EG UK Computer Graphics & Visual Computing (2015)
Rita Borgo, Cagatay Turkay (Editors)

Using sketching to control heterogeneous groups

T. Allen, A. Parvanov, S. Knight, S. Maddock

s.maddock@sheffield.ac.uk, University of Sheffield, UK

Abstract

The basic methods of interaction in strategy games with regards to controlling groups of units has largely remained the same since the first strategy games were released. Although the control systems in games today are effective and intuitive, they are somewhat limiting for the user in terms of achieving more complex goals. Recently, there has been research into using sketch-based systems as an alternate means of controlling a crowd, granting a higher level of control to the user while maintaining an easy to use and intuitive interface. So far, however, this has only been implemented for homogeneous groups. This paper describes the implementation of a sketch-based crowd control system for strategy games, which allows the user to exert a greater level of control over their armies by giving them the ability to control heterogeneous groups by using sub-group sketching to distinguish formations and paths for groups and sub-groups to adhere to.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

While the level of control a user can exert on agents in strategy games has increased over the years, many of the basic methods of control have remained the same. Broadly, one can separate different strategy games and the control systems used within them into two main groups. Firstly, there are those games usually focussed on base building and individual unit production in real time (e.g. StarCraft 2 by Blizzard Entertainment) where the player can control each unit either individually or as part of a larger group with a box selection. The other main type of control can be seen in games such as Total War series (by Creative Assembly). In these games, the player controls their armies directly but they are pre-split into small groups or regiments, which the player can move around. This means that the player can give an order to a regiment to move to a location and all units in that group will move, but the player cannot split the group up during the battle. Similarly, the player can also select multiple groups to move but each regiment will always stay within their own formation.

This paper introduces an alternate system of control, one which allows for individual control of units, but allows the user a greater level of formation control like that of group-based control systems. Specifically, the focus will be on using a sketch-based input system, similar to the one proposed by Gu and Deng [GD11, GD13], but extended to control heterogeneous groups.

In Section 2 on related work, we focus on using sketching for formation control. Other aspects of the project are covered by more general sources. For work on using sketch-based interfaces, such as the problems of dealing with noisy user input, see Olsen et al [OSSJ09]. For sources on using agent-based modelling for crowds or collections of individual units, Reynolds' early work [Rey87] or Pelachano et al [PSAB07] are useful papers.

We evaluate the work by comparing a classic unit-based control approach, as would be found in a strategy game, with our new sketching approach, with both systems described in Section 3. Four different scenarios that might be found in a strategy game are considered in the comparison (see Section 4) and the results of user testing are presented in Section 5. Conclusions are given in Section 6.

2. Related work

The focus of our paper is on formation control, where a formation is considered to represent the positions of a group of agents such that the formation boundaries constrain the positioning of individual agents, and the agents themselves fit

DOI: 10.2312/cgvc.20151249

the formation boundaries in an evenly distributed way while showcasing the formation's shape. This corresponds to a formation in Gu and Deng's work [GD11, GD13], which we base our work on. As such, we briefly include descriptions of some of their approaches to aspects of formation control in the following paragraphs.

Early work on controlling a formation defines the group's motion using vectors [KLLT08] or using navigation fields [PB11]. Most state of the art approaches focus on the different ways of defining the formation's boundaries. Brush sketching [UCT04] represents the space a formation inhibits using a brush stroke, such as the ones used in most popular image processing software, making it a simple to use approach for non-complex formations with simple boundary angles. Texture maps [GD13] define the shape using a preprocessed texture showing the positions of individual agents. This approach is very quick and simple albeit limited - the formation cannot be changed and needs to be known beforehand. Flow tiles label the environment around the agents to describe the path and constraints. While intuitively describing agent behaviour, this approach is very inefficient at dealing with uniform formations as the complexity of the scene rises quickly as the number of agents increases. The final method is boundary sketching [GD11, GD13] where the formation is specified using sketched curves. This approach allows for both complex formation shapes as well as control of the subtle details.

After a formation has been determined by the user the agents need to be positioned within it. One solution is to create an oversampled point space with even distribution of points to generate a balanced formation given the shape. The boundary of the formation is resampled evenly and then the inside of the boundary is filled using a flood-fill algorithm with the same sample size. There are, however, problems with this algorithm as a non-optimally selected sample size could lead to undesired results such as leaving some agents outside of the formation (sample size too big) or having too many agents on the boundaries (sample size too small). Gu and Deng [GD13] solve this by proposing a solution to automate the process of selecting the sample size by using the ratio between the total boundary curve length and the number of boundary agents.

