**References**

1 Jane Austen. *The complete novels of Jane Austen*, volume 4. Chartwell Books, 2016.
2 Claire Dormann and Robert Biddle. Making players laugh: The value of humour in computer games. In *Proceedings of the 2007 conference on Future Play*, pages 249–250, 2007.
3 Michael Mateas and Andrew Stern. Procedural authorship: A case-study of the interactive drama Façade. In *Digital Arts and Culture*, 2005.
4 Josh McCoy, Mike Treanor, Ben Samuel, Aaron A. Reed, Michael Mateas, and Noah Wardrip-Fruin. Prom Week: Designing past the game/story dilemma. In *Proceedings of the Foundations of Digital Games Conference*, 2013.
5 Ben Samuel, James Ryan, Adam Summerville, Michael Matea, and Noah Wardrip-Fruin. Bad news: An experiment in computationally assisted performance. In *Proceedings of the International Conference on Interactive Digital Storytelling*, 2016.
6 Carmen Smith and Laura Mooneyham White. Discerning voice through austen said: Free indirect discourse, coding, and interpretive (un) certainty. *Persuasions: The Jane Austen Journal On-Line*, 37(1), 2016.
7 Ryan Taljonick. Shadow of Mordor's Nemesis system is amazing–here's how it works. `https://www.gamesradar.com/shadow-mordor-nemesis-system-amazing-how-works/`, 2014. accessed 3 July 2022.


## 3.3 Pokegen

*Alexander Dockhorn (Leibniz Universität Hannover, DE), Manuel Eberhardinger (Hochschule der Medien – Stuttgart, DE), Daniele Loiacono (Polytechnic University of Milan, IT), Diego Perez Liebana (Queen Mary University of London, GB), and Remco Veltkamp (Utrecht University, NL)*



The generation of art assets plays a huge part in game development, costing both time and money. We explored how the process of generating game art can be supported using recent advances in generative art.

Machine learning models such as Dall-E 2 [1] and Imagen [2] have demonstrated powerful art generation capabilities. Starting from text prompts, they are able to combine concepts, attributes, and styles to generate artworks of generally high quality. Nevertheless, their usage is restricted and similar projects such as ruDall-E [4] and Mini-Dalle-E [3] do not produce results at the same level of detail, i.e. generating blurry images, struggling to include concepts that are not well represented in the training data, and sometimes creating stock image overlays (c.f. Figure 3). This often results in prompt engineering, a process in which the user adapts the text prompt to guide the black box model to produce the desired outcome [8]. Due to the black-box nature of deep learning models, this process can yield unstable results and is therefore hard to control, making it inefficient and unreliable for creating game assets.

Therefore, we have envisaged several pipelines that may support designers and artists during game development. Starting from possible inputs such as a designer's textual descriptions of the required game asset, some image ideas, sketches, or even expected game mechanics, we have multiple ways to approach the problem of game asset generation. Simple text and image search models may guide the artistic exploration process and spawn new ideas.

Nevertheless, those can only return results that already exist. Given textual descriptions, we can apply text-to-image models for generating new assets. Alternatively, we may use style-transfer models to enforce characteristics described in the text to an existing image (e.g. CycleGAN [9]). The latter may also be used to adjust image characteristics such as drawing style or the choice of colors (e.g. Neural Style Transfer [10]). Especially interesting is the combination of such models, which may allow to tune each component of the processing chain separately.

In our working group, we have worked on implementing a toolchain to generate Pokemon-like creatures. A Pokemon often represents an animal or object and in terms of visual style, does only consist of a few colors as well as simple shapes and textures. Aiming to use existing models without retraining, we started our process by generating images of dragons using ruDall-E [4]. Generated images varied hugely in quality. Since due to our style constraints our final image does not need to include a lot of details we have chosen a rather blurry image of a dragon with a simple background. Having selected a generated image of a dragon we applied style-transfer as a combination of VQGAN [7] and CLIP [6]. Without retraining any of these components to our specific domain (due to time constraints), we were unable to achieve results of high visual quality (see Figure 4). Nevertheless, this process show-cases how mock-ups and ideas may be generated to guide the development process.

