levine, ¶0158.) As explained further below, the "virtual" world in a game encompasses a geographic region corresponding to one or more "Locales." The set of Locales for a game discloses "**a virtual playfield**." (Cooperstock, ¶68.)

Specifically, Levine describes that a game includes one or more "Locales" that together form the "world" of the game:

> An actual game includes at least one Locale, and possibly many Locales, where all the Locales together form a seamless "game world", or simply "world".

(Levine, ¶0177.)

Levine describes that a Locale provides a 3D environment for interactions of "Server Things."

> A Locale is a convex region in three dimensional space, that provides a stage or environment that supports the interactions of one or more Server Things. A Locale represents a place to establish a specific presence as part of the larger game universe. Although a Locale does not have to be rectangular in boundary, in one embodiment, discussed below, it has to fit within a region with the maximum dimension of 65536*65536*65536, as shown in **FIG. 21**.
>
> The Locale is the atomic unit of geography in the game world, and is defined in terms of world coordinates. These values correspond to the POSITION state values transmitted in packets as part of object state (see also discussion of Network Protocol Stack below).

(Levine, ¶¶0371-0372.)

Levine's Figure 21 "illustrates an example of a Locale topology" and indicates how a Locale is three-dimensional and has a size "on the order of magnitude of a small town," illustrating how the "Locale" in the game represents a region of geographical space defining the virtual playfield in the game. (Levine, ¶0052.)

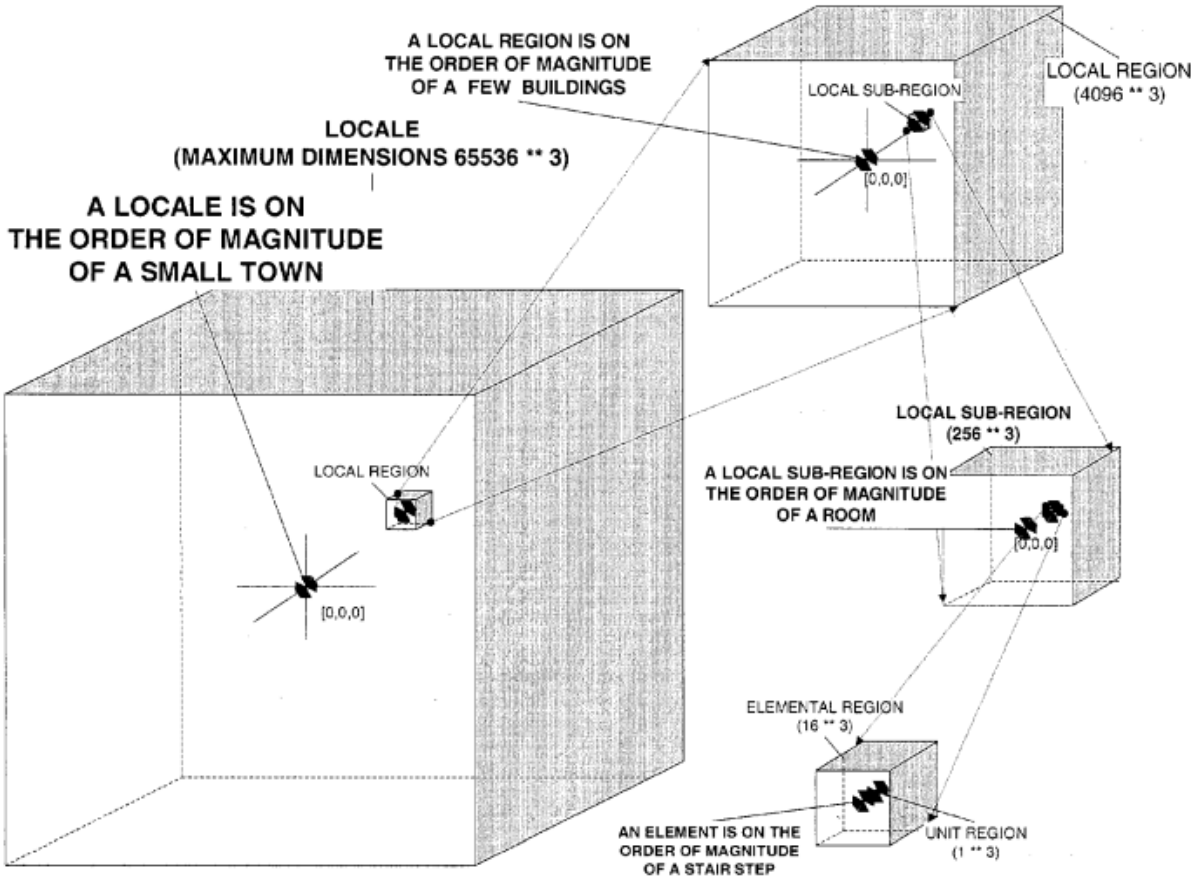


FIG. 21

(Levine, Fig. 21.)

Levine describes four types of Server Things that can interact within a Locale: (1) "Avatars," which are client-controlled characters "ultimately connected to a real

human player," (2) "Active Objects (NPCs)," (3) "Passive Objects," and (4) "Sentinels." (Levine, ¶¶0391-0396.) A player's character in the game (avatar) is initially assigned to a particular Locale but can move between Locales:

> When a client's Avatar is embodied, it is assigned (or bound) to whichever Locale its region of presence is positioned in, that is, the Locale within whose boundary hyperplanes it is completely contained. After discovery of the host location, the Gateway **401** directs (or proxies) client communications to this Game Server of Record. In turn, the Game Server of Record creates an Embodiment-of-Record (called a Server Thing) in the specified Locale and which represents the Avatar within its current context. This binding of the client to Server Thing is dynamic, and as the client roams throughout the Grid, its embodiment can move out of one Locale and into another, as shown in FIG. 24.

(Levine, ¶0379.)

Figure 24 "illustrates an example of a packet used for moving Embodiments of Record between Locales." (Levine, ¶0055.)

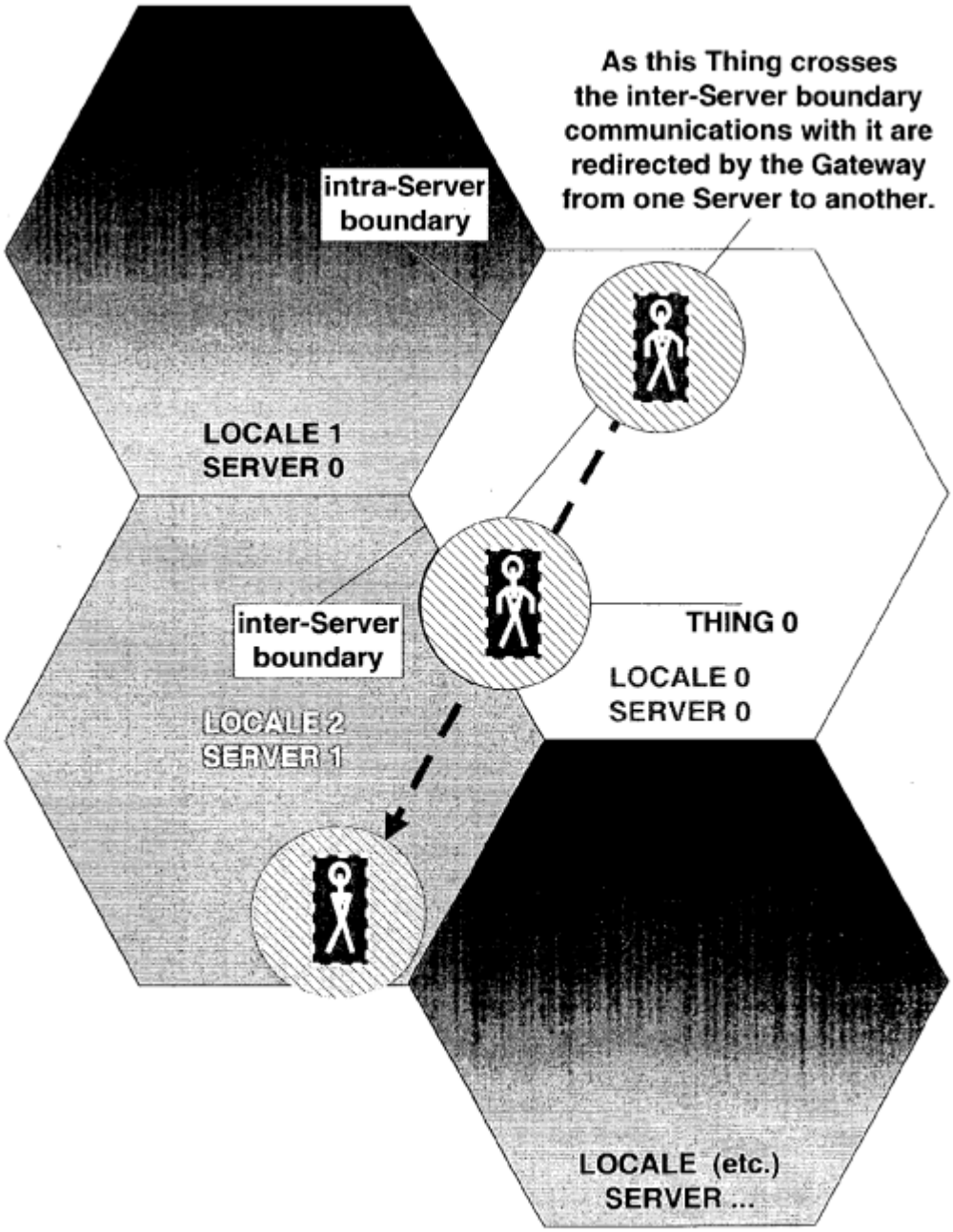


FIG. 24

(Levine, Fig. 24.)