Transitioning between different formations requires that for each agent a corresponding position from the start formation to the target formation is calculated [TYK*09, GD11, GD13]. Following on from the above description, after the oversampled template of the target formation is created, the agents themselves need to be moved to their respective positions. For each agent in the start formation, Gu and Deng [GD13] use a 5-dimensional feature vector containing the agent's direction relative to the formation centre, the normalised distance to the formation's centre, and the absolute formation orientation. These features mean that an agent can keep the same relative position after transition and continue to face in the correct direction for the whole formation.

3. The systems

We have implemented two control systems, a classic unit-based control system for a strategy game and our new sketch-based system, designed to deal with heterogeneous groups, as might be found in a strategy game, e.g. infantry and tanks. The classic control system involves using the mouse to draw a boundary around a group of units by specifying two opposing corners of a bounding box. Any units which fall within this box are then selected and can be controlled by the user. Movement orders can then be issued by clicking on a point in the scene, which makes all selected units move towards this point, allowing for quick movement of units but with little room for complex orders.

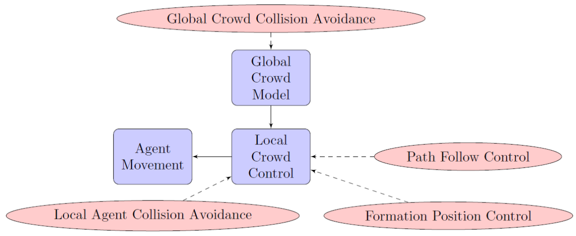


Figure 1: The parts of the sketch-based system

The sketch-based system (the components of which are illustrated in Figure 1) allows for an input sketch consisting of anywhere between two and six strokes, depending on the level of complexity required in the unit movement. Firstly, the user can use a sketched boundary to select a group of units on the screen and give them a formation to move to. This is similar to the Gu and Deng's work [GD13] and can be used with or without a path for the agents to follow (see figure 2). The one additional constraint over that of previous work is that since there are different agent types with varying speeds, all agents contained in the heterogeneous group adjust their speeds such that they all arrive at the target formation at the same time.

The addition of the ability to sketch a subgroup of the main group allows the user to have it behave differently to the main group whilst still remaining part of the same structure. The subgroup can be within the main group or not but the system will treat them as the same structure either way. This allows the user to produce scenarios where the subgroup ends up in their own formation within the main formation. One or two paths can also be supplied to the group for either all agents or the group and subgroup to follow, respectively, as in figures 3 and 4. Animations of these figures can be found at `http://tinyurl.com/qcxhevq`.

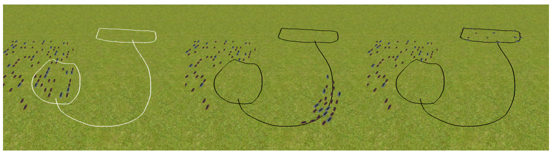


Figure 2: A single group following a path. Three strokes input by the user.

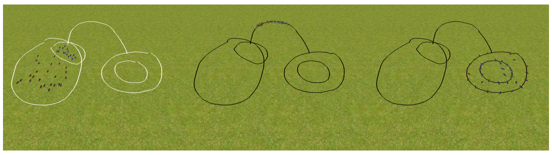


Figure 3: A group and subgroup following a shared path. Five strokes input by the user.

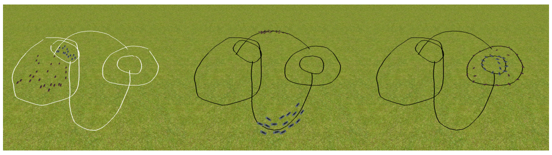


Figure 4: A group and subgroup following separate paths. Six strokes input by the user.

4. Experiment

In order to evaluate the system, tests and a questionnaire were devised and users were asked to evaluate their thoughts on the system using these.

