Finally, it is worth mentioning that the organizers acknowledged at the seminar that that some of these issues would be addressed. In fact, it has been recently announced that at the CIG 2015 competition, the organizers will be introducing two new tracks: Learning Track and Procedural Content Generation Track. Although their information has not yet been revealed at the time of writing this report, the former track apparently will allow a training phase.

3.5 Creativity Facet Orchestration: the Whys and the Hows

Antonios Liapis (IT University of Copenhagen, DK)

Joint work of Bidarra, Rafael; Liapis, Antonios; Nelson, Mark; Preuss, Mike; Yannakakis, Georgios

Creativity facet orchestration aims to combine generation across the multiple creative domains that comprise game design. Literature identifies six core facets existent in games: level design, game design, audio, visuals, narrative and gameplay (involving NPCs or not) [1]. The first two facets are *necessary* for a game to be instantiated whereas the remaining four are *optional*. While there have been a few attempts to integrate more than one facet during the generative process (e.g. Game-o-Matic [2], Angelina [3]) these have been limited to mere hierarchical (linear) procedures. It is only very recently that research on computational game creativity has focused on ways in which more that two facets are interweaved during their generation (such as the work of Hoover et al. [4] orchestrating visuals, audio and gameplay elements).

We argue that to generate novel and valuable games that lie on unexplored regions of the game design space, an *orchestration* approach is needed to automate game generation in a truly integrated manner. We view this approach as an *iterative refining process* particularly suited for the **generation of playable prototypes** for designers to consider and get inspired from. Orchestration requires that the human designer specifies the desired semantics for a query within a (large but manageable) space of possible games. For example, a designer might request a horror game (directly affecting the mechanics of the game), with open-space-style levels (affecting level design), with a warm ambiance (affecting visuals), relaxing music (affecting audio), linear narrative and aggressive NPCs. The generative system blends available concepts (using e.g. ConceptNet [5]) with the aim to deviate from the query in the game design space (e.g. this process could involve searching for novel games from semantically annotated databases of existing games). At this point each facet operates independently, constrained by the semantic information the designer has provided. The facet-specific generator operates using the semantics of all generators, thus providing a high-level context to guide its generative processes. The result is a set of prototypical games with unconventional combinations of facets' outputs, all matching the same underlying semantics. Those games are then presented for designer consideration, along with information about their distance (dissimilarity across several dimensions) to typical games, in order to choose which are to be refined. Refinement tailors parameters of the game space and polishes facets such as visuals and audio.

References

1 Antonios Liapis, Georgios N. Yannakakis, Julian Togelius. *Computational Game Creativity*. In Proceedings of the Fifth International Conference on Computational Creativity, 2014.

2 Michael Treanor, Bryan Blackford, Michael Mateas, Ian Bogost. *Game-o-matic: Generating videogames that represent ideas*. In Procedural Content Generation Workshop at the Foundations of Digital Games Conference, 2012.
3 Michael Cook, Simon Colton, Azalea Raad, Jeremy Gow. *Mechanic miner: Reflection-driven game mechanic discovery and level design*. In Proceedings of Applications of Evolutionary Computation, volume 7835, LNCS, 284–293, 2013.
4 Amy K. Hoover, William Cachia, Antonios Liapis, Georgios N. Yannakakis. *AudioInSpace: A Proof-of-Concept Exploring the Creative Fusion of Generative Audio, Visuals and Gameplay*. In Proceedings of Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMusArt), 2015.
5 Hugo Liu, Push Singh. *ConceptNet – A Practical Commonsense Reasoning Tool-Kit*. BT Technology Journal 22, 4, 211–226, 2004.

3.6 Believable Characters

Brian Magerko (Georgia Tech – Atlanta, Georgia, US)

Joint work of Champandard, Alex; Eladhari, Mirjam; Horswill, Ian; Magerko, Brian; McCoy, Josh; Spronck, Pieter; Treanor, Mike; Young, Michael; Zhu, Jichen

The goal of believable characters in computer games (and other media) is to achieve dramatic believability. In other words, "audiences do not pay for reality." This task group considered two views of the problem of creating dramatically believable characters, as has been considered in film, theater, and academic games research in decades prior: the requirements for "believable behavior" and the outstanding issues in computational architectures for creating said behavior.

As a driving example, we considered a specific dramatic scene of "asking someone out at a bar" and what perceivable individual and social cues were related to the dramatic content of the scene. These externally visible cues included: gaze, feeling, posture, proxemics, dialogue, and action selection & execution. The internal drives that could therefore enable these cues in the scene are functions like: anxiety, a theory of mind, a concept of social contract & cultural norms, character histories, and idiomatics.

The concept of procedural idiomatics seemed to be an avenue of particular interest. Within a system that explores the "asking out on a date" scene (or, conversely, the "Little Red Riding Hood" story that has been a prevalent example story in interactive narrative systems), one could consider automatically reasoning about narrative discourse, genre conventions and tropes, character archetypes, and character status as a means of exploring the scene in dramatically believable fashion with highly different dramatic results.

In terms of architectures, there are a number of important outstanding issues. One important issue is the mid-level consistency problem: arbitrating between high-level plans and low-level action selection so as to avoid trashing behavior. The authorial consistency problem is a related issue. While many character architectures can produce consistent and intelligent behavior within a given character, and high level narrative sequencers such as beat systems and drama managers can coordinate long-term narrative consistency, coordinating between the two systems in a way that produces sensible behavior is difficult. Finally, there is a problem with achieving thematic consistency: preventing low-level systems from choosing locally rational dramatically inappropriate action, such as a Sims character washing the dishes immediately after the death of a loved one.