Self-Organisation in Games, Games on Self-Organisation

Sebastian von Mammen, University of Augsburg, Germany
sebastian.von.mammen@informatik.uni-augsburg.de

***Abstract*—In this paper, we shed light on the phenomenon of self-organisation in the context of computer games. Self-organisation is an important concept that intersperses a broad range of real-world domains—from economy over ecology, the built infrastructure, distributed technologies to the life sciences. Yet, self-organisation is often hard to recognise and especially hard to control. Computer games can amend this problem by training players to cope with self-organising systems interactively. Here, they can continuously interact with self-organising systems, explore them without jeopardy and gain foundational insights in their dynamics. We present several examples of commercial titles that integrate aspects of self-organisation as well as several academically motivated games that explicitly build on top of it. We further propose a taxonomy on the use of self-organisation in gaming contexts and we conclude with an outlook on potential future works in this direction.**

I. Introduction

Schools of fish, flocks of birds, and social insect colonies—these systems consist of large populations of possibly heterogeneous, mostly simple, reactive agents. The interactions of the individuals may result in system-wide emergent phenomena such as efficient mass transport [1], effective foraging [2], population-wide defence strategies [3] or the construction of complex adaptive nests [4]. The lack of a central control, the decentralised, locally acting individuals, together with the possibility of emergent phenomena render swarms a metaphor for self-organising systems. As such, the swarm metaphor bridges between local interactions and global outcomes, between diversity and homogeneity, between the individual and the population. It highlights the discrepancy and the liaison of different levels of abstraction. Due to the spatial and traceable nature of swarms, this metaphor provides a perspective on scientific models that promises accessibility, flexibility, and scalability of complex systems. Consequently, computational swarms are not only a metaphor for self-organisation but for self-organisation at our fingertips, or interactive self-organisation. But how can one interact with a complex system that, by definition, organises itself? How can one guide its individuals to address important tasks and to fulfil globally defined goals of their users? The answers to these questions do not come easily: A scientific approach would utilise formal methods of information theory, graph theory and nonlinear, complex systems in the attempt to answer these questions. An overview of these methods of guided self-organisation (GOS) can be gained from [5]. However, despite great efforts, a universal, optimal way for mastering self-organising, complex systems has not been found, yet. Also, large numbers of interacting agents with numerous properties and the hard and potentially multi-objective goals, often render it infeasible to formalise a given challenge to begin with. Considering the fact that the users might not even know their goals to begin with or that their goals might change over time, as for instance addressed in [6], a one-stop analytical solution becomes outright impossible. Alternatively one may rely on the established method of trial and error, which, of course, is a tedious, iterative process that jeopardises the good use of resources over long periods of time. Computer games have the potential to mitigate the problem. Here, the user/player can systematically explore the dynamics of self-organising systems without negative side-effects and build up a repertoire of solutions for various real-world problems. He can change the self-organising systems' properties on the fly, consider changing goals and changing environments. In this paper, we present several examples of self-organisation in computer games, discuss potential training effects and argue for its deployment to enrich gaming experiences. The remainder of this paper is structured as follows. In Section II, we exemplarily present some gaming titles that appeared over the years in the market and that deployed self-organisation. In Section III, we present several academic gaming titles that are specifically built on top of the notion of self-organisation. In Section IV, we comprehensively discuss preceding and presented works and we propose a taxonomy for interactive self-organisation. We conclude with a summary and an outlook on potential future work in Section V.

II. Related Work

Considering self-organisation, we can turn towards games in which the player needs to take control over a large number of units and concert their interplay to meet globally defined criteria. A number of computer games has been dedicated to infrastructure networks that are the backbone of the engineered society. First released in 1989, the computer game Sim City has a long-time standing in the market [7]. Here, the player learns to layout infrastructure networks while building flourishing cities. Less playful, with a rather tight focus on economics, Power TAC tries to make informed predictions about viable economic settings in a liberalised, decentralised electricity market [8]. With similar goals, the serious game Infrastratego captured the decision-making strategies from more than three thousand played games against human players that tried to optimise (a) policy making and (b) price negotiations in an open energy market [9]. While these and other titles, directly draw benefit from simulating the networked production and dissemination of electrical power, their common theme is rather universal: Diligently configuring nodes to make them serve the functionality needed in a complex interwoven system, adding missing pieces and slimming down whenever possible—these are the general challenges brought about by the network perspective, whether applied to economics, life sciences or engineering. The interaction possibilities offered by countless realtime strategy (RTS) games, including titles such as Dune, StarCraft, or Battle Zone [10], allow the user to select, navigate, control, and manipulate individual (typically military) units and subsets of the system alike. As in self-organising systems, these units would often follow default behaviours without the interference by the player. Yet, he may take control to support their organisation to reach certain goals. The one-to-many relationship between the player and large numbers of units under his control is even more pronounced in games such as Pikmin and Overlord [11], where the player navigates an avatar from 3rd person perspective, which can, in turn, command his followers. Over the decades, games have introduced a great variety of interfaces for dealing with large numbers of (semi-)autonomous units such as zooming in on individuals, reading the status, reverting back to a global view, selecting units of specific kinds or commandeering hand-picked subsets. For all of these interaction tasks, numerous solutions have been presented. Some of which worked rather well, such as selection in 2D views, or selection simply by adjusting the number of required units. The latter technique is, for instance, used in the science fiction strategy game Galcon, and its open-source counterpart Planet Wars [12], where the player directs swarms of units to defend his planets and to conquer new ones. Other interaction tasks, such as the introspection of individual units in Carrier Command: Gaea Mission, may seem bothersome. However, systematic research on these and other questions regarding the usability und user experience for handling self-organising systems has just recently begun, mainly driven by the increasing availability and accessibility of robotic swarm systems. An overview of the current state of the art in the resultant research discipline of human-swarm interaction can be found in [13]. Although self-organisation is often an important game mechanism, to our knowledge no preceding study has been conducted towards the formulation of an according taxonomy of self-organisation in games. Existing classifications around self-organisation mainly focus on the complexity of the arising phenomena themselves. von Neumann's concept of cellular automata, i.e. state-based cells arranged in grids with neighbourhood-based transition functions, is frequently used to provide insights at an abstract generic level of self-organisation [14]. In 1984, Wolfram introduced four according classes of emerging system states: Spatially homogeneous, simple period, chaotic aperiodic, and complicated localised, propagating ones [15]. In this paper, we review several games around self-organisation and formulate a taxonomy for their classification. We consider this an important first step for explicitly and effectively deploying self-organisation in future games.