Accordingly, when playing a game, the players' characters (avatars) may

engage in interactions with other objects within the game's one or more Locales that define a playfield for the game, disclosing the claimed "**virtual playfield**." (Cooperstock, ¶¶69-74.)

Levine further illustrates how Locales can also correspond to real-world locations, so that the virtual playfield mirrors a real-world playfield in a certain geographic area. (Cooperstock, ¶75.) For example, in the "monster" game, the virtual playfield includes a geographic region corresponding to New York City and players may move around within the game's virtual playfield by moving around in the real world, such as moving from the Wall Street area to another area:

> In step **5808**, server **102** would cause the new "monster" character to be delivered to all other users playing the same instance of the interactive multi-user gaming application as the PC user. Such deliver[y] would be affected by translator **108**, under the control of server **102**, via transportation network **103**. Further, **the server would place the new "monster" character in a PC user-dictated location within the synthetic environment, say for example, the Wall Street area of New York City**. . . .
>
> In step **5810**, a user on a laptop client device **112*e*** ("laptop user") would now "see" the new "monster" character on their laptop. More specifically, **the laptop user would see the "monster" character on the synthetic representation of Wall Street in New York City**. Grid system **100** ensures that the "monster" character is properly rendered for each user utilizing a different type of client device **112**.

> In step **5812**, the laptop user sends a message to a user on a mobile phone client device **112*a*** ("mobile user"). Such message, for example, would convey that "a new 'monster' character is two blocks from you." This message may be sent because **the mobile user is represented in the synthetic environment as being on Wall Street in New York City because in the physical world, they are**.

(Levine, ¶¶0661-0664.)

Similarly, Levine also illustrates exemplary scenes of a virtual playfield in the "sniper" game, including visual depictions of characters in different areas of a virtual playfield and a screenshot of the game on a Palm Pilot device, such as shown in Figures 45, 46, and 47 reproduced below. (Cooperstock, ¶76.) As shown in the figures, the "virtual playfield" encompasses "Locale" regions that constitute the playfield in the game where characters (such as "sniper," "victim," and bicyclist) can perform activities and interact as part of playing the game.

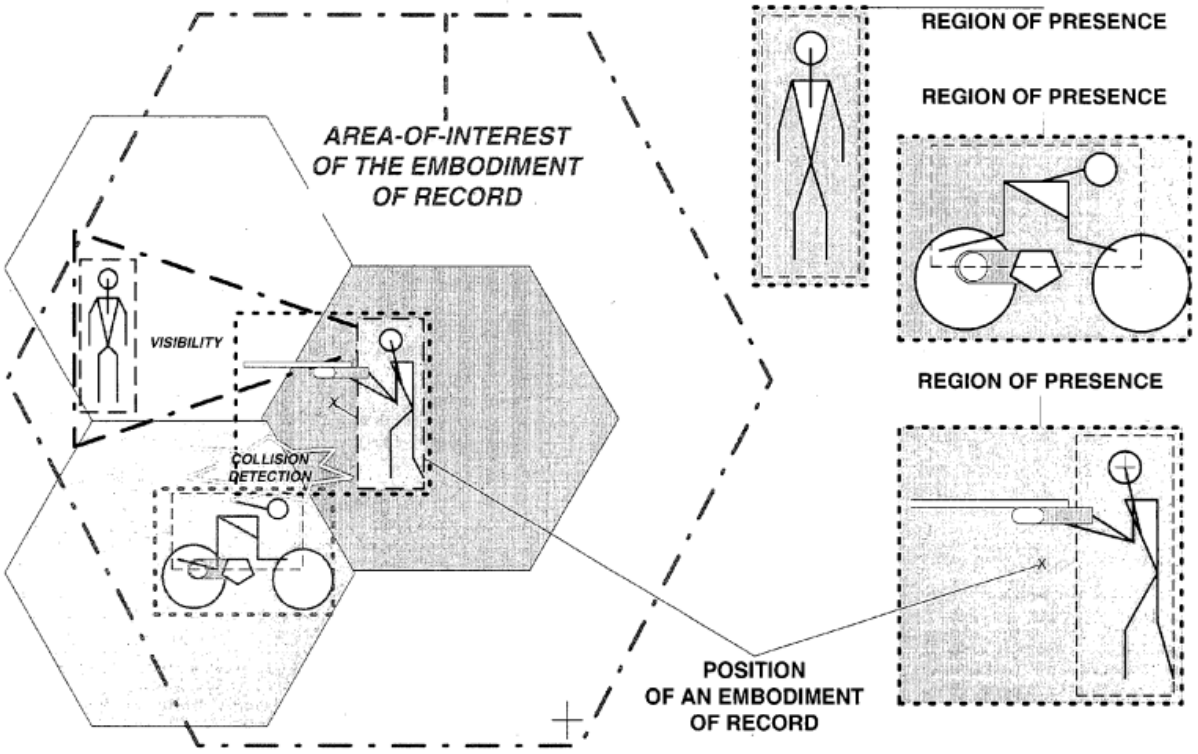


FIG. 45

(Levine, Fig. 45.)

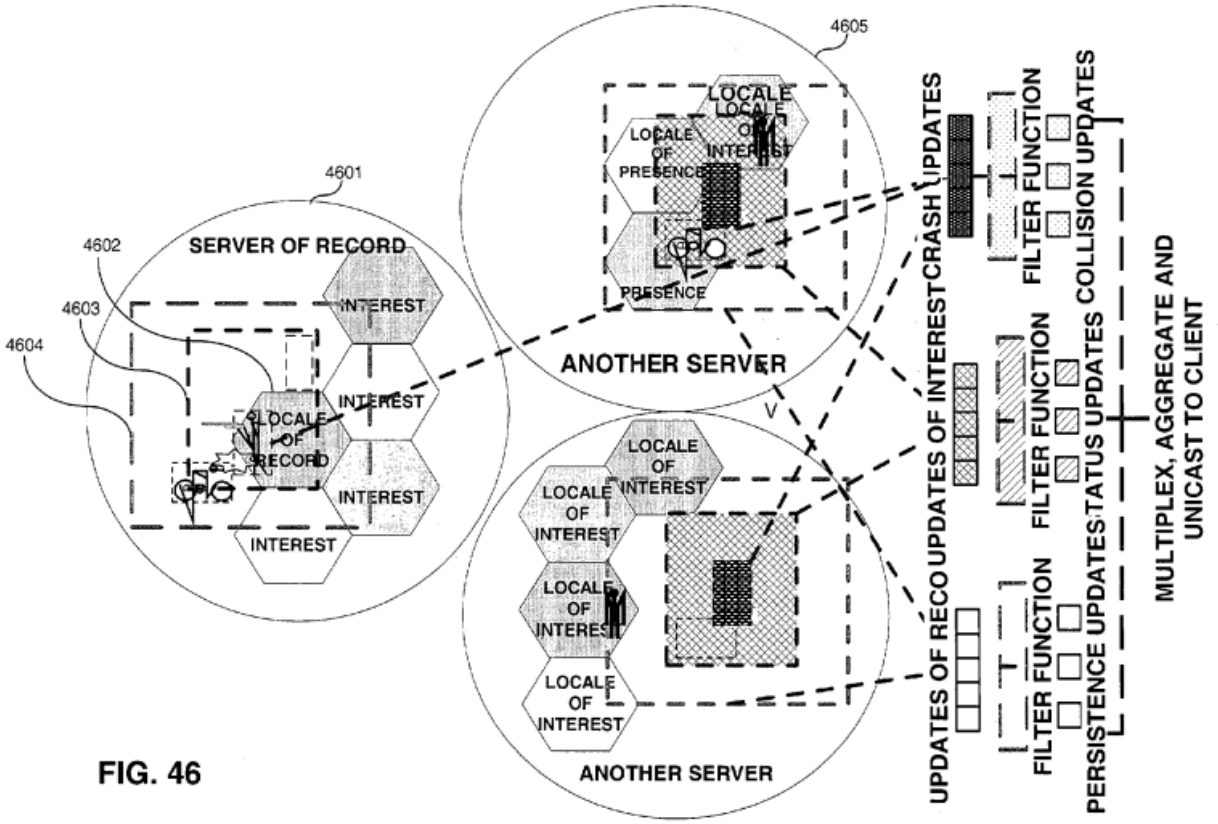


FIG. 46

(Levine, Fig. 46.)

