templates, etc., as the data. Further, the device storage unit 22' may store temporary data relating to predetermined processing.

FIG. 8B depicts a facility table. In the facility table, for each player, an ID of each facility arranged within the game space by the player, a type ID, a position within the game space, etc., are recorded.

The device processing unit 25 includes one or more processors and their peripheral circuits. The device processing unit 25 is, for example, a CPU, and integrally controls an overall operation of the portable device 2. The device processing unit 25 controls operations of the device communication unit 21, the display unit 24, etc., so that various types of processing of the portable device 2 are executed in an appropriate order in accordance with the programs stored in the device storage unit 22', the operation of the operation unit 23, etc. The device processing unit 25 executes processing based on the programs (the operating system program, the driver program, the application program, etc.) stored in the device storage unit 22'. The device processing unit 25 can execute multiple programs (application programs, etc.) in parallel.

FIG. 9 illustrates a concept of applying a template in a multi-player environment.

900 illustrates a game space. Twelve facilities are arranged within the game space 900. Specifically, four facilities illustrated as "black circle", four facilities illustrated as "black triangle", and four facilities illustrated as "black square" are arranged therein. Among these facilities, assume that the one facility illustrated as "black circle" and the two facilities illustrated as "black triangle" arranged in the upper-right three by three squares are those of a player$_1$. Further, assume that the three facilities illustrated as "black square" arranged in the lower-right three by three squares are those of a player$_2$, the two facilities illustrated as "black triangle" and the one facility illustrated as "black square" arranged in the lower-left three by three squares are those of a player$_3$, and the three facilities illustrated as "black circle" arranged in the upper-left three by three squares are those of a player$_4$.

Assume that a template 910 has been applied to an area 901 within the game space 900 by the player$_1$. Similarly, assume that templates 920 to 940 have been applied to areas 902 to 904 by the player$_2$ to player$_4$, respectively. In relation to the player$_1$, the number of types of facilities and the number of facilities in each type arranged within the game space 900 are equal to the number of types of facilities and the number of facilities in each type, respectively, positions of the facilities being defined by the template 910. Thus, all facilities of the player$_1$ are moved to positions of facilities as defined by the template 910. Similarly, all facilities of the player$_2$ to player$_4$ are moved to positions of facilities as defined by the templates 920 to 940, respectively. 900' illustrates the game space 900 after all the facilities have been moved.

In order to achieve the above-described functions, the device processing unit 25 includes a game progression unit 251', a template creation unit 252, a template application unit 253, and a second template application unit 254. All of these units are functional modules implemented by a program executed on a processor provided in the device processing unit 25. Alternatively, these units may also be provided as firmware on the portable device 2.

Since the template creation unit 252 and the template application unit 253 are the same as illustrated in FIG. 2A, a description thereof is omitted.

In the following, processing by the game progression unit 251' will be described.

The game progression unit 251' controls the start and progression of the game, and appropriately gives commands to execute processing to the template creation unit 252, template application unit 253, second template application unit 254, etc.

Specifically, when a command to start the game is given by the player via the operation unit 23, the game progression unit 251' displays the game progression screen 300.

When a command to create a template is given by the player via the operation unit 23, the game progression unit 251' gives a command to execute processing to the template creation unit 252.

When a command to apply a template is given by the player via the operation unit 23, the game progression unit 251' gives a command to execute processing to the template application unit 253.

When a template application command is received from a different portable device 2 via the device communication unit 21, the game progression unit 251' gives the second template application unit 254 a command to execute processing, by using the received template application command as a parameter.

When a command to execute different processing is given by the player via the operation unit 23, the game progression unit 251' executes the different processing.

In the following, processing by the second template application unit 254 will be described.

The second template application unit 254 obtains a template from the server 3, and applies the obtained template.

Specifically, the second template application unit 254 obtains a template from the server 3.

In other words, the second template application unit 254 interprets the received template application command, and specifies the ID of the player, the ID of the template, and the coordinates of the area to which the template is to be applied. Further, the second template application unit 254 sends a template provision request via the device communication unit 21 to the server 3 by using the specified template ID as a parameter. Further, the second template application unit 254 receives thumbnail image data of a corresponding template, as well as the type ID and position of each facility from the server 3 via the device communication unit 21. The second template application unit 254 then stores the received thumbnail image data in the device storage unit 22'. Further, the second template application unit 254 stores the ID of the specified template, the file name of the stored thumbnail image data, the received type ID and position of each facility, etc., in the template table stored in the device storage unit 22'.

The second template application unit 254 applies the obtained template.