III. Games Built on Self-Organisation

In this section, we present a selection of games built around self-organisation that we have developed since 2009. We start our presentation with titles that allow the player to take control of a single element in a self-organising system, thereby influencing the other members of a collective. This perspective can be translated, for instance, to technical systems, where the user controls or configures individual elements, such as robotic units [13] or smart cameras [16], at a time. Next, we focus on the concept of regulating populations in terms of size and in terms of diversity. We then discuss opportunities to establish certain interaction topologies among the agents of a self-organising system. Assuming a self-organising system may deploy different global strategies, choosing the right one at the right time offers another point of access for playful interactions. We conclude this section with two titles where the user determines the success of self-organising systems solely by manipulating their environments.

A. *Playing a Swarm Individual*

In Figure 1, a swarm-based game prototype is shown, in which the player is directly immersed in the process of flocking formation. In particular, the player's task is to influence the overall flock, while only navigating one individual. That being said, all the so-called boid individuals accelerate based on those few neighbours they perceive [17]. In the attempt to concert a flock only navigating one individual, the user quickly gains an intuitive idea of how to control an individual's flight pattern. A simple game setup challenges the player to get a number of swarm agents from a starting area to a goal location [18]. Figure 2 shows a second example of playing

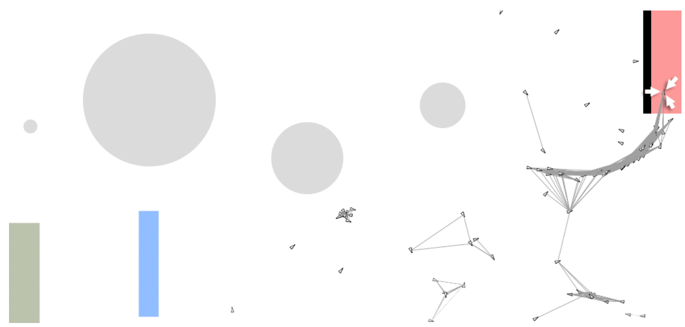

Fig. 1: Screenshot of a flocking game. The player's individual (highlighted by three white arrows) guides its peers from a starting platform (bottom-left) towards the goal (top-right). On their way, various obstacles need to be circumvented.

an individual of a self-organising system. Here, the player is presented with a side-view of a bee hive. He can select a single bee with the mouse, toggle the maintenance default state to defence mode and navigate the bee towards a target. In defence mode, the bee attracts its peers to join the defence patrol by spreading pheromones across the combs. If the player navigates the bee to find a flower outside the hive, it recruits its peers in a similar manner to join its foraging efforts after returning to the hive. These recruits will head towards the last visited flower. In an extended version, we ported the beehive simulator to smart phones and tablets. In addition to a well-structured level-based design with various quests including sustainable population growth and food storage for the winter, we introduced a 3D flight mode (Figure 3). As soon as the player navigates a bee to the outside of the hive, the view switches into this mode. Once the bee has been successfully landed on a flower, it will appear in 2D mode as well and recruited peers can be instructed to fly directly to that flower. The flight performance shown in Figure 3 only contributes to processes of self-organisation as soon as the bee returns to its hive and starts recruiting peers for foraging. But we have also investigated the dynamics of self-organisation that can arise in a collective motion setting in 3D, as seen in Figures 4 and 5. In these examples, similar to the game presented in Figure 1, the

Fig. 2: Screenshot of a 2D beehive simulator. The player selects a single bee, sets its state (maintenance, foraging, defence) and navigates it wherever needed most. Leaving pheromone trails, it recruits other bees to join its cause.