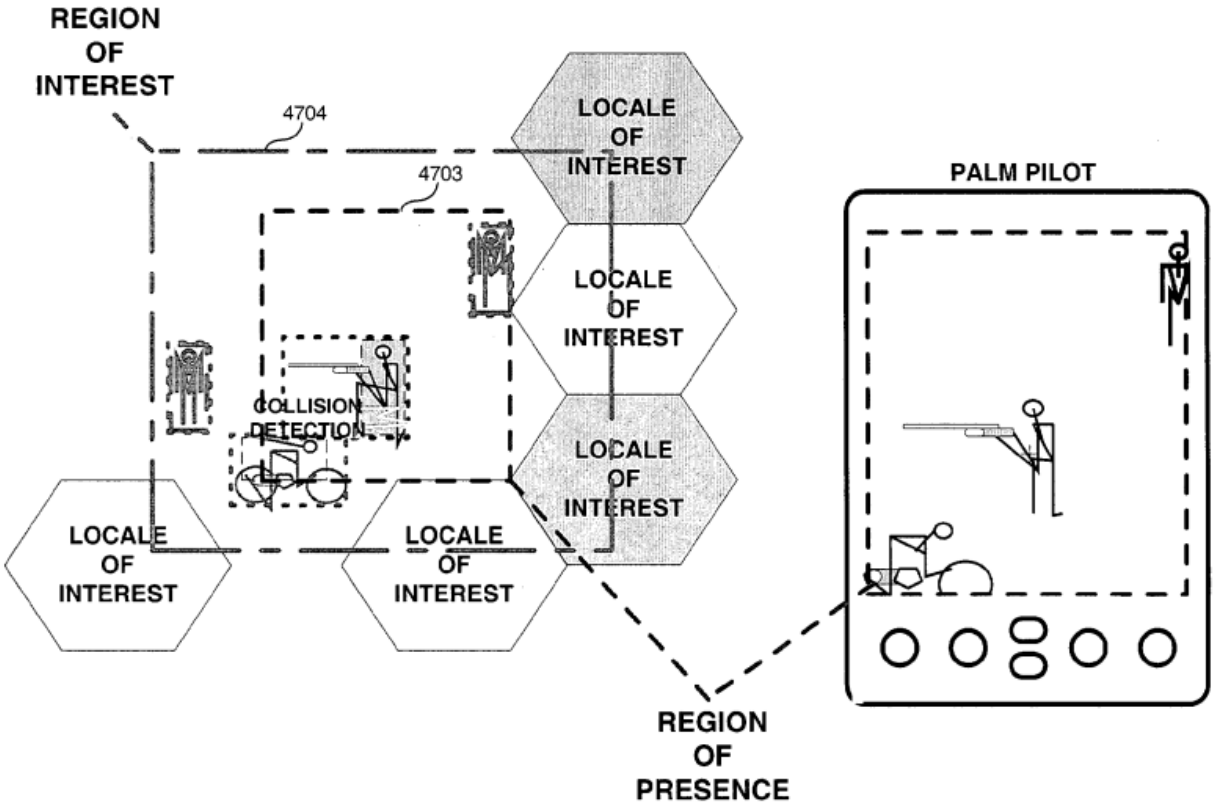


FIG. 47

(Levine, Fig. 47.)

(b) "a first character, wherein the location of said first character in said virtual playfield is displayed on a display and is determined utilizing a first control signal from a first locating device that is based, at least in part, on a physical location determined by said first locating device; and" (Claim 1[b])

Levine discloses and renders obvious "**a first character, wherein the location of said first character in said virtual playfield is displayed on a display and is determined utilizing a first control signal from a first locating device that is based, at least in part, on a physical location determined by said first locating device**."

Due to the length of this element, it is addressed in multiple parts. Levine discloses "**a first character**" in the form of each character in a game (called an "avatar") that is controlled by a human user playing the game. (Cooperstock, ¶78.)

> Avatars **806**—defines a role for the player within a specific game: associates a specific Thing representing the player with its most recent Locale **802**.

(Levine, ¶0224.)

> Normally in the massively multi-player world, there are a multitude of objects. Avatar objects are Things connected to clients (real people pushing buttons and twitching joysticks somewhere out there on the Internet). Passive objects are Things that can be manipulated but aren't connected to any other form of control mechanism (gold coins that can be picked up and put into inventory, flags to capture, etc).

(Levine, ¶0628.)

For example, Levine describes that the "sniper" game includes characters/avatars such as a "sniper," a "victim," and bicyclist avatar that are controlled by players using client device. Each avatar discloses "**a first character**" in the game:

> In the example of **FIG. 45**, there would be a reference to the "sniper" Avatar on the list corresponding to the walking "victim" Avatar, and another reference to the "sniper" on the list corresponding to the bicycling "courier" Avatar. In order for the "sniper" to see the "victim", he or she must receive messages as the walking Avatar moves back and

> forth. This implies that the victim is a packet source for messages to the sniper, which becomes a packet sink for messages about the changes in state of the Server Thing representing the walking Avatar. For the "sniper" to collide with the courier, it must receive messages as the bicycling Avatar pedals here and there. This implies that the courier is a packet source for messages to the sniper, which becomes a packet sink for messages about changes in state of the Server Thing representing the bicycling Avatar. Thus, there is a Server ThingRef maintained on the internal list of the victim and the courier that is used to route messages from these Server Things (as sources) to the sniper Avatar (as sinks). Packets routed in this way and rebroadcast to the Gateway **401** handling the login session for the sniper Avatar, and are proxied back to the client controlling the sniper.

(Levine, ¶0571.)

Similarly, Levine describes that players can control characters in the "monster" game that can take actions such as "killing" the monster or running away from the "monster." (*See* Levine, ¶¶0661-0666, 0670-0672.) Each of the player-controlled characters in the "monster" game discloses "**a first character**" in the game. (Cooperstock, ¶80.)

Levine further discloses and renders obvious that "**the location of said first character in said virtual playfield is displayed on a display**." For example, Levine describes and illustrates how the location of each character in the "sniper" game's virtual playfield is displayed on a display. As shown below, Levine's Figure

45 conceptually illustrates how different characters (including sniper, victim, and bicyclist) are located in nearby areas in the virtual playfield of the game. Figure 46 illustrates the location of characters in a "Locale" within the "**virtual playfield**" of the sniper game. (Levine, ¶0557 ("**FIG. 46** shows a Game Server of Record **4601** that includes a Locale of Record **4602** with a sniper standing inside the Locale of Record **4602**."). Figure 47 likewise shows "Locale" regions in the sniper game's "**virtual playfield**" and illustrates how **the location of each character in the virtual playfield is displayed on a display** of the player's client device (a Palm Pilot in this illustration):

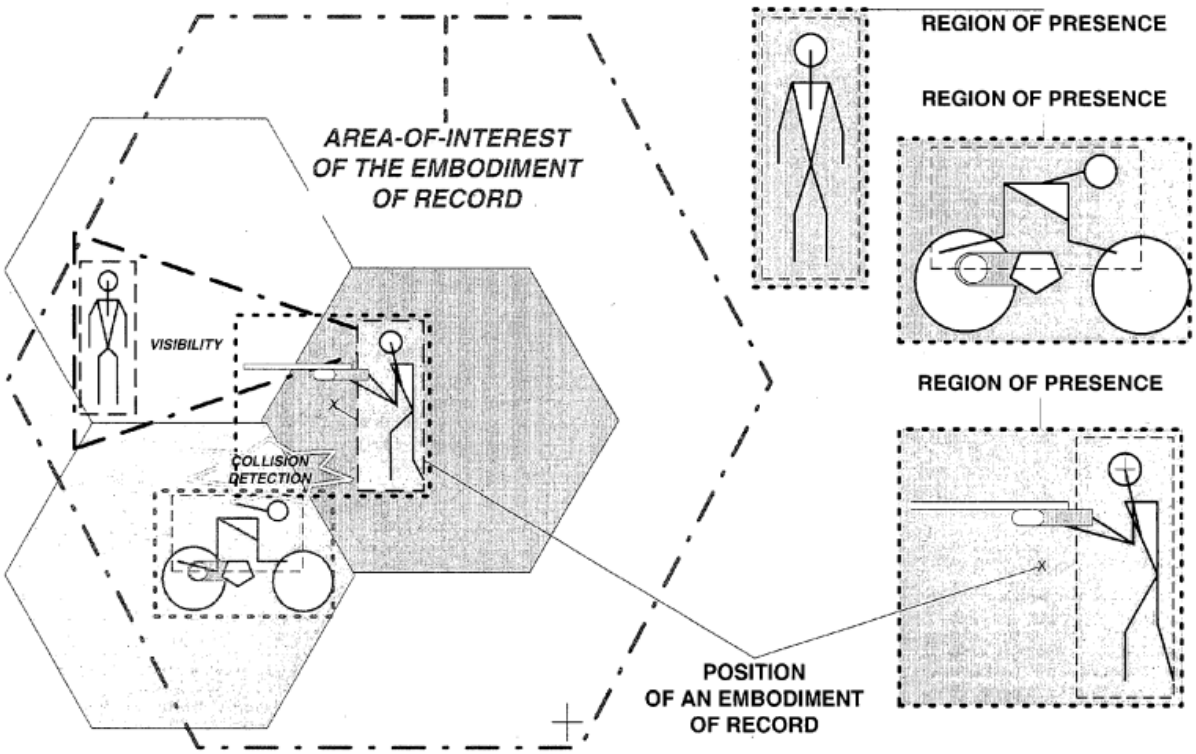


FIG. 45

(Levine, Fig. 45.)