4.1. Scenarios

Four scenarios were used, each of which involves a formation or movement that would be useful in real-time strategy (RTS) games:

1. Given a group of 5 blue VIPs, and a larger group of 50 red bodyguards, surround the VIPs with the bodyguards, as in figure 5. This scenario would be useful if an RTS player wants to protect a group of important soldiers with more expendable units, offering protection from all sides.
2. Given a group of elite soldiers in blue, and a larger group of regular troops in red, move the entire unit along a curved path, with the elite troops leading the way, as in figure 6. This would be useful to a player who wants to clear a path through resistance with a number of strong troops, while having weaker ones picking off any stragglers.
3. Given a group of infantry in blue, and a group of tanks in red, move both groups south, having them arrive at the same time. This would be useful if there is an area directly south of the start that infantry cannot pass but tanks can, so the infantry must take a longer route around the outside, as illustrated in figure 7. This would be useful for attacking an enemy on the other side of a minefield, for example, or poison gas. Either unit arriving before the other would get focused on and whittled down, so both units should arrive simultaneously for maximum firepower.
4. Given a single, large group of blue infantry, execute a pincer manoeuvre to an area to the south, with both sides arriving at the same time as shown in figure 8. This is a very common strategy seen in many RTS games, as it allows players to rain heavy fire down on an enemy force, while having that force's attention divided onto two fronts at the same time. Arriving at the same time is vital - as with the previous test, the danger is that either side arriving first will be weakened before the other side can assist them.

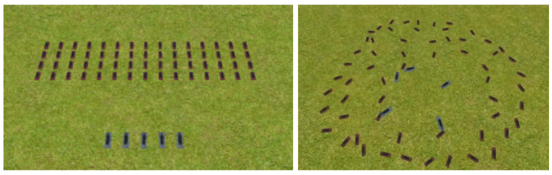


Figure 5: Initial configuration and desired output for test scenario 1

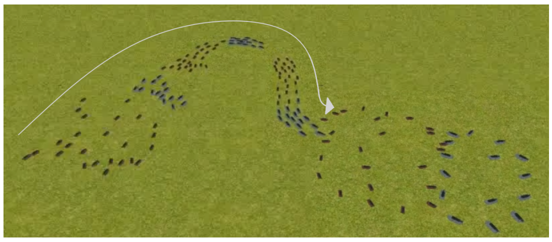


Figure 6: Desired path for test scenario 2. Four frames from the animation sequence are overlaid.

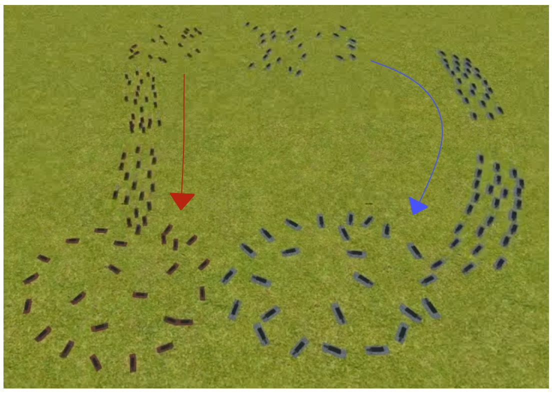


Figure 7: Desired paths for test scenario 3. Four frames from the animation sequence are overlaid.

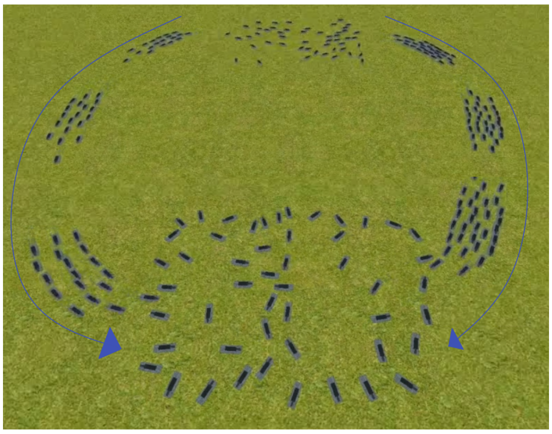


Figure 8: Desired paths for test scenario 4. Five frames from the animation sequence are overlaid.