Fig. 4: The user can navigate a specific individual (the orange coloured bird in the center of the view) to guide the flock (the birds in the background), if it is close enough.

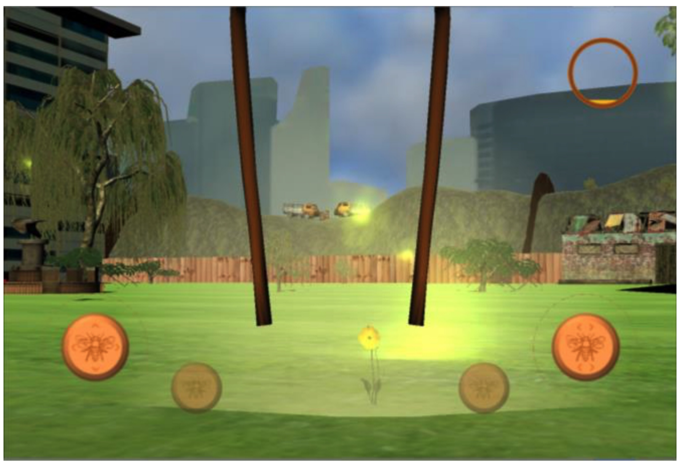

Fig. 3: In the 3D mode of the extended beehive simulator, the user can control a single bee to find resources in a feature rich environment.

Fig. 5: A mobile game where the general flight trajectory is predefined but the player needs to navigate the white bird in order to pass targets and to guide, maintain and utilise a great number of swarmettes (the pink cones).

player takes control of a single individual to guide a collective. It poses a multi-objective challenge which includes keeping the collective together, shaping its formation as well as guiding the collective's direction and speed. Generally speaking, the player needs to continuously exploit the behaviours of the collective's individuals in order to reach certain transient, system-wide states.

B. *Regulation of the Population*

Instead of influencing the emergent behaviours of self-organising systems by controlling a single individual bottom-up, there are numerous titles that allow the player to influence them top-down by regulating the population of involved individuals. In the aquarium game shown in Figure 6, the player is tightly limited to only adding and removing individual components of the system [19]. As, for instances, the (pre-defined) sizes of possible fishes result in different metabolic rates, they impact the ecosystem to varying degrees. Provision of food and the neutralisation of excrements can be indirectly promoted by planting algae and introducing underwater snails. Figure 7 shows a screenshot of an explorative in-browser construction game [20]. Here, the player can visually program individual, swarming builders. As part of the programming, the individuals are instructed to procreate, differentiate or die off. As a result, different from the game shown in Figure 6, the individuals' configuration adds another dimension of population regulation. The parametric configuration of the individuals that is also offered in the game further extends the players' freedom of interaction.

C. *Establishing Topologies*

The composition of populations of self-organising systems is an important mode of influence for facilitating certain global states or processes. However, it does not take the situational relationships of the individuals into account, e.g. their neighbourhood relations, which are a major determinant of the emergent state of a system. In the examples shown above, the interaction topologies emerged from the interplay

Fig. 6: An aquarium game in which the player is tasked to keep an ecological balance by regulating the constituents of the system.

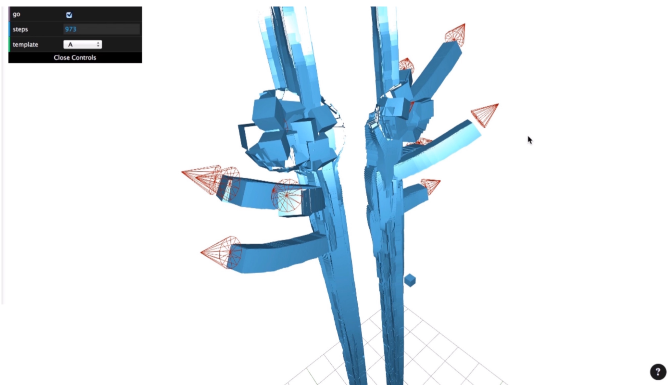

Fig. 7: A construction game in which the user may configure individual elements that themselves regulate the population of interacting components.

of agent parameterisation, starting configurations and player interference. In this section, we show instances of games on self-organisation where the player can interfere with existing and establish new topologies. The game displayed by Figure 8 urges the player to battle the outbreak and spreading of diseases. Individuals travel across the infrastructure that connects buildings serving different needs, e.g. residential complexes, pubs, grocery stores and hospitals. In case one of the inhabitants of this virtual world gets infected the player can implement measures to contain and, ideally, revert the outbreak. To this end, he can modify the predefined infrastructural network by setting up barricades or locks that only allow passage in one or the other direction. Over the course of the game, the infrastructural layouts and the properties of the spreading diseases are altered to increase the challenge but also the individuals' AI adjusts to maximise its well-being. In Figure 9, a screenshot of a serious game on power grids is shown [21]. Depending on the playing mode and the level, the user may need to alter the parameters of existing units, build new consumer or producer nodes of the network and connect them properly. The flow of power among producers, relay stations and consumers is animated by offsetting the striped texture on the connections between nodes and by scaling their geometries. Figure 10 shows a screenshot of a serious game

