Visual Procedural Content Generation with an Artificial Abstract Artist

Matthew Guzdial, Duri Long, Christopher Cassion, Abhishek Das
College of Computing
Georgia Institute of Technology
Atlanta, GA 30308 USA
{mguzdial3, duri, ccassion3, abhshkdz}@gatech.edu

Abstract

We present Pollite (Pollock-lite), an artificial abstract artist with the capability to evaluate, augment, and generate video game visual elements. Our system is based on a cognitive model of abstract artists built from self-reports and interviews. The main intelligence behind our system is a Convolutional Neural Net (CNN), a deep neural network approach that has shown great success in image tagging tasks and that can learn associations between shapes, colors, and concepts. We demonstrate initial results with our system across three case studies.

Introduction

Procedural content generation (PCG) refers to the body of techniques for algorithmically generating video game content through either designer-defined rules and heuristics or based on machine learned models of existing content (Hendrikx et al. 2013; Summerville et al. 2017). The majority of PCG systems, with some notable exceptions (Cook, Colton, and Gow 2016a), do not attempt to replicate the internal processes of human creative designers, but instead focus on a particular game development task (e.g. video game level creation). In addition, the majority of PCG work focuses on function over form in large part due to functional elements affording easier evaluation (Cook and Smith 2015). By function vs. form we indicate the divide between focusing on structural, quantifiable elements in games like level generation focused on playability over altering the visual appearance of a game.

A PCG system capable of generating, altering, and evaluating the visual aesthetics of a video game could be an invaluable tool for video game developers, particularly for those developers who are not themselves skilled in visual design or do not have access to quality visual design knowledge. Such a PCG system would benefit in being modeled after a human creative visual designer, in order to facilitate better collaboration with human developers. Thus a human developer could communicate their aesthetic intentions to the automated visual designer in natural language.

In this paper we present an artificial abstract artist, Pollite (Pollock-lite). Pollite is capable of generating, altering, and evaluating visual components of video games, and is based on a cognitive model of a human abstract artist. We focus on abstract art for our cognitive model in order to avoid tying our system to a particular style of game art. We learn the visual aesthetic knowledge for our system from real-world images (photographs) with a convolutional neural network (CNN), a deep learning approach.

The rest of this paper is organized as follows. First, we cover relevant background work from a variety of fields related to artificial visual aesthetics. Next, we provide an overview of our system, including the cognitive model and deep neural network approach. We then present three case studies showing initial results of our system applied to evaluating, generating, and altering visuals for video games. We end the paper with a discussion of these results and future avenues for research.

Background

In this section we cover work from a set of related fields. We discuss the existing prior examples of procedural content generation applied to visual game artifacts as well as prior work in artificial visual artists. We then cover style transfer and texture generation, two related fields within the wider discipline of computer graphics, both with features relevant to our work.

Visual Procedural Content Generation

Visual procedural content generation, the generation of visual components of a video game, has to a large extent been focused on the generation of photo-realistic textures for 3D games (Hendrikx et al. 2013). This means that the majority of prior work has focused on a single aesthetic (photorealism) and a single type of visual component (textures). There are notable exceptions to this trend, including generation of weapon particle effects in a space shooter (Hastings, Guha, and Stanley 2009), avatar generation for a research game (Lim and Harrell 2015), and generation of character and item art for a rouge-like RPG (Johnson 2016). Similar to our work, the game-creation and playing mobile application Gamika (Colton et al. 2016) makes use of abstract generated art, but with no communicated intention or aesthetic.

Cook's ANGELINA system stands as a particularly important reference point to our work, with ANGELINA being an artificial game designer capable of expressing design intent and with substantial focus on aesthetics (Cook, Colton, and Gow 2016a). For example, one iteration of ANGELINA read in news articles and created games with an aesthetic intended to match the tone and substance of the article (Cook, Colton, and Gow 2016b). Notably, instead of generating art, ANGELINA collected images from online repositories. More recently Cook has focused on vision-driven procedural level design, in part to guide a player's play via aesthetics (Cook 2015).