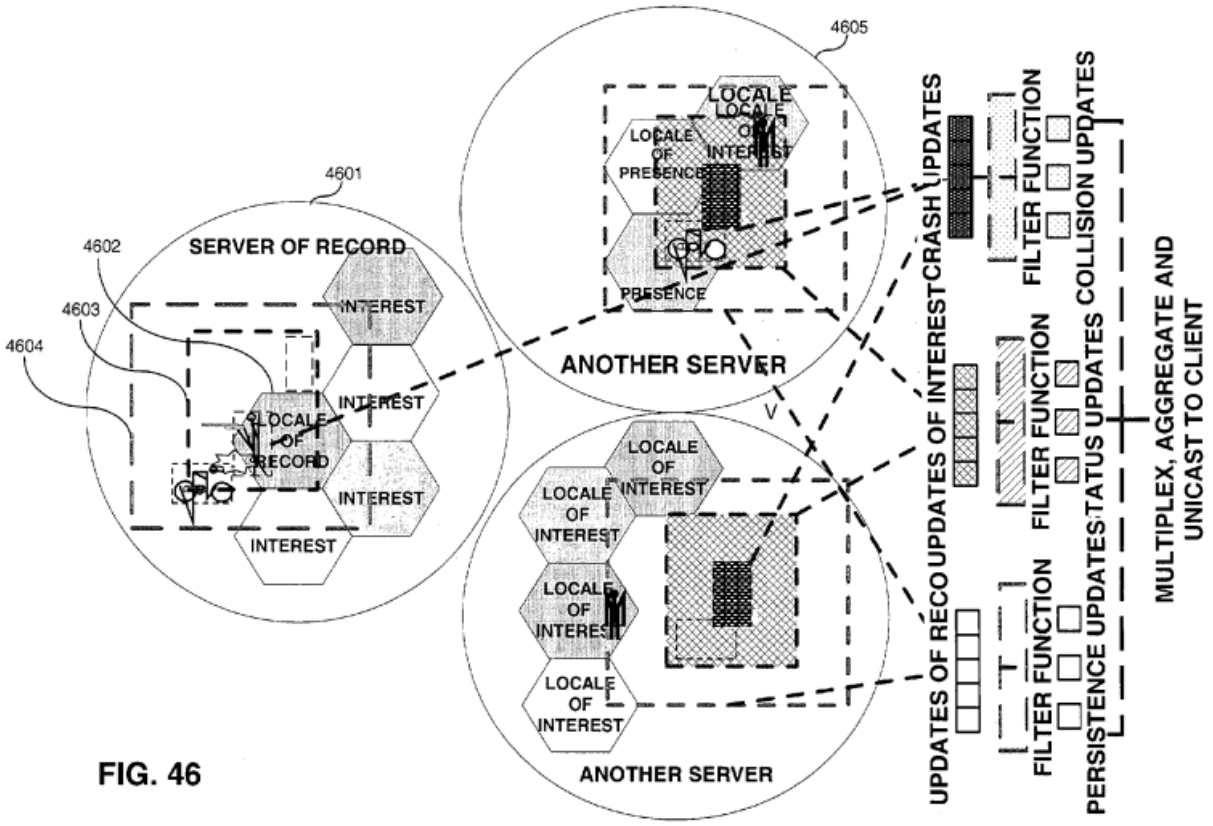


FIG. 46

(Levine, Fig. 46.)

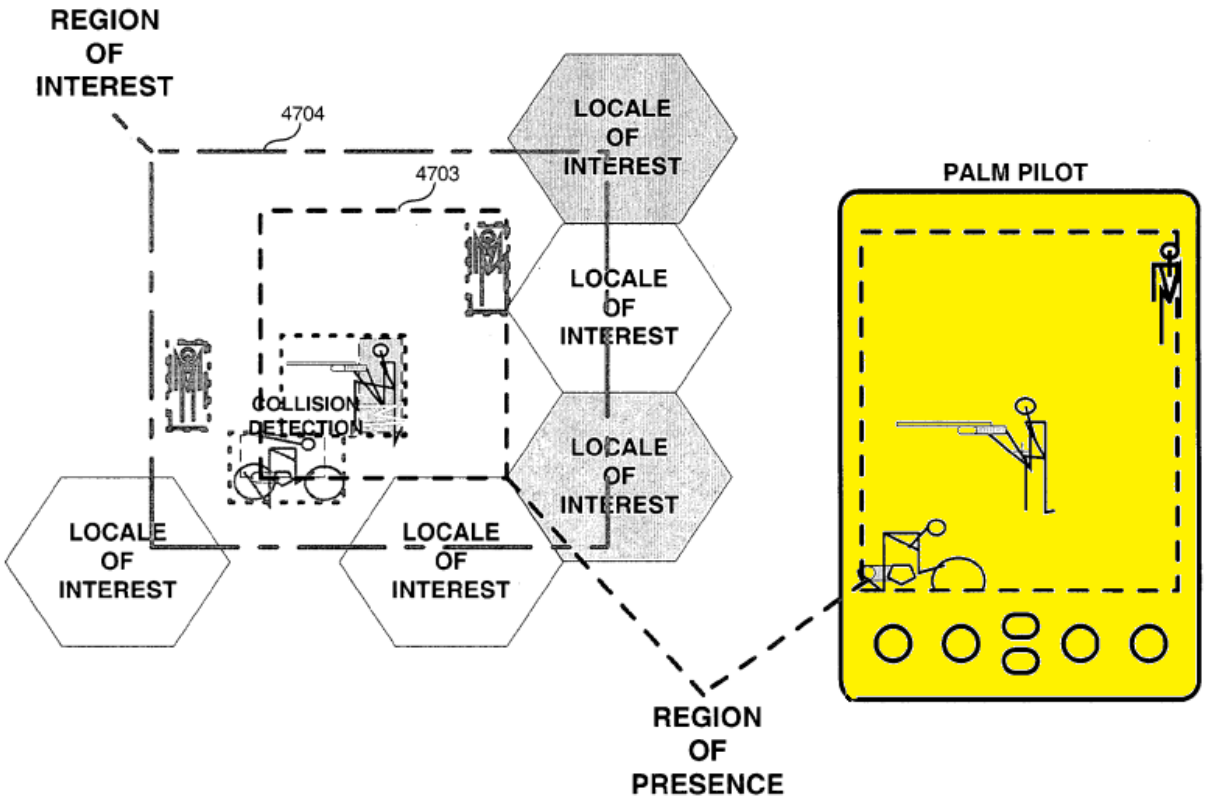


FIG. 47

(Levine, Fig. 47 (highlighting added).)

As shown in Figure 47 and highlighted above, the display on the client device (Palm Pilot) shows the location of each character in the game, such as showing the sniper located near the bicyclist and the victim in the exemplary screenshot. The left portion of Figure 47 illustrates the location of each character in the game with respect to Locale regions in the virtual playfield, including the sniper located near the bicyclist and the victim, and the right portion of Figure 47 illustrates how the location of each character is displayed on the client device. Levine thus discloses in the "sniper" game "**the location of said first character in said virtual playfield is**

displayed on a display." (Cooperstock, ¶82.)

Levine also renders obvious that a game in Levine's system would similarly display the location of each character on users' client devices. For example, the description of the "monster" game does not expressly specify that the location of each character is displayed on players' devices, but a person of ordinary skill in the art would have understood and found it obvious that the location of each character in the "monster" game would similarly be displayed. Levine describes that a visual representation of each player is presented to other players in the "monster" game. (*E.g.*, Levine, ¶0671 (when a mobile user leaves the area of Wall Street, "then the synthetic representation of the mobile user would disappear from the PC user's and laptops user's client devices").) It would have been obvious given the location-based gameplay of the monster game, such as where players' characters may "kill" or "run away" from the "monster" and effects such as audio content may be played when a player "enters a specific area," that the **location** of each character would be displayed so that the player could see the relative location of the character vis-à-vis the monster and other objects and characters that the character interacts with. (*E.g.*, Levine, ¶¶0666, 0670-672.) (Cooperstock, ¶83.)

Levine discloses "**a locating device**" in the form of an orientationally-aware peripheral (OAP) device contained in each player's client device. Specifically, Levine teaches that each client device **112** may include within it an OAP for

"tracking the locating and orientation of users within the system."

> In one embodiment, the system also includes an orientationally-aware peripheral device within the client devices for tracking the locating [sic] and orientation of users within the system of the present invention.

(Levine, ¶0155.)

> In an embodiment of the present invention, an orientationally-aware peripheral (OAP) device, described in detail below, may be included within Grid system **100** within each client device **112**.

(Levine, ¶0197.)

