1. Workflow
The simulation space consists of six objects: the field object, the avoidance object, the goal object, the event object, the character object, and the parameter object.

The **field object** is a subset of the scene objects where the characters are animated.

The **avoidance object** is an obstacle that the characters avoid.

The **goal object** sets the route of the character. Each character has the goal object and advances toward it. When the character reaches the goal object, he advances toward the new goal object. The new goal object can be chosen by user’s directions. The route can be set freely and easily by changing the goal objects.

The **event object** is an object for the character to generate the event such as changing animation. The user gets necessary animations ready beforehand in a library. Each character loads animation from the library at the event object's request. The character can express various behaviors with the event object, and the user can simply make a work similar to the artificial intelligence.

The **character object** is the character that moves around in the simulation space. The user can select the best from several methods of setting the initial positioning of the characters. Those methods are converting the particles to the characters, manual control, arrangement based on UV of the field object and random arrangement.

The **parameter object** has the parameters as attributes. Therefore, the user can change the values according to the time. Moreover, the output values according to the distance can be edited by using the graph editor. So, the user is not annoyed by complex calculations, and can operate it intuitively. The user can export the parameter to other scenes by exporting the object. And the user can store the old parameter by copying the object.

2. Implementation
The process of making the street which has an intersection with traffic lights

The user begins to make the field object (ground and pedestrian bridge) where the characters walk.

The user make the avoidance object (building and roadway).

The user sets the goal object which is a turning point of the route in each corner.

The user sets the event object which the character stops in front of the crosswalk according to the color of the signal.

We create particles, and assign each character to one of the particles.

We tune the parameters by using the graph editor.

Then we can create the crowd animation (see the movie as well). If the result is not satisfactory, we can simulate the crowd behavior with other parameter settings during any period of time during animation.
3. Demonstration movie
Demonstration movie is QuickTime (H.264/MPEG-4) format. (Duration 4:59, Size 85MB)

The workflow of making the scene
The characters stop in front of the crosswalk according to the color of the signal.
Number of characters: 80
Animation time: 20s (30fps)
Computation Time: 193s

Crowd who wanders in maze
Each character can follow a different route.
Number of characters: 40
Animation time: 20s (30fps)
Computation Time: 70s

Battle scene
Various movements are made by combining some animations.
Number of characters: 160
Animation time: 12s (30fps)
Computation Time: 45s

Human letters formation by characters
The letters can be freely changed by local editing in time and space.
Number of characters: 1000
Animation time: 40s (30fps)
Computation Time: 50m

Appendix: Background technique
We have developed character’s reactive behavior with combination of several methods. First, the crowd’s behavior is based on [Rey87]. The animation consists of three simple behaviors (Separation, Alignment and Cohesion). Next, the character’s behavior uses a part of the basic reactive behavior routines of [Shao05]. But, this method cannot express the relations between characters. Then, it is operated based on personal space of [Unuma95]. It is a method for division into three areas of zone A, zone B, and zone C according to the distance from the character (Fig. 1), and judging other characters that approach. Zone A is an area where the character avoids all people. Zone B is an area where the character accepts friends and avoids other people. Zone C is an area where the character approaches friends and avoids other people. This setting makes the relations such as enemies and friends between the characters. The areas are concentric circles in this method. But, our method makes the areas oval according to the direction of the character. Behaviors of avoiding collision are adjusted by the parameter which combined these methods.

Our method easily can correct and add data to the simulation result. The simulation requires so many parameters to input that sometimes it's very hard to get a desired result. However, our approach allows us to deal only with the characters who should be recalculated in the simulation. We can add and animate more characters, if desired. Then the crowd simulation result is added to an older animation data, so that we get a "global" result where some characters are simulated separately from the others. Or, our method can simulate the surrounding crowd after the user creates manually the main character. Therefore the user can make an effective scene in such a short time as explained above in this note.

References