Artificial Visual Artists

Generative Systems Current work in developing artificial visual artists can be broadly divided into one of three categories (Davis 2015). *Generative systems* are systems that automatically create artwork based on a pre-determined corpus of knowledge. Generative systems like AARON (McCorduck 1991), The Painting Fool (Colton 2011), MONICA (de Silva Garza and Lores 2004), and KANTS (Fernandes, Mora, and Merelo) have succeeded in generating novel and interesting works of art both in abstract and more realistic aesthetic styles.

Two generative systems that are particularly relevant to our work are Horn et al.'s Visual Information Vases (Horn et al. 2015) and the DARCI system (Norton, Heath, and Ventura 2013). Horn et al. developed a cognitive model of transmedia inspiration, using color palettes in photographs to inspire a generative AI agent's creation of 3D printed vases (Horn et al. 2015). This is similar to the model of inspiration that we implemented in Pollite, which draws on inspiration from photographs and stories to create abstract paintings.

DARCI is a generative system that can classify images using labels (e.g. lonely, peaceful) (Norton, Heath, and Ventura 2010) in addition to using a genetic algorithm to modify images according to these labels in order to create unique art pieces (Norton, Heath, and Ventura 2013). This approach is similar to how Pollite learns how to express emotions visually from emotion-labeled photographs, with a key difference being that Pollite creates completely novel abstract paintings rather than applying filters to existing images.

Creativity Support Tools The second area of work in artificial visual art is the area of *creativity support tools*, or systems that help humans to be more creative. While somewhat less relevant to Pollite than generative systems, several creativity support tools have been developed that draw on cognitive models as our system does. For instance, Machado and Amaro's system uses a model of ant cognition to assist humans in transforming photos into non-photorealistic images (Machado and Amaro 2013).

Computer Colleagues Finally, *computer colleagues* are agents that work in collaboration with a human to generate artwork. Systems like the Drawing Apprentice are capable of creatively collaborating with human participants, in this case to create both abstract and more realistic sketch-based visual art (Davis et al. 2016). These colleagues are arguably the best cognitive match for human creativity, as they do not decouple the perception-action feedback loop that is central to the creative process (Davis 2015). In other words, computer colleagues are able to perceive changes in their environment and then act accordingly upon the artifact they are making, allowing real-time creative collaboration rather than just isolated rule-based generation.

Our system falls in the space between generative systems and computer colleagues. Pollite is similar to generative systems like DARCI in that it creates artwork on its own without external interaction with humans. However, unlike DARCI, we attempt to model the iterative perception-action feedback loop that is central to the creative process, making the cognitive model we have developed more similar to that of a computer colleague despite the lack of interaction with a human partner (Davis 2015). Thus, while our system is not currently applied to real-time collaboration, we imagine this application as a natural extension.

Texture Generation and Style Transfer

Within the field of graphics there exists a tradition of work in generating textures and transferring visual style. Both fields rely on processes that feed some input image(s) to a system, build a model of the visual aesthetic of these input image(s), and then express this model through output images. In texture generation this process typically takes small, low-resolution images and outputs higher-resolution textures (Fleischer et al. 1995; Wei and Levoy 2000; Cohen et al. 2003).

Style transfer approaches involve altering an input image to represent a specific visual style (e.g. a photograph in the style of Van Gogh's *Starry Night*) (Johnson, Alahi, and Fei-Fei 2016). Similar to our system's ability to palette swap there exist approaches in this area to alter the hue and saturation of images to match an input image (Neumann and Neumann 2005). Recently, there has been a focus on deep neural networks for style transfer, particularly convolutional neural nets, which we also make use of (Gatys, Ecker, and Bethge 2016). While our system can generate textures and alter the style of existing visuals, we focus on learning a more abstract, general model based on natural language concepts instead of replicating a specific visual style.

System Overview

At a high level, Pollite is an artificial abstract artist that learns to visually depict natural language concepts by training on real world images. Pollite is based on a cognitive model of an abstract artist, derived through prior research and interviews with such artists, in order for it to best fill the role that a visual designer might otherwise have in a video game development team. The underlying intelligence of Pollite is based on a convolutional neural net (CNN), a deep neural network architecture that learns to associate colors and shapes with natural language concepts based on real-world training data. In this case, our real-world training data is a collection of thousands of pre-tagged photographs downloaded from Flickr.