Levine further details how the OAP device is used to track the "**location**" of players in the system by incorporating an OAP device into each client device **112** used by players:

> Latitude, longitude and other sets of **location** data are often integral to the applications executed within Grid system **100**. Such location awareness allows software agents to traverse physical terrains and physical entities such as people, buildings and vehicles to be represented in virtual worlds. Therefore, in addition to existing systems such as GPS and the like, inertial tracking can be used to track the **location** and orientation of players within system **100**.
>
> In an embodiment of the present invention, an orientationally-aware peripheral (OAP) device, described in detail below, may be included within Grid system **100** within each client device **112**.
>
> Referring to **FIG. 3**, a block diagram representing the architecture of an orientationally-aware peripheral (OAP) device **300** according to an

> embodiment of the present invention is shown. OAP device **300** includes an inertial tracking subsystem **330** and a communication subsystem **320**. Inertial tracking subsystem **330** employs six accelerometers that will track the placement and orientation of the peripheral device in six degrees of freedom ("6-DOF"). Such a design eliminates the need for separate gyroscopic sensors to determine orientation information. The six accelerometers are divided into three groups of two sensors each (i.e., accelerometers pairs **302**, **304** and **306**) oriented along each of three perpendicular axes. Each pair of accelerometers is separated as far as is possible on the platform. By correctly integrating the acceleration of all six sensors, both position and angular orientation can easily be calculated.

(Levine, ¶¶0196-0198.)

As noted in the excerpted passage above, the OAP device provides inertial tracking, which "can be used to track the location and orientation of players" within the system. (Levine, ¶0196.) Levine further reiterates that "users' locations can be geographically tracked" by inertial tracking as provided by the OAP device:

> Another feature of the present invention is that users' locations can be geographically tracked, via a Global Positioning Satellite (GPS) system, cell-based triangulation, dead-reckoning (i.e., inertial tracking) or the like as described herein, in order to provide more realistic content, more realistic interactive experience to users, or data which is more contextually relevant to the user.

(Levine, ¶0162.)

Levine further notes that the OAP device calculates "position." (Levine, ¶0198 ("both position and angular orientation can easily be calculated").) Levine notes that "position" is recorded as "where this object is located in the game world."

> Each object in every game has an entry in the Thing table. The Thing table controls the behavior of objects across the Grid, and maintains their common basic states: position, orientation, range, presence, region of interest type, whether they are active or passive in nature. It includes definitions for the following properties: . . .
> d) Position—where this object is located in the game world. Also provided are Velocity and Acceleration for rectilinear motion.

(Levine, ¶¶0232-0236.)

Levine further discloses and renders obvious that "**the location of said first character in said virtual playfield . . . is determined utilizing a first control signal from a first locating device that is based, at least in part, on a physical location determined by said first locating device**." Specifically, the position information that is determined and transmitted by the OAP (the "**first locating device**"), recorded as "Position" information in the system for the corresponding character object for the player (Levine, ¶¶0232-0236), and communicated to other client devices discloses and renders obvious the claimed "**first control signal**." As discussed previously, Levine describes that "inertial tracking can be used to track the location and orientation of players within system **100**" and teaches that the OAP

device "within each client device **112**" contains an inertial tracking subsystem **330** using accelerometers from which "both position and angular orientation can easily be calculated." (Levine, ¶¶0196-0198.) (Cooperstock, ¶88.)

Levine further describes that the "output of the inertial tracker" is communicated to other client devices playing the same game:

> As mentioned above, OAP device **300** also includes communication subsystem **320**, where the output of the inertial tracker is received by a data translator **314** and communicated to other client devices **112** participating in the same instance of the multi-player, interactive game. One embodiment of the communication subsystem **320** would employ wireless communication protocols (such as Bluetooth, IEEE 802.11 or the like) to communicate with a nearby computer or base-station (and thus with translator **108**) via a transmitter **312**.

(Levine, ¶0204.) Figure 3 illustrates the OAP device including its inertial tracking subsystem and communication subsystem:

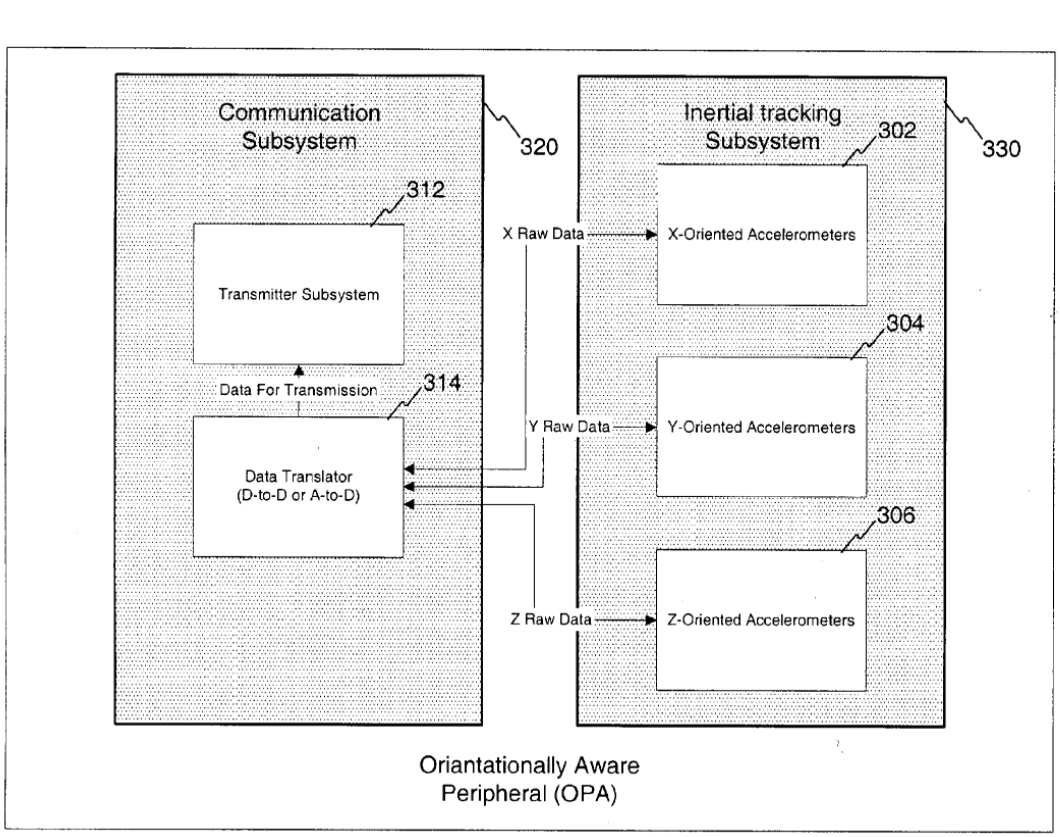


FIG. 3

(Levine, Fig. 3.)

The position information that is transmitted by the OAP, which is recorded in the system as "POSITION" information indicating the location of the character object within the game and communicated to other client devices, thus discloses and renders obvious the claimed "**control signal**" that is utilized to determine the location of each character in the virtual playfield. A person of ordinary skill in the art would have understood that the information transmitted by the OAP device in each player's client device discloses and renders obvious a "**control signal**" that is used by the system to control the determination of the "POSITION" of the player's

character in the game, which discloses the "location" of the character in the virtual playfield. Levine thus discloses and renders obvious that "**the location of said first character in said virtual playfield**" is determined based on the position information transmitted from the OAP in each client device, which is in turn "**based, at least in part, on a physical location determined by said first locating device**" (OAP device) as described previously. As an illustrative example based on Levine's disclosures, when a player is playing the "sniper" character in the "sniper" game, the player's client device (such as a Palm Pilot) can display on its display the location of each other character in the game (such as the bicyclist and "victim" characters). Each other character (such as bicyclist or victim) discloses the "first character" as claimed. The location of each character is based on the location information from the OAP device in the client device of the corresponding player, which is based on the physical location of the OAP device. Levine specifically describes how the POSITION of each avatar (player character), which is output from the OAP device of the player's client device, is recorded in the "sniper" game and used to determine the location of each character in the game: "In the example shown in **FIG. 45**, the area-of-interest of the 'sniper' Server Thing is the region centered about the POSITION of the embodiment-of-record of that Avatar on its Server-of-Record in its Locale." (Levine, ¶0569.) (Cooperstock, ¶90.)

Similarly, in the "monster" game, a person of ordinary skill in the art would have understood and found it obvious that the position of each character that would be displayed on players' client devices, corresponding to their real-world location such as Wall Street in New York City, would similarly be determined based on the physical location determined by the OAP device resulting in a control signal determining the location of the corresponding character in the game. For example, after a player in the real world takes a taxi, "the mobile user's taxi ride takes them outside of the Wall Street area of New York City, then the synthetic representation of the mobile user would disappear from the PC user's and laptops user's client devices." (Levine, ¶0671.) This illustrates how the change in the physical location of the player results in a change of the corresponding location of the player's character in the virtual playfield. A person of ordinary skill in the art would have understood the change of location in the virtual playfield to be determined by the OAP device in the player's client device sending a control signal based on the physical location it determined. (Cooperstock, ¶91.)

(c) "a second character, wherein said second character is computer controlled via artificial intelligence and said artificial intelligence utilizes said first control signal from said first locating device for controlling, at least in part, said second character; and" (Claim 1[c])

Levine discloses and renders obvious "**a second character, wherein said second character is computer controlled via artificial intelligence and said**

artificial intelligence utilizes said first control signal from said first locating device for controlling, at least in part, said second character."