In other words, the second template application unit 254 refers to the facility table stored in the device storage unit 22' by using the ID of the specified player as key, and extracts an ID, a type ID and a position within the game space of each facility of the corresponding player. The second template application unit 254 counts the number of extracted types of facilities and the number of facilities in each type. The second template application unit 254 further refers to the template table stored in the device storage unit 22' by using the ID of the specified template as key, and extracts a type ID and a position within the template of each facility in the corresponding template. The second template application unit 254 counts the number of extracted types of facilities and the number of facilities in each type. Moreover, the second template application unit 254 converts the extracted positions within the template to positions within the game space based on coordinates of the specified area. For each type of facility, the second template application unit 254 compares the number of facilities of this type within the game space and the number of facilities of this type within the template, and, according to the result, moves the facilities of this type within the game space to the positions of the facilities of this type within the template.

Then, the second template application unit 254 terminates the processing.

Since the schematic configuration of the server 3 is the same as illustrated in FIG. 5A, a description thereof is omitted.

FIGS. 10A and 10B illustrate examples of the operational flow of the portable device 2.

The below-described operational flow is executed, based on a program that is stored in advance in the device storage unit 22', mostly by the device processing unit 25 by working together with each component of the portable device 2.

FIG. 10A illustrates an example of the operational flow of the game progression unit 251'.

Since Steps S100 to S108 are the same as illustrated in FIG. 6A, a description thereof is omitted.

When a template application command is received from a different portable device 2 via the device communication unit 21 (Step S300--Yes), the game progression unit 251' gives the second template application unit 254 a command to execute processing, by using the received template application command as a parameter (Step S302).

FIG. 10B illustrates an example of the operational flow of the second template application unit 254.

The second template application unit 254 obtains a template corresponding to the template ID included in the received template application command, from the server 3 (Step S310).

The second template application unit 254 applies the obtained template (Step S312). Then, the second template application unit 254 terminates the processing.

On the other hand, when a command for different processing is given by the player via the operation unit 23 (Step S300--No), the game progression unit 251' executes the different processing (Step S110).

As have been described above, by allowing each player to change the arrangement of facilities by using templates in a multi-player environment, the usability of city building games is improved, and it becomes possible to continuously attract players to the game.

It should be noted that the present invention is not limited to the above-described embodiment. For example, in the above-described embodiment, in order to combine multiple templates to create a single template, it is assumed that a player applies multiple templates to predetermined areas within the game space, or multiple players apply a template each to predetermined areas within the game space, and then a template for a predetermined area that encompasses all these areas is created. However, a player may designate multiple templates or multiple players may designate a template each, and then a template may be created by directly joining these templates.

FIG. 11 illustrates a concept of combining templates in a multi-player environment.

Assume that the $player_1$ has specified a template 1110 for an area 1100. Further, assume that the $player_2$ to $player_4$ have specified templates 1120 to 1140 for areas 1101 to 1103, respectively. 1150 illustrates a template obtained by combining the templates 1110 to 1140.

In order to achieve the above-described functions, the portable device 2 may perform processing as described below.

When a command to create a template is given by the player via the operation unit 23, the portable device 2 displays a predetermined screen and receives designations of template and area. In the same manner, the portable device 2 receives designations of template and area from a different portable device 2. Then, the portable device 2 obtains the designated templates from the device storage unit 22 or the server 3, and creates a new template by arranging the obtained templates on the designated areas.

In other words, the portable device 2 corrects the position of each facility defined by the obtained templates based on the coordinates specified by the designated areas. The portable device 2 then stores the type ID and corrected position, etc., of each facility defined by the obtained templates, in the template table stored in the device storage unit 22 under a newly assigned template ID.

(Third Embodiment)

In the above-described embodiment, it is assumed that templates are created by the player. However, preexisting templates may also be distributed by a server or the like. In the present embodiment, a preexisting template is distributed by a server depending on an event (e.g., protecting the city from an enemy character) happening in the city building game. The player applies the template distributed by the server, to a predetermined area within his/her own game space, moves and adds facilities as necessary, and thus prepares for the event. After a certain time has passed, the event happens, and the player is given various rewards (e.g., templates, facilities, etc.) depending on the outcome. Such a template can also be understood as a task given to the player in an event.

Since the schematic configuration of the game system 1 is the same as illustrated in FIG. 1, a description thereof is omitted.

FIG. 12 illustrates an example of a schematic configuration of the portable device 2.

The portable device 2 progresses the game in response to an operation of an operation unit 23 by a player. When necessary, the portable device 2 is connected to the server 3 via the base station 4, the mobile communication network 5, the gateway 6, and the Internet 7, to communicate with the server 3. In order to implement the foregoing functions, the portable

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.