Term Project: Team and Topic Formation

Cole Granof, Joseph Petitti, Matthew Puentes

November 17, 2019

- Team members: Cole Granof, Joseph Petitti, and Matthew Puentes
- Topic for our project: "procedurally generated levels and enemies for a top-down roguelite twinstick shooter."
- We hope to create an engaging and unique game, similar to other procedurally generated games like *The Binding of Isaac* and *Nuclear Throne*, using the principles of parameterized design we explored in this class. Levels will be generated to form natural-looking cave-like structures, enemies will be randomized by assigning them points in various attributes and giving them random facial characteristics, and power-ups will also be randomly generated. In this way the game will be different each time it is played.
- This project is interesting to us because not many existing roguelites utilize both procedurally generated levels and procedurally generated enemies. The combination of these two techniques can allow for a wide variety of different encounters and unique gameplay scenarios. We believe that cellular automata can be used to create really cool levels, and this technique is underused in existing games.
- We just want to make a fun game that other people can enjoy. Hopefully the procedural nature of the levels and enemies will give the game longevity. There isn't a high potential for generating offensive content based on the constraints we use on the engine.
- Three external sources we have read:
  - Liu et al. (2017). The researchers of this paper used an AI agent to iteratively adjust the difficulty of a game based on the AI agent's performance. The researchers adjusted various parameters that dictate different aspects of difficulty in order to create a balanced experience. Essentially, this paper provides an outline for automating play testing, as well as a breakdown of gameplay elements that increase difficulty when individually scaled.

    Performing this kind of AI-powered play testing is likely outside the scope of our project. However, this paper separates the difficulty of their "pace-battle game" into various parameters, such as speed of the player, the speed of enemies and the amount of bullets fired. Breaking down difficulty into these discrete components will help us generate enemies with specific levels of difficulty. Furthermore, with enough well-chosen parameters, we can create enemies that challenge the player in different ways. This will hopefully encourage players to change up their strategies.

  - Pell (1992). This paper details a way to fully specify and generate chess-like games such as Western Chess, Xiangqi, Shogi and checkers. Capture rules and piece movement can be specified to encompass any existing chess like games, and, of course, define completely new games as well.

    This inspired the idea of randomly generating a rogue-like where the behavior of the enemies is generated. This forces the player to learn a new set of completely novel enemies each run. The specification of chess-like games will help guide us in designing a flexible specification to use for generating enemy aesthetics and behavior.

  - Emilio et al. (2010). This paper describes an algorithm for playing *Ms. Pac-Man* based on the behavior of ant colonies. The paper lists several useful behaviors of ants in ant colonies, and how their heuristics can be applied to solving game problems such as guiding Ms. Pac-Man through a maze filled with hostile ghosts. The researchers used a genetic algorithm to optimize the parameters of these ant behaviors, and were able to create an effective game-playing AI.

    The researchers' technique for designing agent-based AI inspired us to try using a similar method for our enemy AI.

- Six external sources we want to read next:
  - Jain et al. (2016)
  - Van Der Linden et al. (2013)
  - Shaker et al. (2016)
  - Smith et al. (2009)
  - Dewsbury et al. (2016)
  - Bizopoulos et al. (2014)

- Three questions or concerns we have related to the project
  - Starting from scratch without using any existing engine makes a lot more work for us, such as implementing a graphics manager, physics system and collision system.
  - We're concerned with making sure the level designs and generated enemies are both interesting and fun to experience, without sacrificing one for the other.
  - How do we make sure the game is playable without being too easy or too hard?

References

Paschalis A Bizopoulos, Antonis I Sakellarios, Dimitrios D Koutsouris, Dimitra Iliopoulou, Lampros K Michalis, and Dimitrios I Fotiadis. 2014. Randomly generated realistic vessel geometry using spline interpolation and 2D Perlin noise. In *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, 157–160.

Neall Dewsbury, Aimie Nunn, Matthew Syrett, James Tatum, and Tommy Thompson. 2016. Scalable level generation for 2d platforming games. In *Proceedings of the FDG Workshop on Procedural Content Generation*.

Martin Emilio, Martinez Moises, Recio Gustavo, and Saez Yago. 2010. PacmAnt: Optimization based on ant colonies applied to developing an agent for Ms. Pac-Man. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*. IEEE, 458–464.

Rishabh Jain, Aaron Isaksen, Christoffer Holmgård, and Julian Togelius. 2016. Autoencoders for level generation, repair, and recognition. In *Proceedings of the ICCC Workshop on Computational Creativity and Games*.

Jialin Liu, Julian Togelius, Diego Pérez-Liébana, and Simon M Lucas. 2017. Evolving game skill-depth using general video game AI agents. In *2017 IEEE Congress on Evolutionary Computation CEC*. IEEE, 2299–2307.

Barney Pell. 1992. Metagame in symmetric chess-like games. (1992).

Noor Shaker, Antonios Liapis, Julian Togelius, Ricardo Lopes, and Rafael Bidarra. 2016. Constructive generation methods for dungeons and levels. In *Procedural Content Generation in Games*. Springer, 31–55.

Gillian Smith, Mike Treanor, Jim Whitehead, and Michael Mateas. 2009. Rhythm-based level generation for 2D platformers. In *Proceedings of the 4th International Conference on Foundations of Digital Games*. ACM, 175–182.

Roland Van Der Linden, Ricardo Lopes, and Rafael Bidarra. 2013. Procedural generation of dungeons. *IEEE Transactions on Computational Intelligence and AI in Games* 6, 1 (2013), 78–89.