While having struggled to develop a multi-stage model for generating Pokemon-like creatures, it has helped us to better understand the main challenges for generating game assets in general. The following challenges have been identified by us and may guide further research in this domain:

- **Copyright:** Generating art from machine learning models poses the question of who owns the copyright of the final result. This may be a complicated question to answer since the result itself is likely to be a product of an enormous training corpus on which the model is based and the user's input. While there is no definitive answer to this question yet, the current suggestion seems to be an evaluation on a case-by-case basis [5].
- **Training data:** Depending on the stage of production, the amount of available training data may be minor in comparison to the variety of elements that need to be generated. Especially in the early stages of development, machine learning models may merely be used to generate interesting mock-ups or explore ideas. Later development stages may allow to train specialized models or refine existing models to produce desired results.
- **Costs:** Creating your own machine learning models or using the models provided by others can come with non-negligible costs. The required hardware, energy, and time for training and inference should be kept in mind while planning a pipeline. Reducing these costs is already a key aspect of machine learning research and further advancements may considerably reduce the related costs.
- **Usability and Explainability:** Each of the envisaged pipelines comes with its own unique challenges. Especially, the usability of black box models may become a problem in case the input space is not well understood. We have tackled this problem by splitting the asset generation into multiple sub-tasks which we were able to control independently with limited success. Better explaining a model's relation between in- and output as well as its parameter space may help in increasing the usability of such models.

While there are still many steps ahead of us, supporting the generation of game assets using machine learning models may have huge impact on the field. At the current stage, existing models may already be used to support the prototyping stage or generate mock-ups and ideas for the human-guided generation process. Having further advanced on the models' capabilities, it may be possible to learn from just a few examples and produce game assets