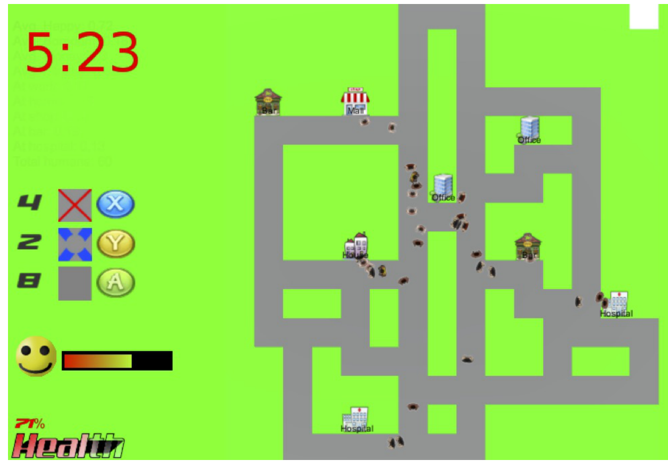

Fig. 8: In this game, the player prevents the outbreak of diseases by establishing barriers and locks along the everyday paths of the player AIs.

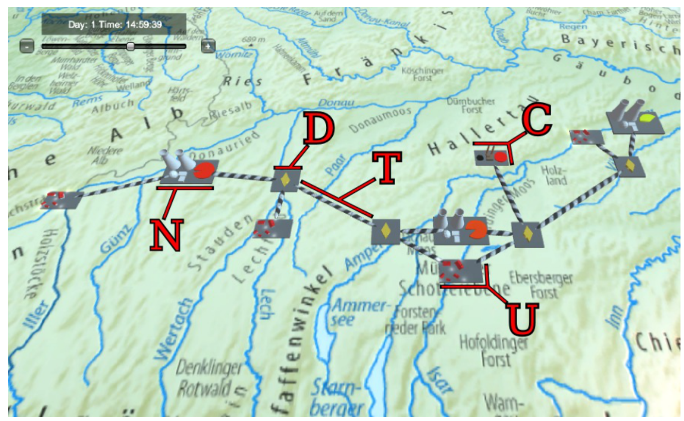

Fig. 9: In this power grid network simulator, the player builds producer and consumer nodes and hooks them up. Challenges are as diverse as repairing existing infrastructures, building them from scratch, or revamp them, e.g. aiming at higher levels of sustainable power production.

around routing networks. Conceptually, it is very similar to the title shown in Figure 9, albeit its different application domain: Data (visualised as postal packages) needs to be wired across the network and find its proper recipient. The player can facilitate his efforts by improving the servers' hardware and by installing various routing protocols. Different from the previously shown title on power networks, the servers' limited numbers of connections are an important gaming mechanism that drives the continuous expansion of the network to handle an increasing data load.

D. Manipulating the Environment

Another important direction of concerting self-organising systems builds on indirect communication (or stigmergy [22]). In biology, self-organisation often relies on local cues. Cues in biology are often chemical signals that spread and evaporate over time or *templates* in the built environment. We focused

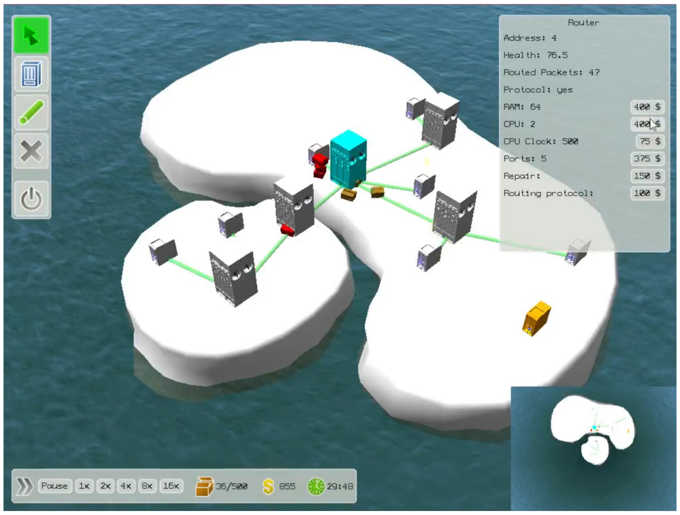

Fig. 10: The player is tasked with setting up, extending and maintaining a communication network in order to deliver packages to their recipients reliably and fast.

on the first in the title presented in Figure 11. The game pursues two different goals: (1) Facilitation of the exploration and education about indirect communication by means of pheromones and (2) design and evaluation of an interface for commandeering large numbers of agents by means of digital pheromones. Starting from the player's base, the agents explore the flat landscape randomly. If they find resources along their way, they pick some up and directly return to the base. On their way back, they release pheromones that would decay after some time, but first they increase the probability of other agents following the trail and contributing to the foraging. In addition, the player can deploy pheromones to quickly guide the agents towards resources, to avoid enemies or to stage an attack. The game seen in Figure 12 multiplies and enriches the

Fig. 11: Foraging and defence agents are guided by the player by spraying digital pheromone trails on the map. The goal is to harvest as many resources as possible within a given time.

available user interactions [23]. Here, the player cannot only lay out and remove pheromone trails but he can also place resources and build physical barriers. We even implemented a virtual reality version of the game featuring a head-mounted display, immersive GUIs and game controller input to increase immersion and presence. The goal of this exploratory game is to keep the colonies of a native and an invasive, superior ant species in check.