4.2. The testing procedure

Twelve students volunteered to test the system, with a range of experience of using strategy games. No payment was given. The following procedure was performed individually on each user:

1. First, a consent form is signed. Then, the user is shown the system, taught how to use it and invited to use it until s/he is satisfied s/he understands it well enough.
2. The questionnaire is then given out, and the user is asked to answer the first question, which self-evaluates their proficiency with RTS games on a scale from one to ten.
3. The user is shown the first test scenario, along with a video and verbal explanation. S/he has to attempt to qualitatively recreate the end formation shown. This must be done in both the classic and sketch-based control systems. For different users, the order of this is alternated, as discussed below. The user is allowed as many attempts as s/he wants, until satisfied with the outcome.
4. Following the first test with both systems, the user returns to the questionnaire and fills in the next two questions, which ask how difficult s/he found achieving the desired result, for each system, on a scale of one to ten.
5. Steps 3 and 4 are repeated with the remaining three tests, with the same questionnaire used throughout. In all of these, the qualitative movement path is important for a correct result, in addition to the end formation as with the first test.
6. Once the tests are done, the user is asked to state which system s/he preferred, and explain why, again as part of the same questionnaire used throughout. S/he is then asked if there is anything s/he can think of that would benefit the sketch system, and the test is concluded.

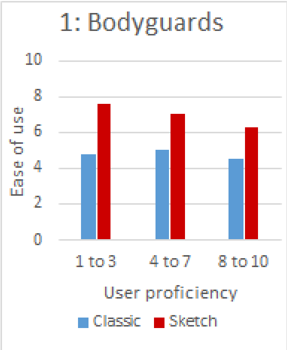
The main point of bias identified in the above procedure is familiarity with each test when performing it with the second control system. If a user correctly performs a movement with one system, s/he will find it easier with the other afterwards. For this reason, the system presented first was alternated on a user-by-user basis. Users with odd-numbered IDs were required to do each test with the classic system first, followed by the sketch-based system, whereas users with even-numbered IDs used the sketch-based system first. An animation comparing the classic and sketch-based approaches is available at `http://tinyurl.com/qcxhevq`.

5. Results and discussion

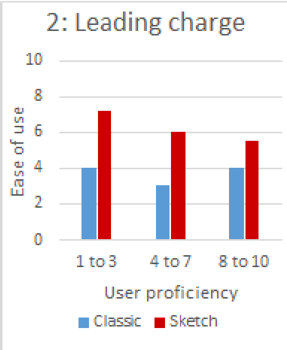
The main difference between using the classic control system and the sketch-based system is the time taken versus the accuracy of the result. The sketch-based system offers a more precise level of control over the units in the scene but pays a price for this precision by taking longer to draw. Conversely, the classic control system makes complex manoeuvres difficult, requiring many mouse strokes in sequence, and resulting in lower accuracy. The advantage of the classic system is that the time taken to produce a single movement is faster (e.g. draw a diagonal line representing a box around a set of units and then make one click to move them all to a new location), meaning that, when speed is concerned, the sketch-based system may be too slow to use (e.g. sketch the entire shape for start and end formations). A hybrid scheme could be beneficial, e.g. use the classic system to select units, when precision is not an issue, and the sketch system to define a specific end position/formation, or vice versa.

Figure 9 shows the graphs charting the ease of use of each system by proficiency. The ease of use presented is simply a score from 1-10, where one is the value assigned to the hardest tasks, ten the easiest. While the question provided had a scale from one to ten, the small number of participants (ten in total) necessitated combining these into fewer groups, so that several users' results could be averaged. It was split into three sections - the first averaged all users with a proficiency of 1-3, the second 4-7 and third 8-10. For a small sample size, the graphs suggest that the sketch-based system is easier to use in comparison to the classic control system. In order to verify whether or not this holds for a larger population, a larger sample and a variety of tests is needed. It also appears as though there is no correlation between proficiency and ease of completing any of the tasks. However, users self-assessed their proficiency and judged ease of use, both of which could be regarded as subjective processes. It would be interesting to consider ways to more objectively assess these aspects, e.g. using timing data or video analysis.