In this iteration of Pollite we make use of Ekman's six basic emotions (Ekman 1992) (anger, disgust, fear, joy, sadness, and surprise), as an initial set of natural language concepts to train the system. We note that this paper represents an initial experiment into the approaches presented. Thus our deep neural network approach is not all that deep,

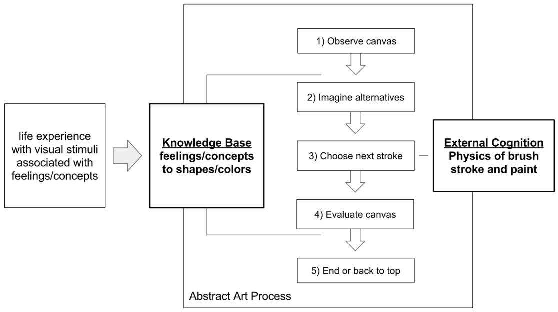

Figure 1: Our model of the process an abstract artist undertakes in creating art.

our brush stroke model is relatively simple, and we pull on a well-known rather than well-verified emotional model in Ekman. However taken together, this is still a system capable of evaluating, adapting, and generating visual components of video games according to Ekman's six basic emotions.

Cognitive Model

We based our approach on the cognitive process of an abstract artist. We noted a natural corollary between altering and generating game visuals to target certain emotions/concepts and the way abstract art is able to describe emotions/concepts using aesthetic elements such as shape, color, and composition. An investigative exploration into the artistic and cognitive processes of artists like Jackson Pollock (Pollock 1950; 1951; Plowshares Media 2010b), Franz Kline (Plowshares Media 2010a), and Mark Rothko (Plowshares Media 2010c) led us to three main conclusions that we incorporated in our system implementation. These conclusions do not necessarily model the cognition of all artists, but instead reflect the cognition of the specific artists working in the style that we were aiming to emulate.

1. **Artistic cognition is iterative.** That is, artists typically do not develop a complete mental image of their artwork before beginning to paint; instead, they compose and evaluate the painting as they work (Pollock 1951). This intuition has previously been modeled in artistic agents like MONICA, which utilize evolutionary algorithms to iteratively self-evaluate (de Silva Garza and Lores 2004).

2. **Artistic cognition is embodied.** As an example, Jackson Pollock's painting style is often described by art historians as "action painting" because the physical movement of the body and the paint contribute significantly to the artistic process (Plowshares Media 2010b). This ties in closely with embodied cognition, the idea that cognition is takes place not only in the mind but also in the interactions of one's body with the outside world (Wilson 2002).

3. **Artists draw on multi-modal sources of inspiration.** These sources can include personal experiences, historical or current world events, other artwork, literature, and/or music (Plowshares Media 2010a; 2010c; Pollock 1950).

These three principles led to the development of a cognitive model of an abstract artist that iteratively reconceptualizes its paintings while it works, utilizes brush stroke physics as an aspect of its cognition, and is capable of drawing on both literature and prior visual experience as sources of inspiration.

We illustrate our process model derived from this background research in Figure 1. We note that abstract artists build up a knowledge base of mappings between emotions/concepts to shapes/colors throughout their lives. We instantiate this in knowledge base and learning process as a convolutional neural net architecture trained on photographs tagged with emotions, which we expand further below. When actually creating art an abstract artist makes use of iterative evaluation and alteration, making use of the external cognition of the physics of the brush stroke. We model this as a greedy hill climbing process, with the system generating hundreds of variations according to a simple brush physics model (made up of velocity and inertia), and choosing the variation that maximally activates its knowledge base (the trained CNN). While we do not picture it within Figure 1 we anticipate that artists choose palettes according to the concept they wish to visualize in a similar iterative fashion, and we address this further in the next section.

