USPTO Patent Application # 63/656,043 06/04/2024

Mikhail Klimusha, Los Angeles, California

Title of Invention:

Online gaming platform for electronic versions of board games with changeable digital in-game assets

ABSTRACT

A system and method for creating and playing online videogame versions of board games which contain identifiable digital in-game assets, for example games of chess family, to enrich the gaming experience, further deepen the gaming theory through bringing new diverse categories of players and systemically analyzing data from interrelated games and provide economic income for the plentitude of creators of digital in-game assets. The digital in-game assets may be player-tokens, characters, items, environments, special visual and audio effects, modes of play, GUIs, and other programming products. Digital in-game assets are created by independent creators on the electronic online gaming platform in accordance with rules and protocols of the platform, by the computational means and tools provided by the platform to be negotiated in the marketplace of the platform and to be transacted and further imported into the gameplays of the online videogames provided by the platform.

TECHNICAL FIELD

The present technology pertains to videogaming, and to creative digitizing those tabletop board games which have simple mechanics and physics and do not require vast computational resource in the electronic embodiments of gameplays. More specifically, the present technology may provide various techniques for deepening and individualizing gaming experience through alternating digital in-game assets and ludemes of the various versions of board games, especially so for the games of chess family.

BACKGROUND

Video gaming is an increasingly popular activity worldwide.

In technical regard, modern video games can be represented as consisting of two major components: game engine and game assets (also often called digital assets, in-game, game-ready, game entities, in-game objects). Whilst definitions of the modern game engine can incorporate game assets, these two components can be immediately distinct for those skilled in the art.

Game engine is the software backbone infrastructure for creation of the videogame and operation of the gameplay, being software framework for the development and operation of the gameplay. Such framework, adapted for a particular game, normally includes protocols of the gameplay rules, or ludemes, which create the game logic. A game engine generally consists of a tool suite and a runtime component.

Game assets are contextual game objects, i.e. elements perceived by the player as belonging to the virtual world of the game. These objects are implemented by specific software programs and data structures developed for a particular game and interrelated with the game engine through specified rules and arrays of conditional commands and relationships. Game assets ordinarily include user interfaces representing commands and information enabling or helping a player to

implement the gameplay, settings representing virtual environments where gameplay takes place, in-game player characters representing virtual embodiments of a human player (player-tokens) in form of virtual characters managed by a player to implement gameplay, in-game non-player characters representing virtual characters not operated directly by a human player but by built-in game AI which may interact with a human player, various virtual artifacts used in the gameplay to implement the gameplay and to provide individual player experience (virtual apparel, weapons, objects, terrain, ambient elements, special visual and sound effects, etc.), cinematics/cutscenes representing artistic segments of a game that are not controlled by a human player but used to draw attention of a player to specific data, and other.

All these components are specialized software programs interrelated by their codes and run in unison in accordance with the game rules and reacting to the player commands, where game engine functions as a governing legislative and executive center.

By analogy, game engine can be compared to motor vehicle base represented by an assembly of chassis, transmission, steering system, conduits, circuitry and engine, while game assets can be compared to a body, interior, and other customizable additions to the vehicle base. The plentitude of customizable addons to the same vehicle base are infinite and unlimited, creating a variety of models and versions related to the same base.

Be other analogy, game engine can be compared to the foundation of a house, with all engineering infrastructure (incoming plumbing, electricity, transport communications) connected to it and carrying walls, floors, ceilings and roof of the house, while game assets can be compared to sidings, deco, interior design, flow coverings, inhouse electricity circuits and plumbing, doors and windows of the house, and furniture and appliances inside of it.

Yet by another brave analogy, game engine can be compared to a human, while game assets can be compared to apparel, jewelry, gadgets, food, and lifestyle such a human use, consume, or deploy. Whilst the garbs, personal decorations, diet, and lifestyle can vary, the basic anatomy and biochemical processes in the human body stay fundamentally the same and run by the once established rules and principles.

Information handled by the player during the interactive cycle can be divided into two categories: functional and aesthetic. Functional information allows the player to undertake the activities they are supposed to carry out to win the game, the logic of the game, its intellectual core. Aesthetic information defines most aspects of the context in which the game takes place and is manly aimed at rendering an atmosphere capable of drawing and maintaining players' attention on an emotive basis, making them feel part of an immersive and entailing virtual world. The context of any video game encompasses the storyline, which can be abstracted to the level of predefined rules the players are required and allowed to follow; the setting of the game, i.e. the backdrop for the storyline; and the goals.

Thus, the context must do mainly with aesthetic information, although the rules and goals (logic) have certainly a crucial functional importance. The setting can condition activities of the players significantly, for the sake of realism, emotional involvement, or immersion.

When analyzing and judging a game, players focus on gameplay, playability, and the game's context. Therefore, these three elements must be the main foci of attention for game designers and can be safely considered true cornerstones of player-centered game design.

The interplay of the logic (coherent system of ludemes) and aesthetical context of the game create double-play effect.

On one hand, the players concentrate on the core of the gameplay presented by its rules. Combinatorial decisions of tactical and strategical nature successfully immerse the players on the high level of abstraction of the representation of the game.

On the other hand, even at the highest level of gameplay abstraction, most of the players fill the missing aesthetical components of representation by their imagination, immersing into the roles of commanders, or even particular characters whom they command, creating mental pictures of battlefield where opponent forces contend. Such double-play is typical for the thematical games, and especially for the videogaming.

Taking first-person shooter (FPS) genre, it is clear that the game of this genre can be decomposed in two layers of the game: (1) abstract logic (rules, tasks and objectives, such as: move, search for target, find target, aim, shoot, evade being shot, find cover, escape, etc.) and (2) aesthetic context (quality and number of digital in-game assets and quality of their interplay). That is why some FTS games with poor mechanics, physics and graphics are less popular then those with better ones, though both have the same abstract rules, tasks, and objectives.

The logic of FTS game can be represented at various levels of abstractions, for example, in form of the simple abstract geometrical figures moving and responding to the commands and actions of the player in accordance with relevant and understandable rules.

Early videogames of this genre were, in fact, good examples of the high levels of abstraction, though rather by necessity than by choice. As the computational technology developed, the second layer of the double-play was becoming increasingly important and found it realization in sophisticated in-game assets such as realistic and complicated terrains, advanced weaponry, advanced tokens of the players and non-player characters, etc. Aesthetic, artistic layers of the game were developing along with, and often overtaking the developments of the core play. Thus, many successful video games have been developed based on the same logic and even the same logic-bound (not versatile) engines through development and integration of distinctly different in-game assets, on the level of aesthetic play, which created experiences essentially different from game to game while the underlying logic, philosophy, and software of such families of the games remained the same. The success of such games has been proved commercially. Development of the versions, modifications, and separately positioned games reusing the same engine and logic through scripting the new storylines and creating new systems of in-game assets, of various degrees of sophistication, is now an industry standard.

In real-life, boardgames in general, and chess in particular, do have simple and fascinating logic, clear goals and at least high-level abstraction of the storyline and context, which make such games playable and popular often for very prolonged periods of time, such as centuries. The main fascination of the games lays in their combinatorial algorithms of purposeful decision-making, strategical and tactical, directed at achieving goals in course of competition, like in chess family of games, or simply continuous competitive trying of chance. The context of the traditional board games thus have being defined by the overall systems of ludemes (rules of the gameplay), actual environment where games take place (outside, in premises, with friends, etc.) and individual mental imagery of the players.

Many attempts have been made in the past to advance the contextual and aesthetic components of the board games. Some of the editions of the games may have elaborate artistic design. Good example of such attempts is given by the artistic gift sets of pieces and boards in chess. However, such artistic contextual solutions made in bricks-and-mortar forms are still static in the sense that they cannot be changed at will and are bound to once created embodiment, however beautiful it is. The increase of flexibility through collecting a library of various versions of sets can be prohibitive expensive. Similar situations have been encountered in attempts to advance the existing versions of the games, such as creating 3D chess, which in real life run into excessive complication of playability.

The videogame technology provides powerful instruments to saturate the aesthetic component of the boardgames. And yet, despite to continuous convergence of the traditional boardgames and videogaming, the main improvements achieved so far lay not in the area of aesthetic part of the double-game, but in advancing the logical scenarios and analytical support of a gameplay (such as new advanced online AI chess engines), improving the territorial and temporal accessibility to gameplay (online versions of almost all popular boardgames are currently available 24x7).

As the traditional board games, and especially chess family of the game, resolutely entered the world of videogaming, the analytical engines have being developing the complexity and variety of the logical component of the games with astonishing speed. The artistic component of the game has been left far behind.

Some half-hearted attempts to leverage the aesthetic component of the electronic versions of the traditional board games, - such as inclusion of chess subsection into popular Street Fighter videogame, etc. - have been either restricted by the bounds of a bigger mother-videogame, or implemented in inflexible way (selection of several fixed settings for electronic chess), or with a moderate artistic quality. All these brought limited success to such attempts.

The tastes, social and intellectual backgrounds of the players differ. These differences are often so profound, that they do set psychological barriers for attempting to and enjoyable play of certain games stereotypically considered to be “highly intellectual” and “not for all”, such as chess. At the closer analysis, it is obvious, that the rules of games such as chess are accessible to everyone. The main point of self-expulsion for many players lays in not in the inner logic of the games, but in image and academic-like form of the game assets. Traditional Staunton set of chess, most popular all over the world, bears the seal of Ivory League classicism and exclusivity, unattainable and scary, or boring, scientific lore hidden in it. However, imagine the chess set implemented on the multicolor board with fancy animated characters, such as can be found in anime, cinema, fashion, fantasy, historical, political, social or economic cultural realms. Will such novelty bring more new players into the game? It certainly will.

Many classical board games have not yet gained global popularity due to the culture-specific orthodoxy in design of their in-game assets. The screaming examples include Eastern varieties of chess such as xiangqui (Chinese variant of chess) and shogi (Japanese variant of chess). For western player it is difficult even to discern the playing pieces as they are represented by identical tokens marked by different hieroglyphs. If the game assets of the Chinese and Japanese game will be designed taking into account tastes and preferences of western users, will these games become more popular? They certainly will.

SUMMARY

Aspects of the present technology include systems and methods for meaningfully enriching the experience of playing the game of chess and other board games and providing new venue of economic income for independent digital creators through creating, trading, and using in electronic versions of traditional and innovative board games of various digital assets organically related to and being integral elements of the gameplay. The digital assets may be in-game digital assets, such as in-game characters, items, surroundings, visual and audio effects, specific instruments, interfaces, modes, etc.

Whilst in the traditional board games in the present disclosure the elements of gameplay and playability are relatively simple, being predefined by the limited sets of strict and easily understandable rules, the contexts of such games provide ground for unlimited creativity in design and implementation, especially in part of contributing to the liveliness and aesthetic embodiments of the virtual worlds of the games, their artistic and psychological aspects.

In one example, a system of online game platform is provided. The system includes three major components:

(1) game engine runtime component, which will include industrial-strength production interrelated systems and subsystems needed to implement basic logic, physics and mathematics of the game flow of the video game. Existing as a complex of interrelated software programs comprising vast arrays of software development toolsets, programming code libraries, functional programming codes, data structures, algorithms, and software interfaces to implement them, such engine provides optimal techno – and ecosystems for enjoyable online creation and play of the diverse variety of board games and their modifications;

(2) game engine programming tool suite component, which will include industrial-strength production interrelated systems and subsystems needed to develop the versions of the game and in-game assets. Existing as a complex of interrelated software programs comprising vast arrays of software development toolsets, programming code libraries, functional programming codes, data structures, algorithms, and software interfaces to implement them, the game engine programming tool suite component is located in the memory storage coupled with one or more processors (e.g. implemented in circuitry) accessible by the creators of game versions and in-game digital assets;

(3) marketplace for the digital in-game assets, which is essentially one of the major systems channeled with the game engine, where the administrator of the platform and independent creators can trade/negotiate with human players and collectors their game assets compatible with the games playable on the platform, i.e. eligible for smooth importation into and problem-free use in the games. The marketplace will also allow the administrator of the platform and independent providers to offer a variety of auxiliary services such as bespoke creation and modification of in-game digital assets, training, mentoring, etc. Such market place of the online gaming platform will provide new economic opportunities for the independent creators within the gaming ecosystem.

The line between the game engine runtime component and programming tool suite components is blurry as the systems and subsystems of these components overlap and intertwine to considerable degree, often running in unison and necessarily used one with other. However, such distinction can be stipulated functionally wise for the purpose of the clarity.

In another example, a method for creating and negotiating digital game assets is provided. The method includes:

(1) creating a digital game asset usable in the video game by application of the programming tool suite of the game engine of the online platform in accordance with rules, protocols, technical requirements and other documentation established by the online gaming platform;

(2) submission of the created digital asset to the online game platform requesting admission to list the created digital game asset in the marketplace of the platform;

(3) implementation by the online gaming platform of testing and quality assurance of the digital asset submitted for the listing in the online market place of the online gaming platform;

(4) denial of admission for listing or admission to listing of the digital game assets in the online marketplace of the online gaming platform;

(5) negotiating of the digital asset in the online marketplace of the online gaming platform, wherein negotiating includes transfer of the rights of ownership or rights to use through selling, renting, exchange, or granting of the assets to interested and able parties on predefined terms in accordance with rules of the online marketplace;

(6) importing and using the digital in-game assets into gameplay session on the online gaming platform by the legitimate owner or user of the assets.

In another example a plurality of new versions and methods of the game is provided for being implemented in the format of the videogames as the most efficient medium for the implementation of the gameplay.

In another example, a non - transitory computer readable medium is provided that has stored thereon instructions that, when executed by one or more processors, cause the one or more processors to perform the method.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG . 1 is a block diagram illustrating an example network environment in which online gaming platform may be implemented, according to an aspect of the present disclosure;

FIG.2 is a block diagram illustrating an example of major functional components of the online gaming platform, according to an aspect of the present disclosure;

FIG.3 is a block diagram with dynamic flow elements illustrating an example network environment in which online gaming platform may be implemented, according to an aspect of the present disclosure;

FIG.4 is a block diagram illustrating an example of architecture of the runtime engine of the online gaming platform, according to an aspect of the present disclosure;

FIG. 4-1 is a block diagram illustrating an example of game specific subsystems of the runtime engine of the online gaming platform, according to an aspect of the present disclosure;

FIG. 4-2 is a block diagram illustrating an example of visual effects, scene/graph culling optimizations, online multiplayer and audio components of the runtime engine of the online gaming platform, according to an aspect of the present disclosure;

FIG.4-3 is a block diagram illustrating an example of low-level renderer, profiling and debugging, and human interface devices components of the online gaming platform, according to an aspect of the present disclosure;

FIG.4-4 is a block diagram illustrating an example of collision and physics and skeletal animation components of the runtime engine of the online gaming platform, according to an aspect of the present disclosure;

FIG. 4-5 is a block diagram illustrating an example of game resources and core systems components of the runtime engine of the online gaming platform, according to an aspect of the present disclosure;

FIG.4-6 is a block diagram illustrating an example of components of platform independence layer, software development kits (application programming interface), operational system, drivers, and user hardware components of the runtime engine of the online gaming platform, according to an aspect of the present disclosure;

FIG.5 is illustrating an example taxonomical classification of digital in-game assets 500 which may be created for and deployed on the online-gaming platform 110, according to an aspect of the present disclosure;

FIG.5-1 is illustrating an example taxonomy of the characters class 510 of digital in-game assets 500 which may be created for and deployed on the online-gaming platform, according to an aspect of the present disclosure;

FIG.5-2 is illustrating an example taxonomical classification of the player-token subclass 520 the characters class 510 of digital in-game assets 500 which may be created for and deployed on the online-gaming platform 110, according to an aspect of the present disclosure;

FIG.5-2-A, -B, -C, -D, -E, -F, - G, - H, -I, -J are illustrating conceptual examples of representations of player-token subclass 520 of characters class 510 of digital in-game assets 500 in example GUI 521, 521-1 of the online gaming platform 110 at the start and during the gameplay;

FIG.5-3 is illustrating an example taxonomical classification of the non-player subclass 530 of the characters class 510 of digital in-game assets 500 which may be created for and deployed on the online-gaming platform, according to an aspect of the present disclosure;

FIG.5-3-A, 5-3-B, 5-3-C are illustrating conceptual examples of representations of the non-player subclass 530 of the characters class 510 of digital in-game assets 500 in GUI 521 of the online gaming platform 110 during the gameplay;

FIG.5-4 is illustrating an example taxonomical classification of the semi-player subclass 540 of the characters class 510 of digital in-game assets 500 which may be created for and deployed on the online-gaming platform, according to an aspect of the present disclosure;

FIG.5-4-A is illustrating conceptual examples of representations of the semi-player subclass 540 of the characters class 510 of digital in-game assets 500 in GUI 521 of the online gaming platform 110 during the gameplay;

FIG.5-5 is illustrating an example taxonomical classification of the environments (settings) class 550 of digital in-game assets 500 which may be created for and deployed on the online-gaming platform 110, according to an aspect of the present disclosure;

FIG.5-5-A, 5-5-B, 5-5-C are illustrating conceptual examples of representations of the environments (settings) class 550 of digital in-game assets 500 in GUI 521 of the online gaming platform 110 during the gameplay;

FIG.5-6 is illustrating an example taxonomical classification of the visual special effects class 560 of digital in-game assets 500 which may be created for and deployed on the online-gaming platform 110, according to an aspect of the present disclosure;

FIG.5-6-A is illustrating conceptual example of representation of the options of the visual special effects class 560 of digital in-game assets 500 in GUI 590 of the online gaming platform 110;

FIG.5-7 is illustrating an example taxonomical classification of the audio class 570 of digital in-game assets 500 which may be created for and deployed on the online-gaming platform 110, according to an aspect of the present disclosure;

FIG.5-7-A is illustrating conceptual example of representation of the options of the audio class 570 of digital in-game assets 500 in GUI 590 of the online gaming platform 110;

FIG.5-8 is illustrating an example taxonomical classification of the modes class 580 of digital in-game assets 500 which may be created for and deployed on the online-gaming platform 110, according to an aspect of the present disclosure;

FIG.5-9 is illustrating an example taxonomical classification of the graphical user interfaces (GUI) class 590 of the digital in-game assets 500 which may be created for and deployed on the online-gaming platform 110, according to an aspect of the present disclosure;

FIG.6 is illustrating an example taxonomical classification of the versions of the game developed and run on the online-gaming platform 110, according to an aspect of the present disclosure;

FIG.7 is a flowchart illustrating the process of developing digital in-game assets 500 created for and deployed on the online-gaming platform 110, according to an aspect of the present disclosure;

FIG.8 is a flowchart illustrating the process of starting the gameplay by the player 250 on the online-gaming platform 110, according to an aspect of the present disclosure.

DETAILED DESCRIPTION

The detailed description set forth below is intended as a description of various configurations of the subject technology and is not intended to represent the only configurations in which the technology can be practiced.

The appended drawings are incorporated herein and constitute a part of the detailed description.

The detailed description includes specific details for the purpose of providing a more thorough understanding of the technology. However, it will be clear and apparent that the technology is not limited to the specific details set forth herein and may be practiced without these details. In some instances, structures and components are shown in block diagram form in order to avoid obscuring the concepts of the subject technology.

Techniques and technologies are described for creating, modifying, authenticating, transferring, and using unique digital assets associated with a videogame. The digital assets may be in - game digital assets, such as in – game items, characters, environments (settings), special effects, modes, GUIs, etc. The digital assets may be video game digital media assets with media representations of moments of gameplay of a video game, such as video clips, images, or audio clips.

The digital asset is created, traded, stored, and used on the same gaming online platform which provides adequate protocols, instruments, and ecosystems for such operations.

The digital asset may be provided with a unique identifier for the digital asset and metadata identifying properties of the digital asset.

Pending or completed changes to properties of the digital asset, such as ownership, visual appearance, or metadata, can be identified as a request to update the history of the digital asset stored on the online gaming platform.

The techniques and methods described herein expand the gaming experience for the board games by leveraging the opportunities provided by computational technologies and instruments associated with videogames, particularly the opportunities for unlimited customization of the plentitudes of digital in-game assets that create the general immersive environment of the game and perfect the aesthetic enjoyment of gameplay through artistic representations not available outside of computational virtual realms created in cyberspace.

Such computational technologies and instruments are also deployed for the practical implementation of techniques and methods of further modifying the rules and logic of the gameplay described herein without impeding the playability of the games, as would happen when similar modifications were attempted to be practiced outside of computational environment.

Such computational technologies and instruments are also deployed to advance the proficiency of the players and creators through advanced social communication and meaningful interaction with artificial intelligence.

Such computational technologies and instruments are also deployed to create new sources of revenue, advance economic welfare and increase levels of life satisfaction of the creators of the digital in-game assets, providers of auxiliary services, players, and collectors.

The techniques and technologies described herein expand the depth of the gameplay experience and functionality of digital in-game assets associated with the board games, and of systems that create and manage such digital assets, by basing the modifications of the games on the lore and time-proved logic of the fundamental paradigms of the game and meaningfully relating the digital assets to the gameplay.

Such relation of the digital assets to the game can include, for example, creation of the individual style of the player, specific mood and context of the gameplay, appreciation of the digital assets as a result of artistic value, technical functionality and history, and other aspects.

The techniques and technologies described herein shall provide sound basic for the further scientific research of the game theory, including theories of specific games, and improve our understanding of games phenomena through continuous modification and relational analysis of various modifications of gameplay and various versions of games.

Fig.1 illustrates an example computational network environment 100 in which the online gaming system associated with the games may be implemented, according to an aspect of the present disclosure.

The network environment 100 may include online gaming platform 110 consisting of game engine 120 that provides whole set of programming tools necessary for development of the videogames, digital in-game assets 500, and source code for implementing gameplay, digital in-game assets 500 storage vault 130 where digital in-game assets 500 are located and allocated to their contemporary owners, platform's ecommerce marketplace 140 where digital in-game assets 500 and auxiliary services are negotiated (sold, bought, rented, leased, licensed, exchanged, etc.), one or more interactive content servers 150 that provide online streaming content (e.g., interactive video, podcasts, videogame content, real-time gameplay, etc.), one or more platform servers 160, one or more data structures 170, plurality of application programming interfaces (APIs) 180 related to the functional modules of the platform 110, access control and management layer 190, one or more players' devices 193, one or more collectors and observers' (spectators) devices 195, one or more creators' devices 197, or a combination thereof.

The online gaming platform system 110 of the present disclosure has three major reasons of existence:

(1) provide beneficent internet videogaming environment and ecosystem for rich gaming experience and comfortable social interaction between the players and enthusiasts of the games hosted on the platform 110, and further advance research and understanding of the theories of the relevant games;

(2) provide efficient online software development environment where creators can design, produce and store their digital in-game assets 500, share their experience and expertise, and form collaborative development projects;

(3) provide reliable and vivid ecommerce marketplace associated with the game where players, creators, collectors, and specialists can trade and otherwise negotiate digital in-game assets and auxiliary services, interact socially, and collaborate professionally. Such a platform shall not only essentially and continuously advance the gaming experience, but also provide new beneficial venue of generating income and fostering economic welfare of the professionals and enthusiasts of the game.

Interactive content servers 150 may maintain, stream, and host interactive media available to stream on user devices 193, 195, 197 over a communication network.

Such interactive content servers 150 may be implemented in the cloud (e.g. , one or more cloud servers). Each media may include one or more sets of object data that may be available for participation with (e.g., viewing or interacting with an activity) by a user. Data about the object shown in the media may be stored by the interactive content servers 150, platform servers 160 and the user devices 193, 195, 197 in an object file.

The platform servers 160 may be responsible for communicating with the different interactive content servers 150, data structures 170, and user devices 193, 195, 197.

Such platform servers 160 may be implemented on one or more cloud servers.

Game engine 120, digital in-game assets library vault 130, platform marketplace 140, may be located on the interactive content servers 150 and platform servers 160.

The interactive content servers 150 may communicate with multiple platform servers 160 , though the media interactive content servers 150 may be implemented on one or more platform servers 160.

The platform servers 160 may also carry out instructions, for example, receiving a user request from a user to stream streaming media (i.e. , video games , activities , video , podcasts , User Generated Content (“ UGC ”) , publisher content, administrator content, etc.).

The platform servers 160 may further carry out instructions, for example, for streaming the streaming media content titles. Such streaming media may have at least one object set associated with at least a portion of the streaming media. Each set of object data may have data about an object (e.g., activity information, zone information, actor information, mechanic information, physics information, graphic information, game media information, etc.) displayed during at least a portion of the streaming media.

The streaming media and the associated at least one set of object data may be provided through an application programming interfaces (APIs) 180, which allows various types of interactive content servers 150 to communicate with different platform servers 160 and different user devices 193, 195, 197.

APIs 180 may be specific to the particular computer programming language, operating system, protocols, etc., of the interactive content servers 150 providing the streaming media content titles, the platform servers 160 providing the media and the associated at least one set of object data, and user devices 193, 195, 197 receiving the same.

In a network environment 100 that includes multiple different types of interactive content servers 150 (or platform servers 160 or user devices 193, 195, 197…), there may likewise be a corresponding number of APIs 180.

The user device 193, 195, 197 may include a plurality of different types of computing devices. For example, the user device 193, or 195, or 197 may include any number of different gaming consoles, mobile devices, laptops, and desktops. In another example, the user device 193, or 195, or 197 may be implemented in the cloud (e.g., one or more cloud servers) .

Such user device 193, or 195, or 197 may also be configured to access data from other storage media, such as, but not limited to memory cards or disk drives as may be appropriate in the case of downloaded services. Such devices 193, 195, 197 may include standard hardware computing components such as, but not limited to network and media interfaces, non - transitory computer - readable storage (memory), and processors for executing instructions that may be stored in memory. These user devices 193, 195, 197 may also run using a variety of different operating systems (e.g. , iOS , Android, etc.) and applications or computing languages (e.g. , C ++ , JavaScript, etc.).

The data structures 170 can include, for example, one or more databases (DBs), one or more one or more tables , one or more hash tables, one or more hash sets, one or more heaps , one or more trees, one or more lists, one or more arrays, one or more array lists, one or more dynamic arrays, one or more dictionaries, one or more matrices, one or more counts, and tools for organizing complex game data such as backgrounds, objects, levels, and characters, or a combination thereof.

The data structures 170 may be stored on the platform server 160, the interactive content servers 150, across one or more different servers, on a single server, across different servers, on any of the

user devices 193, 195, 197, on devices identified by network locations identified by pointers (e.g., uniform resource identifiers) stored in the distributed ledgers or other outside computational resources, or a combination thereof.

Such data structures 170 may store digital assets associated with video games, such as the streaming media, portions thereof, and associated set (-s) of object data. Such streaming media may depict one or more objects (e.g., activities) that a user can participate in and UGC (e.g. , digital in-game assets, screen shots, videos, commentary, mash ups, etc.) created by peers, creators, observers, publishers of the media content titles, third party publishers, and other legitimate parties.

Portions of the streaming media may include images, video clips, audio clips, or combinations thereof.

Digital assets may include metadata by which to search for such digital assets. Such digital assets may also include information about the media and peer. Such peer information may be derived from data gathered during peer interaction with an object of an interactive content title (e.g., a video game, interactive book , etc.) and may be bound to and stored with the digital asset.

Such binding enhances digital assets as the digital asset may deep link (e.g. , directly launch) to an object, may provide for information about an object and a peer of the digital asset , and may allow a user to interact with the digital asset.

One or more user profiles may also be stored in the data structures 170. Each user profile may include information about the user (e.g. , user progress in an activity and media content title, user id, user game characters, user gaming rating, user gaming statistics, digital in-game assets belonging to the user, digital in-game assets offered by the user for negotiation, etc.) and may be associated to media.

In some examples, an object and an object file are an example of a digital in-game asset associated with the video game of the gaming platform, such as in-game characters, objects, environments, settings, special visual and audio effects, audio files, AI analyzers, AI helpers, interfaces, toolbars, score bars, modes, etc.

In some examples, a portion of media, such as a video clip or image or audio clip of one or more moments of gameplay, is an example of a digital asset. The portion of media may be generated, recorded, and streamed using the interactive content servers 150, platform servers 160, and the user devices 193, 195, 197.

In some examples, such digital game assets can be controlled by, and restricted to, use for a single video game. In some examples, digital assets can be controlled by, and restricted to, use for a set of video games, such as a particular modifications and versions of the video game of the online gaming platform 110.

Fig. 2 illustrates conceptually the functional structure of the online gaming platform and types of users associated with each functional component.