Levine discloses "**a second character, wherein said second character is computer controlled via artificial intelligence**" in the form of Non-Player Character (NPC) objects, which are "character" objects (as the name indicates) that are controlled via "artificial intelligence" as Levine describes. (Cooperstock, ¶93.) At the system level, Levine describes that online games include "Active objects" which include NPCs and are controlled via "artificial intelligence" as opposed to being "directly controlled by users," as follows:

> Grid system **100** also includes a Daemon Controller **108** which acts as a privileged client for managing the activities of elements of the application not directly controlled by users, such as artificial intelligence or aspects of a simulation that run on their own internal logic and react to other aspects of the simulation.

(Levine, ¶0186.)

> Active Objects (NPCs)—non-player controlled objects that operate as multiple sources of packets to others and provide a single sink for packets from others. The originator of NPC packets is the daemon (discussed below), a computer controller login account for each Locale with special privileges. The daemon manipulates active objects within a specific Locale, and their embodiments-of-record provides a multiple sources of packets to other clients. An example of an active object might be a Dragon or a Troll.

(Levine, ¶0394.)

> Active Objects are objects, some of which are also known as Non-Player Characters (NPCs) that may have an independent life of their own; that walk and talk, or run and hide, or perform other changes of state actively of their own accord. These Non-Player Characters are not necessarily human characters. They may be animals, enchanted swords, or magic portals that take some positive role in directing game play. Some sort of Artificial Intelligence (AI) is attributed to this class of objects, and their object state changes appear to be directed by some sort of intelligent agent. Those changes of object state do not have to be physical ones. They may range from a proximity alarm that sounds a warning beacon if an Avatar approaches too closely to a morning glory that furls its petals at the setting of the sun. In other words, Active objects do something on their own or respond to external stimuli without having to be controlled by a real person sitting at the controls.

(Levine, ¶0629.)

Each NPC in a game that includes a "**first character**" as claimed thus discloses "**a second character, wherein said second character is computer controlled via artificial intelligence**." Further, a person of ordinary skill in the art would have understood and found it obvious that each game, including for example the "monster" and "sniper" games, implemented using Levine's online multi-player game system could include at least one such NPC. For example, in the "monster" game, Levine describes that the "monster" character is created with certain pre-

defined characteristics including movement behavior that is not controlled by manual-input, and does not describe that the activities of the "monster" are controlled by manual-input from a player. (*E.g.*, Levine, ¶¶0659-0666 (creation of character to "define how the 'monster' character moves within the synthetic environment (e.g., x,y position to x′,y′ position at z rate)" and corresponding gameplay).) A person of ordinary skill in the art thus would have found it obvious to implement such a "monster" as an NPC controlled by artificial intelligence, in view of Levine's teachings of "active object" characters that are controlled in a manner "not directly controlled by users, such as artificial intelligence" (Levine, ¶0186) and may include non-player characters of a nature similar to a "monster" such as "Dragon" or "Troll" (Levine, ¶0394). (Cooperstock, ¶94.)

Levine also renders obvious that the "sniper" game would include NPC active objects controlled by artificial intelligence. Levine describes that the "sniper" game is associated with one or more "Locales" among which players can move around, and Levine describes that a given "Locale" may have NPCs associated with it. (*E.g.*, Levine, ¶¶0557 ("FIG. 46 shows a Game Server of Record **4601** that includes a Locale of Record **4602** with a sniper standing inside the Locale of Record **4602**."), 0562 (discussing "when a player moves from one Locale to another" in the sniper game), 0394 (describing "Active Objects (NPCs)" and "active objects within a specific Locale"), 0637 (discussing "NPCs within a given Locale").) A person of

ordinary skill in the art would have found it obvious to apply these teachings regarding Locales to embodiments of games that use Locales. "Combining two embodiments disclosed adjacent to each other in a prior art patent does not require a leap of inventiveness." *Boston Scientific*, 554 F.3d at 991. Levine thus renders obvious that one or more of the Locales encompassed within the "sniper" game would implement one or more NPCs controlled with artificial intelligence, as a matter of design choice to implement Levine's teaching that Locales may have associated NPCs. (Cooperstock, ¶95.)

Levine further renders obvious that "**said artificial intelligence utilizes said first control signal from said first locating device for controlling, at least in part, said second character**" as claimed. (Cooperstock, ¶96.) Specifically, Levine renders obvious that the location of each player's character in the game as indicated by a control signal, as discussed for Claim 1[b], also is used to control, at least in part, the actions of NPCs controlled by artificial intelligence. Levine describes examples of how the activity of NPCs may be controlled, at least in part, based on the location of a player's character. For example, Levine describes NPCs that "talk," "run and hide," "sound[] a warning beacon," and generally "respond to external stimuli." (Levine, ¶¶0629-0630.) A person of ordinary skill in the art would have therefore understood and found it obvious that the "artificial intelligence" Levine describes as controlling the behavior and actions of an NPC would use a player's

location information to control NPC behavior and actions (such as talking, running away/hiding, or sounding an alarm when a player approached). Levine describes these features as follows:

> Active Objects are objects, **some of which are also known as Non-Player Characters (NPCs)** that may have an independent life of their own; that **walk and talk**, **or run and hide**, or perform other changes of state actively of their own accord. These Non-Player Characters are not necessarily human characters. They may be animals, enchanted swords, or magic portals that take some positive role in directing game play. Some sort of **Artificial Intelligence (AI)** is attributed to this class of objects, and their object state changes appear to be directed by some sort of intelligent agent. Those changes of object state do not have to be physical ones. **They may range from a proximity alarm that sounds a warning beacon if an Avatar approaches too closely** to a morning glory that furls its petals at the setting of the sun. In other words, Active objects do something on their own or **respond to external stimuli** without having to be controlled by a real person sitting at the controls.

(Levine, ¶0629.)

As described in this passage, an NPC (claimed "**second character**" controlled by "**artificial intelligence**") may be controlled, at least in part, by the location of nearby player characters (each an avatar disclosing the claimed "**first character**"). For example, where an avatar (player character) approaches too closely to an NPC,

that proximity **controls** the NPC to change its object state to a proximity alarm that sounds a warning. A person of ordinary skill in the art would have understood that the determined location of the player character (avatar) would be determined based on the control signal transmitted from the OAP in the player's client device, in view of the disclosures discussed previously for Claim 1[b]. (Cooperstock, ¶97.) Similarly, a person of ordinary skill in the art would have understood that the described functions of an AI-controlled NPC such as to "talk" with a player character, or "run and hide" from a player character, obviously would be **controlled** in part by the location of the player character as determined by the player's location as transmitted by the OAP in the player's client device. For example, it would have been understood that an NPC character programmed with its artificial intelligence to "run and hide" under predefined conditions would be caused to "run and hide" when a player character approached the NPC within a certain proximity in the game, as determined based on the location of the character in the game that is in turn determined based on a control signal transmitted by the OAP device in the player's client device as described for Claim 1[b]. (Cooperstock, ¶97.)

(d) "an impenetrable object, wherein said first locating device is operable to travel through a physical location that correlates to a virtual location of said impenetrable object on said virtual playfield, and said first character is impacted when said first character contacts said impenetrable object." (Claim 1[d])

Levine discloses and renders obvious "**an impenetrable object, wherein said first locating device is operable to travel through a physical location that correlates to a virtual location of said impenetrable object on said virtual playfield, and said first character is impacted when said first character contacts said impenetrable object**."

The analysis will first address the "**impenetrable object**" and "**impacted**" limitations and then address the "**location**" limitations.

As discussed below, Levine discloses and renders obvious "passive objects" which are each a "**virtual object**" in the game such as a "sword," an "axe," walls, "rubble," trees, and other objects that exist only in the game, are "**impenetrable**" in the game, and cause an impact (a "collision") on the character when the character contacts them. (Cooperstock, ¶100.)

The '582 patent describes "impenetrable" objects as objects in a game that may impact a character in the game, such as a wall that impedes movement or a car that runs into the frog in the classic video game FROGGER:

> A location-based game where a user's actual physical location on a physical playfield correlates to a video game character's location on a

> virtual video game playfield may require that the operational rules of current game be changed. Using PACMAN as an example, in the classic PACMAN, PACMAN is not operable to stray close to maze walls. In an actuality PACMAN game, however, a player may stray close to maze walls and even come into contact with maze walls. Hitting a maze wall may be similar to a user contacting, or passing through, a perimeter of the video game playfield. In this manner, functionality is provided for this scenario. For example, a user may be deducted a particular number of points if that user contacts a maze wall, a game's perimeter/boundary, or an object that a video game character cannot pass through and/or cannot use. For the purposes of this application, such objects will be referred to as impenetrable objects. One alternative to reducing points is to stun the character for a period of time (e.g., not allowing the video game character to move for a period of time), reduce the number of lives of that character (e.g., reduce the number of character lives left by 1), or end the game (e.g., display "GAME OVER" to the user).

('582, 11:51-12:20.)

> Multiple types of impenetrable (or interactive) objects may be provided in a single game. For example, in FROGGER cars may be impenetrable objects that cause FROGGER to die (e.g., lose a life). Alternatively, the perimeter of the playfield may be an invisible impenetrable object such that when a user leaves a portion of the actual playfield that corresponds to a location of the virtual playfield outside of the perimeter, FROGGER is left behind at the perimeter.

('582, 15:49-56.)

