February 2020

SYSTEM AND METHOD FOR NAVIGATION SYSTEMS OF VIDEO GAMES

HP INC

System and Method for Navigation Systems of Video Games

This disclosure describes a system to help video game players navigate within the game world. This navigation system is built upon maps of video games, of which this disclosure also provides a method for creating.

Video game players with busy real lives want to immerse into gaming environment in their spare time. They want to experience the gaming mechanisms, plots, or sense of accomplishment, but do not necessarily have the leisure to explore the game world or solve puzzles. A good navigation system built on top of a map can better help game players experience games. Similarly, game players with poor sense of direction can also better enjoy games with help from a navigation system. And while video games may come with built-in maps or similar mechanisms, the information is restricted to within the game and difficult to extract geolocation data to outside the game, causing enhancements to the map to be severely limited.

Navigation System

Traditionally, before the Internet became mainstream, video game players would use video game strategy guidebooks to help them navigate through games. Later, players would search on bulletin board systems or websites for strategy guides posted by video game enthusiasts or book publishers. In either case, video game players would have to identify their geolocation and/or progress within the game first before looking up the corresponding section in the strategy guide. They then must digest the content of the strategy guide and map the context back to the game. This introduces additional shift of attention and interrupts video game players from being absorbed in the zone. This approach is also undesirable for video game players who have difficulty reading maps.

The navigation system from this disclosure offers experiences similar to car GPS navigation systems, such as routing information and step-by-step directions, by overlaying information on top of the video game players' screen. With reference to Fig. 1, the system identifies character's surge **100**, sway **105**, heave **110**, roll **115**, pitch **120**, and yaw **125** movements, as well as recognizing in-game geolocation coordinates **130** (if available) or feature points in the background **135**, to resolve the current geolocation within the game world.

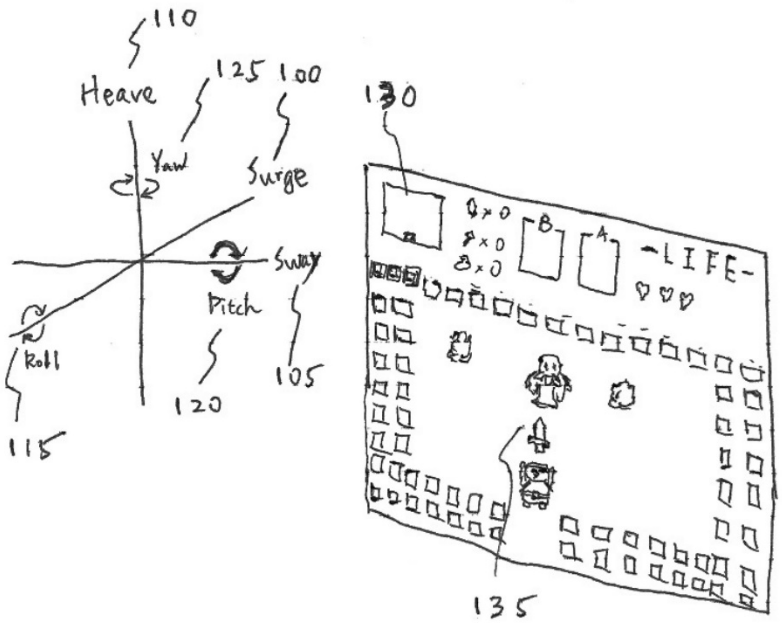

Fig. 1 Resolve characters' current geolocation within game world.

The system tracks the current geolocation information within the game **200** and fetches corresponding map data **205** from a remote service **210** in the background, as shown in Fig. 2.

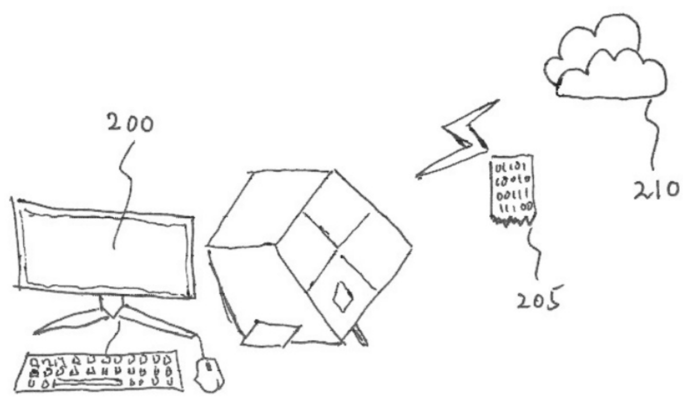

Fig. 2 Fetch map data from a remote service.

With reference to Fig. 3, video game players bring up the navigation system **300** as an in-game overlay and query their destination **305**. The system sends the current geolocation information within the game and the query to a remote service and retrieves the query results back. Video game players can also inspect the points of interest **310** in the region and select one as their destination.

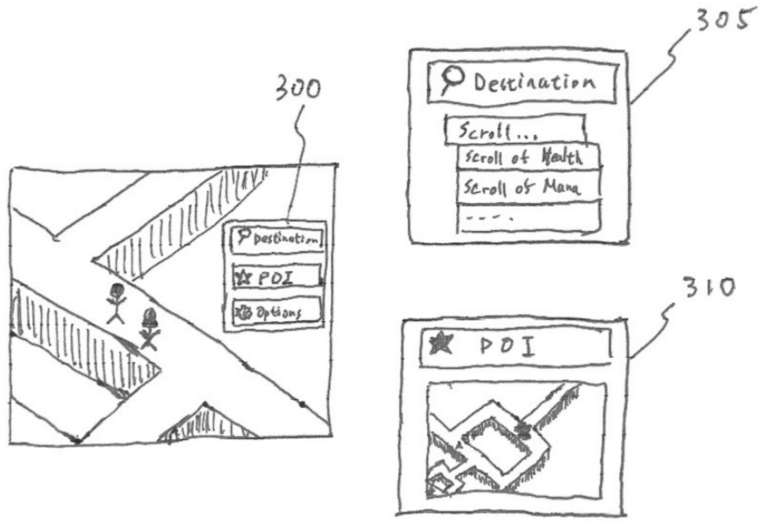

Fig. 3 Navigation system.