The online gaming platform system 110 of the present disclosure includes three major functional components 200 defined by the reason of the platform existence described herewith above:

(1) runtime game engine component 210 represents that part of the game engine 120 which will include industrial-strength production interrelated systems and subsystems needed to

implement basic logic, physics, and mathematics of the game flow of the video game. Existing as a complex of interrelated software programs and code snippets comprising vast arrays of software development toolsets, programming code libraries, functional programming codes, data structures, algorithms, and software interfaces to implement them, the runtime component 210 provides and secures conditions of the software development and digital environment necessary to implement the gameplay online;

(2) game engine programming tool suite component 220 represents that part of the game engine 120, which will include industrial-strength production interrelated systems and subsystems needed to develop the versions of the game and digital in-game assets. Existing as a complex of interrelated software programs comprising vast arrays of software development toolsets, programming code libraries, functional programming codes, data structures, algorithms, and software interfaces to implement them, the programming tools suite component 220 provides and secures conditions of the software development environment necessary to develops high-quality in-game assets fully compatible with the runtime game engine component 210 and game engine 120 and ready to be integrated into gameplay of the associated version of the game on command of the players;

(3) marketplace 140 for the digital in-game assets, which is essentially one of the major systems channeled with the game engine where the administrator of the platform, independent creators, players and other groups of users can negotiate (buy, sell, exchange, lend, rent, license, pledge, etc.) their digital in-game assets compatible with the game, i.e. eligible for smooth importation into the game and problem-free use in the gameplay. The marketplace will also allow the administrator of the platform and independent providers to offer various auxiliary services such as bespoke creation and modification of in-game digital assets, training, mentoring, combinations and puzzle composing, etc.

The components 210, 220, 140 are located in one or more non-transitory memory storages coupled with one or more processors (e.g. implemented in circuitry). The one or more processors are configured to and can:

receive a request to gain access to the game/gameplay,

receive a request to gain access to start the game/gameplay,

receive a request to pause, continue and end the game/gameplay,

receive a request to gain access to the programming instruments and libraries,

receive a request to create, change, store, and transfer digital in-game assets,

receive a request to list, negotiate, and transfer digital in-game assets,

receive a request to access digital in-game assets,

receive a request to import digital in-game assets into gameplay session,

receive a request to create, change, access any user created content,

receive any other requests associated with the game, gameplay, and in-game digital assets.

As the administrator 240 of the online gaming platform 110 creates, maintains, develops, provides access and is responsible for functioning of all the components of the platform 110, they have full access to all the components of the online gaming platform 110.

Players 250 have access to the components and functions of the online gaming platform 110 which are needed to implement enjoyable gameplay, and thus they have, in various degrees defined by such purpose, access rights to all of the components of the system except for programming tools 220.

Creators (of digital in-game assets) 270 have access to programming tools 220 to a degree needed for creating digital in-game assets. Creators 270 also have access to platform marketplace component 140 through requesting administrator 240 to list their digital in-game assets in the marketplace 140 and, if the access is granted and the assets are listed, to negotiate such assets with players 250 and collectors 280.

The platform administrator 240 can provide access to the online platform marketplace 140 to:

players 250 to negotiate the digital in-game assets (buy, rent, get license for, or sell, exchange, lend or give sublicense to the digital assets belonging to them, etc.);

observers 260, who may not negotiate the assets but simply observe the content of the marketplace 140 for aesthetic or analytical reasons;

creators (of digital in-game assets) 270 to negotiate their digital assets on the primary and secondary markets (sell, buy, exchange, lend, provide license, etc.);

collectors 280, who may not play the game but simply collect digital in-game assets for aesthetic or economic reasons, for negotiating the digital in-game assets (buy, sell, exchange, lend, get or give license, etc.);

providers of auxiliary services 290 which may list and negotiate in the marketplace 140 their professional services and expertise associated with the game, such as training, consulting, composing, support, etc.

The roles of these groups can overlap and are fluid, as a collector 280 may also become a player 250 and a creator 270; a creator 270 may become player a 250 and a collector 280, etc. Thus the marketplace 140 of the online gaming platform 110 is the functional component of the platform 110 where all types of users get access to in various roles and to various degrees defined by the administrator 240 of the platform 110.

FIG . 3 is a block diagram illustrating a generalized example network environment 300 in which a gaming platform may be implemented, according to an aspect of the present disclosure.

In the example network environment 300 of FIG . 3, where example user devices 193, 195, 197 and example servers 390 may be varieties of platform servers 160 (e.g. , game engine programming tools server 391, game engine runtime server 392, digital in-game assets vault server 393, marketplace server 394, activity feed server 395, UGC server 396, an object server 397, and streaming server 398).

The user devices 193, 195, 197 may be implemented on the platform server 160, a cloud server or on any of the servers 390. The user devices 193, 195, 197 may further include a content recorder

310 and an object recorder 340, where content (e.g. , gameplay and digital game assets) may be uploaded and output through the user devices 193, 195, 197 . The interactive various content titles 300 may be executed on the user devices 193, 195, 197.

Alternatively, or in addition to, the content recorder 310 may be implemented on any of the platform servers 160, or a cloud server. Such content recorder 310 may receive and record content (e.g., a gameplay) from an interactive content title 300 (e.g., interactive content source servers 150) onto a content buffer 320. Such buffer 320 may store multiple content segments.

Concurrent to the content recorder 310 receiving and recording content from the interactive content title 300, an object library 330 may receive object data from the interactive content title 300, and an object recorder 340 may track the object data to determine when an object begins and ends. Such object data may be uploaded periodically or in real - time or close to real - time.

The object library 330 and the object recorder 340 may be implemented on the platform server 160, including any of the servers 390, or a cloud server.

When the object recorder 340 detects an object beginning, the object recorder 340 receives object data (e.g., if the object were an activity, user interaction with the activity, activity ID, activity start times, activity end times, activity results, activity types, etc.) from the object library 330 and records the activity data onto an object buffer 350. Such activity data recorded onto the object buffer 350 may be stored in an object file 380. For example, an object file 380 may store data regarding an item used during the activity.

Such object file 380 may be stored on the object server 397, though the object file 380 may also be stored on any server, a cloud server, or any user device 193, 195, 197.

Such object files 380 ... may represent digital in-game assets, as completed and ready to be used in the gameplay, so work in progress of developing such assets by the platform administrator 240 or creators 270, which may also be represented by draft files 360.

Fig. 4 illustrates the general example architecture of the engine runtime component 210. This is a large-scale, complex software system.

Modern game engines are some of the most complex software programs ever written, often featuring dozens of finely tuned complicated systems and subsystems interacting to ensure a precisely controlled user experience.

The continued evolution of game engines has created a strong separation between gameplay scripting, rendering, artwork, and level design. It is now common, for example, for a typical modern game development team to have several times as many artists as actual programmers.

In the broader sense of the term, game engines themselves can be described as middleware - a robust physics simulation system, along with a suite of tools and features (animation, behavior, virtual environment, and other applications) for game developers to build their games.

Regardless of specific game contents, while playing a game, a human player interacts with a virtual universe represented by the game during the gameplay, with its well-defined rules, objectives, limitations, and artistic expression. Such a virtual universe receives player's inputs and responds by changing its status.

Information regarding the outcome of the interaction is then conveyed to player (-s), and eventually analyzed by them to rationally decide what to do next to achieve their objectives. This conveyance means can be represented by changing properties of contextual game objects perceived by player (-s) as part of the virtual world of the game, or explicit informative interfaces, such as in-game graphic user interfaces (UI/GUI), etc.

This cycle is repeated iteratively, until the player wins, loses or draws the game, or simply decides to suspend temporarily their play session. The play experience interactive cycle is centered on a decision-making process that relies on the information conveyed to the player. Information is transmitted through visual (graphics and text), aural and even tactile means (in case the game relies on force-feedback interface devices).

The term gameplay refers to the action that takes place in the game, the rules that govern the virtual world in which the game takes place, the abilities of the player character (-s), known as player mechanics and physics, and of the other characters and objects in the world, and the goals and objectives of the player (-s).

Gameplay of various modifications and versions of the electronic versions of the board games may be typically implemented either in the native language in which the rest of the engine is written, or in a high-level scripting language, or both. To bridge the gap between the gameplay code and the low-level engine systems, the online gaming platform 110 may introduce a layer called a gameplay foundations layer 410 which provides a suite of core facilities upon which game-specific logic can be implemented conveniently and efficiently.

The gameplay foundations layer 410 introduces the notion of a game world, containing both static and dynamic elements. The contents of the world are modeled in an object-oriented manner. The collection of object types that make up a game may be called the in-game assets model. The in-game assets model provides a real-time simulation of a heterogeneous collection of objects in the virtual game world.

Typical types of in-game assets include: static background geometry, like buildings, roads, terrains, etc.; dynamic background environments; dynamic rigid bodies, such as rocks, trees, furniture, etc.; player characters (PC); non-player characters (NPC); semi-player-characters (SPC); weapons; projectiles; vehicles; lights which may be present in the dynamic scene at a run time, or only used for static lighting off line; special effects, in-game cameras; and so on. The digital in-game assets 500 specific to aspect of the present disclosure are described in detail later.

The game world model is intimately tied to a software object model, and this model may be pervading the entire engine. The term software object model refers to the set of language features, policies, and conventions used to implement a piece of object-oriented software. In the context of the present disclosure, the software object model defines such aspects as: game engine design in an object-oriented manner; programming languages used; organization of the static class hierarchy (one monolithic hierarchy, plentitude of loosely coupled components, or combination of the both); templates and policy-based design, traditional polymorphism, or combination of the both; reference system to in-game assets (straight pointers, smart pointers, handles, etc.); identification system for in-game assets (by address in memory only, by name, by a global unique identifier (GUID), etc.); management of the lifetimes of in-game assets; temporal simulation of the states of the in-game assets; etc.

Gameplay foundations component 410 includes event/message system. Game objects invariably need to communicate with one another. This can be accomplished in various ways. For example, the object sending the message might simply call a member function of the receiver object. An event-driven architecture, typical for graphical user interfaces, may also be deployed for inter-object communication. In an event-driven system, the sender creates a little data structure called an event or message, containing the message's type and any argument data that are to be sent. The event is passed to the receiver object by calling its event handler function. Events can also be stored in a queue for handling at some future time.

Gameplay foundations component 410 also contains scripting system to make development of game-specific gameplay rules and content easier and more rapid through elimination of the need to recompile and relink game executables every time a change is made to the logic or data structures used in the engine. As a scripting language is integrated into the game engine 120, changes to game logic and data can be made by modifying and reloading the script code with much faster turn-around time. The runtime game engine 210 will allow script to be reloaded while the games continue to run on the online gaming platform 110, the games need not to be shut down prior to script recompilation.

Gameplay foundations component 410 also contains artificial intelligence system which is AI engine upon which game-specific logic and protocols for AI characters and functions can be developed. An architecture of the AI decision layer includes, but not limited to the concept of imitating thinking activity in tactical, operational and strategic decision-making, agent functions each of which is responsible for executing a specific task, such as moving from point to point, expressing emotions, etc.), and functions of actions responsible for allowing the character to perform a fundamental movement, which often results in playing animations on the character's skeleton.

Because most of the versions of the game will predictably have digital in-game assets in the form of organic or semi-organic characters (humans, animals, cartoon characters, sci-fi characters, fantasy characters, cyborgs, robots, other) the runtime engine 210 needs animation systems. There may be five or more basic types of animation used in games: sprite/texture animation, rigid body hierarchy animation, skeletal animation, vertex animation, and morph targets.

Skeletal animation system 412 permits a detailed 3D character mesh to be posed by an animator using a relatively simple system of bones. As the bones move, the vertices of the 3D mesh move with them.

The animation system 412 produces a pose for every bone in the skeleton, and then these poses are passed to the rendering engine 418 as a palette of matrices. The renderer 418 transforms each vertex by the matrix or matrices in the palette to generate a final blended vertex position in the process known as skinning. The renderer 418 and the animation system 412 may be bridged by skeletal mesh rendering subsystem.

Skeletal animation system 412 may also be tightly coupled with collision and physics system 421 through process known as rag doll animation, when bodily motions of limp animated character are simulated by the physics system 421. The physics system 421 determines the positions and orientations of the various parts of the body by treating them as a constrained system of rigid bodies. The animation system 412 calculates the palette of matrices required by the rendering engine 418 in order to draw the character on-screen.

Like most games, the game versions of the online gaming platform 110 will employ various kinds of 2D graphics overlaid on the 3D scene for various purposes, such as the game's heads-up display (HUD), in-game menus, a console, and other development tools, an in-game graphical user interface (GUI), allowing the player to manipulate his or her character's inventory, configure units for battle, and perform other pre-game and in-game tasks. The engine may also have the full-motion video (FMV) system responsible for playing full-screen movies that have been rendered with the game's rendering engine or using another rendering package. A related system is the in-game cinematics (IGC) system. This component may allow cinematic sequences to be choreographed within the game itself, in full 3D. For example, as the player avatar or semi-player character moves through a game world, a conversation between two or more key characters might be implemented as an in-game cinematic. IGCs may or may not include the player character(s). They may be done as a deliberate cut-away during which the player has no control, or they may be subtly integrated into the game without the human player even realizing that an IGC is taking place. All these functions are handled in the front end component 411 of the runtime game engine 210.

Figures from 4-1 to 4-6 further detail the example subsystems of the engine runtime component 210, with layered design.

Fig. 4-1 illustrates exemplary set of game specific subsystems 413, which lays on top of the gameplay foundation layer and the other low-level engine components to allow gameplay programmers and designers cooperate to implement the features of the game itself. Gameplay systems are numerous, highly varied, and specific to the version and embodiment of game being developed, and may include, but not limited to, the mechanics of the player and semi-player characters, various in-game camera systems, artificial intelligence for the control of non-player characters (NPCs), weapon systems effects, environment modelling, and so on. If a clear line could be drawn between the engine and the game, it would lie between the game-specific subsystems and the gameplay foundations layer, though this line is never perfectly distinct.

Fig. 4-2 further illustrates several key modules of the runtime game engine 210. Visual effects module 414 of the engine will support a plentitude of various visual effects such as, but not limited to: particle systems (for smoke, fire, water splashes, etc.), decal systems (for bullet holes, foot prints, etc.), light mapping and environment mapping, dynamic shadows, full-screen post effects (applied after the 3D scene has been rendered to an off screen buffer) such as a high dynamic range (HDR) lighting and bloom, full-screen anti-aliasing (FSAA), color correction and color-shift effects, including bleach bypass, saturation and de-saturation effects, etc.

In other words, the system of visual effects module 414 of the runtime game engine 210 manages the specialized rendering needs of particles, decals, and other visual effects. The particle and decal subsystems may be distinct components of the rendering engine and act as inputs to the low-level renderer 418. Light mapping, environment mapping, and shadows may be handled internally within the rendering engine proper. Full-screen post effects may be either implemented as an integral part of the renderer or as a separate component that operates on the renderer's output buffers.

Scene graph culling / optimization module 415 is a higher-level component needed to limit the number of primitives submitted for rendering, based on designated form of visibility determination. Depending on the size of the game world of the particular version of the game, it may use simple removal of excessive objects from the world-camera view, and advanced spatial

subdivision data structures, and portal or occlusion culling methods to improve rendering efficiency.

Many versions of the games permit multiple human players to play within a single virtual world, or simultaneous different visualizations of the same virtual world. This in-game phenomena may be called simultaneous multiverse, and happens, for example, when a player selects for his version of the game world a representation of his opponent or opponent characters different from the representations selected by his opponent. Such instances are described in greater detail in section related to the digital in-game assets 500, particularly to assets class kingdom 510 classes 520 player-token, 530 non-player characters, 540 semi-player characters.

Multiplayer versions of the game may come in at least four basic variants:

(1) Single-screen multiplayer when two or more human interface devices (joypads, keyboards, mice, etc.) are connected to a single machine, PC, console, or other apparatus. Multiple player characters inhabit a single virtual world, and a single camera keeps all player characters (player tokens) in frame simultaneously;

(2) split-screen multiplayer when multiple player characters inhabit a single virtual world, with multiple HIDs attached to a single game machine, but each with its own camera, and the screen is divided into sections so that each player 250 can view his or her character (player token);

(3) networked multiplayer when multiple computers or other playing devices are networked together, with each machine hosting one of the players 250;

(4) massively multiplayer online games (MMOG), when large numbers of players 250 can be playing simultaneously within a giant, persistent, online virtual world hosted by a powerful battery of central servers, such as, for example, platform servers 390 - such variant may be presented in “all-vs-all” or “battle royale” versions of the game described herewith below. Support for multiple players 250 can have a profound impact on the design of certain game engine components. Such support will be provided by online multiplayer system 416.

No great game is complete without a stunning audio engine as audio is just as important as graphics in any game engine. Audio engine system 417 may be used by the online gaming platform administrator 240, creators 270 and providers of auxiliary services 290 to develop proper in-game audio assets. Even with powerful pre-existing audio engines, high-quality in-game audio asset requires a great deal of custom software development, integration work, fine tuning, and attention to detail.

Fig. 4-3 also illustrates example low-renderer 418 which encompasses all the raw rendering facilities of the runtime game engine 210. At this level, the design is focused on rendering a collection of geometric primitives as quickly and richly as possible, without much regard for which portions of a scene may be visible. Low-renderer 418 may include such components as graphics APIs used to manage the graphics devices (graphic device interfaces); components used to collect submissions of geometric primitives (render packets), such as meshes, line lists, point lists, particles , terrain patches, text strings, and other, and render them as quickly as possible; camera-to-world matrix and 3D projection parameters, such as field of view and the location of the near and far clip planes which provide viewport abstraction; material system and its dynamic lighting systems which manage the state of the graphics hardware and the game’s shaders; etc.

The low-renderer 418 relates to profiling and debugging module 419.

Profiling and debugging module 419 allow to profile performance of real-time versions of the game to optimize the performance. Profiling and debugging module 419 also encompasses memory analysis tools and in-game debugging facilities, such as debug drawing, an in-game menu system or console, and the ability to record and play back gameplay for testing and debugging purposes. Module 419 may also include a mechanism for manually instrumenting the code for timing specific sections of the code; a facility for displaying the profiling statistics on-screen while the game is running; a facility for dumping performance stats to a text file, an Excel spreadsheet, or other format; a facility for determining how much memory is being used by the engine, and by each subsystem, including various on-screen displays; the ability to dump memory usage, high-water mark, and leakage stats when the game terminates or during gameplay; tools that allow debug print statements to be peppered throughout the code, along with an ability to turn on or off different categories of debug output and control the level of verbosity of the output; the ability to record game events and then play them back.

Human interface devices (HIDs) system 420 serve to obtain, preprocess, and transfer various inputs from the players 250 to process in the gameplay, from the creators 270 to process in the in-game assets development process, from observers 260 to process during their surfing and observations, or from all categories of users during their browsing and negotiating in the online platform marketplace 140 and in-platform social activities. HIDs may include, but not limited to the keyboard and mouse, a joypad, other specialized game controllers, cybernetics wearables, and other devices. Such HIDs 420 may also be called the player I/O component, because they may also provide output to the player through the HID, such as force feedback /rumble on a joypad, the audio feedback, etc. HIDs systems may provide subsystems allowing the player to customize the mapping between physical controls and logical game functions, detect chords (multiple buttons pressed together), sequences (buttons pressed in sequence within a certain time limit), and gestures (sequences of inputs from the buttons, sticks, accelerometers, etc.).

Fig. 4-4 demonstrates Collision and physics module 421 and its direct coupling with skeletal animation module 412. Collision and physics module 421 is concerned with design and implementation of the motion of rigid body objects and the forces and torques dynamics which cause this motion to occur which is important for the scenery of many board game.

Fig. 4-5 further illustrates two important modules of the runtime game engine 210 - game resources module 422 and core game systems module 429.

Game resources 422 are all types of game assets and other engine input data, which may include, but not limited to, 2D and 3D models, textures, materials, fonts, skeletons, collision detection and physic dynamics, maps, etc. These assets and inputs are accessed through a suite of unified interfaces collectively called resource manager.

Core systems 429 are software utilities designated to optimize operation of the runtime engine 210. Such utilities may include, but not limited to, error-checking codes used to locate logical mistakes and violations of programmer's original assumptions; memory allocations systems; mathematical libraries providing facilities for vector and matrix math, quaternion rotations, trigonometry, geometric operations with lines, rays, spheres, frusta, etc., spline manipulation, numerical integration, solving systems of equations, and other mathematical facilities required in game programming; a suite of tools for managing fundamental data structures (linked lists, dynamic arrays, binary trees, hash maps, etc.) and algorithms (search, sort, etc.) to minimize or eliminate dynamic memory allocation and to ensure optimal runtime performance; other similar systems.

Fig. 4-6 demonstrates several core lower-levels systems of the runtime engine 210.

Platform independence layer 424 is a complex of software systems that ensures the smooth development of digital in-game assets and problem-free runtime of the gameplay across multiple third-party software platforms.

The online gaming platform 110 will leverage a number of third-party software development kits (SDKs) 425. The functional or class-based interfaces provided by an SDKs 425 may also be called application programming interfaces (APIs). Such SDKs (APIs) 425, some of specific instances of which have been earlier mentioned in Fig.1 as APIs 180, may include, but not limited to data structures and algorithms programming libraries, graphics hardware interface libraries, collision detection and rigid body dynamics libraries and packages (also known collectively as 'physics'), character animation packages, artificial intelligence packages, etc.

The operating system (OS) 426 of the runtime engine 210 is running all the time as it orchestrates the execution of multiple programs on a single user device 200, or 210, or 220, some of which are associated with game.

Device drivers 427 are low-level software components provided by the operating system 426 or hardware vendor. Drivers manage hardware resources and shield the operating system 426 and upper layers of online gaming platform 110 from the details of communicating with the myriad variants of hardware devices available.

The user hardware layer 428 represents the computer systems or consoles on which the game will run. Typical platforms include Microsoft Windows- and Linux-based PCs, mobile devices, the Apple iPhones and Mac computers, Microsoft 's Xbox and Xbox 360, Sony's PlayStation versions, and other computational and communicational devices. The game is platform-agnostic.

Any ludic activity involves the interaction with concrete or abstract objects.

The use of the objects, both in terms of modes and purposes, and their relationships are regulated by rules, which organize a set of ludic activities and turns it into a complete and coherent game.

In-game assets, with all their aesthetic content, in any game, and especially in the game of the present disclosure, enrich the gameplay by creating "double-play" effect. In addition to enjoying the process of the bare gameplay represented by pure planning, analysis, tactical, operational and strategic decision-making regulated by the strict set of rules of the game, which does not depend on the artistic implementation of the in-game assets, the in-game assets create appealing and motivating opportunities for the players to extend the mental scope of the game by creating specific virtual environments fitting individual tastes, preferences and moods of the human players.

For example, the player may enjoy artistic embodiment of their player character player-token (avatar). Equally, they may enjoy artistic embodiment of the player-token of their opponent (-s) and interactions of their players tokens with tokens of their opponents in course of the gameplay. The artistic implementation of the non-player or semi-players characters of the player and their opponent (-s) further enhances the aesthetic satisfaction, suspense, immersion, emotional involvement, excitement and thus overall satisfaction from the gameplay. The artistic implementation of the setting surrounding the gameplay and creating the virtual environment where the core actions take place are no less important for the overall game satisfaction that player-tokens or non-player/semi-player characters. Such is the universal nature of the double-

play – for human players, it cannot be reduced to the sanitized levels of abstraction. The plot of the movie may be highly abstract, but plot is by far not the only reason why we watch movies. In this sense, videogames are very similar to the cinematography and other dynamic visual arts. Even in table games traditionally embodied in the highly abstract assets, such as traditional checkers or Staunton set of pieces in chess, or dices, or playing cards in card games, the aesthetic component is normally layered by the real surroundings where the gameplay takes place and mental imageries proceeding in the consciousness and subconsciousness of the players, whether they realize it or not. Playing Western chess on the beach in Florida with friends is quite different experience from playing chess with the President of United States in the White house. Whilst the rules of the game are same, the overall experience may be different.

The artistic form of the in-game assets may be considered by some of the orthodox human players as a factor distracting from the mathematics of the gameplay itself (and the online gaming platform administrator will surely accommodate such purists them with a choice of classical minimalist version of the game with ascetic design of digital in-game assets 500 making it just possible to comfortably operate the gameplay in accordance with rules, in “black and white”, or “red and black”). Such stance has its merit, but not more than motion pictures may be considered as distracting compared to the books evaporated to the very rational essence of the plot of script (synopsis they call it sometimes) or conventional artistic implementation of the first-person shooter games may be considered distracting for shooting the moving bull-eye target in empty space from nothingness-gun. For those acquainted with the art it is immediately obvious that artistic implementation of the in-game assets is a crucial component of successful video game experience. Even in the example of brick-and-mortar table game of chess in real-world there is a phenomenon of the market for art design of chess board and pieces, with offers ranging from mass-market sets on AMAZON and eBay to highly artistic products with prices of dozens of thousands of the U.S. Dollars. Thus, one of the missions of the online gaming platform 110 will be to converge the fascinating science of the game with the artistic expression of it. Nothing can do it better than properly produced digital videogame.

Fig. 5 is a conceptual diagram illustrating an example taxonomical classification of digital in-game assets (DIAs) 500 which will be opened for the creative development of the creators 270, to be further negotiated in the platform marketplace 140 for the purposes of earning income, deployment in the gameplay and collecting.

Such DIA 500 may be created by administrator 240 or the creators 270 of the online gaming platform 110 in form of data files fully compatible with the data structures, commands, rules and procedures of the game in the online gaming platform 110 and may be associated with a player 250 entity with behaviors in the game, and can be player controlled or game - controlled , and can change dynamically during gameplay; or may be not associated with player 250 entity with behaviors in the game but be player-independent entity to create static or dynamic environment, or separate element (-s) of environment, where the game takes place and can be player controlled or game - controlled , and can change dynamically during gameplay. Such data files of DIA 500 may include a character ID for the DIA 500, a localizable name for the DIA 500, an image of the DIA 500, and a short description of the DIA 500. Such data files of DIA 500 may be associated with an asset select event that indicates that the player's selected DIA 500 have changed. The selected data files of DIA 500 may represent the characters the player 250 is controlling in the game and may be displayed on the player's profile and other spaces via the online gaming platform 110. There may be more than one of the DIA 500 selected at a time and each game may replace its list of the DIA 500 upon loading save data .

Players' interactions with a virtual world of the game are mediated by the digital in-game asset most important for player of any game: the player-token. Fig. 5-1, 5-2.

Such player-token can be a true avatar, thus representing the embodiment of the player in the virtual world (as it happens in games Quake and the like), or it can be an "invisible hand" (as it happens in games Tetris and the like), or it can be a hybrid (as it happens in games Sim City, Assassins' Creed, Tomb Raider, Warcraft, League of Legends and the like).

In all cases, the player-token receives inputs from the human player and mediates all the attempts of interaction with the rest of the virtual world.

The rules that govern the player-token determine what the player can do with it, and with the rest of the virtual world through it.

In other words, the player-token is the key to what is called "core gameplay" in the game design jargon.

The online gaming platform 110 will provide as free-of-charge, so payable player-tokens 520 to use in the gameplay. Such payable player-tokens 520 can be either bought, bartered, or rented directly from the online gaming platform 110 libraries or through the marketplace 140 of the online gaming platform 110.

Creators 270 will be provided with the choice of creating player-tokens 520 using the online gaming platform 110 toolsets. Such a process is schematically shown in Fig. 7.

The online gaming platform 110 will allow the human players 250 to choose their player-tokens 520 either from the public libraries of the online gaming platform 110 or from their own libraries. The players 250 always have a choice not to use any player token 520 at all, like the most of the contemporary digital and online versions of the game do now, which provide neither option, nor a single variant of player-token 520 except for the static or quasi-animated avatars of the players.

Further, the human players 250 will also be able to choose what would be the visual representation of their opponent (-s). The players 250 will have a choice to see during the gameplay session either the player-tokens 520 chosen by their opponents, or to select visual representation of their opponents on the player 250 individual screens of video game interfaces on their devices 193 by other player-token 520 representations from the platform's libraries or from their own libraries similar to how they choose their personal player-tokens 520 for the play. Thus two players may see the their visual representations in the world of game in different ways, in multiverse mode. The opponent (-s) shall not necessarily know about the choices of the players they oppose. After all, we all see the world differently, so we can provide an additional degree of individual comfort for the scenery of our version of the surrounding world, at least in the virtual world of gameplay, without disturbing our counterparts too much by own choices.

Fig.5-2-A is a conceptual diagram illustrating an example video game interface 521 before the start of the gameplay in which a player 250 may select their visual representation in the gameplay by player-token 520, according to an aspect of the present disclosure. The visual interface 521 may prompt the player 250 to select the player-token 520 from the list available to the player 250 as shown by the radio buttons 522. After the player 250 selects player-token 520 they want to represent them visually in the gameplay (in the example this is player-token 520-1 based on Garry Kasparov image), the video game interface 521 switches to window 521-1 showing the selected player-token 520-1.

Fig.5-2-B is a conceptual diagram illustrating another example of video game interface 521 before the start of the gameplay in which a player 250 may select their visual representation in the gameplay by player-token 520, according to an aspect of the present disclosure. The visual interface 521 may prompt the player 250 to select the player-token 520 from the list available to the player 250 as shown by the radio buttons 522, similar to Fig.5-2-B. After the player 250 selects player-token 520 (in the example this is player-token 520-2 based on Gandalf character image), the video game interface 521 switches to window 521-1 showing the selected player-token 520-2 which will represent them visually in the gameplay.

Fig.5-2-C is a conceptual diagram similar to Fig. 5-2-A and Fig.5-2-B illustrating another example of video game interface 521 before the start of the gameplay in which a player 250 may select their visual representation in the gameplay by player-token 520, according to an aspect of the present disclosure. The visual interface 521 may prompt the player 250 to select the player-token 520 from the list available to the player 250 as shown by the radio buttons 522. After the player 250 selects player-token 520 (in the example this is player-token 520-3 based on Joseph Stalin artistic image), the video game interface 521 switches to window 521-1 showing the selected player-token 520-3 which will represent them visually in the gameplay.