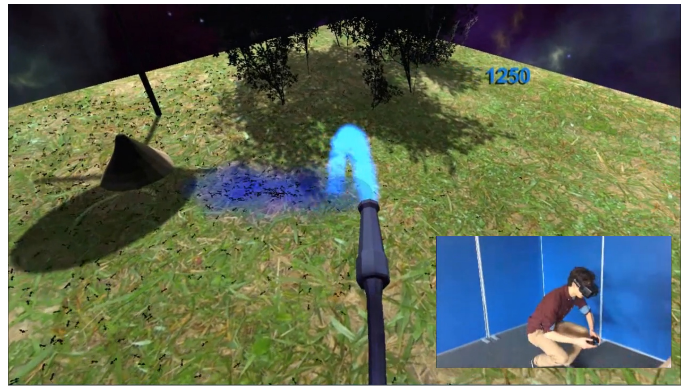

Fig. 12: In this virtual reality game, the player can spray and wash away pheromone trails of two ant species (one native, one invasive) to keep them in balance. He may also build physical barriers with bricks or set out new food sources.

E. Alternating Strategies

In this section, we present four titles that stress the need for self-organising systems to alter strategies when facing changes to their environment. In Figure 13, the player directly changes the parameters of a homogeneous self-organising swarm that flocks in a 2D space [18]. These parameters impact, for instance, the individuals' attraction to each other or the configuration of their fields of view. Concrete settings may result in loose or tight clusters of swarms as well as more directed behaviours, including line formations, spreading into different directions, etc. The player then utilises these respective behaviours in order to maintain as many individuals for as long as possible, avoiding traps (in red), and picking up as many resources as possible (in yellow). As fine-tuning the

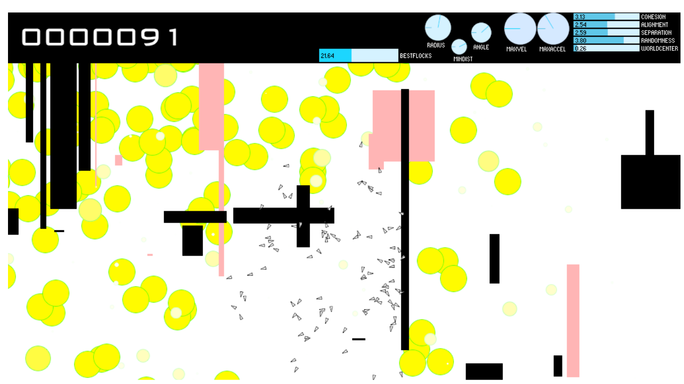

Fig. 13: While the environment is constantly moving in one or the other direction, the swarm needs to be adjusted to harvest as many yellow particles as possible, avoid contact with red areas and overcome physical obstacles (in black).

parameters of a self-organising system is hard and requires lots of experience, especially in fast-paced, arcade-style games as the one presented in Figure 13, we introduced a means of support. In Figure 14, one sees a screenshot of the supporting sub-game. In order to retrieve desirable flocking behaviours, the player may place tiles on the ground and an evolutionary optimisation engine runs in the background to automatically find parameter settings that make the swarm cover those tiles for as long as possible. Next, the user can select his previously evolved configurations in a flocking game such as the one shown in Figure 13. To further support the player

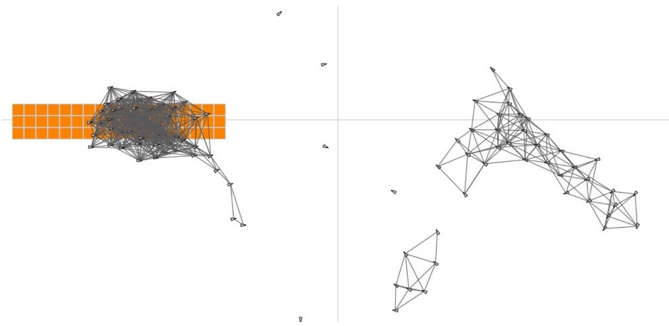

Fig. 14: The player can evolve certain flocking strategies that fulfil globally defined goals, such as maximising the swarm's exposure to the manually placed orange tiles. These flocking strategies can be utilised for the game in Figure 13.