Levine discloses and renders obvious the same concept of "**impenetrable**" objects contemplated by the '582 patent.

For context, Levine teaches that the "Grid" in its game system contains "objects" that represent "physical" things in the game:

> Central to the operation of the database **104** is the concept of the object. An object is a "physical" item that is part of a shared environment.

(Levine, ¶0286.)

Levine describes multiple different types of "object," including a character controlled by a player (an avatar), an "active object" such as a non-player character (NPC), or a "passive object" such as "an Enchanted Castle." (Levine, ¶0393-0395.) Levine describes various passive objects such as a sword, an axe, a wall, and other objects. (Levine, ¶¶0211 ("When a player picks up a sword, the database **104** must record this fact and store it . . ."), 0526 ("Every sword has its own GUID . . ."), 0633 ("If the Non-Player Character is low on health points, the daemon knows it. If it is carrying an axe, the Daemon Controller can swing it.").)

Levine further teaches that each object in its game system may occupy "space" in the game's virtual playfield, resulting in a "collision" when a character contacts the object. Levine describes:

> In terms of computation complexity, area-of-interest management is essentially an O (n^2) process, since each Server Thing in a region may potentially interact with every other Server Thing in that region. Every

> time some Avatar takes a step, they may come into range, collide with, or drop out of sight of some other object.

(Levine, ¶0574.)

> EVENT_HERE—this Thing is in close proximity to an existing secondary object: a collision is immanent.

(Levine, ¶0646.)

To determine whether two objects collide, Levine's Grid (virtual playfield) maintains the position and "presence" of each object, which is used to determine "collision detection" between objects:

> Each object in every game has an entry in the Thing table. The Thing table controls the behavior of objects across the Grid, and maintains their common basic states: position, orientation, range, presence, region of interest type, whether they are active or passive in nature. It includes definitions for the following properties: . . .
> Position—where this object is located in the game world. Also provided are Velocity and Acceleration for rectilinear motion. . . .
> Presence—how far this objects "extends" in space for collision detection.

(Levine, ¶¶0232-0239.)

> Each Server Thing (see discussion above) interacts with others in its proximity through its area-of-interest. For example, each object on the Game Server **405** can have a range of vision (of block data subtype RANGE) within which other objects are visible, and a presence (of subtype PRESENCE) with which other objects can collide.

(Levine, ¶0568.)

Levine describes numerous examples of passive objects in the game and with which a character may collide, such as a wall, trees, or rubble:

> For performance reasons, a particular client device **112** may perform certain use-logic calculations locally, but the results of these calculations will not be transmitted unmediated to other clients within system **100**. For example, collision detection (i.e., a player collides into a wall within a shared environment) may be performed locally, but the back-end servers **102** must perform heuristics to ensure that collision constraints are met before transmitting updated position-states to other clients **112**.

(Levine, ¶0681.)

> A group of civilians may form a molecular type of civilian called a crowd. Other objects may be purely passive, such as trees or rubble.

(Levine, ¶0386.)

A person of ordinary skill in the art would have understood and found it obvious that an object such as a sword, axe, wall, or rubble in the Grid would be understood to be an **impenetrable** object that the character could not penetrate in the game. As Levine describes, when the "presence" of two objects corresponds such that they **contact** each other, there is a "collision" between the two objects. A person of ordinary skill in the art would have understood and found it obvious that for realistic and satisfying gameplay when a player's character collides with a

passive object, the object is **impenetrable** in the game such that the character cannot pass through or occupy the same space as the object. (Cooperstock, ¶¶104-108.)

Furthermore, Levine discloses that the character is **impacted** by contact with the wall because the contact forms a "collision," as shown in the passages cited above. The "collision" is an event that impacts the character.

In addition, it would have been obvious to implement the game such that a "collision" event may negatively impact the health of the character, providing an additional disclosure of **impact** on the character. Levine describes that characters in the game may have a "health" attribute (such as "health points"), like many adventure games in the prior art where the player's character has a variable "health" attribute. (Levine, ¶¶0255 ("Some networked games, especially first-person shooters, may get by with only a handful of states, such as health, damage, and strength. Other, more strategic games, will require an extensive list of special powers, items, and abilities (the palette of choices available to the game designer is illustrated in **FIG. 10**)."), 0633 ("If the Non-Player Character is low on health points, the daemon knows it.").) (Cooperstock, ¶110.)

Levine further describes that the objects in the game may include weapons such as swords and axes, and specifically describes "striking a sword blow" as an action between two players: "striking a sword blow, losing stamina, or exchanging goods or services do not necessarily affect the norm or distance metric between two

players." (Levine, ¶0574.)

A person of ordinary skill in the art would have understood and found it obvious that where a player's character (the "first character") contacts an impenetrable object (such as a sword swung by another player or an NPC, or running into a wall), the "collision" event would impact the character, such as by causing a loss of health points. A person of ordinary skill in the art would have understood that these are basic principles of adventure games that involve physical weapons and boundaries, where the game maintains the health status of the player's character, as reflected in Levine's teachings. (Cooperstock, ¶112.)

For these reasons, Levine discloses and renders obvious "**an impenetrable object, wherein . . . said first character is impacted when said first character contacts said impenetrable object**."

Levine further renders obvious "**wherein said first locating device is operable to travel through a physical location that correlates to a virtual location of said impenetrable object on said virtual playfield**." As described previously, Levine's games may correlate a real-world geographic area, such as the Wall Street area of New York City, with the "virtual playfield" of the game as defined by the Locales in the game, using location information from the OAP device in each player's client device. (*E.g.*, Levine, ¶¶0158 ("coordinating activities in the real (i.e., physical) and virtual worlds"), 0196-0204 (describing OAP device

functionality and location information transmission), 0661-0664 ("the mobile user is represented in the synthetic environment as being on Wall Street in New York City because in the physical world, they are"), 671 ("if the mobile user's taxi ride takes them outside of the Wall Street area of New York City, then the synthetic representation of the mobile user would disappear from the PC user's and laptops user's client devices.").) (Cooperstock, ¶114.)

However, the passive objects created and defined in the game, such as a wall or "rubble" or a sword or axe or gold coin, do not correspond to real-world objects (except perhaps in unusual circumstances where someone might create an object in the game to match a real-world object). (Levine, ¶¶0225 ("Things **807**—the basic description of an object in the game world. The Thing table distinguishes active objects from passive objects."), 0395 ("Passive Objects—non-controlled objects that operate as multiple sinks of packets to the Locale daemon and do not provide packets to others."), 0628 ("Passive objects are Things that can be manipulated but aren't connected to any other form of control mechanism (gold coins that can be picked up and put into inventory, flags to capture, etc[.])."). (Cooperstock, ¶115.)

As such, since the virtual object does not exist in the real world, the player's client device including its OAP ("**first locating device**") can pass through a physical location that correlates to a virtual location of that object. For example, in a game set in Wall Street, the Position of a virtual object (for example, sword or axe or

rubble) located on Wall Street in the game ("**virtual location**") would have a corresponding real-world **physical location** on Wall Street. The human player with their device could walk along Wall Street through the location correlating to the virtual location of the object in the game without any impediment. Levine thus renders obvious "**wherein said first locating device is operable to travel through a physical location that correlates to a virtual location of said impenetrable object on said virtual playfield**." (Cooperstock, ¶116.)

2. Independent Claim 2: "A non-transitory computer-readable medium having program logic provided thereon for providing a location-based game comprising:" (Claim 2[pre])

The preamble of claim 2 recites "[**a**] **non-transitory computer-readable medium having program logic provided thereon for providing a location-based game comprising**." To the extent the preamble is a limitation, Levine discloses and renders obvious the preamble for the same reasons discussed for the preamble of Claim 1. (Cooperstock, ¶117.)

(a) "a virtual playfield," (Claim 2[a])

Levine discloses "**a virtual playfield**" for the same reasons as discussed for Claim 1[a]. (Cooperstock, ¶118.)

(b) "a first character, wherein the location of said first character in said virtual playfield is displayed on a display of a portable device and is determined utilizing a first control signal from a first locating device of said portable device that is based, at least in part, on a physical location determined by said first locating device" (Claim 2[b])

Claim 2[b] is the same as Claim 1[b] except that it adds a "portable device" limitation as shown below.

Claim 1[b]	Claim 2[b]
a first character, wherein the location of said first character in said virtual playfield is displayed on a display and is determined utilizing a first control signal from a first locating device that is based, at least in part, on a physical location determined by said first locating device; and	a first character, wherein the location of said first character in said virtual playfield is displayed on a display <u>of a portable device</u> and is determined utilizing a first control signal from a first locating device <u>of said portable device</u> that is based, at least in part, on a physical location determined by said first locating device;

Levine discloses and renders obvious the two "portable device" limitations of Claim 2[b] and discloses and renders obvious the remainder of Claim 2[b] for the same reasons explained for Claim 1[b]. (Cooperstock, ¶¶119-121.)

Levine teaches that each client device operated by each user playing the game may be a **portable device**, such as a mobile phone or personal digital assistant (PDA):

> Connected to the transportation network (e.g., global Internet **103**), outside of the LAN **101**, includes a plurality of external client devices **112** that allow users (i.e., players) to remotely access and use Grid system **100**. External client devices **112** would include, for example, a mobile phone **112a**, a video game console (with Internet connection) **112b**, a personal digital assistant **112c**, a personal area network with retinal projection displays and/or ear piece **112d**; a laptop **112e**, and a desktop computer **112f**.