Fig.5-2-E is a conceptual diagram illustrating an example video game interface 521 before the start of the gameplay in which a player 250 is notified about the player-token 520-4 which was selected by his opponent to represent the opponent in the incoming gameplay. The game interface 521 may provide the option to the player 250 through radio button 522-1 to change the visual representation of the opponent in the version of the game flow which will be seen by the player 250. The player 250 may further choose alternative player-token 520 through option radio buttons 522. In the example the player 250 chooses play-token 520-5 based on image of Napoleon. After the player 250 has made the choice, the game interface 521 switches to window 521-2 showing the selected player-token 520-5 which will represent the opponent of the player 250 in their version of the game play world.

Fig.s 5-2-F through 5-2-J are conceptual diagrams illustrating various examples of game interfaces 521 before start or during the gameplay showing the visual representation of the opponent of the player 250 (521-6) and the player 250's own visual representation (521-7) in forms of player-tokens 520 digital in-game assets. Such player-tokens 520 may be animated, backed by AI, and in a variety of other forms in accordance with the taxonomical classification provided in Fig. 5, 5-1, 5-2.

The human players 250 may be provided with additional options of digital in game assets 500 such as, for example, placing a virtual mirror behind the back of their opponent's player-token 520 in virtual environment 550 where the chosen settings allow such arrangement to see during the gameplay not only their opponent (-s) tokens 520 in the virtual environment 550, but also the visual representation of their own play-tokens in the same environment 550 as a reflection in the mirror during the gameplay.

As shown in Fig. 5-1, 5-2, the variety and taxonomy of player-token 520 embodiments is essentially limitless within the technical requirements and specifications imposed by the online gaming platform 110 to ensure smooth import and functioning of the player-tokens 520 in the game. For example, player-tokens 520 can be implemented with various degrees of artistic value and craftmanship; various degrees of realism; various degrees of detail; various degrees of

animation; various degree of contextual emotional reaction, which will be especially important for the player-tokens 520 representing AI in the game, i.e., non-human players 530 – for example, the player-token 520 can express sadness, meaningful anger of frustration after the unfavorable move or loss of controlled piece, happiness and merriment after successful move, or, alternatively, maintain demeanor of "poker face", etc.

Player-tokens 520 may be implemented as in humanoid, so not in humanoid style (animals, fictional creatures, etc.).

Player-tokens 520 may be implemented as personification of characters from various historical epochs, ethnicities, sexes, ages, occupations, artistic genres, psychological and behavioral architypes, etc.

For example, player-tokens 520 may be implemented as personification of famous historical figures, legendary creatures, etc.

For example, player-tokens 520 may be implemented as virtual personifications of the real human-players depending on their characteristics, preferences, tastes, and perceptions of themselves.

For example, player-tokens 520 may be implemented on demand as personification of particular persons or creatures familiar and significant to the human players, such as friends, relatives, colleagues, neighbors, supervisors, bosses, real-life opponents or adversaries, pets, etc. Such player-tokens may be created on demand by creators 270 based on the orders and visualization materials (personal photos, videos, descriptions, specifications, etc.) of players 250 and collectors 280.

Depending on the decision of the creator 270 of the player-token 520 of the digital in-game assets 500, the asset may be serial (identical copies of the asset will be sold in the marketplace 140 of the online gaming platform 110 without quantitative limitations or en masse), limited edition (only limited quantity of copies will be sold in the marketplace 140 of the online gaming platform 110) or unique (only one copy of the game asset will be offered for sale in the marketplace 140 of the online gaming platform 110), or having other attributes of rarity.

Purchasing, exchange, lending, licensing, and other negotiation of the player-tokens 520 will be implemented in the marketplace 140 of the online gaming platform 110. Only those play-tokens 520 which are admitted for the negotiation in the marketplace 140 of the online gaming platform 110 by the platform administrator 240 will be allowed to be imported into and used in the games on the online gaming platform 110.

FIG.5-3 illustrates an example of the general taxonomical classification of nonplayer and semi-nonplayer characters (pieces) 530 of the digital in-game assets to be used in the games on the online gaming platform 110.

One of the specific features of the digitization of the board games is that during the gameplay the opposing players 250 manipulate not only their player-tokens 520, but also a variety of other in-game characters or pieces, primarily the pieces whose movements pronounce the results of the decision-making of the controlling players 250 and are meaningful for the game, as in part of the logic of the game, so in part of its aesthetic expression.

As the human player 250 manipulates some of the in-game pieces under their control in accordance with the protocols of the game, such pieces partially become functional embodiments

of the players 250, though in a degree and aspects different from personal player-token 520. This is why we call the characters commanded by a human player 250 semi-player characters 540.

The characters which are not manipulated by any of the opposing human players 250 are called nonplayer characters 530 and may be manipulated by in-game AI as part of the game-specific systems 413 of the online gaming platform 110.

In terms of the variety and taxonomy of artistic expression, the implementation of the nonplayer characters 530 and semi-player characters 540 are identical as to each other, so to player-tokens 520, with the only distinction that functions and capacities of the nonplayer characters 530 and semi-player characters 540 are defined in the protocols of the gameplay, and such functions and capacities do cause requirement of distinct visualization of the particular characters/pieces depending on their functions and capacities with the aim to facilitate easy differentiation between the characters 530 and 540 by the human players 250 to make the game comfortably playable.

Accordingly, the taxonomical classification laid out in FIG. 5, FIG. 5-1, FIG. 5-2 for player-tokens 520 is equally applicable to the non-player characters 530 and semi-player characters 540, as illustrated by FIG. 5-3, FIG.5-4.

Some examples of possible implementation of the nonplayer characters class 530 and semi-nonplayer characters class 540 of the digital in-game assets 500 are presented in FIG. 5-3-A, FIG. 5-3-B, FIG. 5-3-C. Such examples are equally applicable to semi-player characters 540 of the digital in-game assets 500.

Similar to as they may do the selection of the player-tokens 520 described above and illustrated by examples in FIG.5-2-E, the human player 250 can either accept the visual implementation of the nonplayer characters 530 and semi-player characters 540 proposed by their AI or human opponent (-s) or choose other available DIA 500 of the nonplayer characters 530 and semi-player characters 540 which fit their preferences better to ensure the desirable level of comfort and enjoyment in gameplay. Thus, the opposing players 250 can simultaneously see different visualizations of the nonplayer characters 530 and semi-player characters 540 depending on the DIA 500 chosen by each player 250, though other functional attributes of such DIA 500 in the gameplay, for example the real-time states of the characters 530 and 540, their positional locations, movements and outcomes of other actions involving and effecting such characters 530 and 540 will be seen by the opposing characters in the same fully synchronized view of the game world.

Depending on the decision of the creator 270 of the nonplayer characters 530 and semi-player characters 540 DIA 500, the asset may be serial (identical copies of the asset will be sold in the marketplace 140 of the online gaming platform 110 without quantitative limitations or en masse), limited edition (only limited quantity of copies will be sold in the marketplace 140 of the online gaming platform 110) or unique (only one copy of the game asset will be offered for sale in the marketplace 140 of the online gaming platform 110), or having other attributes of rarity.

Purchasing, exchange, lending, licensing, and other negotiation of the nonplayer characters 530 and semi-player characters 540 DIA 500 will be implemented in the marketplace 140 of the online gaming platform 110. Only those nonplayer characters 530 and semi-player characters 540 DIA 500 which are admitted for the negotiation in the marketplace 140 of the online gaming platform 110 by the platform administrator 240 will be allowed to be imported into and used in the game.

FIG. 5-5 shows example of general taxonomical classification of environments (settings) 550 of the digital in-game assets 500.

The environments, or settings of the game, which create the virtual environment in which the gameplay takes place, may be implemented in an infinite variety of artistic and thematic choices.

Such implementations can vary by the degree of artistic value, complexity, sophistication, detail, amination, realism, level of interactivity and responsiveness to players' actions. Thematical contexts can also vary greatly, and be represented by such choices as, for example, interiors of various epochs, social status locations and territories (ancient palaces, Egyptian deserts, medieval castle, royal courts, Victorian fireplace, scenic vistas, etc.), battlefields of various epochs, as historical so fictional (as on land, so in the sea, underground, submerged, in the air, in the space, etc.), various genres (realism, surrealism, fantasy, sci-fi, etc.), various degrees of expansion (limited spaces, limited worlds, limited universes, unlimited worlds, etc.).

Some examples of possible implementations of the environments (settings) class 550 of the digital in-game assets 500 are presented in FIG. 5-5-A, FIG. 5-5-B, FIG. 5-5-C.

Similar to as they may select to do in case of the player-tokens 520, nonplayer characters 530, or semi-nonplayer characters 540, human players 250 may either assent to the choice of settings 550 for the gameplay made by their opposing players, or to choose their own settings 500 among available options depending on personal preferences. Thus, different human players 250 can see different settings during the same gameplay session which will not affect the synchronicity of the gameplay for various players – different players will see different environmental versions of the same gameplay world at the same time, but the flow of the game will be fully synchronized.

Depending on the decision of the creator 270 of the environments (settings) in-game asset 550, the asset may be serial (identical copies of the asset will be sold in the marketplace 140 of the online gaming platform 110 without quantitative limitations or en masse), limited edition (only limited quantity of copies will be sold in the marketplace 140 of the online gaming platform 110) or unique (only one copy of the game asset will be offered for sale in the marketplace 140 of the online gaming platform 110), or having other attributes of rarity.

Purchasing, exchange, lending, licensing, and other negotiation of the environments (settings) digital in-game assets 550 will be implemented in the marketplace 140 of the online gaming platform 110. Only those environments (settings) in-game asset 550 which are admitted for the negotiation in the marketplace 140 of the online gaming platform 110 by the platform administrator 240 will be allowed to be imported into and used in the game.

Special effects in the game, as visual 560 so audible 570, are essentially implements for the environments (settings) 550 designed to enrich the playing experience emotionally and aesthetically, and as such can be coupled either to play-tokens 520, nonplayer characters 530 or semi-player characters 540, or environments (settings) 550. Visual special effects 560 and audio 570 audio digital in-game assets may be offered either in bundle with any of the other related digital in-game assets 500, or imported into game separately. Such visual special effects 560 and 570 audio digital in-game assets can include visualization and sounds of lighting and sea, forest, shadowing, weather conditions, day-night conditions, elements (sunshine, clouds, rain, wind, tornado, etc.), urban, military, battle, etc.

The variety of implementations of visual special effects 560 and audio digital in-game assets 570 is virtually limitless, within the requirements imposed by the administrators 240 of the online

gaming platform 110 to ensure aesthetical, ethical and technical compatibility of the visual special effects 560 and audio 570 audio digital in-game assets with the games.

General taxonomical classification of the visual special effects in-game assets 560 is presented in Fig. 5-6.

Fig.5-6-A is a conceptual diagram illustrating example options menu in graphical user interface 590 of the game with options of visual special effects 560 which may be available to accompany the game play.

General taxonomical classification of the audio digital in-game assets 570 is presented in Fig. 5-7.

Fig.5-7-A is a conceptual diagram illustrating example options menu in graphical user interface 590 of the game with options of audio 570 digital in-game assets which may be available to accompany the game play.

Similar to as they may do with the player-tokens 520, nonplayer characters 530, or semi-nonplayer characters 540, environments (settings) 550, human players 250 can either assent to the choice of visual special effects 560 and audio 570 in-game assets by their opposing players, or to switch them off for their version of the virtual world of the gaming session, or choose their own settings among available options depending on personal preferences. Thus, different human players 250 can see different settings during the same gameplay session which will not affect the synchronicity of the gameplay for various players – different players will see different environmental versions of the same gameplay world at the same time, but the flow of the game will be fully synchronized.

Depending on the decision of the creator 270 of visual special effects 560 or audio 570 in-game asset, the asset may be serial (identical copies of the asset will be sold in the marketplace 140 of the online gaming platform 110 without quantitative limitations or en masse), limited edition (only limited quantity of copies will be sold in the marketplace 140 of the online gaming platform 110) or unique (only one copy of the game asset will be offered for sale in the marketplace 140 of the online gaming platform 110), or having other attributes of rarity.

Purchasing, exchange, lending, licensing, and other negotiation of the visual special effects 560 or audio 570 in-game assets will be implemented in the marketplace 140 of the online gaming platform 110. Only those in-game assets which are admitted for the negotiation in the marketplace 140 of the online gaming platform 110 by the platform administrator 240 will be allowed to be imported into and used in the game.

Game modes 580 of the digital in-game assets 500 are snippets of code (software programs or data structures) which modify either mechanics, rules, and protocols of gameplay, or certain presentational (visual or audio) aspects of the game.

FiG.5-8 illustrates general taxonomical classification of game modes 580 digital in-game assets.

The online gaming platform 110 will provide two major modes of each version of the game: in 2D graphics and 3D graphics for eligible versions of the games, the latter including subclasses of augmented reality and virtual reality. These major modes establish two major classes for all versions of the game, including the version with 3D mechanics of the game such as Kubikschach, Raumschach and other original non-videogame 3D versions. These major modes will be supported by the platform's 110 runtime engine 210 and represent one of the primary factors

defining the compatibility of the in-game assets 500 with the chosen major mode of a game, as not all 2D in-game assets 500 can be imported into and used in 3D modes of the games, and not all of the 3D in-game assets 500 can be imported into and used in the 2D modes of the games.

Within the limits established by the platform administrator 240, the game modes 580 of the digital in-game assets 500 may modify the mechanics, logic, and mathematics of the games, i.e. modify the gameplay materials, flow, rules, and objectives. Such modifications may either have minor effect and not to cause mutation of the certain version of the game into qualitatively distinct version of the game or impact the game flow in such a considerable degree, that will cause the change of the moded game into qualitatively different game.

Examples of the game modes 580 which have minor effects not causing transition to the new versions of the game may be represented by such modes as introduction of the time limits for decision-making (chess clock), introduction of chance factor (dice or spinner giving players 250 certain advantages such as additional move in sequence, etc.) imitating force-major and unpredictable human-factor circumstances and effects in the real world, and other similar modifications within the established mechanics of the version of the game and flow of the gameplay.

In some examples, the modes 580 can include, but not limited to, different perspectives of in-game camera views, for example to include views of the gameplay as if seen by different semi-player characters 540, with options to change the perspectives and vantage points during gameplay or, if the gameplay was recorded, during playing the recorded media of the gameplay.

Examples of the game modes 580 which have major effects causing transition of the modified game to the new versions of the game may be represented by such modes as introduction of the new dimensions and number of the playing board (-s) (e.g., change of the number of files and ranks of the two dimensional chess board, replacement of the two-dimensional chess board with a three-dimensional array of cells between which the pieces can move, etc.), introduction of new number and functional materiel (lists) of the playing pieces (e.g., introduction of new types of pieces with specific rules of movement, etc.), introduction of the new rules, rights of movement and other ludemes for the playing figures, introduction of new number of the players (e.g., one player vs. two or more players, alliance of two or more players playing against alliance of two or more other players, battle royale when more than two players play versus each other with right of alliances, etc.).

Such major modifications can include, but not limited to, introductions of the game versions similar to Kubikschach (German for "Cube chess") game invented by Lionel Kieseritzky (1806–1853) in 1851 (an 8×8×8 board, labelling the third dimension with Greek letters alpha through theta, etc.), Raumschach (German for "Space chess") invented by Ferdinand Maack (1861–1930) in 1907 (an 8×8×8 board, or 5×5×5 board with introduction of additional pieces, etc.), Star Trek Tri-Dimensional Chess seen in many Star Trek franchise TV episodes and movies (with rules developed by Andrew Bartmess and Franz Joseph in 1970-ies, etc.), etc. Such game versions, for example, may include a board represented as a cube sliced into five, eight or other number of equal spaces across each of its three major coordinal planes, such spaces having embodiments of smaller cubical cells alternating in color in all three dimensions.

Complex modes 580 or the combination of modes 580 thus may develop the existing versions of the game into new versions. The border between the modes 580 modifying existing game and creating new versions of the game is blurred.

The modes 580 may be further implemented and negotiated as stand-alone modes for particular functions or sets of functions or may be created and negotiated in bundles (being coupled to) other in-game assets 500, such as player-tokens 520, non-player characters 530, semi-player characters 540, environments (settings) 550, visual effects 560, audio assets 570, and graphical user interfaces (GUI) 590.

The class of game modes 580 of the digital in-game assets 500 can be further subdivided into modes used within the gameplay, and modes used in the online gaming platform 110 but outside of the gameplay flow (e.g., for surfing the online gaming platform 110, negotiating digital in-game assets 500 in the online marketplace 140, attend trainings, interact socially with the gaming community, etc.).

Graphical user interfaces (GUI) are forms of users' interfaces that allow users to interact with electronic devices, or, to put it in other context, with virtual worlds, through graphical icons and visual indicators such as secondary notation. In gaming parlance, GUIs are often called head-up displays (HUDs), and sometimes referred to also as "chrome".

FiG.5-9 illustrates general taxonomical classification of GUI 590 digital in-game assets.

Generally, all GUI 590 of the digital in-game assets 500 class can be subdivided into GUI implemented in 2d and GUI implemented in 3D format.

The GUI 500 of the digital in-game assets 500 can be further subdivided into GUI used within the gameplay, and GUI used in the online gaming platform 110 but outside of the gameplay flow (e.g., for surfing the online gaming platform 110, negotiating digital in-game assets 500 in the online marketplace 140, attend trainings, interact socially with the gaming community, etc.).

Graphical user interfaces 590 can be considered as part of the game modes 580, but due to their importance and specifically identifiable commercial value are defined as a separate class of digital in-game assets 500.

GUI 590 is a visual language of the players 250 (especially in game flow to reconnaissance the situation and articulate and implement their decisions) and other users of the online gaming platform 110.

User experience design is important for the overall gameplay experience, and thus GUI 590 represent important digital in-game assets with high potential for customization and artistic creativity along with functional efficiency. Designing the visual composition and temporal behavior of a GUI 590 is an important part of software application programming in any human–computer interaction, and especially in dynamic environment of gaming applications, including those having as moderate rate of dynamism as the main versions of the board games normally do.

User centered design of GUI 590 enhance the efficiency and ease of use for the underlying logical design of a stored programs, their usability. Methods of user-centered design are normally used to ensure that the visual language introduced in the design is well-tailored to the tasks.

Typically, players 250 and other users will interact with the game and other information in the online gaming platform 110 by manipulating visual widgets that allow for interactions appropriate to the kind of data they hold.

The widgets of a well-designed interface 590 shall be selected to support the actions necessary to achieve the goals of users. Normally, a model–view–controller approach may allow flexible

structures in which the GUI 590 is independent of and indirectly linked to application functions, so the GUI can be customized easily. This may allow players 250 to select or design a different skin or theme at will and eases the creators 270 work to develop new GUIs as users' needs evolve.

Good designs of GUI 590 will relate to users more, and to system architecture less.

The most common elements in GUI 590 may be windows, icons, toolbars, text fields, canvases, menus, radio buttons, handles, dashboards, pointers of WIMP paradigm - especially in personal computers, pointers of post-WIMP paradigm - especially in mobile devices, etc.

The variety of the GUI 590 designs may be limitless, varying by degrees of functionality, artistic expression, realism, 2D vs. 3D solutions, etc.

The example of taxonomical classification of the versions of the games to be supported by the platform 110 is shown in FIG. 6. The online gaming platform 110 may provide plentitude of the electronic versions and modifications of the traditional and newly designed board games based on the fundamental model of collaboration professed by the platform 110: the online gaming platform 110 provides to players 250 an access to the online runtime gaming engine 210 which delivers the basic variants of the games facilitating smooth gameplay for each of them. Such variant normally does not offer extensive selection of digital in-game assets 500, but additional assets 500 of choice may be acquired or rented in the online marketplace 140 and easily imported into compatible game version on demand in real-time. The online gaming platform 140 also provides access to all the tools necessary for the creators 270 to develop such assets and negotiate them in the online marketplace 140.

Based on the above, there are three ways how new versions of the game may be created:

(1) created by the online gaming platform 110 (i.e., platform administrator 240);
(2) created by the creators 270 and approved for negotiation by the online gaming platform 110;
(3) resulting from the deployment, in particular version of the game, of the modes 580 as part of the digital in-game assets 500 which cause major changes to the gameplay to the extend that allows to call such moded version of the game to be a qualitatively new version distinctly different from the original (unmoded) version.

FIG. 7 is a flowchart illustrating example process of developing digital in-game assets 500 by creators 270 created for and deployed on the online-gaming platform 110, according to an aspect of the present disclosure.

FIG.8 is a flowchart illustrating example of digital assets 500 negotiation in the online marketplace of the online-gaming platform 110, according to an aspect of the present disclosure.

FIG.8 is a flowchart illustrating the process of starting the gameplay by the player 250 on the online-gaming platform 110, according to an aspect of the present disclosure.

What is claimed is:

1. a system of online gaming platform (platform) for electronic (digital) versions of traditional and perspective board games with changeable in-game digital assets such system comprising:

a. subsystem of game engine a complex of software programs allowing to create, modify and run electronic versions of the board games;
b. subsystem of software instruments accessible for the participants of the platform to create various digital in-game assets that may be imported into electronic versions of the board games to enrich the gameplay experience;
c. subsystem of marketplace for digital in-game assets where participants of the platform may negotiate digital in-game assets (buy, sell, rent, exchange, license, modify, etc.);
d. plentitude of other subsystems to facilitate and manage access, security, users' accounts, transactions, digital in-game assets, social interaction and business collaboration between users;
e. plentitude of servers, processors and communication devices to host the subsystems of the system.

2. The system of Claim 1 wherein various designs and sets of the digital in-game assets may be imported to and used by the users in the games supported by the platform to diversify and enrich gaming experience.

3. The system of Claim 1 wherein various designs and sets of the digital in-game assets include electronic player-tokens which are visual representations of the human players or non-human (AI) players such player-tokens created, negotiated and used by players in the gameplay.

4. The system of Claim 1 wherein various designs and sets of the digital in-game assets include non-player and semi-player characters or pieces.

5. The system of Claim 1 wherein various designs and sets of the digital in-game assets include environments (settings) for creation of in-game virtual environments.

6. The system of Claim 1 wherein various designs and sets of the digital in-game assets include special visual effects and audio.

7. The system of Claim 1 wherein various designs and sets of the digital in-game assets include graphical user interfaces.

8. The system of Claim 1 wherein various designs and sets of the digital in-game assets include modes of the games.

*****

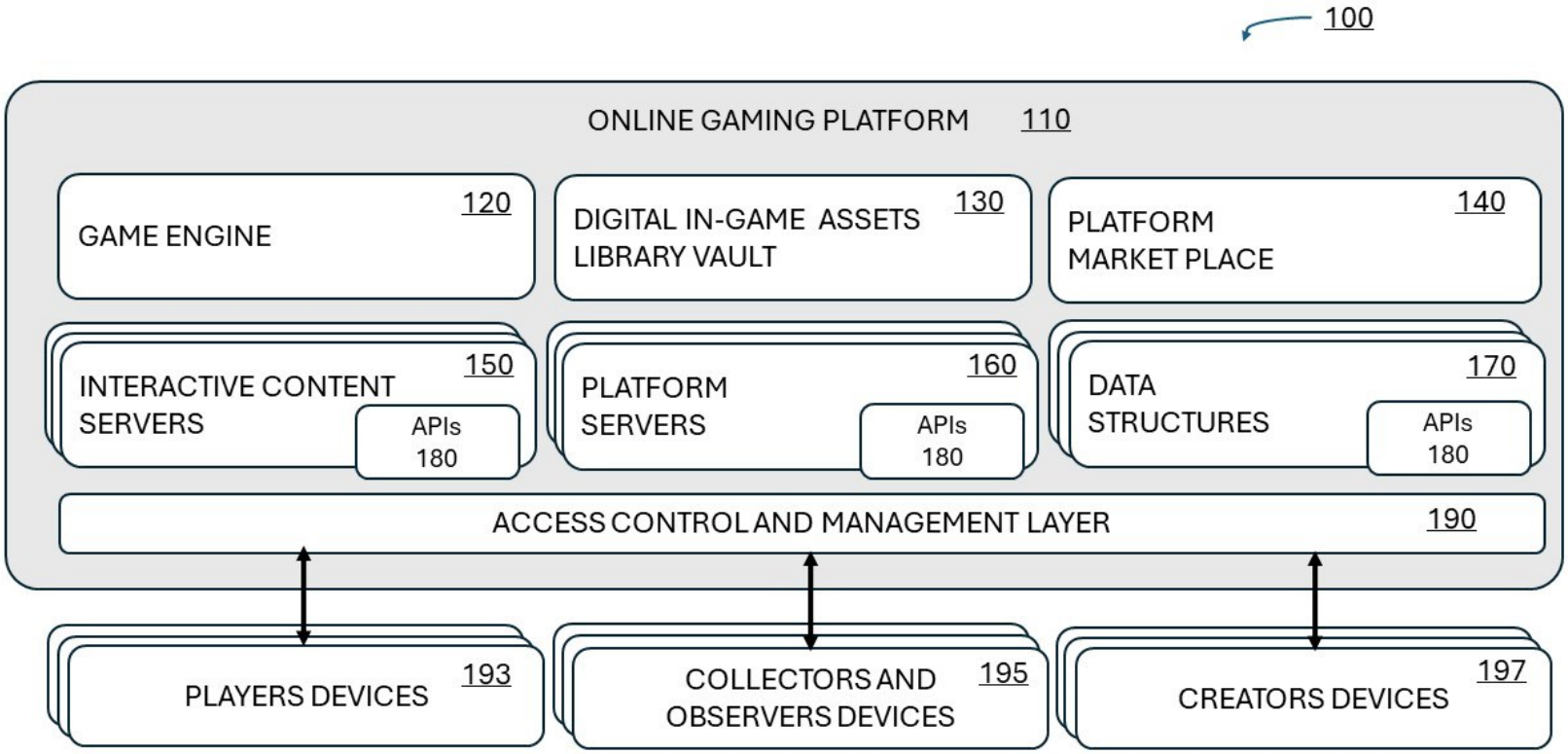
100
ONLINE GAMING PLATFORM 110
GAME ENGINE 120
DIGITAL IN-GAME ASSETS LIBRARY VAULT 130
PLATFORM MARKET PLACE 140
INTERACTIVE CONTENT SERVERS 150
APIs 180
PLATFORM SERVERS 160
APIs 180
DATA STRUCTURES 170
APIs 180
ACCESS CONTROL AND MANAGEMENT LAYER 190
PLAYERS DEVICES 193
COLLECTORS AND OBSERVERS DEVICES 195
CREATORS DEVICES 197

FIG. 1

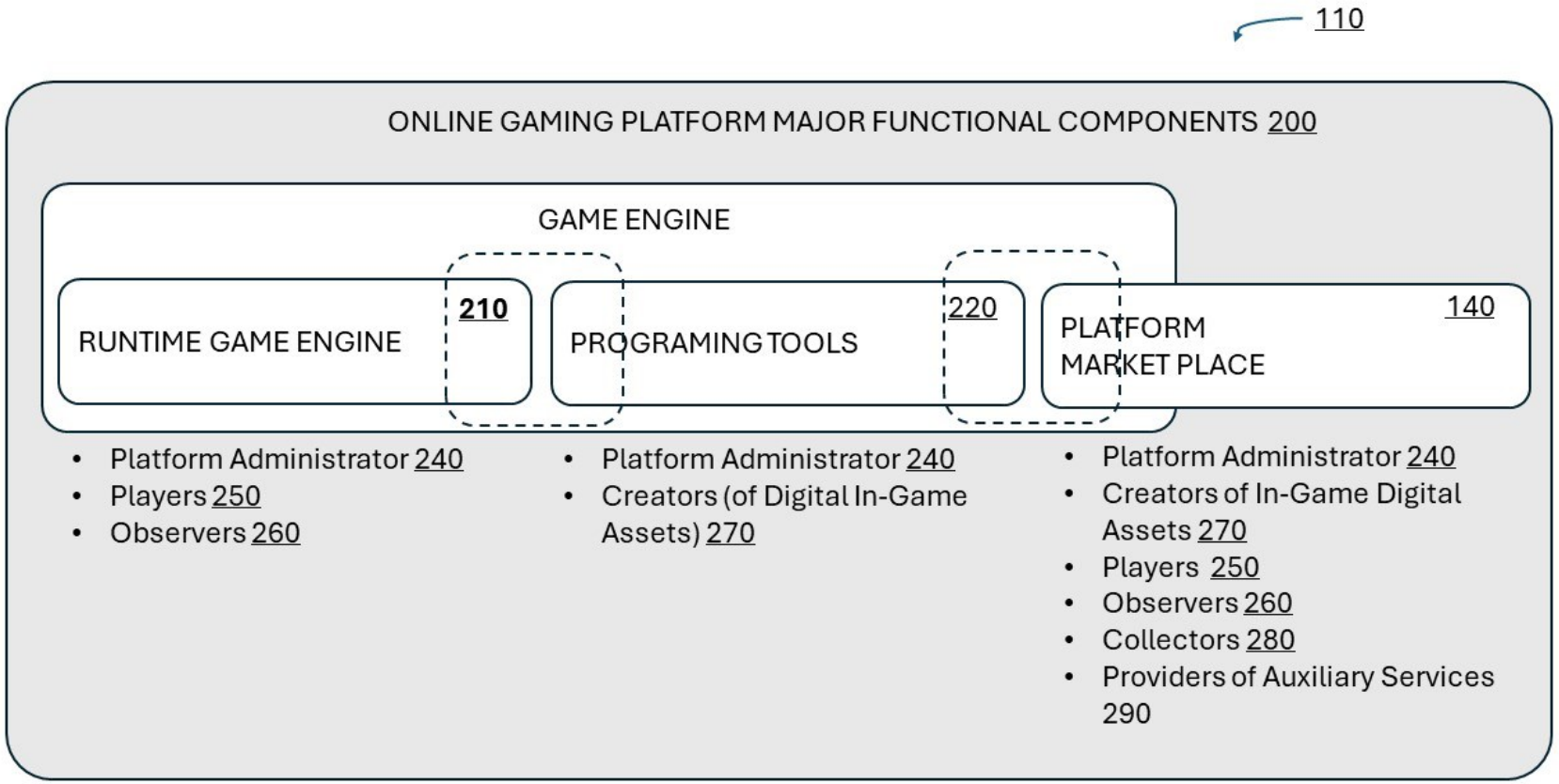
110
ONLINE GAMING PLATFORM MAJOR FUNCTIONAL COMPONENTS 200
GAME ENGINE
RUNTIME GAME ENGINE
210
PROGRAMING TOOLS
220
PLATFORM MARKET PLACE
140
• Platform Administrator 240
• Players 250
• Observers 260
• Platform Administrator 240
• Creators (of Digital In-Game Assets) 270
• Platform Administrator 240
• Creators of In-Game Digital Assets 270
• Players 250
• Observers 260
• Collectors 280
• Providers of Auxiliary Services 290