in mastering self-organising systems, different configurations of self-organising systems can be manually setup or learned offline and made accessible to the player during the game, as seen in the following example. Figure 15 depicts the screenshot of an early prototype of a game in which the user navigates a flock of firefighting drones. It is his task to find sources of fire and extinguish the flames in due time. In order to achieve this goal, the player needs to take control of the swarm by steering one *leader* individual (as in Section III-A), by assigning flight targets in direct sight, by dividing the flock into subgroups to cover several spots at the same time—for extinguishing fire or for refilling the drones' water tanks. While taking direct control of one individual, the player can instruct the swarm by performing certain gestures to alter its flight formation between v-formation, line-formation and dense/loose cluster (the game is played using either multi-touch or hand gestures). In this way, the player ensures that the locations of the swarm individuals are perfectly attuned to the task (releasing/picking up water) during those periods of time the flock is in a target zone. As pointed out in Section II, besides flocking games, realtime strategy games are especially interesting in the context of self-organisation. Typically, in an RTS, the player aims at exceeding an opponent in terms of resource gathering, number of deployable units and stationary buildings. StarCraft Broodwars is an established commercial RTS-title which provides an elaborate programming API for implementing artificial players, or artificial intelligences (AIs). Furthering the autonomy of self-organisation still, we implemented four AIs with basic default characteristics and with the ability to learn [24]. We trained them in hundreds of matches (three hundred matches for each AI) and validated their training successes in another few hundred rounds (150 for each AI). In the end, the player

Fig. 15: In order to refill water tanks and extinguish fires, the player needs to commandeer a swarm of firefighting drones: Navigate them, split them into subgroups, assign them to different targets, bring them back together and change their flight formations.

could use any of the four AIs, let them play autonomously or enrich their behaviours in accordance with his own ideas. A screenshot of an evolved aggressive AI is shown in Figure 16.

Fig. 16: We designed and evaluated several learning AIs for the RTS game StarCraft Broodwars.

IV. A Taxonomy of Self-Organisation in Games

In the context of games, we are particularly interested in the manner in which interactions between human users and self-organising systems are realised as well as their impact. Examining the examples of Section III, we observe commonalities and differences with respect to the following characteristic dimensions: **level of control**, i.e. comprehensive sets of self-organising system components, subsets and individuals, **target of control**, i.e. manipulation of the environment, population management, definition of topologies, instruction of strategies or activities, changes in behaviours and parameters, **granularity of control**, i.e. global goals, subgoals, strategies, activities, parameters, **view**, i.e. top-down 2D visualisation, isometric views, 3D immersive, first or third person views, fixed cameras, predefined trajectories with limited degrees of freedom, unhindered camera navigation, **interface**, i.e. parameter GUIs, visual programming UIs, interactive maps, multi-touch interfaces, gamepad or mouse, **time of interference**, i.e. discrete (before or after a single step or playing phase) or continuous (throughout the game or during specific phases). Clearly, the choice of the respective features greatly depends on the contents and mechanics of the game and it is not limited to games on or driven by self-organisation. Yet, the level, target, and granularity of control can be detached from the concrete application context and therefore deserve closer inspection.

A. Level of Control

We would like to start our discussion of the dimension of level of control by referencing another line of work that attempts to capture the basic features of self-organising systems. The level of control directly opposes the degree of autonomy of a self-organising system. Measures of autonomy have, for instance, been introduced in the context of organic computing [25], a research initiative with the goal of harnessing biological principles for technical systems. To measure the degree of autonomy of a system, one first describes its *variability* V, i.e. the binary logarithm of the magnitude of the set of different configuration states of the system, which is equivalent to the number of bits in a vector that fully describes the system's configuration. Second, we shed light on the number of configuration bits that the user proactively manipulates and infer the so-called *external variability* V_e that is controlled by the user and the *internal variability* V_i that is controlled autonomously by the system. The degree of autonomy of a system is then defined as $\alpha = \frac{V_i - V_e}{V_i}$. For a concrete application of this concept, let us consider the aquarium game presented in Section III-B. Here, for each organism the player adds to the ecosystem, he also introduces roundabout five variables that interact with the remainder of the system. Hence, the system's autonomy is constant at $\alpha = \frac{5-1}{5} = 80$, resulting in a level of control of 20%. The calculations are more involved for the pheromone-based RTS game displayed in Figure 11. The degrees of freedom of each agent comprises two basic states (attack or foraging mode), whereas in foraging mode, the agent may be searching or bringing in resources. While searching, an agent may take a turn into any direction in two-dimensional space. Both, state and direction can be controlled by the user; a state can be switched directly, the direction can be imposed by laying out pheromone trails. Accordingly, the autonomy of the system may theoretically be evened out by the user, i.e. $\alpha = 0$. Yet, this result is not fully satisfying, as it does not take the target of interaction into account. Consider, for example, flocking games with one leader agent, as illustrated in Figures 1, 4 and 5—the application of pheromone trails implies a more immediate impact on the respective self-organising system than taking control over a single agent. For this reason, we will investigate the target of control next.