Deep Neural Network Implementation

In this section, we go over the deep neural network architecture implementation details of our system. We specify two neural network architectures within our system. First, a three-layer fully-connection neural network trained to classify color palettes, sets of colors used for creating art, associated with a particular emotion. Second, a three-layer convolutional neural network (CNN) trained to classify images according to a particular emotion, which serves as the knowledge base discussed in the prior section.

The palette neural network takes as input a list of thirty-two colors, with each color represented in an RGB format. That is each color is made up of a vector of length three, with the values representing the amount of red, green, and blue within the color (each value between 0 and 256). We make use of RGB as our color space representation due to its typical application in video games, as opposed to other color space representations. This palette is passed through three fully connected layers of sizes 32, 64, and 126 with ReLU activation, before a final soft-max layer.

We use CNNs to model the knowledge base for our system. CNNs are powerful pattern recognizers that have gained recent popularity for a variety of computer vision tasks (Krizhevsky, Sutskever, and Hinton 2012; Bengio, Courville, and Vincent 2013). More relevant to our system, CNNs have the capacity to encode knowledge and learn concepts of part-related information in the form of distributed representations when trained for a downstream task (Zhou et al. 2014). We first train our CNN to classify emotions from photographs, in the process encoding knowledge of abstract concepts associated with each of these emotions in the network similar to how humans understand abstract concepts from real experiences, such as bright colors being associated with happiness, green tints with disgust, etc.

Figure 2: First returned screenshot from each of the seven games in chronological order from left to right: Doom(1993), Super Mario 64(1996), Super Mario Galaxy(2007), Candy Crush(2012), Bloodborne(2015), Mobile Strike (2015), and Overwatch (2016).

	Ang.	Dis.	Fear	Joy	Sad.	Surp.
Accuracy	37%	11%	60%	20%	24%	49%

Table 1: Accuracy at which the system predicts the correct emotional tag in a test set.

Our CNN takes as input images of 256 by 256 pixels, with each pixel represented according to its RGB value as described above. Thus, a total input shape of 256x256x3. We pass this input through three convolutional layers, each with leaky ReLU activation with filters of sizes 16, 32, and 32, with window sizes of 4x4, 2x2, and 2x2 respectively. Each CNN layer has a max pooling layer of size 2 after it, and the final max pooling layer ends in a soft-max layer.

Once both models are trained, we begin by selecting a target concept/emotion and instantiating an initially random palette. This palette is modified in a greedy manner to maximally activate the target concept/emotion until it reaches some local maxima. We then have our system move on to a blank canvas, and have the trained model draw incremental brush strokes on it. Location and color of the brush strokes is based on a greedy process by which the system simulates the potential impact of random brush strokes based on a simple physics model (only including velocity and inertia of the paint). Each of these simulations is passed through our CNN model as a distinct image, with the image that maximally activates the target concept/emotion having its brush stroke placed on the actual canvas. Intuitively, the network draws artwork in colors and shapes to express concepts that it has learned to associate with certain emotions during training.

We use photographs from Flickr as our initial training dataset. These images have metatags classifying them into one of six emotions – anger, joy, surprise, disgust, sadness, and fear. For the palette network, the top thirty-two most common colors are calculated for each image to create a palette training set. Note that our approach is robust to the specific choice of emotions or concepts, and we choose emotions from Ekman's work to present a proof-of-concept. In total, we use 1000 training images for each emotion, making for a total of 6000 images.

Preliminary Evaluation

We present an preliminary objective evaluation to demonstrate the system's ability to correctly identify emotional tags in novel photographs. For this evaluation we collected a set of one-hundred unique photographs from Flickr for each emotional tag. Thus creating a test set one-tenth in size to our training set. We ran each image of this test set through the entirety of our system and took the maximally activated index of the output vector as the system's predicted emotional tag.

We summarize the results of this preliminary evaluation in Table 1. We note that for this task (choosing one from a set of six) we can expect a random baseline performance of roughly 17%. We note that all emotional tags achieved at least this performance other than disgust. This indicates that the system has learned some helpful features in predicting the other emotional tags. We wish to draw special attention to fear and surprise, both of which should impressive performance for such a simple approach. We expect these results to only improve in future iterations of the system.