FIG. 2

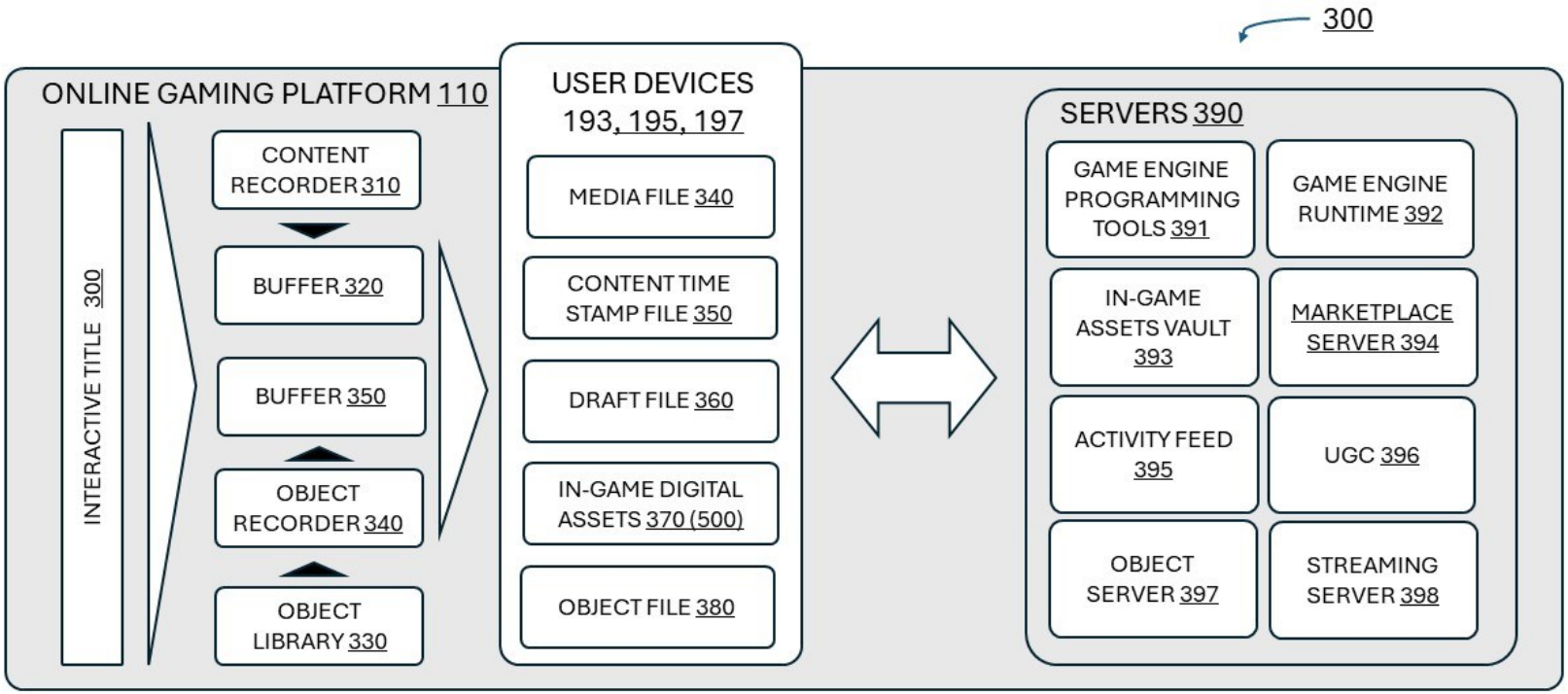
300
ONLINE GAMING PLATFORM 110
INTERACTIVE TITLE 300
CONTENT RECORDER 310
BUFFER 320
BUFFER 350
OBJECT RECORDER 340
OBJECT LIBRARY 330
USER DEVICES 193, 195, 197
MEDIA FILE 340
CONTENT TIME STAMP FILE 350
DRAFT FILE 360
IN-GAME DIGITAL ASSETS 370 (500)
OBJECT FILE 380
SERVERS 390
GAME ENGINE PROGRAMMING TOOLS 391
GAME ENGINE RUNTIME 392
IN-GAME ASSETS VAULT 393
MARKETPLACE SERVER 394
ACTIVITY FEED 395
UGC 396
OBJECT SERVER 397
STREAMING SERVER 398

FIG. 3

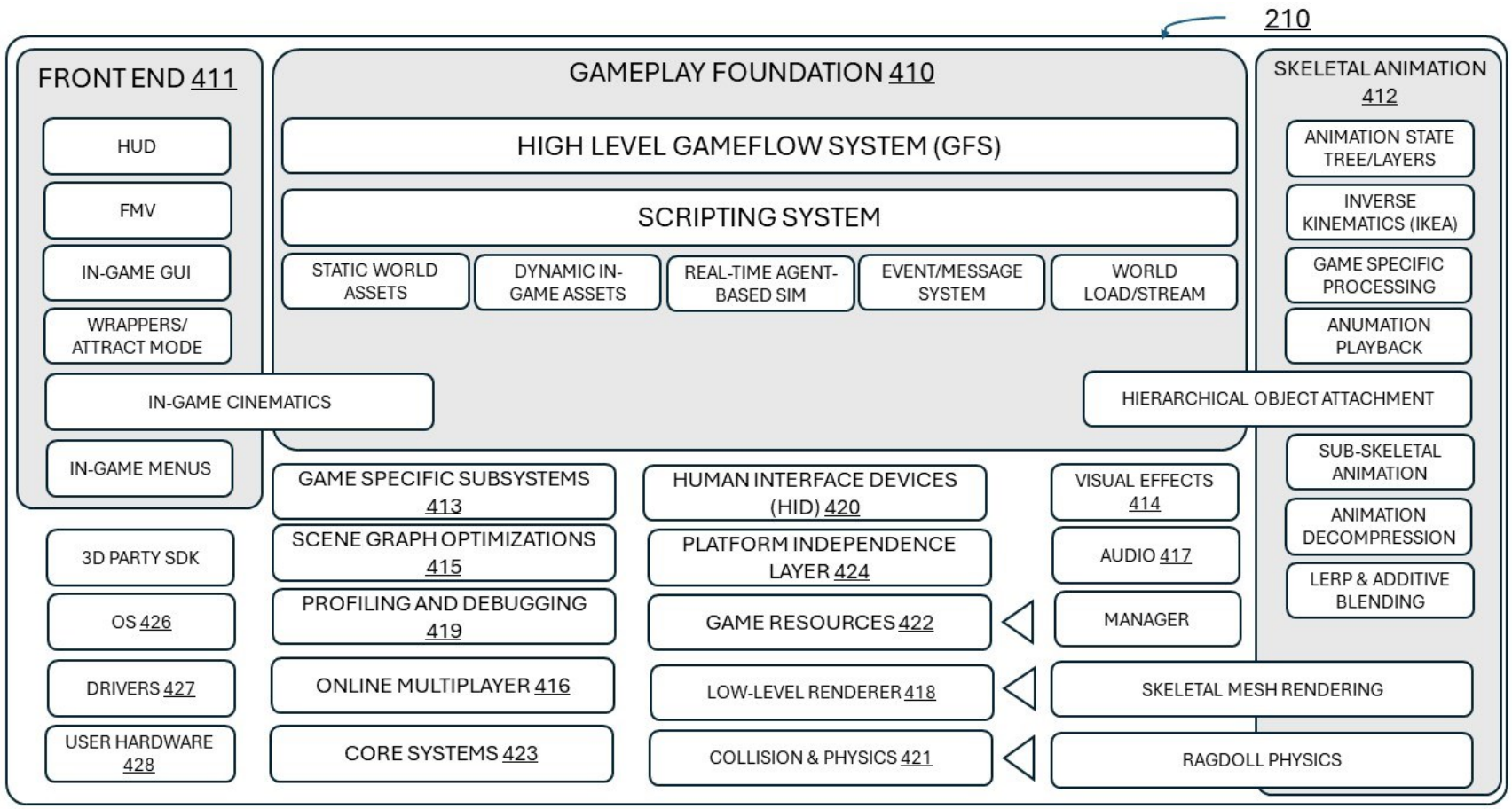
210
FRONT END 411
HUD
FMV
IN-GAME GUI
WRAPPERS/ ATTRACT MODE
IN-GAME CINEMATICS
IN-GAME MENUS
GAMEPLAY FOUNDATION 410
HIGH LEVEL GAMEFLOW SYSTEM (GFS)
SCRIPTING SYSTEM
STATIC WORLD ASSETS
DYNAMIC IN-GAME ASSETS
REAL-TIME AGENT-BASED SIM
EVENT/MESSAGE SYSTEM
WORLD LOAD/STREAM
SKELETAL ANIMATION 412
ANIMATION STATE TREE/LAYERS
INVERSE KINEMATICS (IKEA)
GAME SPECIFIC PROCESSING
ANUMATION PLAYBACK
HIERARCHICAL OBJECT ATTACHMENT
SUB-SKELETAL ANIMATION
ANIMATION DECOMPRESSION
LERP & ADDITIVE BLENDING
GAME SPECIFIC SUBSYSTEMS 413
SCENE GRAPH OPTIMIZATIONS 415
PROFILING AND DEBUGGING 419
ONLINE MULTIPLAYER 416
CORE SYSTEMS 423
HUMAN INTERFACE DEVICES (HID) 420
PLATFORM INDEPENDENCE LAYER 424
GAME RESOURCES 422
LOW-LEVEL RENDERER 418
COLLISION & PHYSICS 421
VISUAL EFFECTS 414
AUDIO 417
MANAGER
SKELETAL MESH RENDERING
RAGDOLL PHYSICS
3D PARTY SDK
OS 426
DRIVERS 427
USER HARDWARE 428

FIG. 4

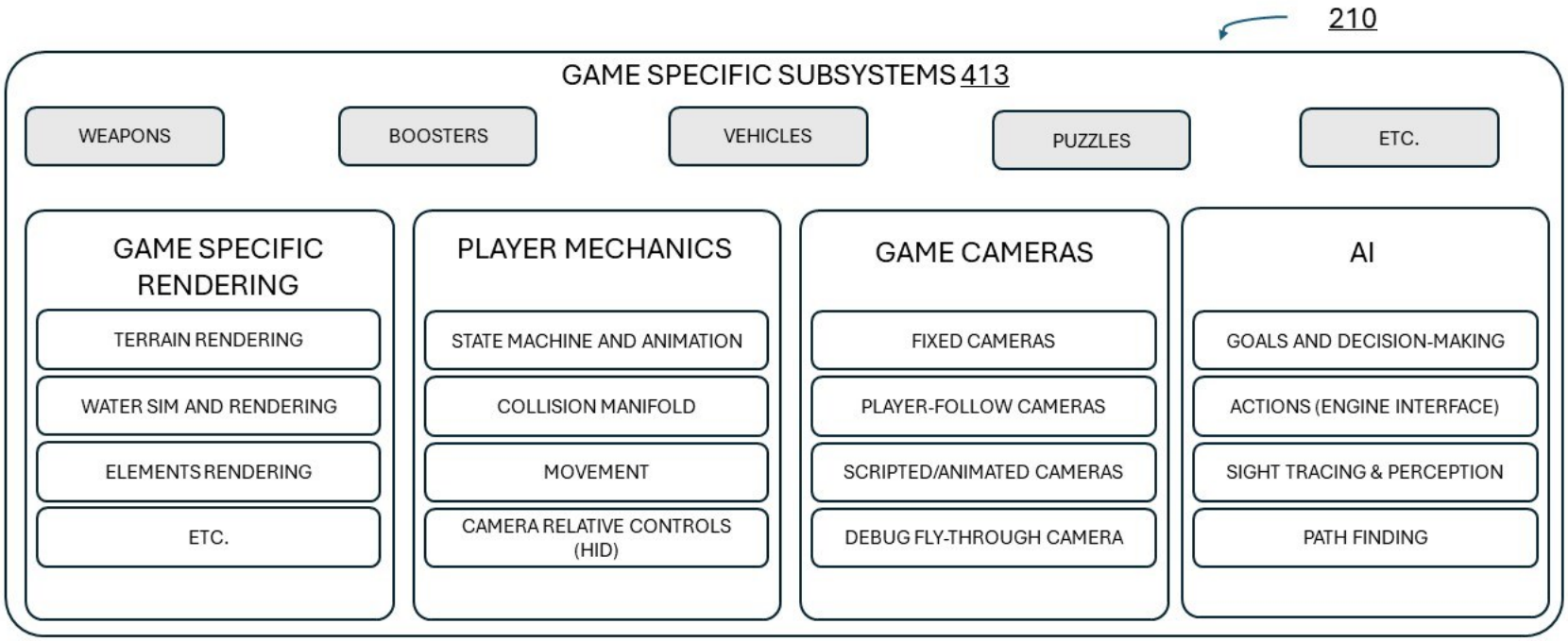
210
GAME SPECIFIC SUBSYSTEMS 413
WEAPONS
BOOSTERS
VEHICLES
PUZZLES
ETC.
GAME SPECIFIC RENDERING
TERRAIN RENDERING
WATER SIM AND RENDERING
ELEMENTS RENDERING
ETC.
PLAYER MECHANICS
STATE MACHINE AND ANIMATION
COLLISION MANIFOLD
MOVEMENT
CAMERA RELATIVE CONTROLS (HID)
GAME CAMERAS
FIXED CAMERAS
PLAYER-FOLLOW CAMERAS
SCRIPTED/ANIMATED CAMERAS
DEBUG FLY-THROUGH CAMERA
AI
GOALS AND DECISION-MAKING
ACTIONS (ENGINE INTERFACE)
SIGHT TRACING & PERCEPTION
PATH FINDING

FIG. 4-1

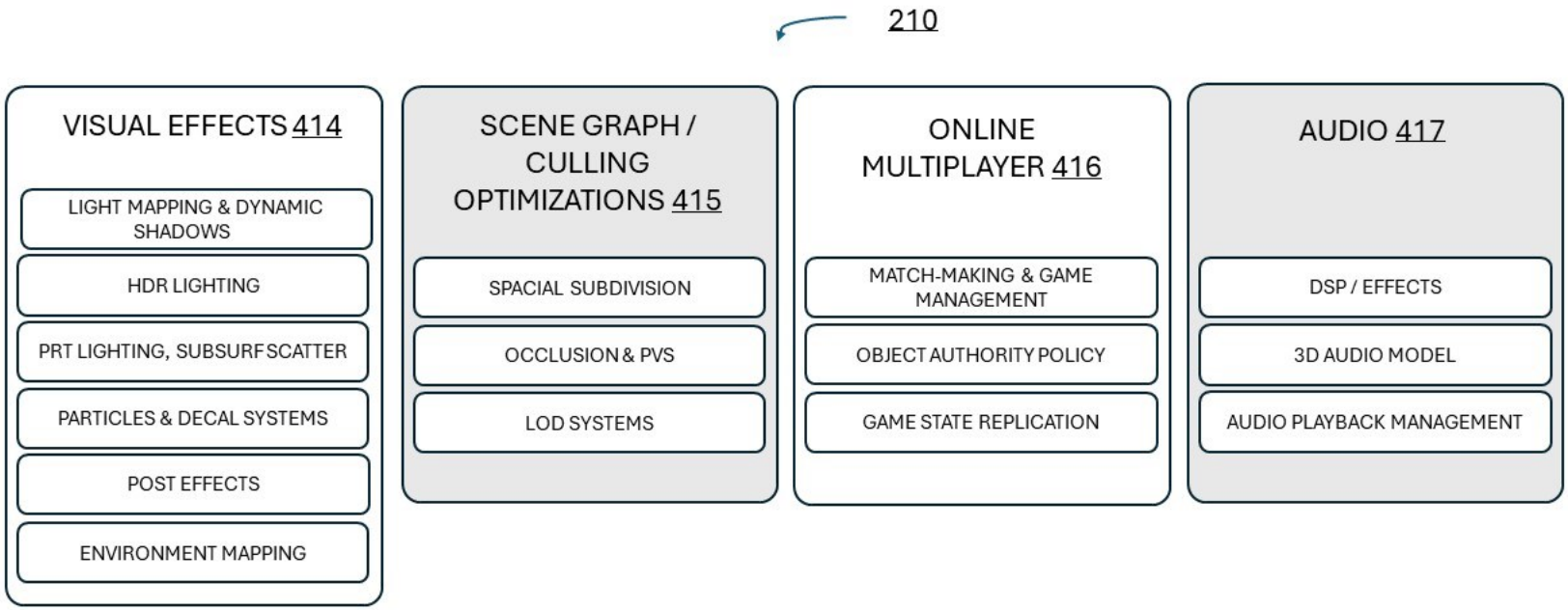
210
VISUAL EFFECTS 414
LIGHT MAPPING & DYNAMIC SHADOWS
HDR LIGHTING
PRT LIGHTING, SUBSURF SCATTER
PARTICLES & DECAL SYSTEMS
POST EFFECTS
ENVIRONMENT MAPPING
SCENE GRAPH / CULLING OPTIMIZATIONS 415
SPACIAL SUBDIVISION
OCCLUSION & PVS
LOD SYSTEMS
ONLINE MULTIPLAYER 416
MATCH-MAKING & GAME MANAGEMENT
OBJECT AUTHORITY POLICY
GAME STATE REPLICATION
AUDIO 417
DSP / EFFECTS
3D AUDIO MODEL
AUDIO PLAYBACK MANAGEMENT

FIG. 4-2

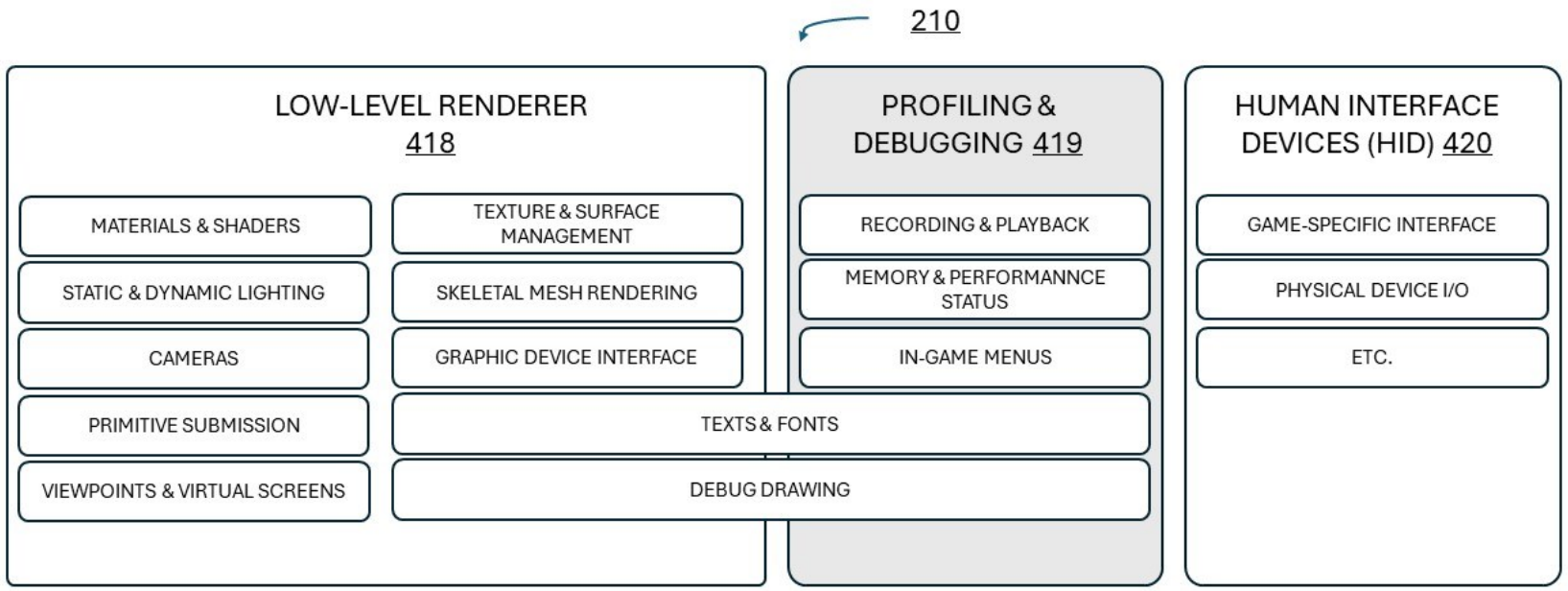
210
LOW-LEVEL RENDERER
418
MATERIALS & SHADERS
TEXTURE & SURFACE MANAGEMENT
STATIC & DYNAMIC LIGHTING
SKELETAL MESH RENDERING
CAMERAS
GRAPHIC DEVICE INTERFACE
PRIMITIVE SUBMISSION
TEXTS & FONTS
VIEWPOINTS & VIRTUAL SCREENS
DEBUG DRAWING
PROFILING & DEBUGGING 419
RECORDING & PLAYBACK
MEMORY & PERFORMANNCE STATUS
IN-GAME MENUS
HUMAN INTERFACE DEVICES (HID) 420
GAME-SPECIFIC INTERFACE
PHYSICAL DEVICE I/O
ETC.

FIG. 4-3

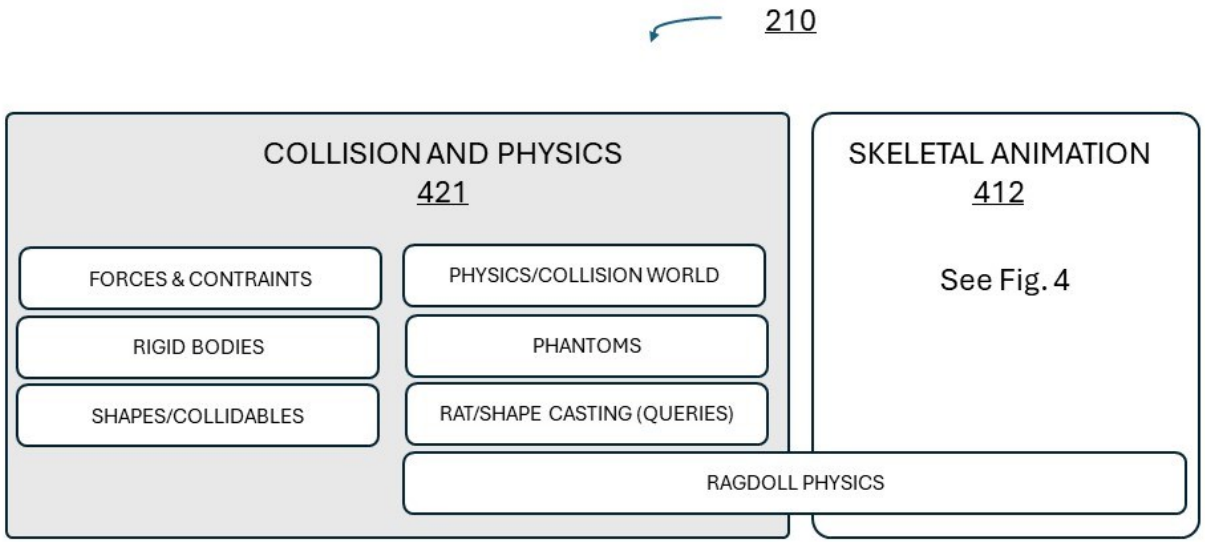
210
COLLISION AND PHYSICS
421
FORCES & CONTRAINTS
PHYSICS/COLLISION WORLD
RIGID BODIES
PHANTOMS
SHAPES/COLLIDABLES
RAT/SHAPE CASTING (QUERIES)
RAGDOLL PHYSICS
SKELETAL ANIMATION
412
See Fig. 4

FIG. 4-4

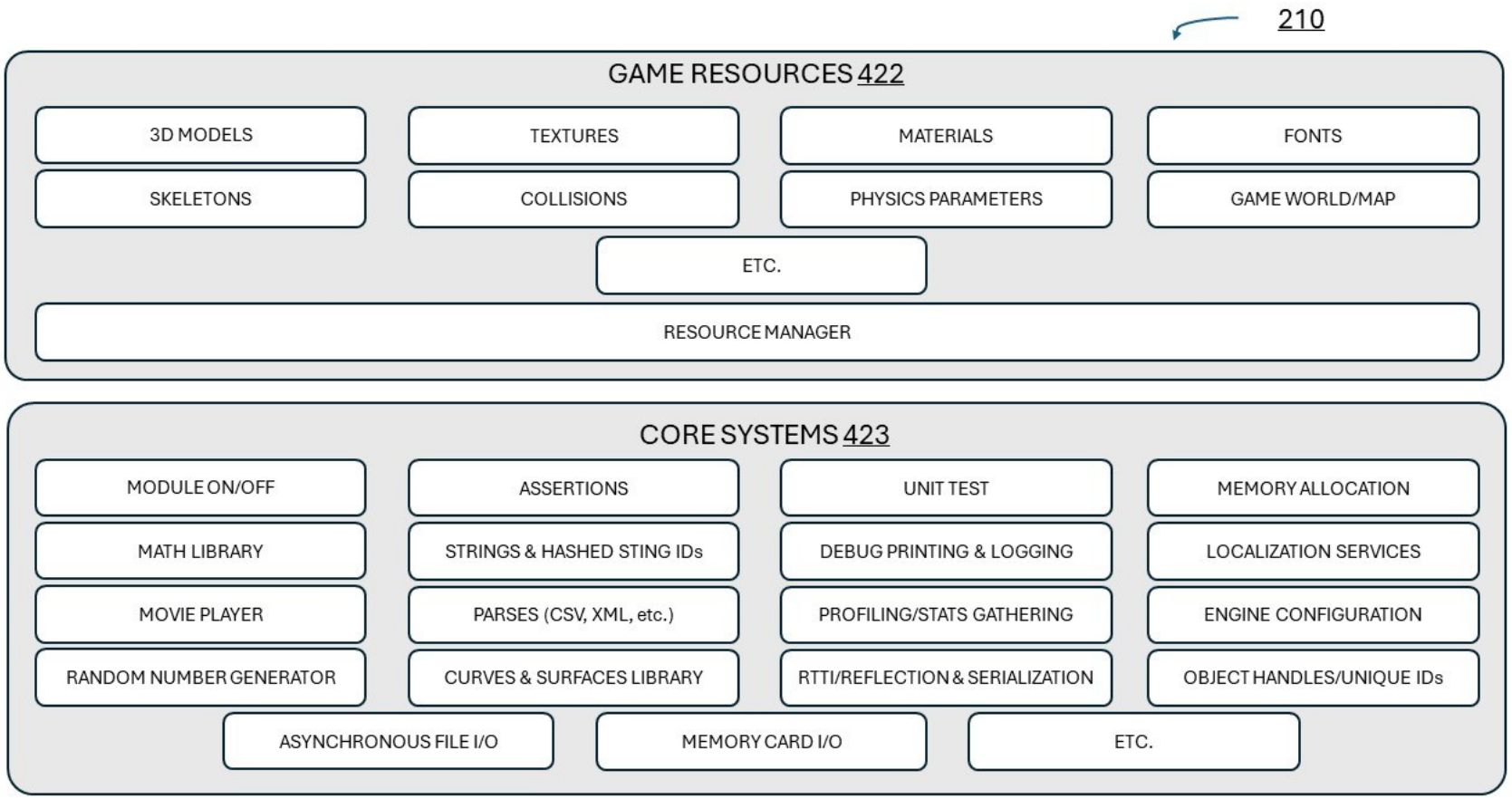
210
GAME RESOURCES 422
3D MODELS
TEXTURES
MATERIALS
FONTS
SKELETONS
COLLISIONS
PHYSICS PARAMETERS
GAME WORLD/MAP
ETC.
RESOURCE MANAGER
CORE SYSTEMS 423
MODULE ON/OFF
ASSERTIONS
UNIT TEST
MEMORY ALLOCATION
MATH LIBRARY
STRINGS & HASHED STING IDs
DEBUG PRINTING & LOGGING
LOCALIZATION SERVICES
MOVIE PLAYER
PARSES (CSV, XML, etc.)
PROFILING/STATS GATHERING
ENGINE CONFIGURATION
RANDOM NUMBER GENERATOR
CURVES & SURFACES LIBRARY
RTTI/REFLECTION & SERIALIZATION
OBJECT HANDLES/UNIQUE IDs
ASYNCHRONOUS FILE I/O
MEMORY CARD I/O
ETC.