B. Target of Control

The potential control space, i.e. the entirety of data the user can manipulate, comprises the self-organising system itself as well as its environment. As pointed out above, the description of this space is a first step, but it requires an additional layer of information that captures the potential impact of any targeted, atomic manipulations. Due to feedback loops, chains of interdependencies and potential phase transitions in self-organising systems, this challenge cannot be addressed in general terms. However, one reasonable way to account for the potential impact of individual manipulations lies in looking at the numbers of directly affected components of the system over a predefined timespan. Returning to the example from Figure 11, let us estimate the impact of spraying a straight line on the game board. Due to the default dissipation rate, the maximal length of the trail is half the dimension of the board. Early in the game, when the agents are leaving the base, at most 50% of them can be attracted to follow the trail. At a later point in time, when the agents might almost be evenly spread, only about 10% of all agents would perceive the line, resulting in a refined, target-specific level of control ranging between 10 and 50%. Applying the same train of thought to the flocking games with one leader agent (Figures 1, 4, 5), the connectivity of the leader agent to its peers ranges from 0 to 100% depending on the peers' fields of perception and their relative locations, which are determined by their autonomous flocking behaviour and the leader's position. Clearly, the more control the user can exercises, the better he can achieve his goals. Therefore, the difficulty for the player lies in exhausting, and at the same time directing, the target-specific level of control. More specifically, the player needs to master a control target with respect to the goals of a game, also taking into account detrimental or boosting elements (such as the obstacles in Figure 1). Choosing and improving the most effective available control target can be another important skill for a game.

C. Granularity of Control

Granularity of control is the last characteristic dimension that we would like to briefly discuss. Again, we propose to pursue a quantitative approach to gain a descriptive handle on self-organisation in games that we can work with. To this end, we first take a look at the biggest (predefined), accessible entity that occurs in a game, for instance whole military fleets as in Figure 16 or flocks as in Figure 15. Then we investigate top-down, whether there are any underlying components that we can access as well. In the given examples, these would be smaller commandos or subsets of swarming drones. Next, we would look at the individuals, then their behaviours, and finally the parameterisation of the behaviours. Step by step, we build a wood of trees whose roots represent the interaction entities of the greatest granularity of control and the leaves those of the finest granularity. Depending on the design and contents of the game, the player might only be allowed to access certain levels of these granularity trees and only under certain circumstances. For the purpose of comparison, properties of the resulting woods can be analysed. The fraction of the number of leaves and the number of roots, $g_d = \frac{l}{r}$, captures the quantitative relationship between the most minute control instructions and the units controlled at the highest level. Accordingly, this fraction expresses the averaged degree of control for each high-level unit. Applying this measure to the matches ran to train self-learning AIs in StarCraft Broodwars (Figure 16), we consider the player's fleet the highest level entity. It consists of 83 individual military units (64 Zerglings, 12 Hydralisks, 2 Ultralisks, 4 Scourges and 1 Queen) that all can, in theory, be individually instructed to move into arbitrary directions on a 2D map (two degrees of freedom) and which have one or two modes of attack. Overall, there are roughly $l = 83 \times (2 + 1, 3) = 273, 9$ micro-instructions available, which in the given case equals g_d. In case not all high-level units can be selected and instructed in unison, as in the pheromone-based RTS (Figure 11), g_d drops fast, here to about $g_d = \frac{75\times(2+1)}{75} = 3$. In the firefighting drones game (Figure 15) a deeper tree emerges, as we cannot only commandeer the whole flock, but because we can recursively divide it in two subswarms, each of which can be sent to specific landmarks on the map (on average about five landmarks at any point in time). Hence, the recursive division scheme that applies here coincides with the game's tree of control granularity. In case of elaborate woods or trees, investigating the trees' depths alongside the branching factors can be understood as indicators of the system's complexity.

V. Conclusion and Future Work

We presented fifteen different games that revolve around self-organisation or in which self-organisation plays an important role. While presenting them, we emphasised commonalities such as the principle of taking control of a single leader individual and thereby impacting a self-organising system. We turned to titles next where the player can influence the composition of self-organising systems at the population level. In addition to local configuration and system-wide population control, we identified the manipulation of the topologies among interacting components as well as their model environments to be important targets of control. The highest level of control of a self-organising system is to merely alternate its strategies over time. Based on examples that entertained these perspectives to varying degrees, we proposed a taxonomy of self-organisation in games. In particular, we introduced the notions of level of control, target of control, and granularity of control. We exemplarily applied the respective measures to some of the presented games and discussed their potential meaning for honing game mechanics. We are fully aware that despite the modest variation of presented examples, our observations cannot make any claims towards completeness. Whole genres, such as puzzlers, in which self-organisation may provide for seminal game mechanics, have not been considered. Although we believe that the measures and perspectives we introduced are helpful for capturing the relationships between players and self-organisation, they can only represent a first step towards a longer term attempt to classify, quantify and exploit mechanisms of self-organisation in games. To this end, we want to propose a rigorous investigation of the suggested measures in terms of player experience and opportunities for deployment.

References

[1] S. Marras, S. S. Killen, J. Lindström, D. J. McKenzie, J. F. Steffensen, and P. Domenici, "Fish swimming in schools save energy regardless of their spatial position," *Behavioral Ecology and Sociobiology*, vol. 69, no. 2, pp. 219–226, 2015.

[2] T. J. Czaczkes, C. Grüter, and F. L. Ratnieks, "Trail pheromones: An integrative view of their role in social insect colony organization," *Annual review of entomology*, vol. 60, pp. 581–599, 2015.