(Levine, ¶0187; *see also* Levine, ¶¶0160 ("universal access to the shared environment is allowed via any device to which a client can be provided (e.g., mobile phones, video game consoles, personal computers, personal digital assistants (PDAs), retinal projection displays, ear pieces, etc.)"), ¶0161 ("Another feature of the present invention is that, aside from personal computers, workstations and terminals connected to the Internet, it allows mobile phones, wireless data devices, PDAs and the like, which are commonly owned by today's consumers, to represent opportunities to where users can participate in multi-person, interactive applications.").) (Cooperstock, ¶121.)

As such, where Levine discloses and renders obvious display of the first character as discussed for Claim 1[b], Levine further discloses and renders obvious

that the display is "**of a portable device**" per Claim 2[b]. As discussed for Claim 1[b] above, Levine also expressly discloses a Palm Pilot—which is a PDA—as an example of a portable device in the "sniper" game which includes a display as shown in Figure 47. (Levine, Fig. 47 ("Palm Pilot" with display).) (Cooperstock, ¶122.)

Further, Levine teaches incorporating the "**locating device**" of Claim 1[b], the OAP device, into each client device playing the game, including mobile phones and PDAs. (Levine, ¶¶0197 ("In an embodiment of the present invention, an orientationally-aware peripheral (OAP) device, described in detail below, may be included within Grid system **100** within each client device **112**."), 0187 ("External client devices **112** would include, for example, a mobile phone **112a**, a video game console (with Internet connection) **112b**, a personal digital assistant **112c**, a personal area network with retinal projection displays and/or ear piece **112d**; a laptop **112e**, and a desktop computer **112f**.").) (Cooperstock, ¶123.)

For these reasons, Levine discloses and renders obvious Claim 2[b] for the same reasons as Claim 1[b] with the claimed "**display**" being provided on "**a portable device**" and the claimed "**locating device**" being contained within that "**portable device**." (Cooperstock, ¶124).

(c) "a second character, wherein said second character is computer controlled via artificial intelligence and said artificial intelligence utilizes said first control signal from said first locating device for controlling, at least in part, said second character" (Claim 2[c])

Levine discloses and renders obvious "**a second character, wherein said second character is computer controlled via artificial intelligence and said artificial intelligence utilizes said first control signal from said first locating device for controlling, at least in part, said second character**" for the same reasons explained previously for Claim 1[c]. (Cooperstock, ¶125.)

(d) "a virtual object, wherein said virtual object is operable of being picked up by said first character; and" (Claim 2[d])

Levine discloses and renders obvious "**a virtual object, wherein said virtual object is operable of being picked up by said first character**."

Levine teaches that in its online games, a player's avatar in the game ("**first character**") can pick up a **virtual object** such as a sword, duck, or gold coin, which are then placed into a record of the avatar's inventory:

> Thus, at the center of every persistent-state, massively multi-player game lies its database **104**. The database **104** manages the persistence of object state across the game world: from login to login, session to session, Avatar to Avatar, property to property, it keeps a record of all significant state changes. **When a player picks up a sword, the database 104 must record this fact and store it**, otherwise the next time that player logs in they will wonder where they lost it.

(Levine, ¶0211.)

> utilities.grab (. . .)—invoke the grab function in the utilities module to **pick up an object and transfer it into your inventory**

(Levine, ¶0548.)

> Every time some Avatar takes a step, they may come into range, collide with, or drop out of sight of some other object. However, many state changes do not involve changes that affect the Server ThingRef list of current packets sinks for this Avatar. For example, **picking up a gold coin**, striking a sword blow, losing stamina, or exchanging goods or services do not necessarily affect the norm or distance metric between two players.

(Levine, ¶0574.)

> This C++ server method is called by the buy_a_duck function to generate an approval dialog with the seller of the 'duck', whose response will control whether or not the transfer actually takes place. If the approval for this action is received, the script will call another C++ server method, **grabByGUID**(. . .), **which will attempt to stuff the purchased 'duck' into the buyer's inventory list**.

(Levine, ¶0609.)

Each "picked up" item added to the player's avatar's inventory is a "**virtual object**" insofar as it exists only in the game and not in the real world. (Cooperstock, ¶128.)

Given these teachings that Levine's system includes programming for each

avatar to pick up items and retain them in inventory, a person of ordinary skill in the art would have found it obvious that in each game implemented in Levine's system, including the exemplary "monster" and "sniper" games, the player's character may pick up one or more virtual objects. For example, a person of ordinary skill in the art, considering Levine's teachings, would have understood and found it obvious that in the "monster" and "sniper" games, a character could pick up virtual items such as a weapon or shield to defend against the adversary in the game. (Cooperstock, ¶129.)

(e) "an impenetrable object, wherein said first locating device is operable to travel through a physical location that correlates to a virtual location of said impenetrable object on said virtual playfield, and said first character is impacted when said first character contacts said impenetrable object." (Claim 2[e])

Levine discloses and renders obvious "**an impenetrable object, wherein said first locating device is operable to travel through a physical location that correlates to a virtual location of said impenetrable object on said virtual playfield, and said first character is impacted when said first character contacts said impenetrable object**" for the same reasons as discussed for Claim 1[d] previously.

3. Dependent Claim 11: "The non-transitory computer-readable medium of claim 1, wherein said first locating device comprises an accelerometer."

Levine discloses claim 11. As discussed for Claim 1[b], Levine describes that

the OAP device ("**first locating device**") contained in each client device comprises an accelerometer:

> Referring to FIG. 3, a block diagram representing the architecture of an orientationally-aware peripheral (OAP) device **300** according to an embodiment of the present invention is shown. OAP device **300** includes an inertial tracking subsystem **330** and a communication subsystem **320**. Inertial tracking subsystem 330 employs six accelerometers that will track the placement and orientation of the peripheral device in six degrees of freedom ("6-DOF"). Such a design eliminates the need for separate gyroscopic sensors to determine orientation information. The six accelerometers are divided into three groups of two sensors each (i.e., accelerometers pairs **302**, **304** and **306**) oriented along each of three perpendicular axes. Each pair of accelerometers is separated as far as is possible on the platform. By correctly integrating the acceleration of all six sensors, both position and angular orientation can easily be calculated.

(Levine, ¶0198.) (Cooperstock, ¶131.)

4. Dependent Claim 13: "The non-transitory computer-readable medium of claim 2, wherein said first locating device comprises an accelerometer."

Levine discloses claim 13 for the same reasons discussed for Claim 1[b], Claim 2[b], and Claim 11. (Cooperstock, ¶132.)

VII. CONCLUSION

Petitioner respectfully requests IPR institution.

Dated: April 5, 2025

Respectfully submitted,

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue NW
Suite 700
Washington, DC 20004
Tel: (650) 843-5001
Fax: (650) 849-7400

By: / *Heidi L. Keefe* /
Heidi L. Keefe
Reg. No. 40,673
Counsel for Petitioner

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), I certify that this petition complies with the type-volume limits of 37 C.F.R. § 42.24(a)(1)(i) because it contains 12,639 words, according to the word-processing system used to prepare this petition, excluding the parts of this petition that are exempted by 37 C.F.R. § 42.24(a) (including the table of contents, a table of authorities, mandatory notices, a certificate of service or this certificate word count, appendix of exhibits, and claim listings).

DATED: April 5, 2025

COOLEY LLP
ATTN: Patent Docketing
1299 Pennsylvania Avenue NW
Suite 700
Washington, D.C. 20004
Tel: (650) 843-5001
Fax: (650) 849-7400

/ *Heidi L. Keefe* /
Heidi L. Keefe
Reg. No. 40,673

CERTIFICATE OF SERVICE

I hereby certify, pursuant to 37 C.F.R. Sections 42.6 and 42.105, that a complete copy of the attached **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 9,662,582 B2,** including all exhibits (**Nos. 1001-1006**) and related documents, are being served via Federal Express on the 5th day of April, 2025, the same day as the filing of the above-identified document in the United States Patent and Trademark Office/Patent Trial and Appeal Board, upon Patent Owner by serving the correspondence address of record with the USPTO as follows:

32733 - JEFFREY D. MULLEN
2212 Hassinger Lane
Glenshaw, PA 15116
UNITED STATES

And, via Federal Express upon counsel of record for Patent Owner in the litigation pending before the U.S. District Court for the Western District of Texas entitled *Mullen Industries LLC v. Meta Platforms, Inc.*, Case No. 1:24-cv-00354-DAE (W.D. Tex.) as follows:

Peter F. Snell
Mintz, Levin, Cohn, Ferris, Glovsky and Popeo, P.C.
919 Third Avenue
New York, NY 10022
212-935-3000
Fax: 212-983-3115
Email: pfsnell@mintz.com

DATED: April 5, 2025

/ *Heidi L. Keefe* /
Heidi L. Keefe
Reg. No. 40,673
COOLEY LLP
1299 Pennsylvania Ave. NW, Suite 700
Washington, D.C. 20004
Tel: (650) 843-5001