FIG. 4-5

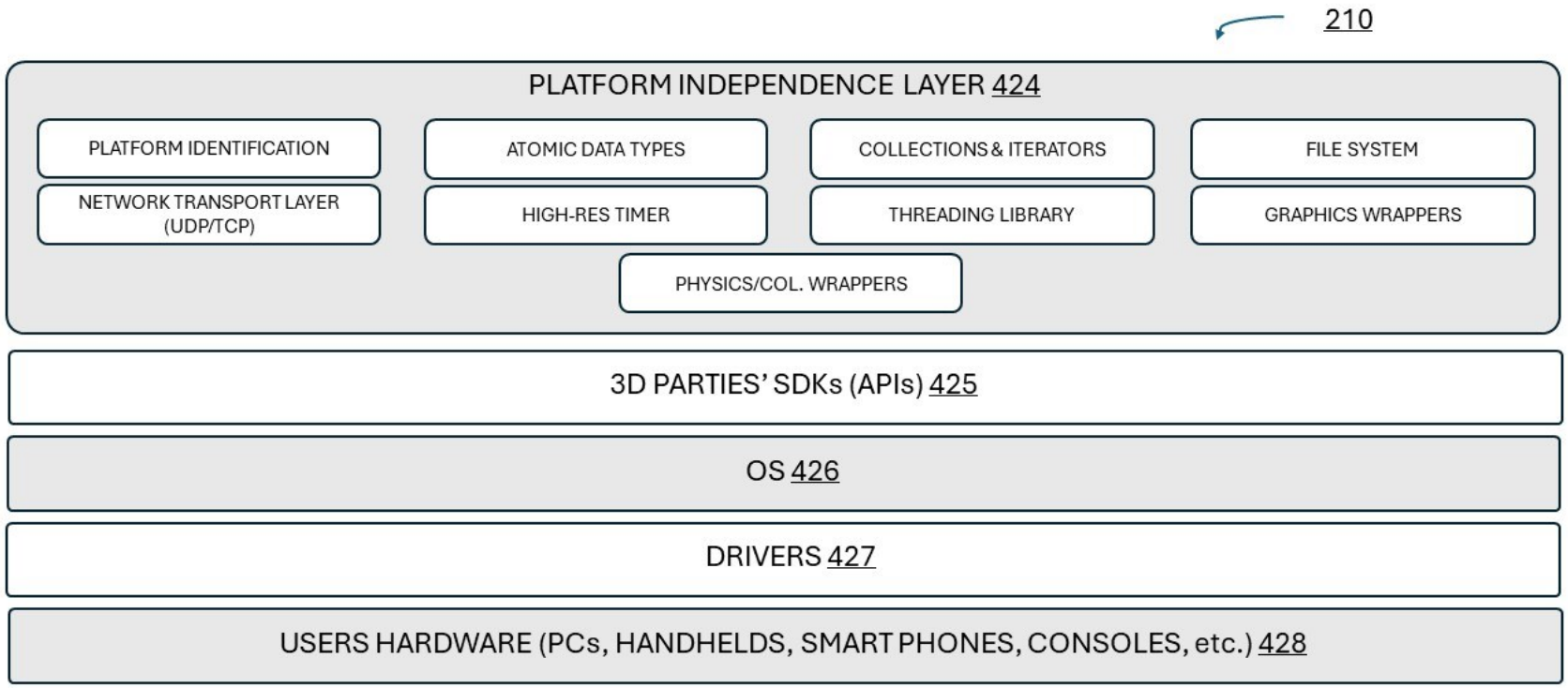
210
PLATFORM INDEPENDENCE LAYER 424
PLATFORM IDENTIFICATION
ATOMIC DATA TYPES
COLLECTIONS & ITERATORS
FILE SYSTEM
NETWORK TRANSPORT LAYER (UDP/TCP)
HIGH-RES TIMER
THREADING LIBRARY
GRAPHICS WRAPPERS
PHYSICS/COL. WRAPPERS
3D PARTIES' SDKs (APIs) 425
OS 426
DRIVERS 427
USERS HARDWARE (PCs, HANDHELDS, SMART PHONES, CONSOLES, etc.) 428