[3] T. Parmentier, W. Dekoninck, and T. Wenseleers, "Context-dependent specialization in colony defence in the red wood ant formica rufa," *Animal Behaviour*, vol. 103, pp. 161–167, 2015.

[4] D. Fouquet, A. Costa-Leonardo, R. Fournier, S. Blanco, and C. Jost, "Coordination of construction behavior in the termite procornitermes araujoi: structure is a stronger stimulus than volatile marking," *Insectes sociaux*, vol. 61, no. 3, pp. 253–264, 2014.

[5] M. Prokopenko, *Guided self-organization: Inception*, vol. eBook. Berlin / Heidelberg: Springer, 2014.

[6] B. G. Bezirtzis, M. Lewis, and C. Christeson, "Interactive evolution for industrial design," in *Conference on creativity & cognition*, (New York, NY, USA), pp. 183–192, ACM, 2007.

[7] J. P. Gee and M. H. Levine, "Welcome to our virtual worlds," *Literacy*, vol. 2, pp. 48–52, 2009.

[8] W. Ketter, J. Collins, and P. Reddy, "Power tac: A competitive economic simulation of the smart grid," *Energy Economics*, vol. 39, pp. 262–270, 2013.

[9] M. Kuit, I. S. Mayer, and M. De Jong, "The infrastratego game: An evaluation of strategic behavior and regulatory regimes in a liberalizing electricity market," *Simulation & Gaming*, vol. 36, no. 1, pp. 58–74, 2005.

[10] S. Dor, "A history of real-time strategy gameplay from decryption to prediction: Introducing the actional statement," in *History of Games International Conference Proceedings* (C. Therrien, H. Lowood, and M. Picard, eds.), (Montreal, Canada), Kinephanos, 2014.

[11] J. Linderoth, "Why gamers dont learn more: An ecological approach to games as learning environments," in *Proceedings of the 2010 International DiGRA Nordic Conference: Experiencing Games: Games, Play, and Players*, (Stockholm), Digra, 2010.

[12] A. Fernández-Ares, P. García-Sánchez, A. Mora, P. Castillo, and J. Merelo, "There can be only one: Evolving rts bots via joust selection," in *Applications of Evolutionary Computation*, pp. 541–557, Springer, 2016.

[13] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *Human-Machine Systems, IEEE Transactions on*, vol. 46, pp. 9–26, Feb 2016.

[14] J. von Neumann and A. W. Burks, *Theory of self-reproducing automata*. Urbana and London: University of Illinois Press, 1966.

[15] S. Wolfram, "Cellular automata as models of complexity," *Nature*, vol. 311, pp. 419–424, October 1984.

[16] B. Rinner, T. Winkler, W. Schriebl, M. Quaritsch, and W. Wolf, "The evolution from single to pervasive smart cameras," in *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pp. 1–10, Sept. 2008.

[17] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of SIGGRAPH '87*, (New York, NY, USA), pp. 25–34, ACM, 1987.

[18] S. von Mammen and C. Jacob, "Swarming for games: Immersion in complex systems," in *Applications of Evolutionary Computing, Proceedings Part II*, Lecture Notes in Computer Science, (Tübingen, Germany), pp. pp. 293–302, Springer Verlag, 2009.

[19] J. Schikarski, O. Meisch, S. Edenhofer, and S. von Mammen, "The digital aquarist: An interactive ecology simulator," in *Proceedings of the European Conference on Artificial Life 2015 (ECAL)*, pp. 389–396, 2015.

[20] S. von Mammen and S. Edenhofer, "Swarm grammars gd: Interactive exploration of swarm dynamics and structural development," in *ALIFE 14: The International Conference on the Synthesis and Simulation of Living Systems* (H. L. et al., ed.), pp. 312–320, ACM, MIT press, 2014.

[21] S. von Mammen, F. Hertwig, P. Lehner, and F. Obermayer, "Powersurge: A serious game on power transmission networks," in *EvoApplications Proceedings of the 18th European Conference on Evolutionary Computation (EvoStar)*, vol. 9028, (Copenhagen, Denmark), pp. 406–417, Springer, April 2015.

[22] P.-P. Grassé, "La reconstruction du nid et les coordinations interindividuelles chezbellicositermes natalensis etcubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs," *Insectes Sociaux*, vol. 6, no. 1, pp. 41–80, 1959.

[23] A. Knote, S. Edenhofer, and S. von Mammen, "Neozoa: An immersive, interactive sandbox for the study of competing ant species," in *Proceedings of the IEEE Virtual Reality*, (Greenville, SC, USA), IEEE Press, March 2016.

[24] S. Rudolph, S. von Mammen, J. Jungbluth, and J. Hähner, "Design and evaluation of an extended learning classifier-based starcraft micro ai," in *Applications of Evolutionary Computation*, pp. 669–681, Springer, 2016.

[25] C. Müller-Schloer, H. Schmeck, and T. Ungerer, eds., *Organic Computing - A Paradigm Shift for Complex Systems*. Autonomic Systems, Birkhäuser Verlag, 2011.