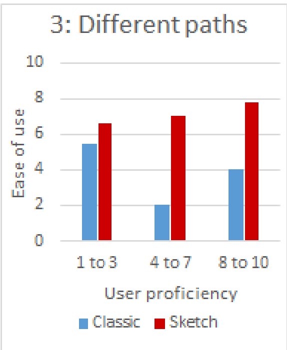
All but one user preferred the sketch-based system over the classic system, and the user who preferred the classic said it was because he was familiar with it. This indicates that, while the sketch-based system appears to be worth learning for newer players (i.e. the benefits appear to outweigh the increased complexity), it is less useful to experienced players who have mastered the use of the classic sys-



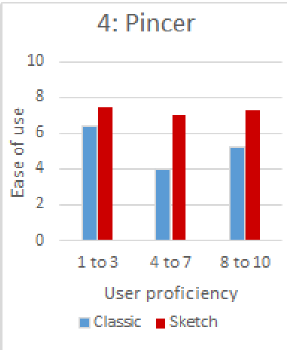
(a) Test scenario 1



(b) Test scenario 2



(c) Test scenario 3



(d) Test scenario 4

Figure 9: Graphs depicting ease of use values for each system in each test

tem. Care must be taken, therefore, if it is to be implemented in a game - experienced RTS players will more likely than not stick to what they know, at least until they become more familiar with the sketching system. In other words, both systems must be able to be run in tandem.

A frequent improvement suggested by users was that the order of shapes drawn should be changed. Every user who mentioned a particular order said they would prefer selecting the first group of agents first, followed by sketching their path, followed by the destination shape. This was not implemented due to the optional nature of the paths, as the second sketch under this system could either be the path or destination - however, this is an extremely important consideration. For the system to be as user-friendly as possible, it must be able to distinguish between paths and shapes, so it can be used in this manner. This would also allow for an arbitrary number of subformations, if extended.

Another improvement suggested was a hybridisation of the systems - several users felt the sketching system would be far too slow to use in the heat of battle. One user suggested, for example, the ability to hold Ctrl when drawing a shape to make the system use sketch controls, but otherwise use the classic system. In addition, integration of more classic features (e.g. double-clicking to select all units of a type) would benefit the system.

6. Conclusions

While the sample size of the user tests is small, it does suggest that a sketch-based control system would allow for a greater level of control within real-time strategy games. Implementing a sketch-based system alongside the classic control system would be necessary due to the speed of being able to perform simpler actions that the classic system brings, which would require a lot of user-interface considerations. Overall, the testing shows that integrating a sketch-based system into strategy games is an area worth spending more time on and would be a useful addition to current control schemes. Nonetheless, future work needs to consider metrics so that more detailed objective analysis could be done on the speed versus accuracy trade-off. In addition, it would be useful to integrate the approach into a real game environment, so as to investigate measures in relation to specific game requirements, e.g. evaluate the accuracy of a pincer movement based on time of arrival and angle of attack of each side of the pincer, and the effect this might have on the game play.

References

[GD11] Gu Q., Deng Z.: Formation sketching: an approach to stylize groups in crowd simulation. *Proceedings of Graphics Interface 2011* (2011), 1–8. 1, 2

[GD13] Gu Q., Deng Z.: Generating freestyle group formations in agent-based crowd simulations. *Computer Graphics and Applications, IEEE 33*, 1 (2013), 20–31. 1, 2

[KLLT08] Kwon T., Lee K., Lee J., Takahashi S.: Group motion editing. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 80:1–80:8. 2

[OSSJ09] Olsen L., Samavati F. F., Sousa M. C., Jorge J. A.: Sketch-based modeling: A survey. *Computers & Graphics 33*, 1 (Feb. 2009), 85–103. 1

[PB11] Patil S., Berg J. V. D.: Directing crowd simulations using navigation fields. *Visualization and Computer Graphics 17*, 2 (2011), 244–254. 2

[PSAB07] Pelechano N., Stocker C., Allbeck J., Badler N.: Controlling Individual Agents in High-Density Crowd Simulation. *ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2007), 99–108. 1

[Rey87] Reynolds C. W.: Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph. 21*, 4 (Aug. 1987), 25–34. 1

[TYK*09] Takahashi S., Yoshida K., Kwon T., Lee K. H., Lee J., Shin S. Y.: Spectral-Based Group Formation Control. *Computer Graphics Forum 28*, 2 (Apr. 2009), 639–648. 2

[UCT04] Ulicny B., Ciechomski P. D. H., Thalmann D.: Crowdbrush : Interactive Authoring of Real-time Crowd Scenes. *Proceedings of the 2004 ACM SIGGRAPHEurographics symposium on Computer animation* (2004), 243–252. 2