FIG. 4-6

EXAMPLE TAXONOMICAL CLASSIFICATION OF DIGITAL IN-GAME ASSETS 500

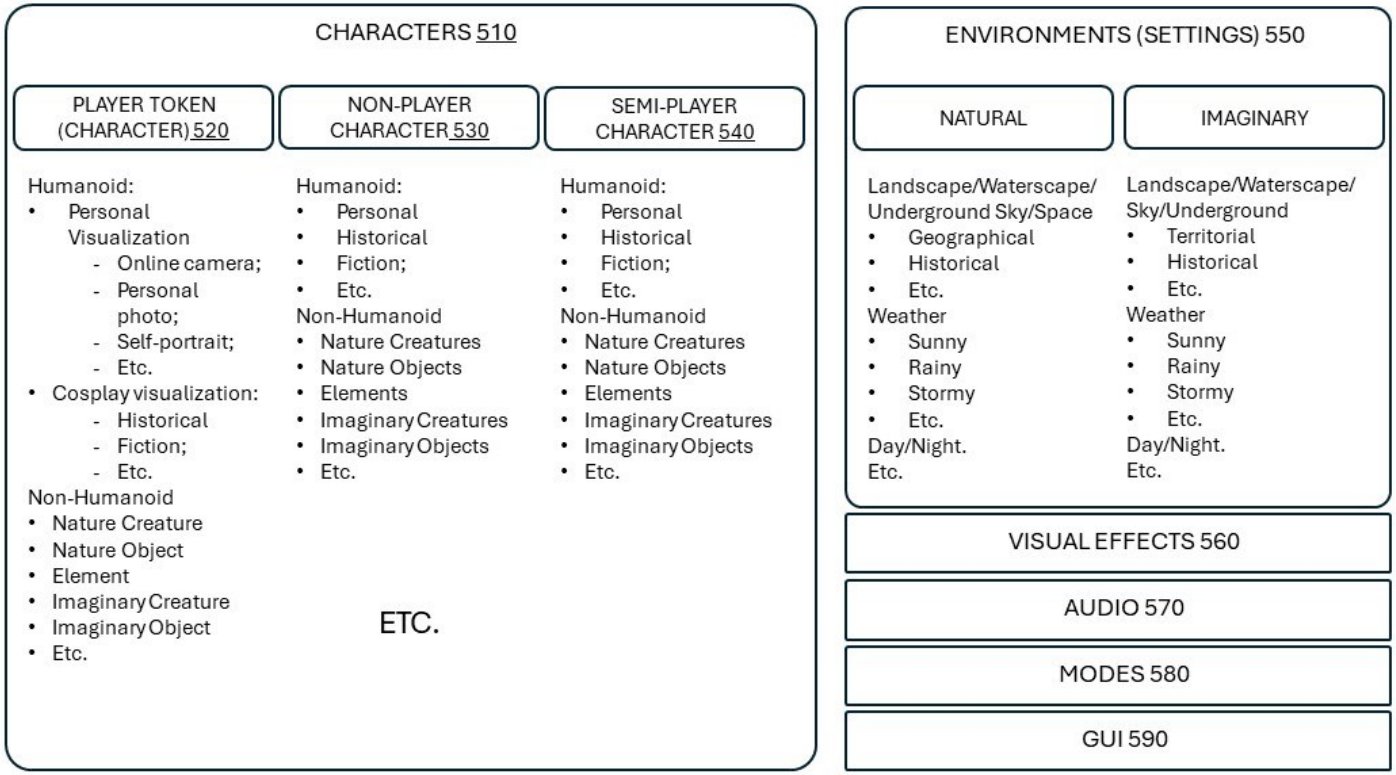

FIG. 5

EXAMPLE TAXONOMICAL CLASSIFICATION OF THE CHARACTERS CLASS OF DIGITAL IN-GAME ASSETS

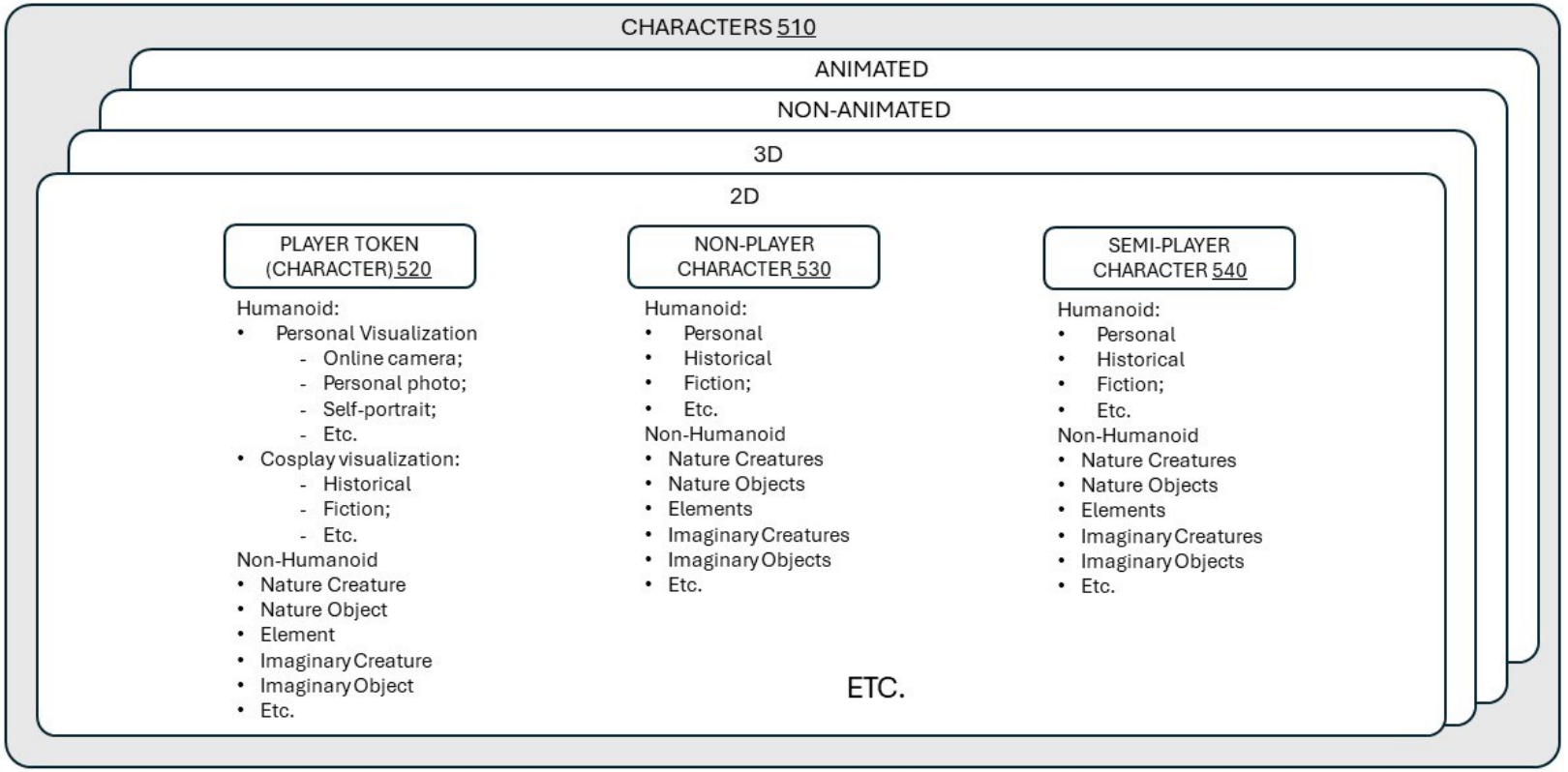

FIG. 5-1

EXAMPLE TAXONOMICAL CLASSIFICATION OF THE PLAYER TOKEN SUBCLASS 520 OF DIGITAL IN-GAME ASSETS 500

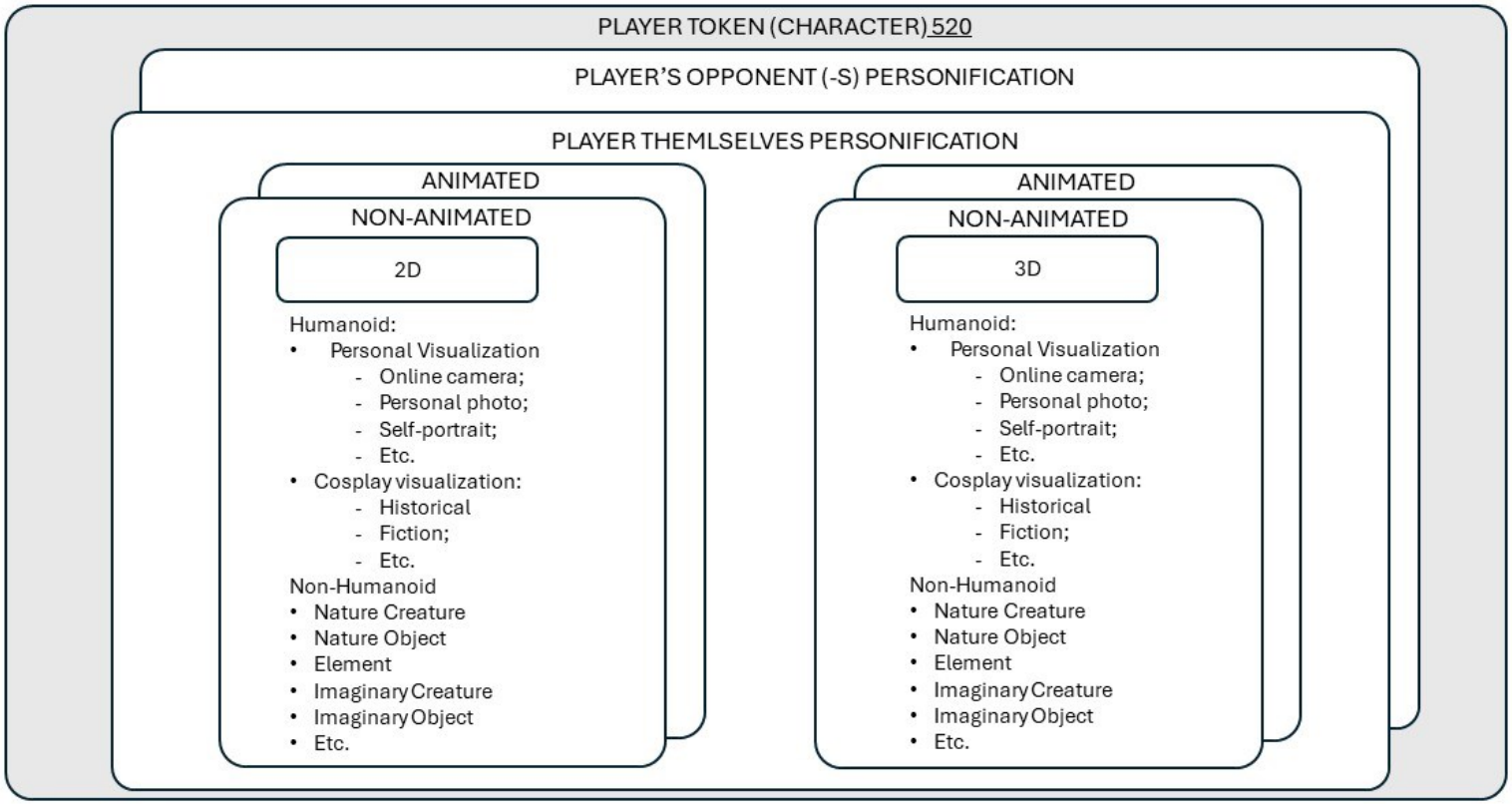

FIG. 5-2

CONCEPTUAL DIAGRAM OF EXAMPLE PLAYER TOKEN IN-GAME REPRESENTATIONS

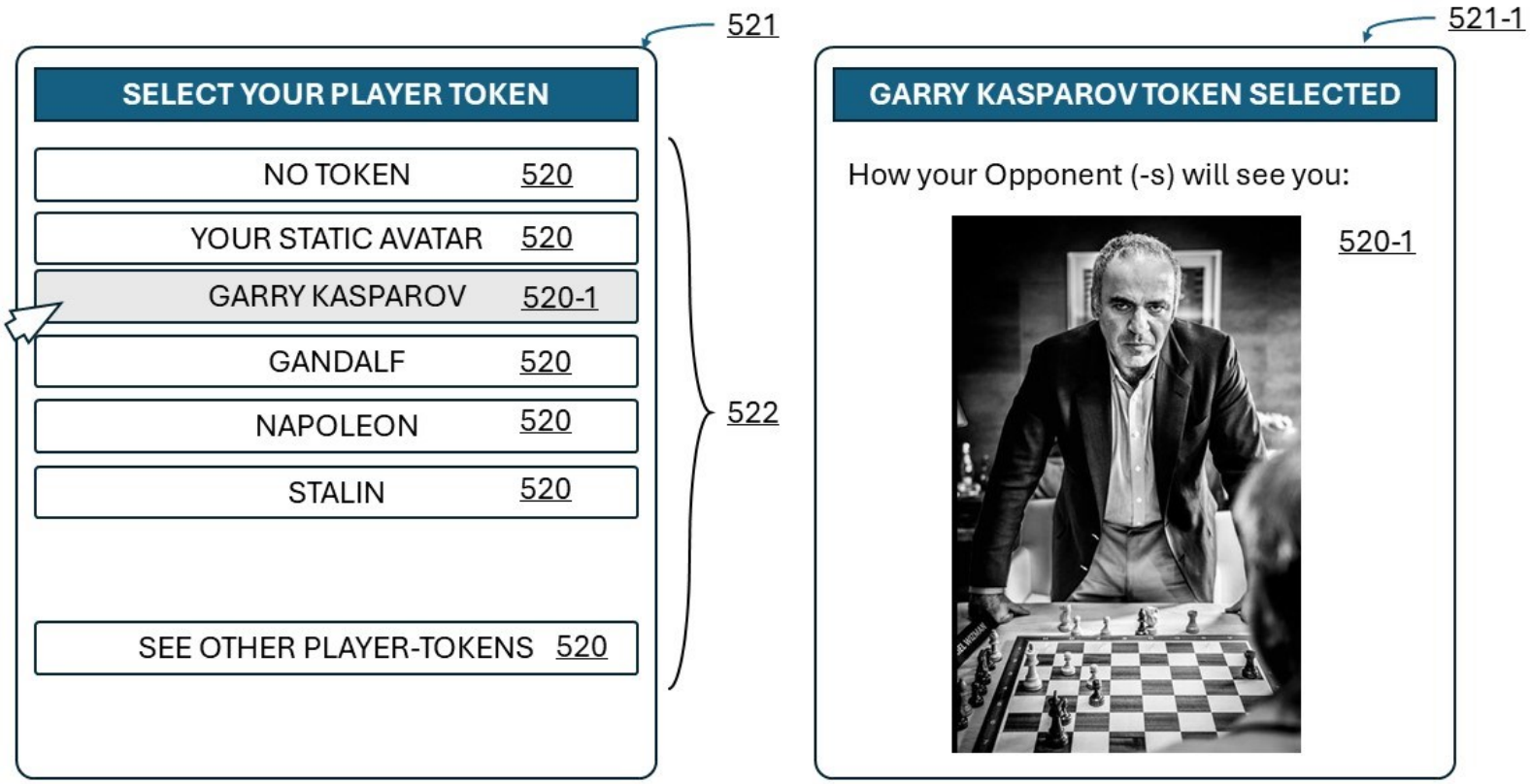

FIG. 5-2-A

CONCEPTUAL DIAGRAM OF EXAMPLE PLAYER TOKEN IN-GAME REPRESENTATIONS

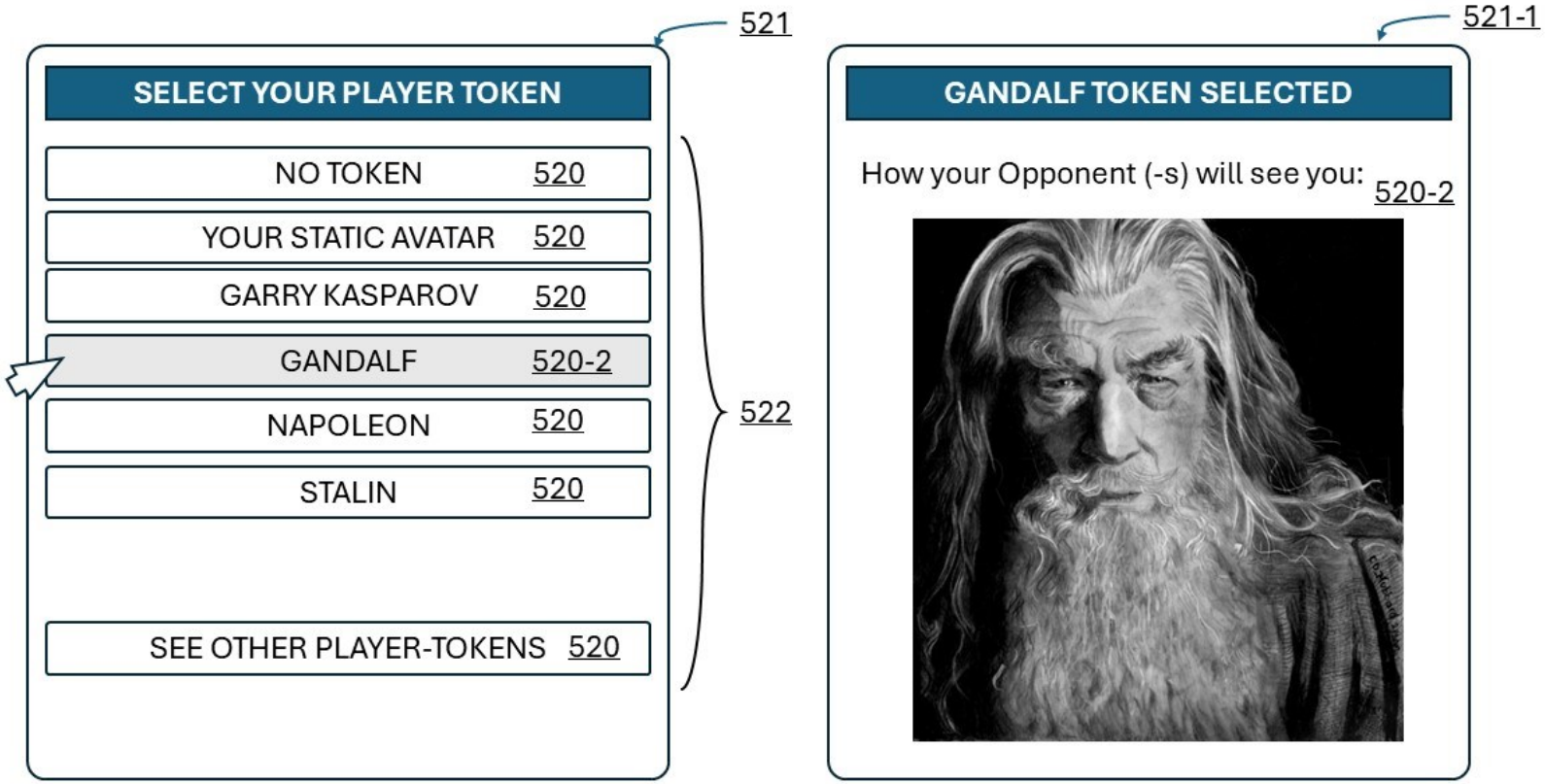

FIG. 5-2-B

CONCEPTUAL DIAGRAM OF EXAMPLE PLAYER TOKEN IN-GAME REPRESENTATIONS

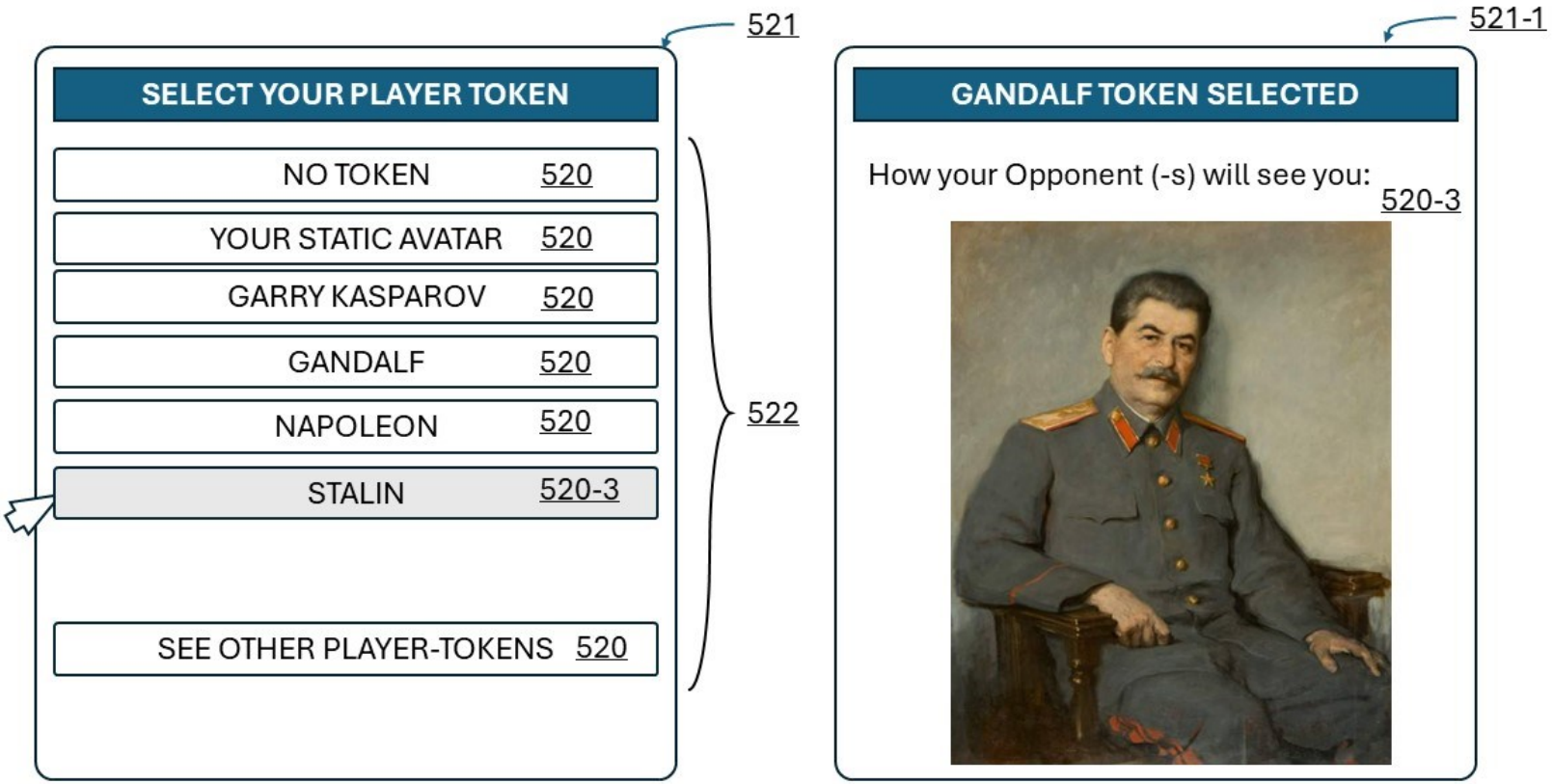

FIG. 5-2-C

CONCEPTUAL DIAGRAM OF EXAMPLE PLAYER TOKEN IN-GAME REPRESENTATIONS

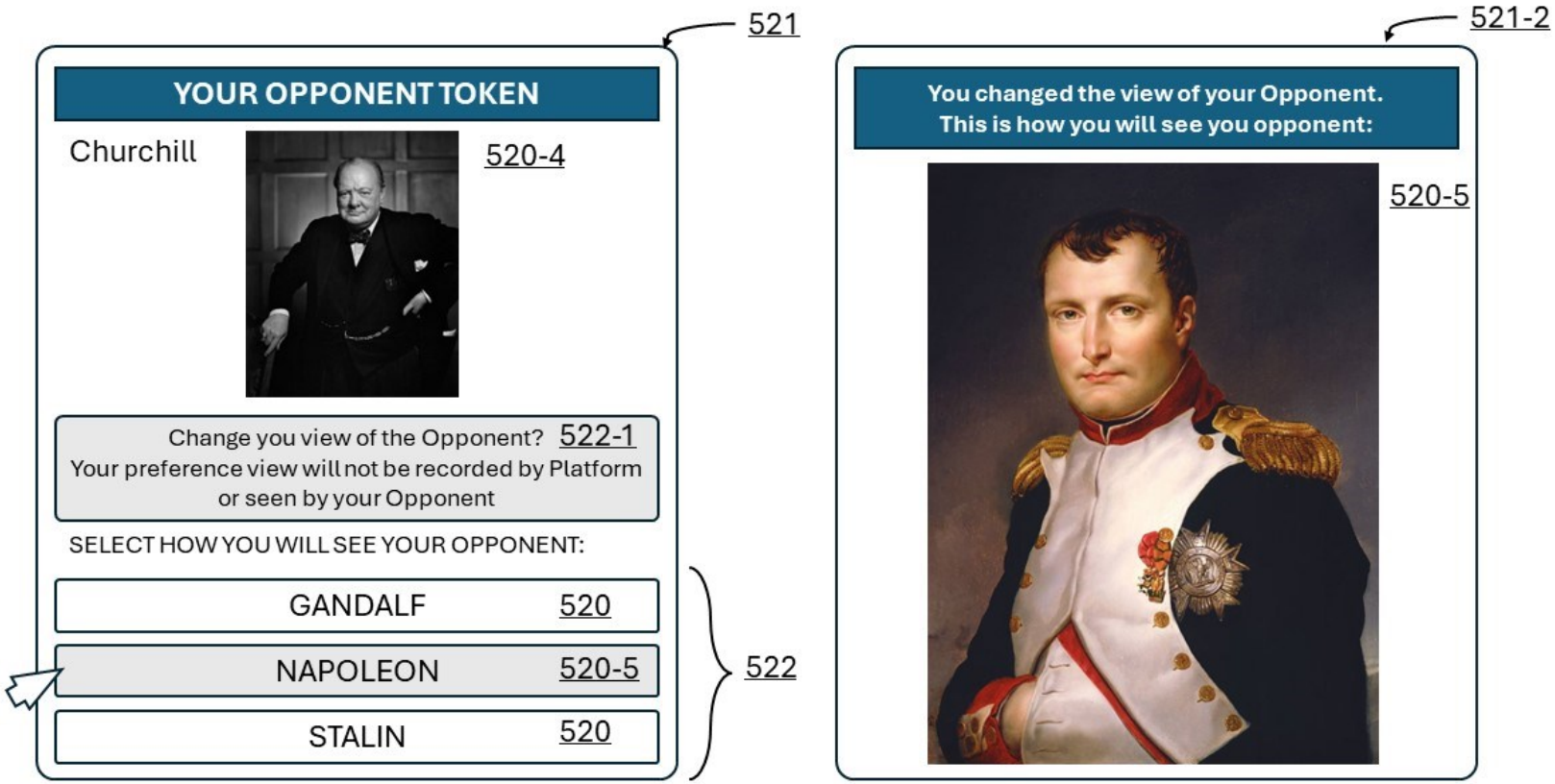

FIG. 5-2-E

CONCEPTUAL DIAGRAM OF EXAMPLE PLAYER TOKEN IN-GAME REPRESENTATIONS

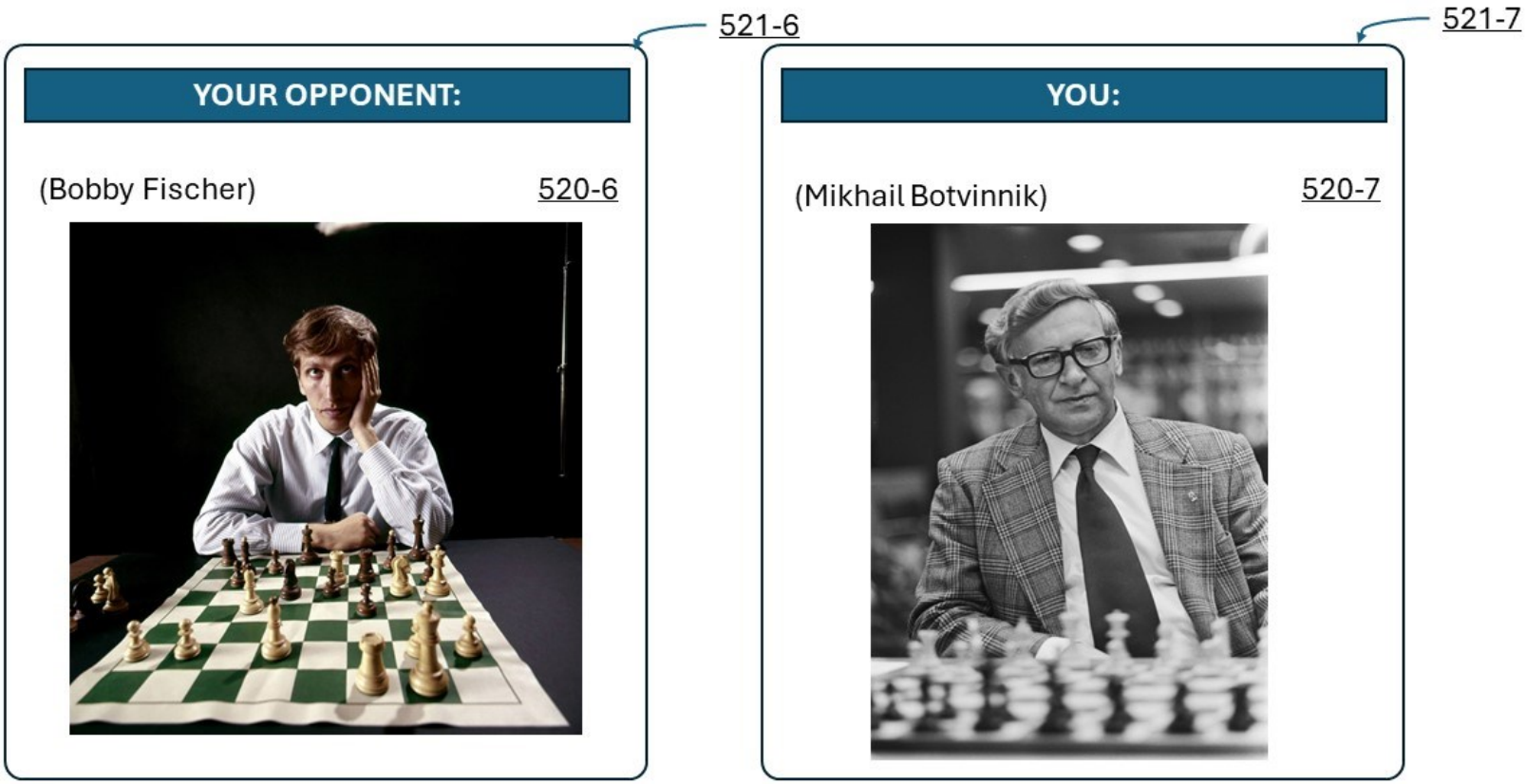

FIG. 5-2-F

CONCEPTUAL DIAGRAM OF EXAMPLE PLAYER TOKEN IN-GAME REPRESENTATIONS

FIG. 5-2-G

CONCEPTUAL DIAGRAM OF EXAMPLE PLAYER TOKEN IN-GAME REPRESENTATIONS

FIG. 5-2-H

CONCEPTUAL DIAGRAM OF EXAMPLE PLAYER TOKEN IN-GAME REPRESENTATIONS

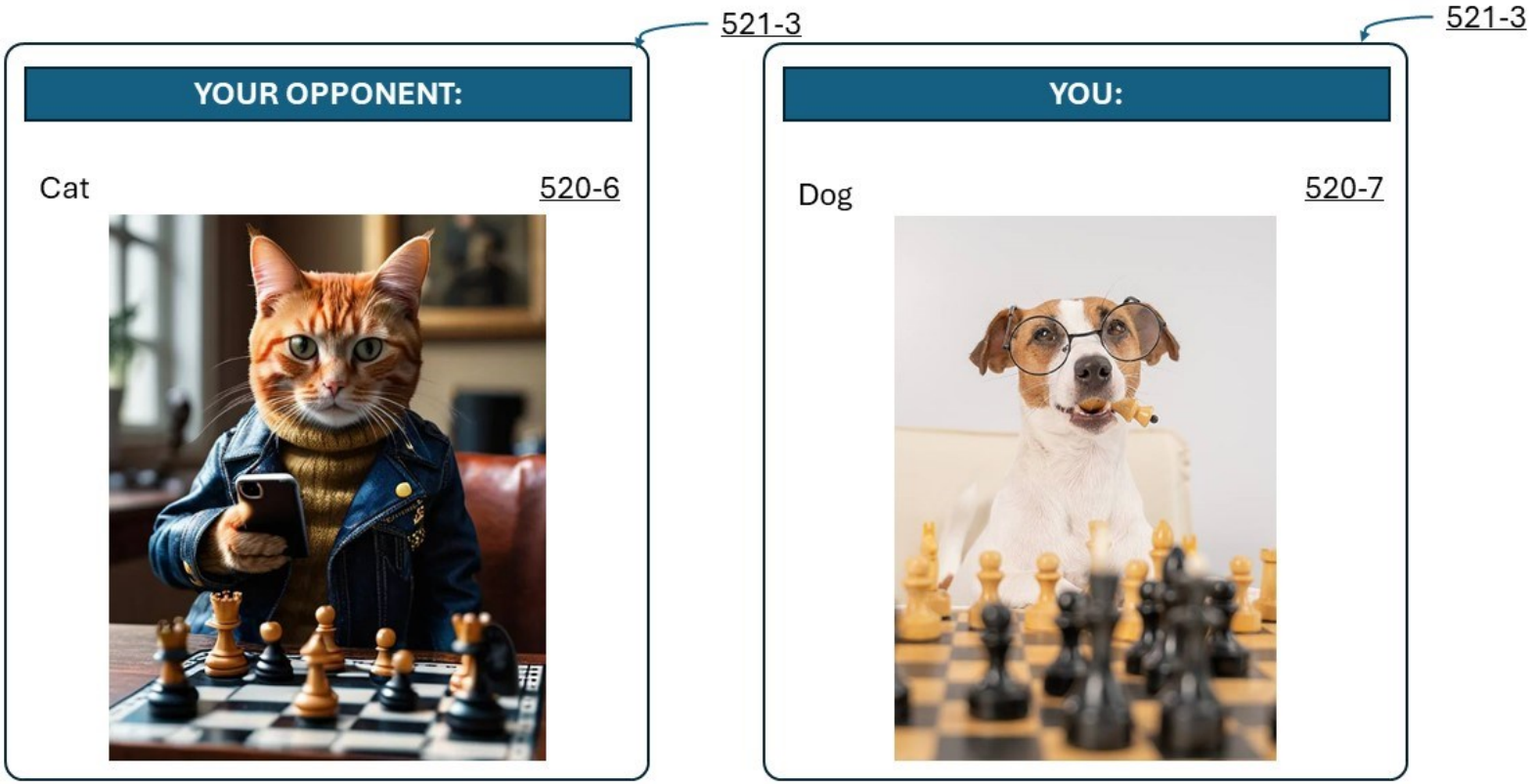

FIG. 5-2-I

CONCEPTUAL DIAGRAM OF EXAMPLE PLAYER TOKEN IN-GAME REPRESENTATIONS

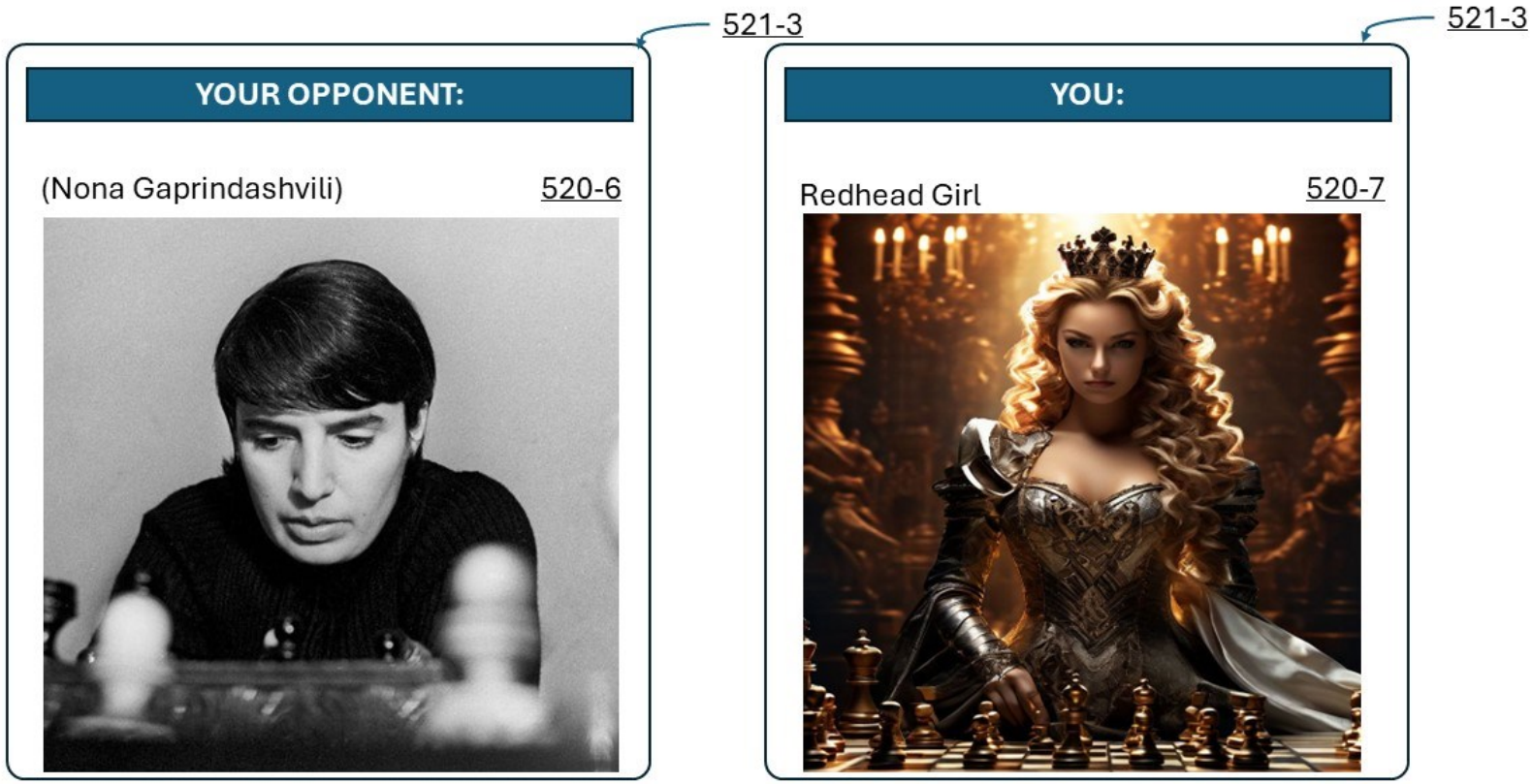

FIG. 5-2-J

EXAMPLE TAXONOMICAL CLASSIFICATION OF THE NON-PLAYER CLASS 530 OF DIGITAL IN-GAME ASSETS