Once video game players selected their destination, the system can show a preview of the route information, as shown in Fig. 4, before the video game players proceed with the navigation. The invention then shows step-by-step direction, as shown in Fig. 5, within the game as overlay.

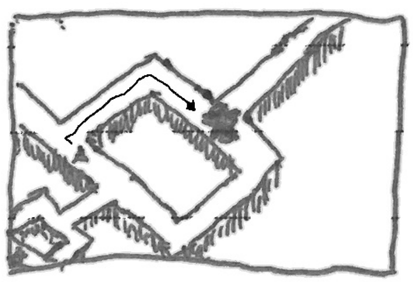

Fig. 4 Preview of route information.

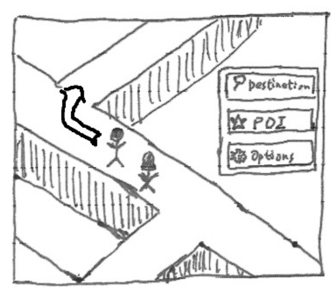

Fig. 5 Step-by step direction.

<u>Maps of video games</u>

As for maps, early video game players have created hand-drawn maps along with their gameplay. However, the process is tedious and error prone. The produced maps are also static and cannot be consumed by computation system easily. Others have also reverse-engineered games at run-time and extracted maps from memory locations. However, this approach requires a functional debugger to access memory containing map data, or an emulator to dynamically output map data. Also, the logic to access or output map data can vary drastically between games.

This disclosure creates maps of video games by analyzing information collected from video recordings and/or live streaming of gameplay. With reference to Fig. 6, streams usually consist of three regions: footage of actual gameplay **600**, footage of the streamers entertaining their viewers **605**, and a region where viewers express their reactions **610**. Video analyzing agents with AI capabilities can recreate maps and extract much information by analyzing these regions from the vast amount of video game streams on livestreaming platforms, which became popular in the mid-2010s. This information can include in-game geolocation data, point of interests, event forecasts, routing information, etc.

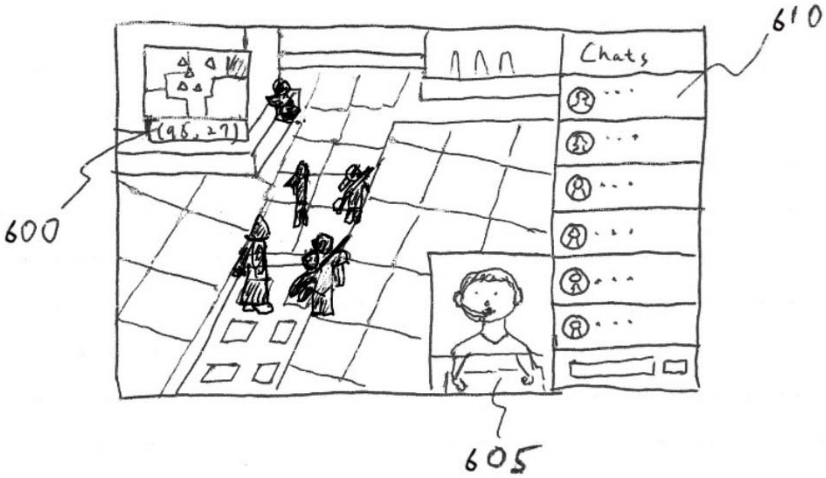

Fig. 6 Footage of actual gameplay.

With reference to Fig. 1 again, the agent can track character's surge **100**, sway **105**, heave **110**, roll **115**, pitch **120**, and yaw **125** movements, as well as recognizing in-game geolocation coordinates **130** (if available), to recreate map and route information from footage of actual gameplay. The agent can also recognize various point of interests, e.g., loots **135**, non-player characters, mobs, pitfalls, obstacles, etc., from the footage.

With reference to Fig. 7, the agent can analyze speech **700** from the streamers and text from the viewers to supplement accuracy and details of the map. For example, viewers may explain what the loot is for. The agent can also detect facial emotion of the streamers **705** and/or reactions from the viewers **710** to determine additional attributes to points of interests. For example, the alignment of the non-player characters.

Fig. 7 Analyze speech from streamer and text from viewers.

And by aggregating analysis from multiple streams, the agent can derive event information and generate insights. For example, the agent can use video stutters or glitches as indication to server overloading for massively multiplayer online games. Or, as shown in Fig. 8, the agent can identify the optimal route **800** to complete quests based on the multiple attempts from various streamers.

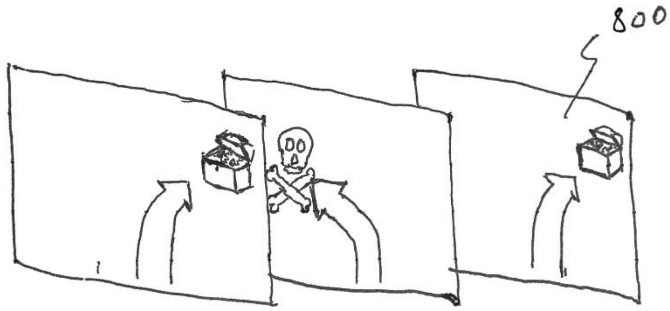

Fig. 8 Optimal route to complete quests based on the multiple attempts from various streamers.

Disclosed by Ron Huang, HP Inc.