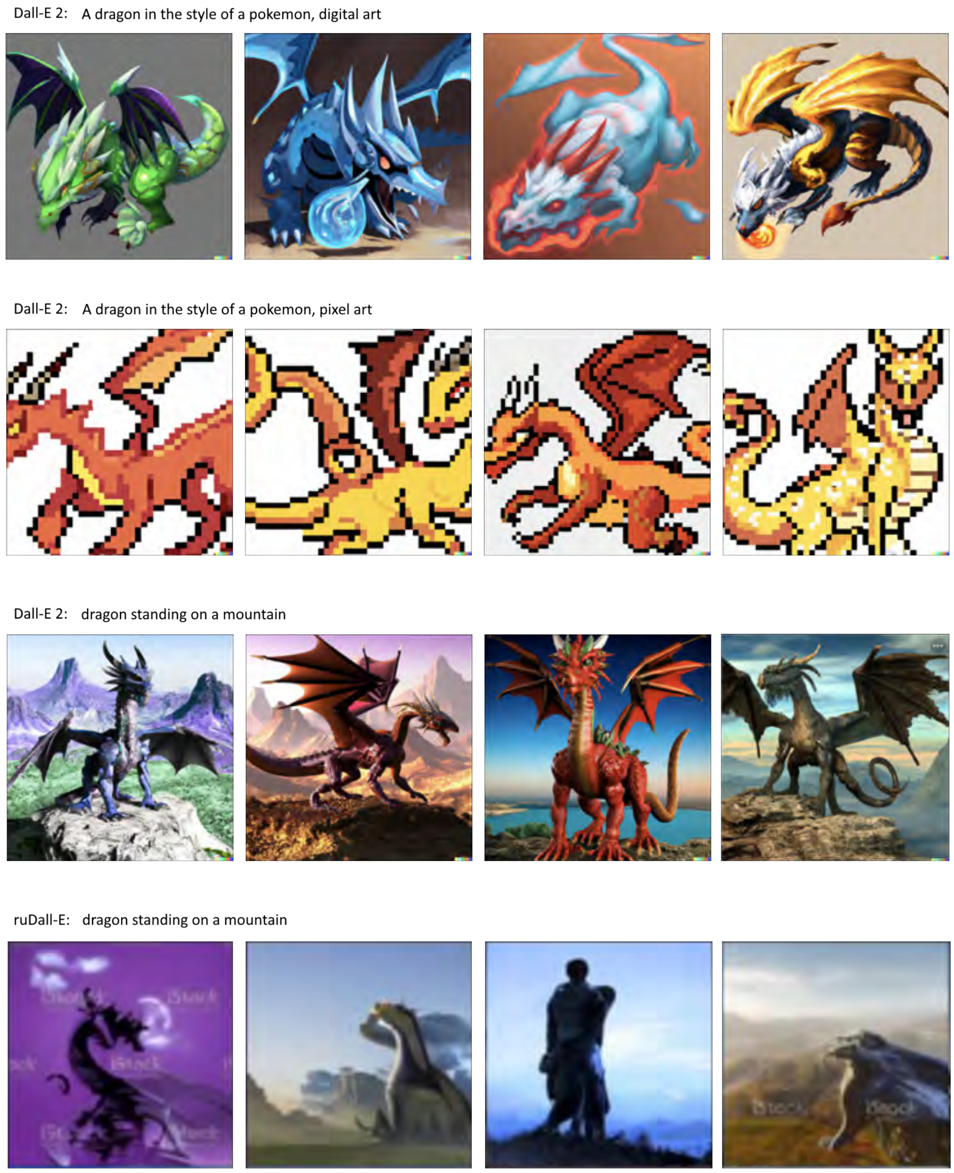


**Figure 3** Comparison of Dall-E 2 and and the openly accessible ruDall-E for generating dragons and pokemon-like creatures. Dall-E 2 is able to produce images of higher quality. However, it requires an invitation from OpenAI to be used.

of matching styles. In the long run, combinations of machine learning models may even guide the development of whole game worlds and game mechanics, allowing us to generate complete game experiences given a user's queries.

**Figure 4** Demo pipeline for generating Pokemon-like creatures. First, generating images of dragons using ruDall-E[4] (left) discarded examples, (middle) chosen example, (right) given the text prompt "A dragon in the style of a pokemon" we used VQGAN [7] and CLIP [6] to produce the final result.

## References


1. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. (arXiv,2022), https://arxiv.org/abs/2204.06125
2. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S., Ayan, B., Mahdavi, S., Lopes, R., Salimans, T., Ho, J., Fleet, D. & Norouzi, M. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. (arXiv,2022), https://arxiv.org/abs/2205.11487
3. Dayma, B., Patil, S., Cuenca, P., Saifullah, K., Abraham, T., Lê Khâc, P., Melas, L. & Ghosh, R. DALL E Mini. (2021,7), https://github.com/borisdayma/dalle-mini
4. Shonenkov, A. ruDall-E. (2021), https://pypi.org/project/rudalle/
5. Chiou, T. Copyright lessons on Machine Learning: what impact on algorithmic art?. (2019)
6. Radford, A., Kim, J., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. & Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. (arXiv,2021), https://arxiv.org/abs/2103.00020
7. Esser, P., Rombach, R. & Ommer, B. Taming Transformers for High-Resolution Image Synthesis. (2020)
8. Liu, V. & Chilton, L. Design Guidelines for Prompt Engineering Text-to-Image Generative Models. *CHI Conference On Human Factors In Computing Systems.* (2022)
9. Zhu, J., Park, T., Isola, P. & Efros, A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *Computer Vision (ICCV), 2017 IEEE International Conference On.* (2017)
10. Gatys, L., Ecker, A. & Bethge, M. Image Style Transfer Using Convolutional Neural Networks. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition (CVPR).* (2016,6)