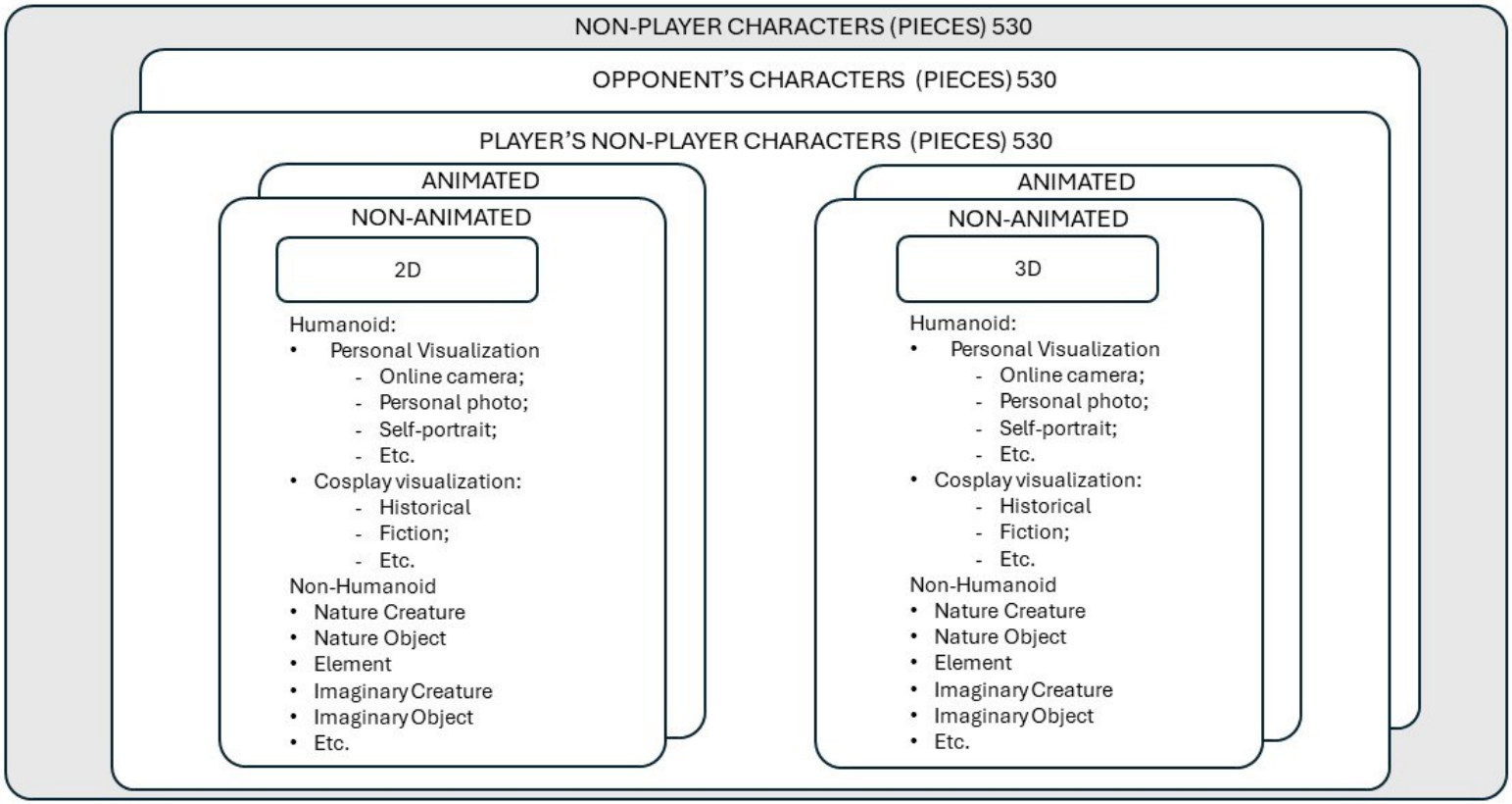

FIG. 5-3

CONCEPTUAL DIAGRAM OF EXAMPLE NON-CHARACTERS 530 AND SEMI-CHARACTERS 540 IN-GAME REPRESENTATIONS

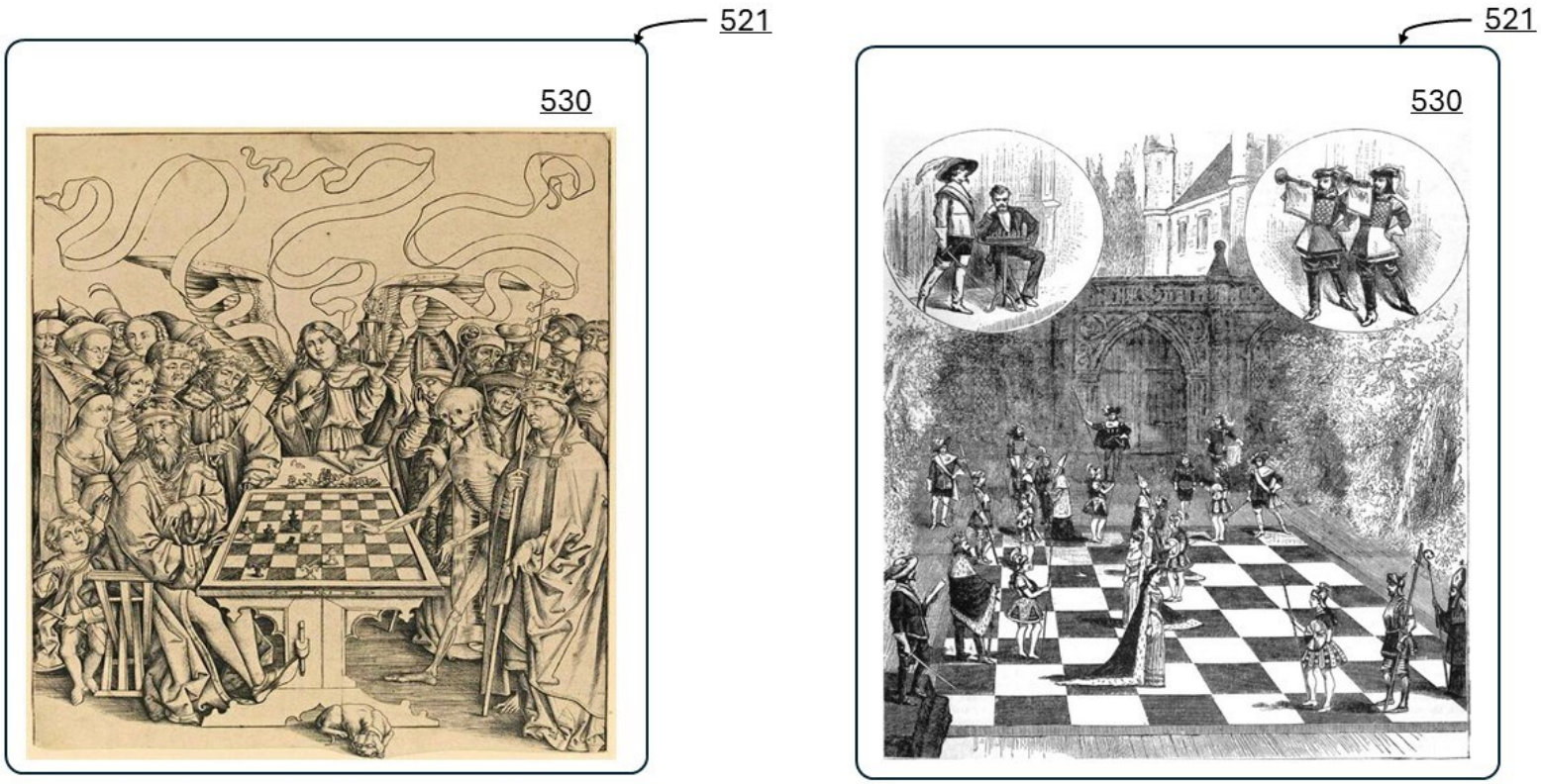

FIG. 5-3-A

CONCEPTUAL DIAGRAM OF EXAMPLE NON-CHARACTERS 530

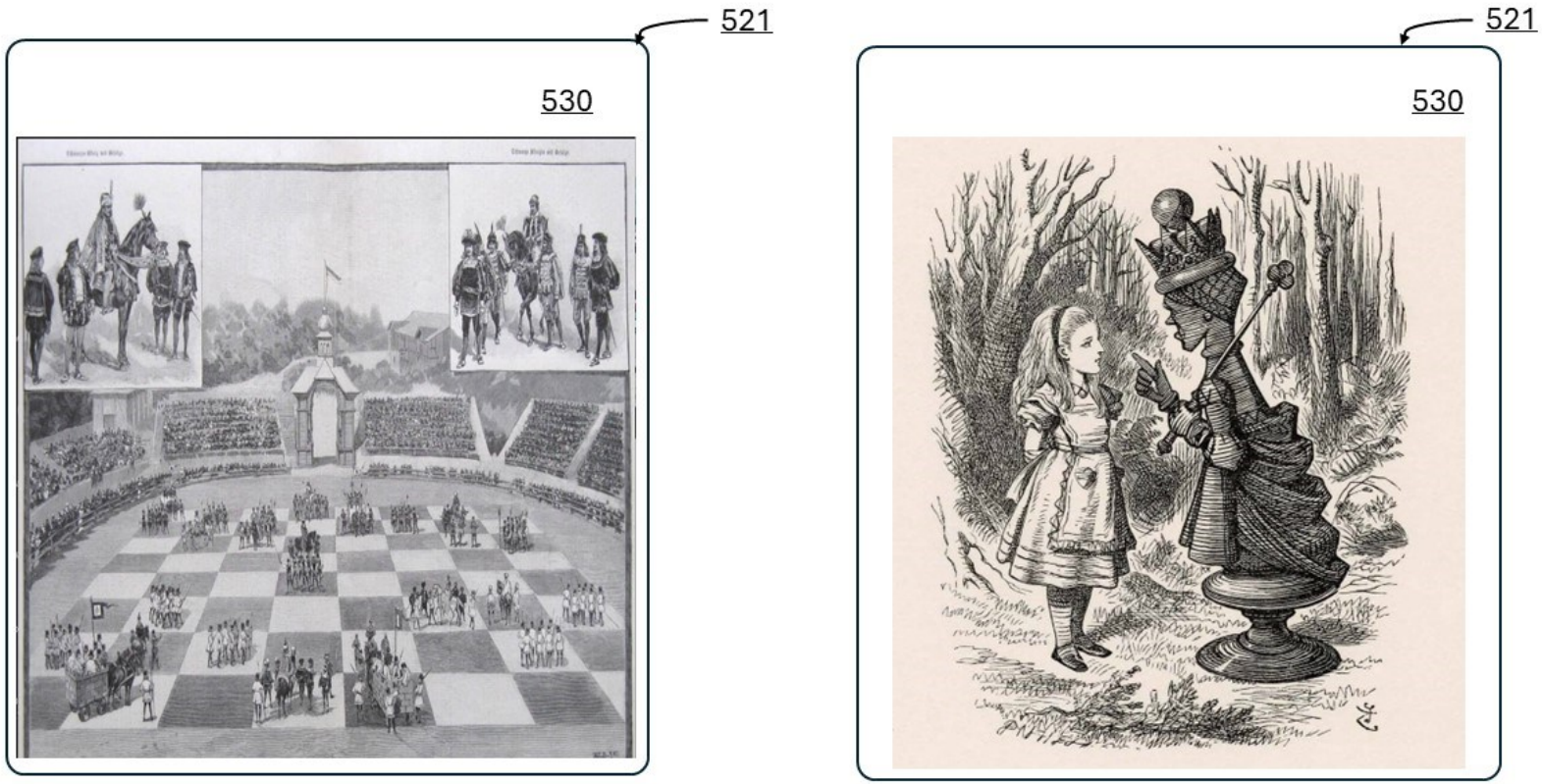

FIG. 5-3-C

CONCEPTUAL DIAGRAM OF EXAMPLE NON-CHARACTERS 530

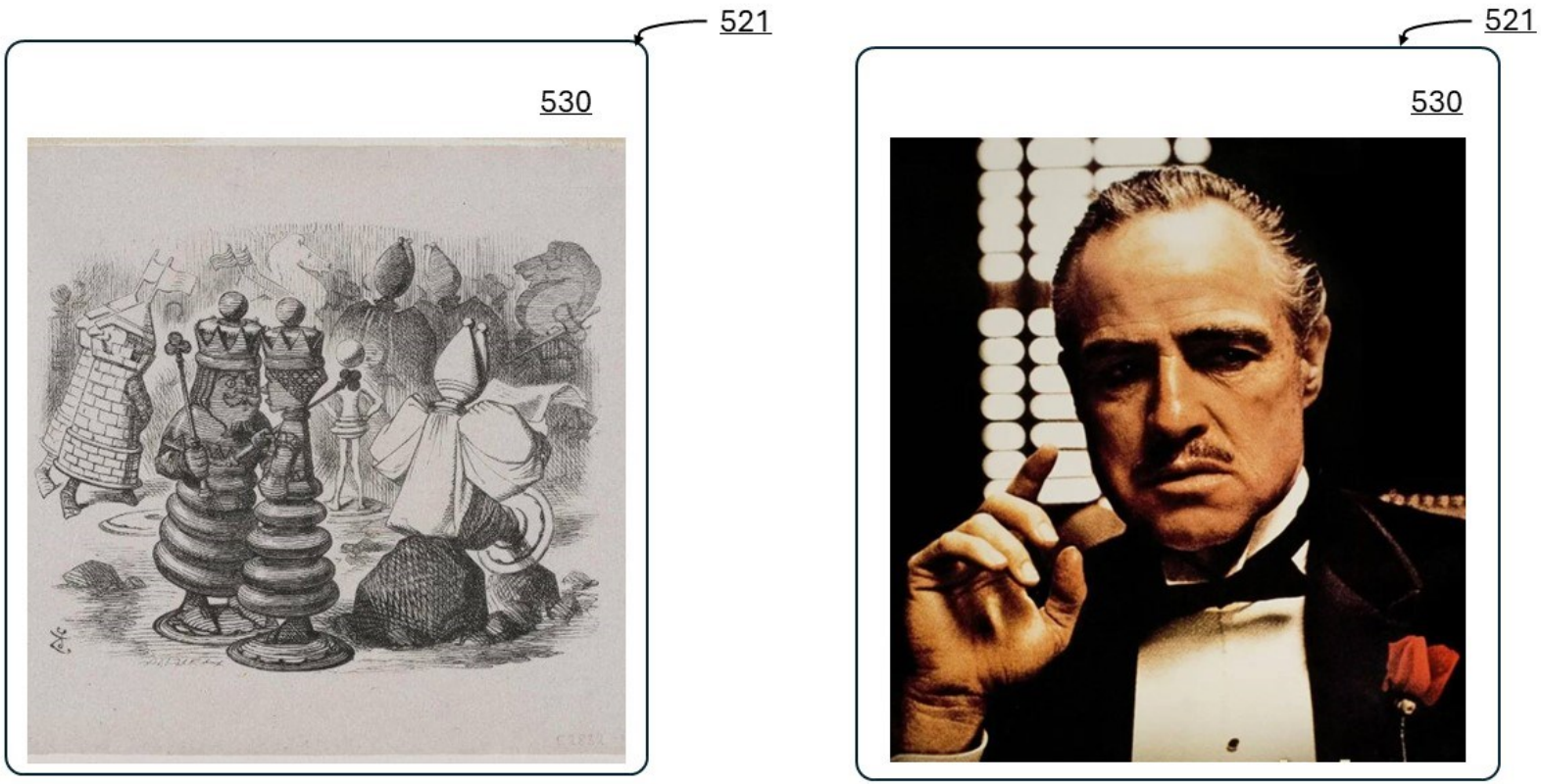

FIG. 5-3-D

EXAMPLE TAXONOMICAL CLASSIFICATION OF THE SEMI-PLAYER CLASS 540 OF DIGITAL IN-GAME ASSETS

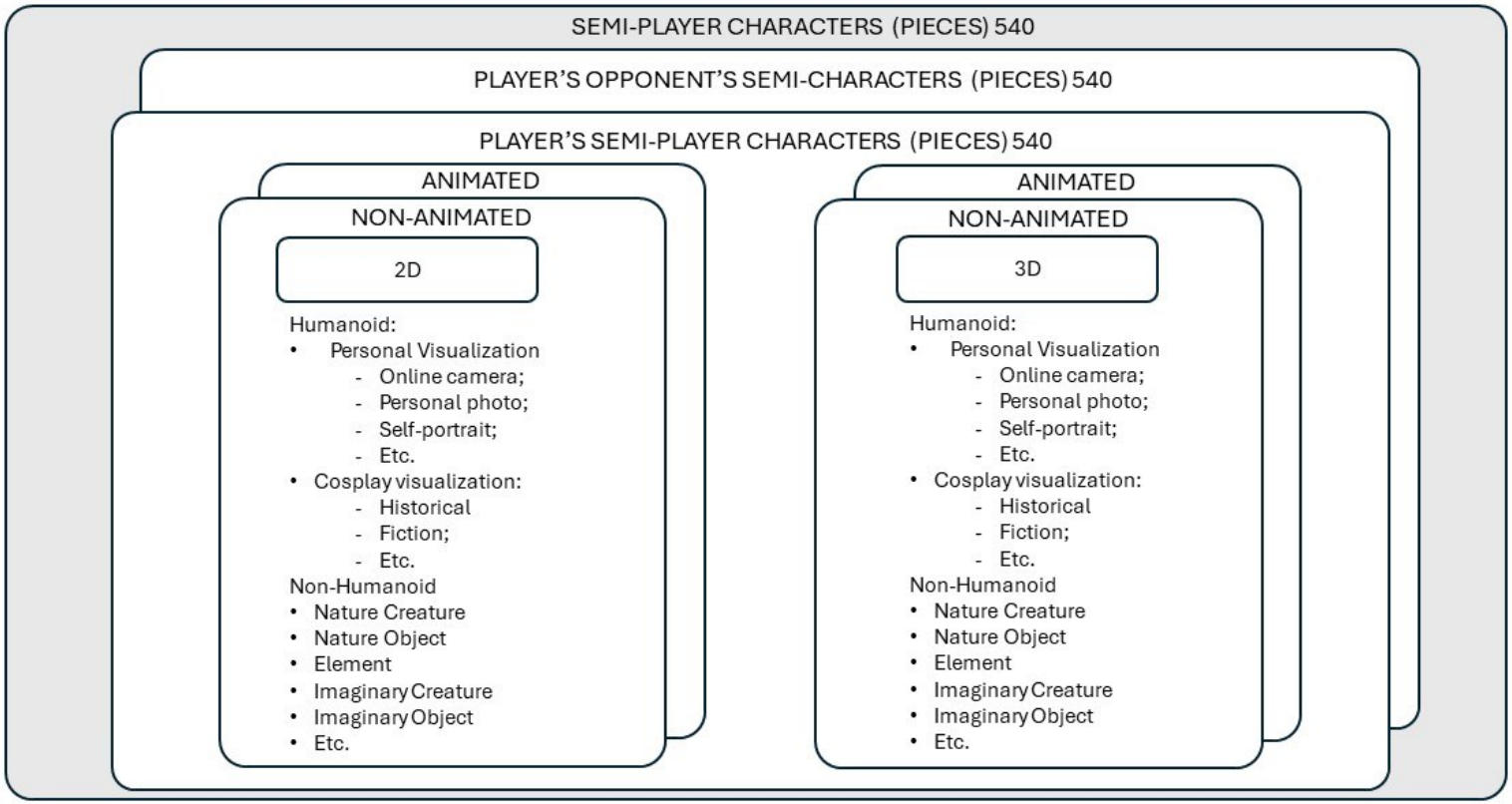

FIG. 5-4

EXAMPLE TAXONOMICAL CLASSIFICATION OF THE ENVIRONMENTS (SETTINGS) CLASS 550 OF DIGITAL IN-GAME ASSETS

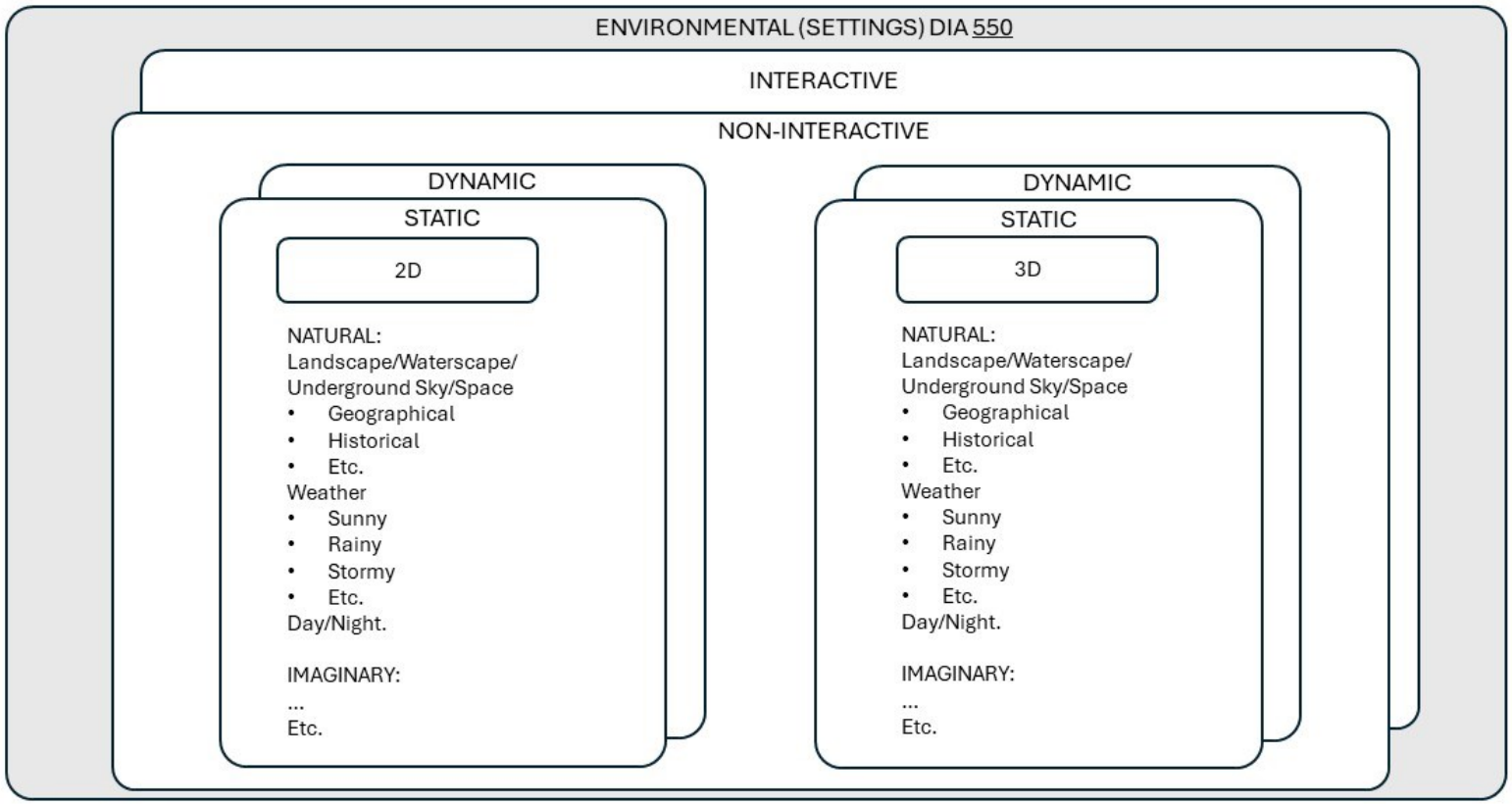

FIG. 5-5

CONCEPTUAL DIAGRAM OF EXAMPLE ENVIRONMENTS (SETTINGS) CLASS 550 OF DIGITAL IN-GAME ASSETS IN-GAME REPRESENTATIONS

FIG. 5-5-A

CONCEPTUAL DIAGRAM OF EXAMPLE ENVIRONMENTS (SETTINGS) CLASS 550 OF DIGITAL IN-GAME ASSETS IN-GAME REPRESENTATIONS

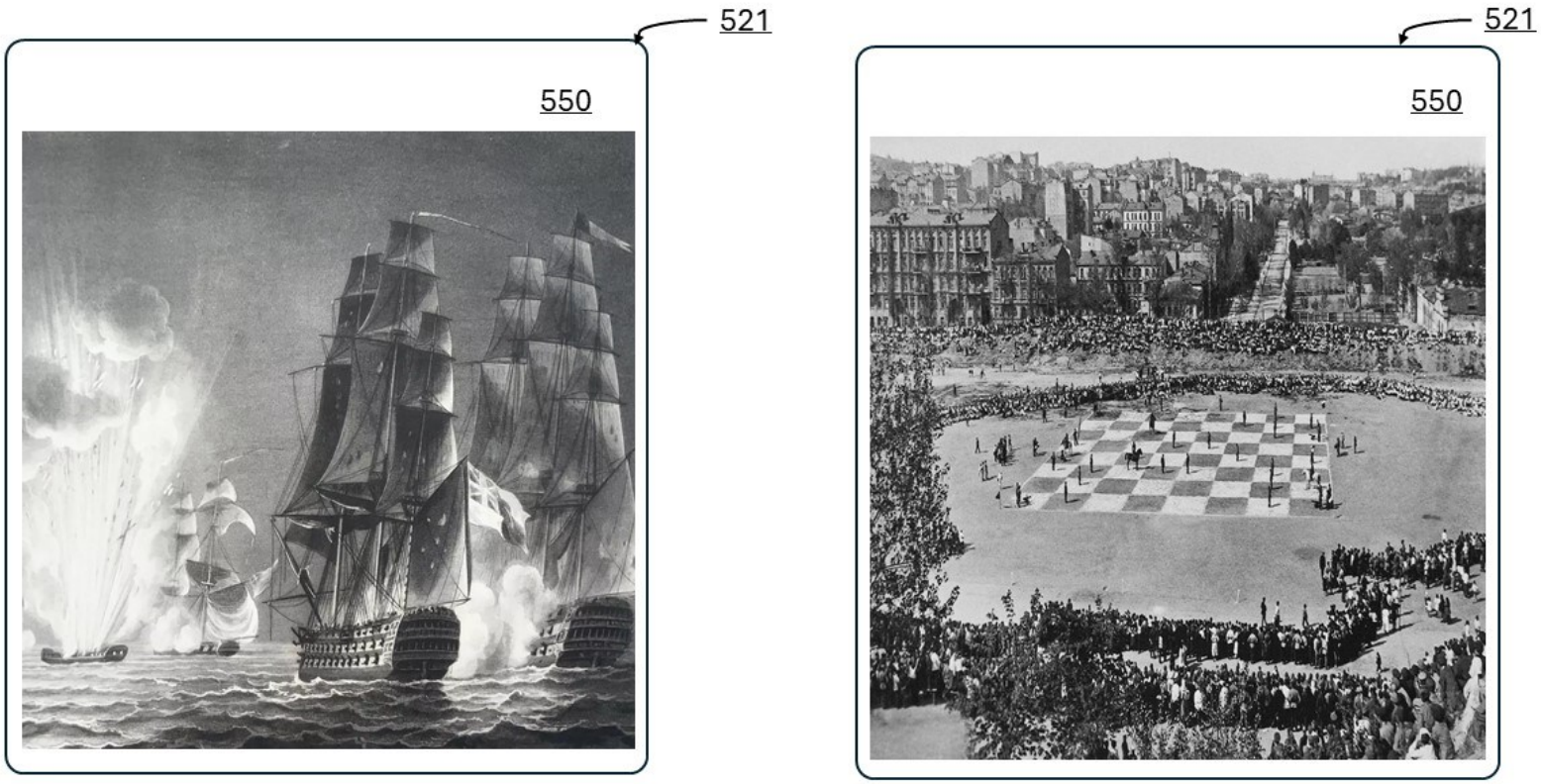

FIG. 5-5-B

CONCEPTUAL DIAGRAM OF EXAMPLE ENVIRONMENTS (SETTINGS) CLASS 550 OF DIGITAL IN-GAME ASSETS IN-GAME REPRESENTATIONS

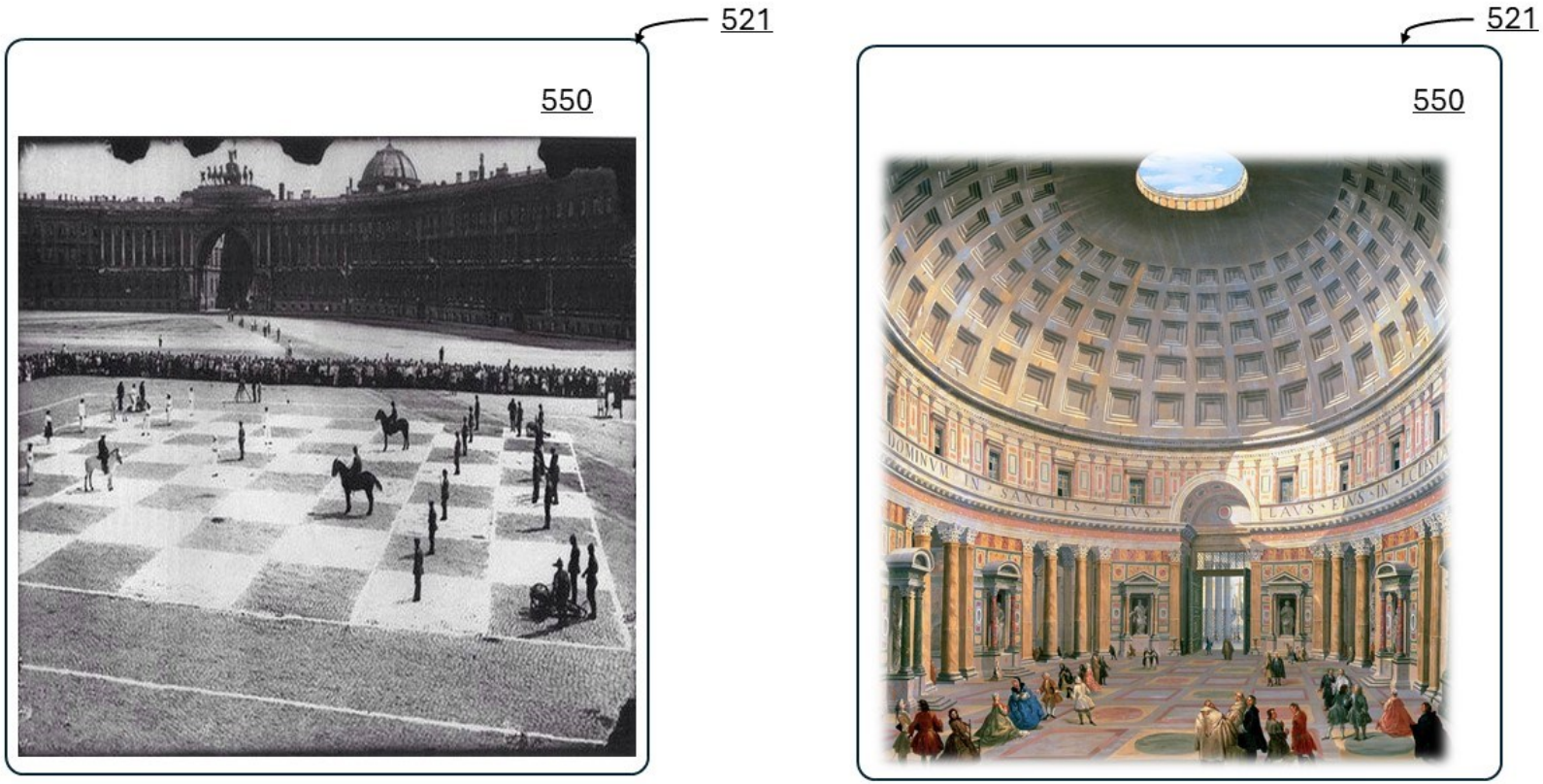

FIG. 5-5-C

EXAMPLE TAXONOMICAL CLASSIFICATION OF THE VISUAL SPECIAL EFFECTS CLASS 560 OF DIGITAL IN-GAME ASSETS

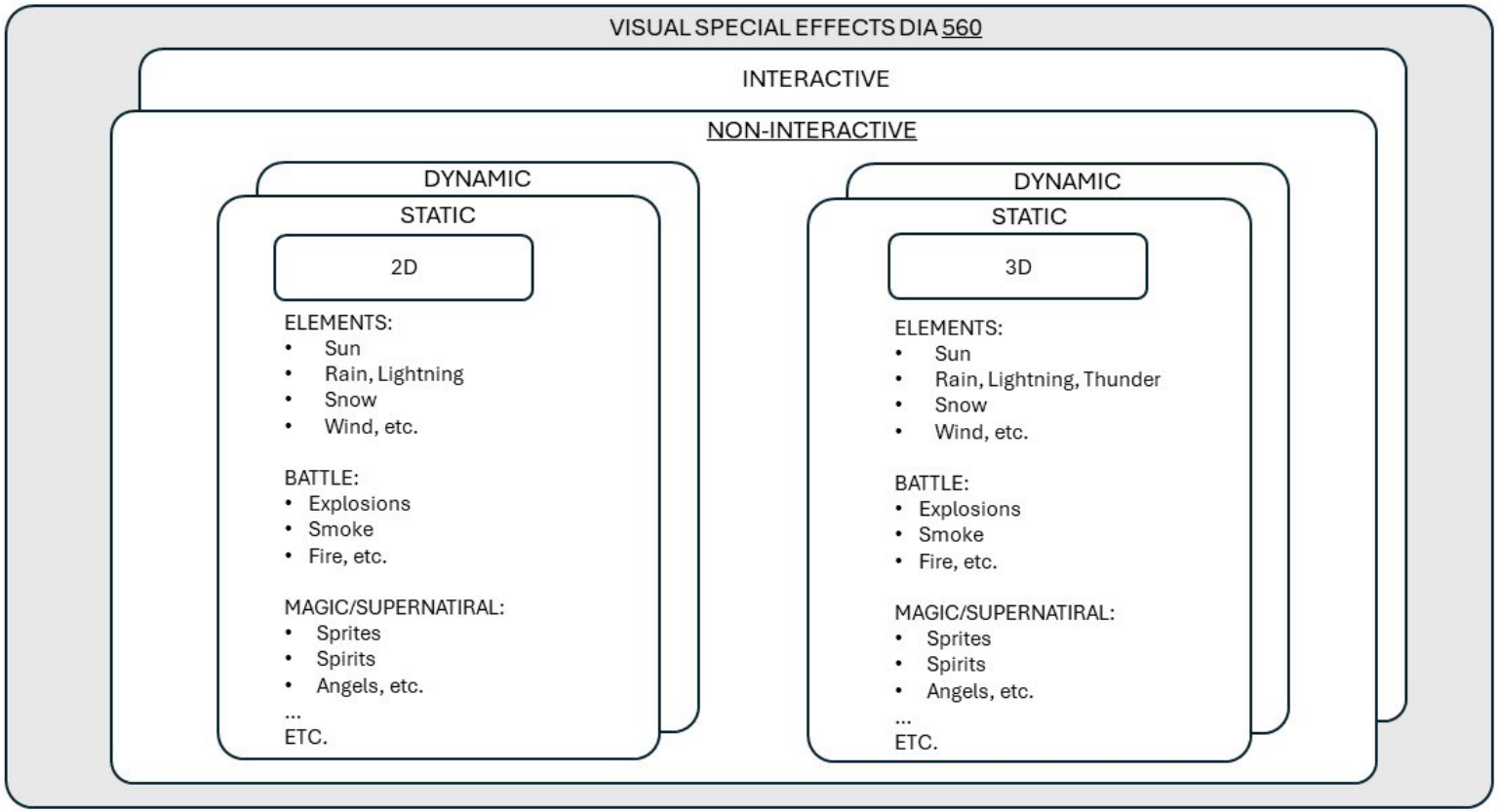

FIG. 5-6

CONCEPTUAL DIAGRAM OF EXAMPLE OF THE OPTIONS MENUE OF GUI 521 FOR THE VISUAL SPECIAL EFFECTS CLASS 560 OF DIGITAL IN-GAME ASSETS 500 OPTION MENUE IN GUI

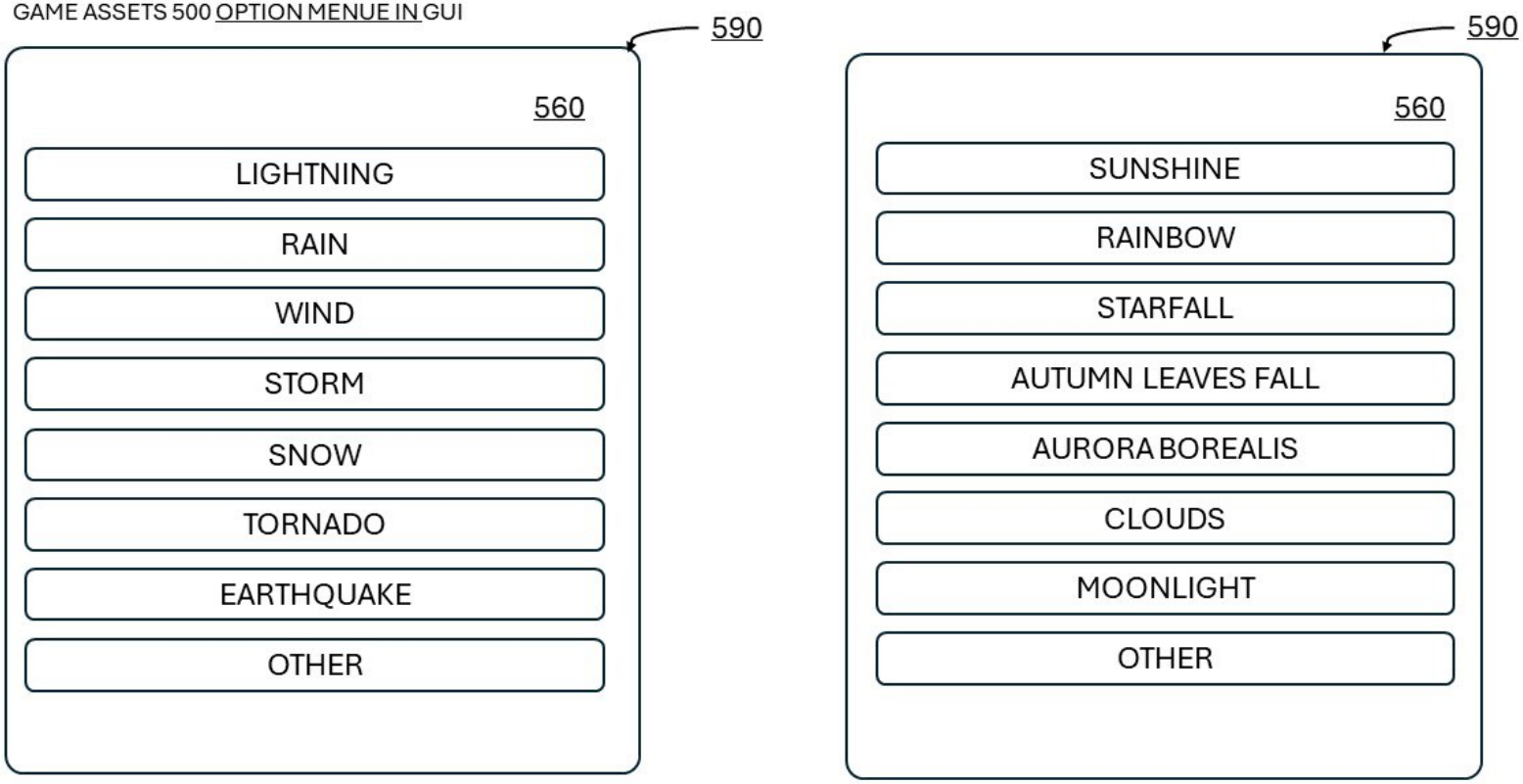

FIG. 5-6-A

EXAMPLE TAXONOMICAL CLASSIFICATION OF THE AUDIO CLASS 570 OF DIGITAL IN-GAME ASSETS 500

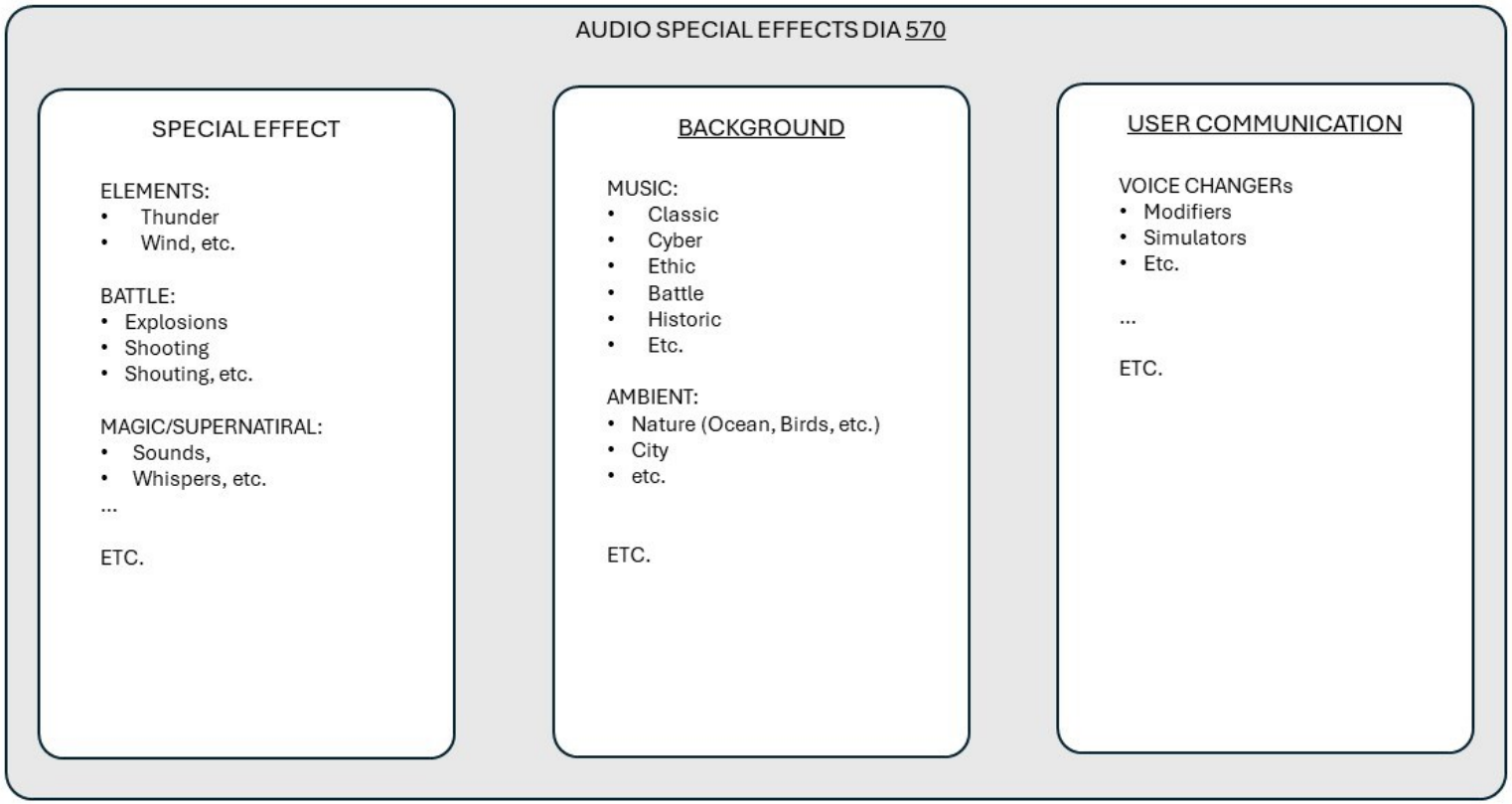

FIG. 5-7

CONCEPTUAL DIAGRAM OF EXAMPLE THE AUDIO CLASS 570 OPTION MENUE GUI

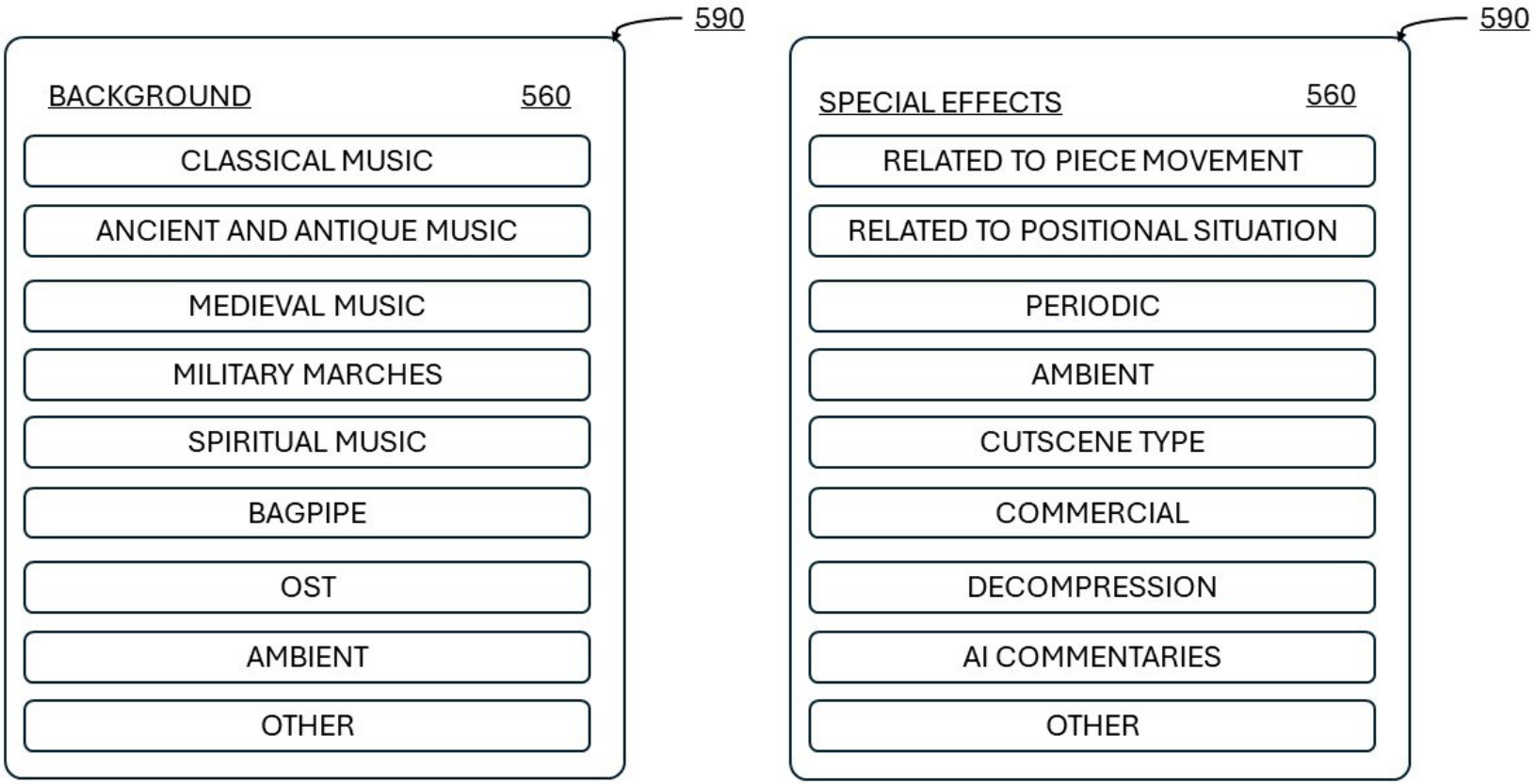

FIG. 5-7-A

EXAMPLE TAXONOMICAL CLASSIFICATION OF THE MODES CLASS 580 OF DIGITAL IN-GAME ASSETS

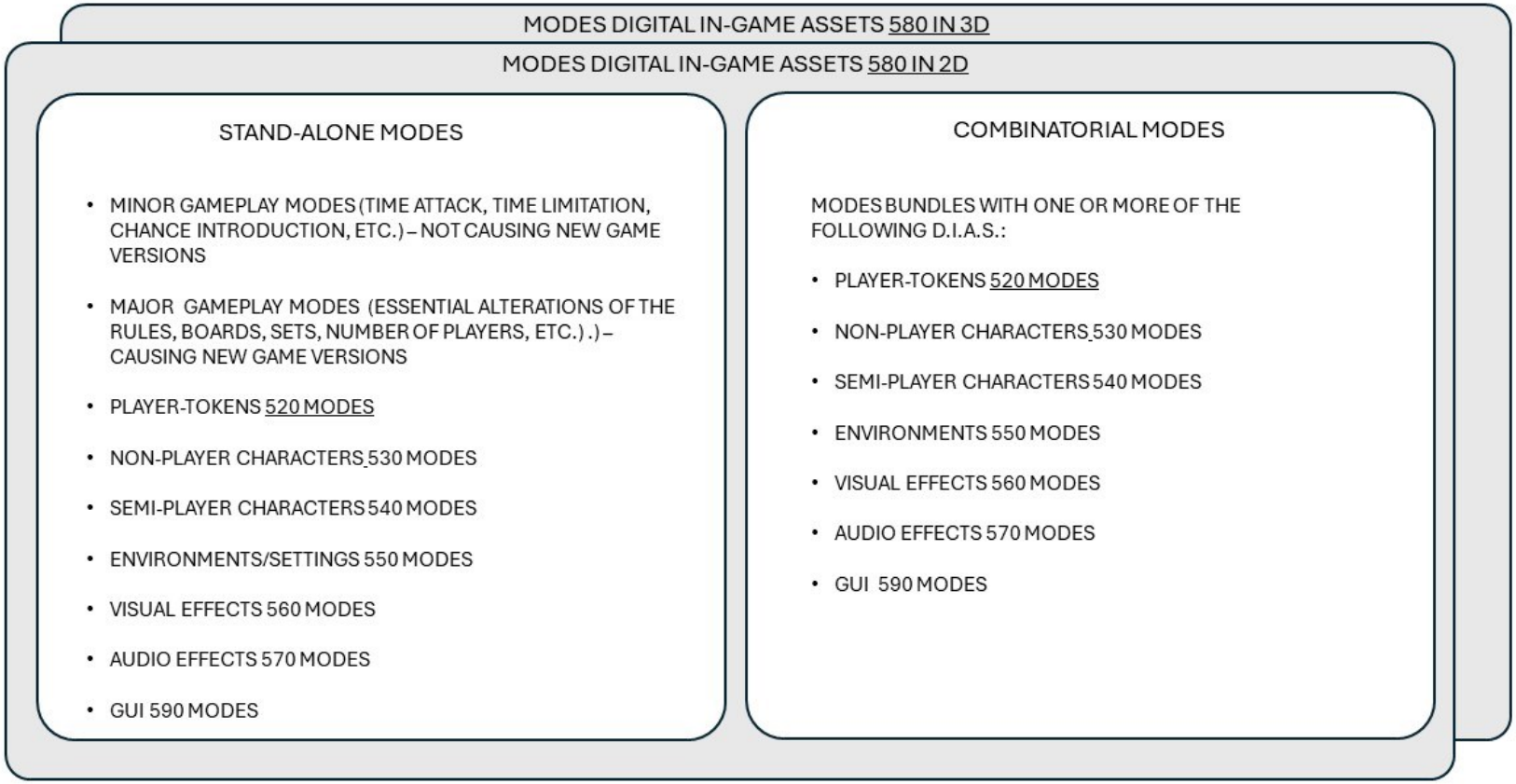

FIG. 5-8

EXAMPLE TAXONOMICAL CLASSIFICATION OF THE GUI AND TOOLSBARS CLASS 590 OF DIGITAL IN-GAME ASSETS

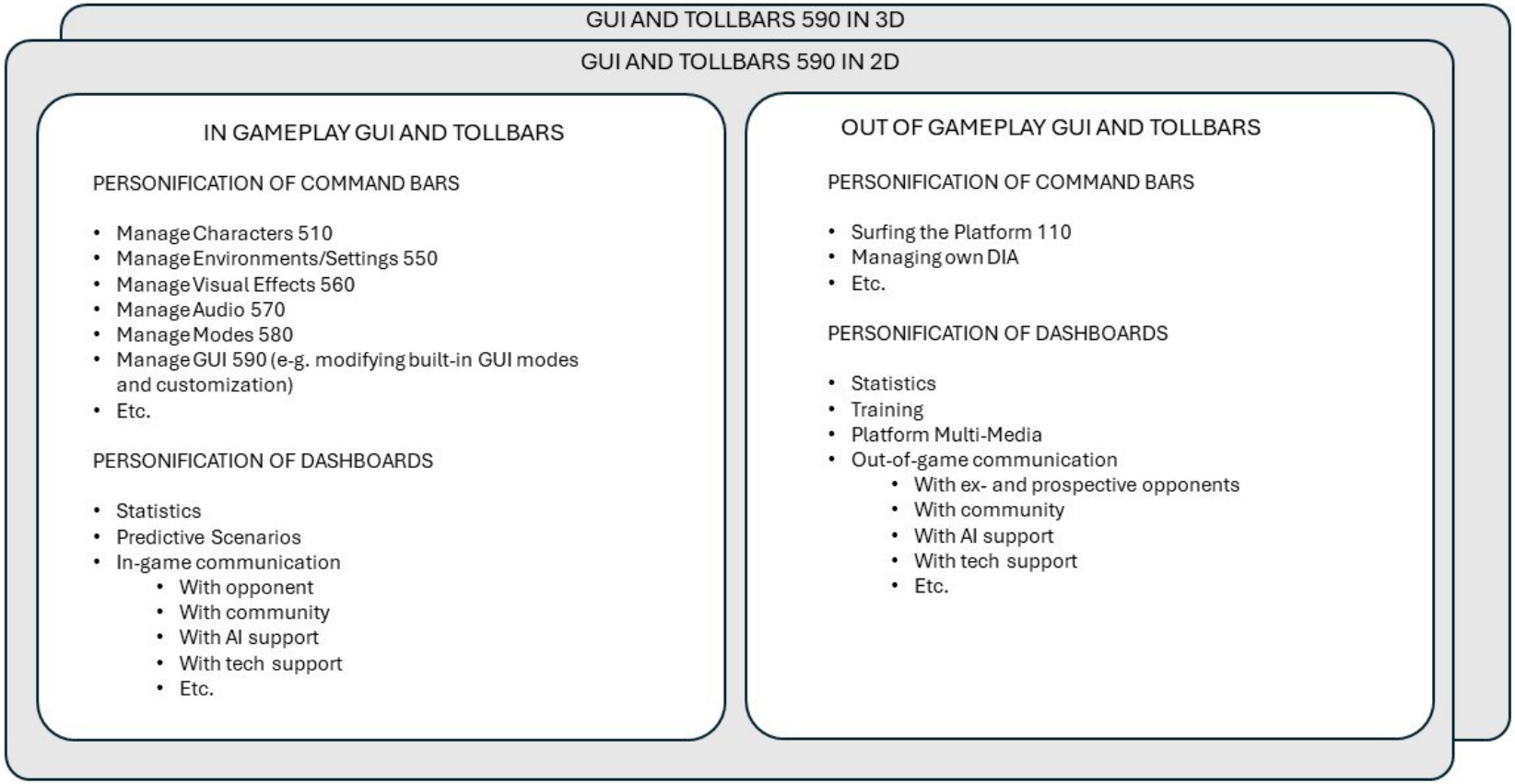

FIG. 5-9

EXAMPLE TAXONOMICAL CLASSIFICATION OF THE VERSIONS OF THE GAME 600

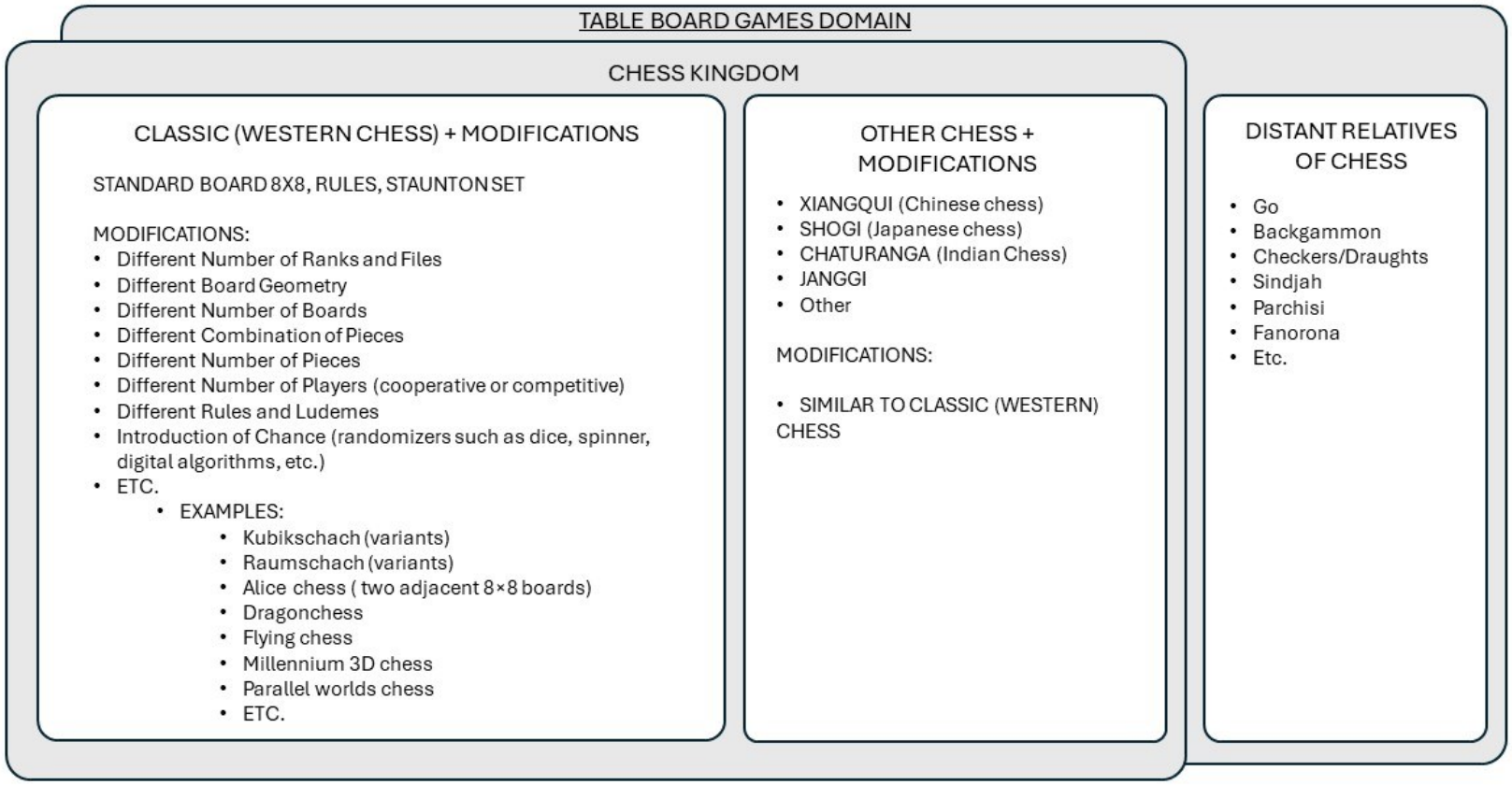

FIG. 6

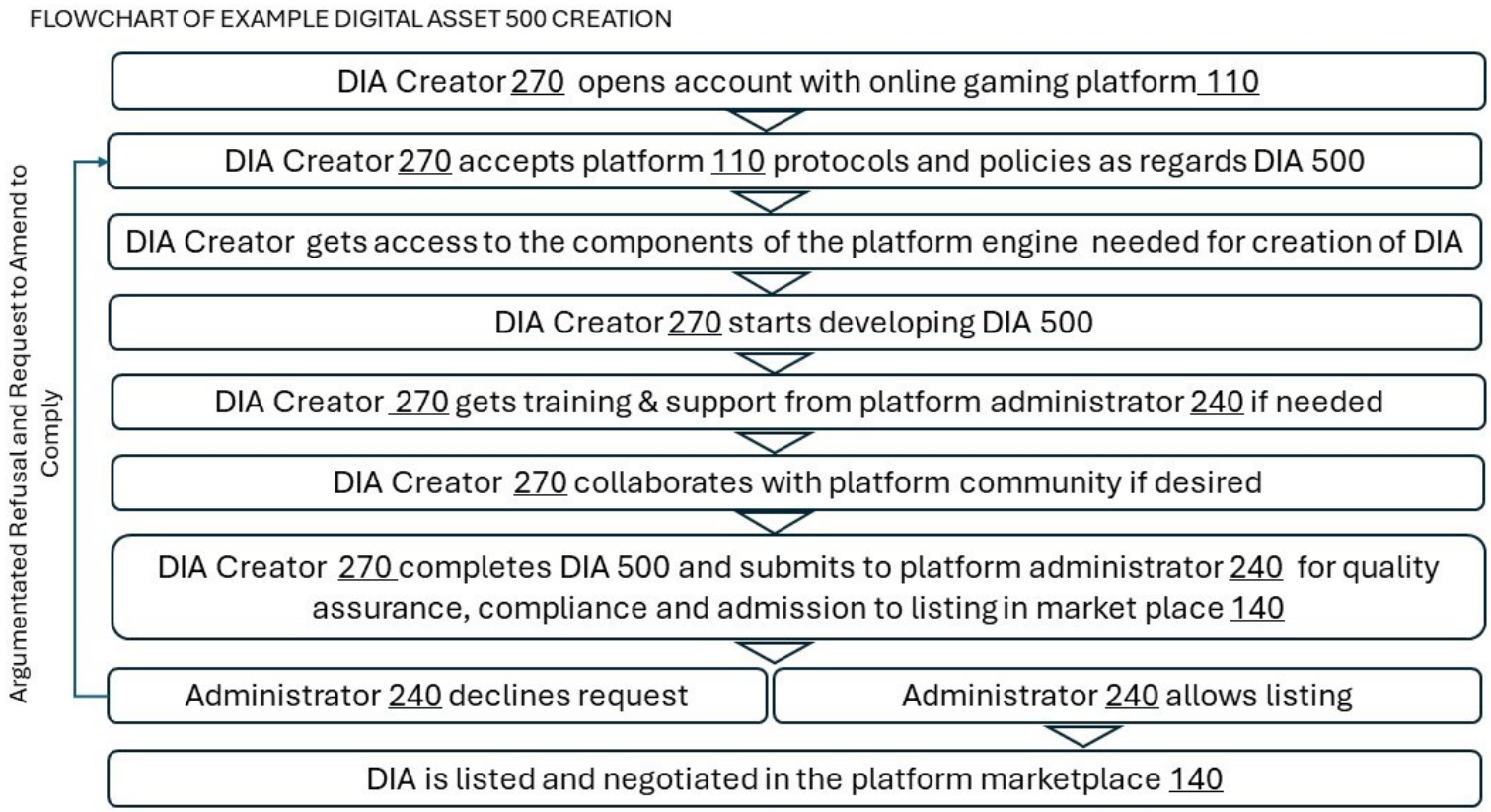
FLOWCHART OF EXAMPLE DIGITAL ASSET 500 CREATION
DIA Creator 270 opens account with online gaming platform 110
DIA Creator 270 accepts platform 110 protocols and policies as regards DIA 500
DIA Creator gets access to the components of the platform engine needed for creation of DIA
DIA Creator 270 starts developing DIA 500
DIA Creator 270 gets training & support from platform administrator 240 if needed
DIA Creator 270 collaborates with platform community if desired
DIA Creator 270 completes DIA 500 and submits to platform administrator 240 for quality assurance, compliance and admission to listing in market place 140
Administrator 240 declines request
Argumentated Refusal and Request to Amend to Comply
Administrator 240 allows listing
DIA is listed and negotiated in the platform marketplace 140

FIG. 7

FLOWCHART OF EXAMPLE PLAYER GAMEPLAY START, FLOW, COMPLETION

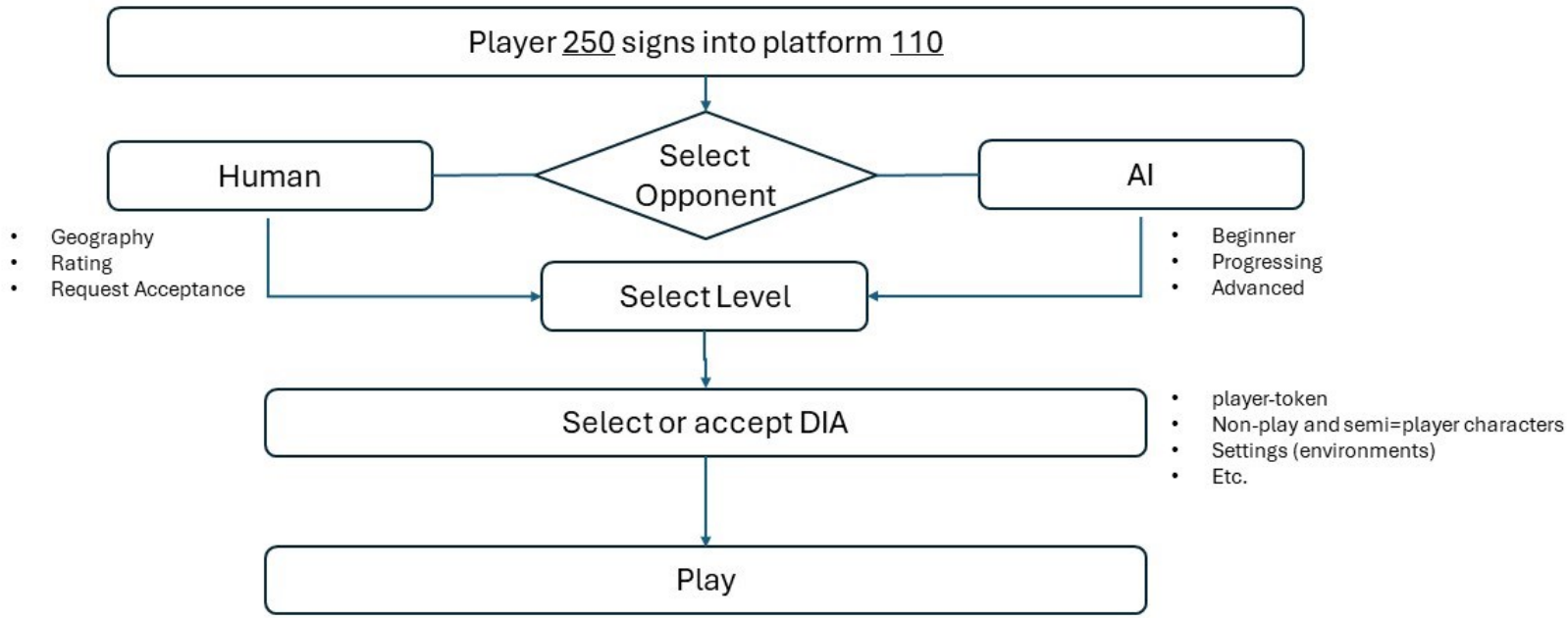

FIG. 8

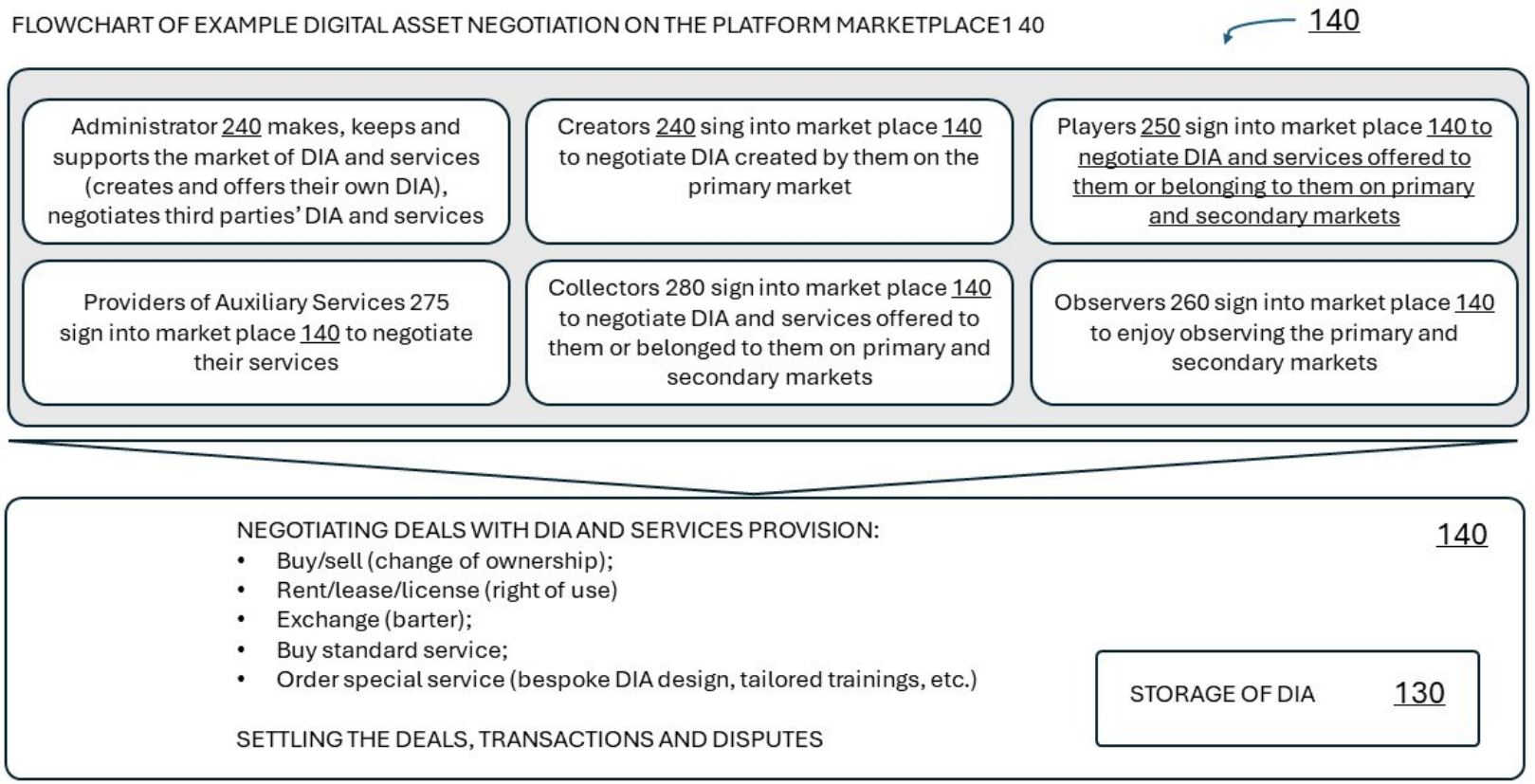
FLOWCHART OF EXAMPLE DIGITAL ASSET NEGOTIATION ON THE PLATFORM MARKETPLACE1 40
140
Administrator 240 makes, keeps and supports the market of DIA and services (creates and offers their own DIA), negotiates third parties' DIA and services
Creators 240 sing into market place 140 to negotiate DIA created by them on the primary market
Players 250 sign into market place 140 to negotiate DIA and services offered to them or belonging to them on primary and secondary markets
Providers of Auxiliary Services 275 sign into market place 140 to negotiate their services
Collectors 280 sign into market place 140 to negotiate DIA and services offered to them or belonged to them on primary and secondary markets
Observers 260 sign into market place 140 to enjoy observing the primary and secondary markets
NEGOTIATING DEALS WITH DIA AND SERVICES PROVISION:
140
• Buy/sell (change of ownership);
• Rent/lease/license (right of use)
• Exchange (barter);
• Buy standard service;
• Order special service (bespoke DIA design, tailored trainings, etc.)
SETTLING THE DEALS, TRANSACTIONS AND DISPUTES
STORAGE OF DIA 130

FIG. 9