Case Studies

To demonstrate the utility of the system, we isolated three visual design tasks associated with games and performed three case studies. The studies (*identifying tone of game visuals*, *abstract texture generation*, and *adjusting game visuals*) were meant to present preliminary results on our systems ability to evaluation, generation, and adjust game visual aesthetics. The goal is to determine if the system can be considered a useful tool for these tasks.

Identifying Tone of Game Visuals

For the first case study, we look at the system's performance versus our intuition in identifying the tone of game visuals. For this experiment, we've selected seven different games from different eras represented a variety of different visual styles and tones. We briefly describe each of these games and our expectation on how the system should categorize each in terms of whether the visual tone is primarily light or dark.

1. **Doom (1993)**: Doom is a first-person shooter game where the player must find low-resolution demons through a variety of hellish environments. We note it has a dark tone.
2. **Super Mario 64 (1996)**: Super Mario 64 is a 3D platformer game where the titular Mario collects bright stars through a set of visually distinct stages to save a princess. We note it has a light tone.
3. **Super Mario Galaxy (2007)**: Super Mario Galaxy is much like its predecessor, but with higher-resolution

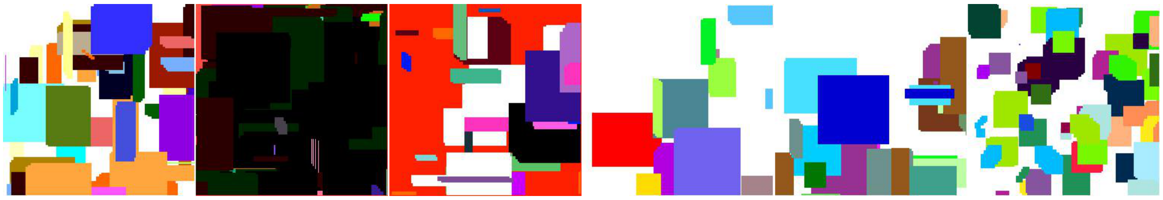

Figure 3: Output of our system given a blank canvas. From left to right the output is intended to represent anger, disgust, fear, joy, sadness, and surprise.

	Ang.	Dis.	Fear	Joy	Sad.	Surp.
Doom	1	3	4	1	7	3
SM 64	4	5	7	3	2	2
Galaxy	3	5	3	2	3	6
Candy Crush	5	5	1	5	4	7
Bloodborne	2	1	6	4	6	1
Mobile Strike	7	2	2	7	1	5
Overwatch	6	4	5	6	5	4

Table 2: A table of the ranking of the median values of each game for each emotion.

graphics and taking place in space. We note it has a light tone.

4. **Candy Crush (2012)**: Candy Crush is a casual mobile puzzle game focused on bright candy and flashy effects. We note it has a light tone.
5. **Bloodborne (2015)**: Bloodborne is an intense, high-definition action game with strong horror elements. We note it has a dark tone.
6. **Mobile Strike (2015)**: Mobile Strike is a gritty mobile strategy game. We note it has a dark tone.
7. **Overwatch (2016)**: Overwatch is a bright first-person shooter with Pixar-like environments. We note it has a light tone overall, but darker than most of the other light tone games.

To collect data for this case study we performed a Google search for each game and collected the top ten unique screen shots. We include a section of the first screenshot for each game in Figure 3. In order to avoid the issue of different screen resolutions, we parsed the screen shots to identify their palettes (see System Overview section), ran the ten palettes through the CNN to get the confidence values of each of the Ekman emotion tags. We ran a Kruskal-Wallis test for each emotion and found that at least one of the game's score distributions differed significantly from the others for all the emotions but anger ($p < 0.01$).

We summarized the results of the experiment in a table of rankings of the median score for each game and each emotion in Table 2. Each row i represents a video game and each column j represents one of Ekman's six basic emotions. To illustrate, row/column combination Doom/Ang represents that Doom was ranked number 1 by the system for anger, while the combination Overwatch/Surp represents Overwatch ranked number 4 out of the video game images analyzed for surprise.

The results summarized in Table 2 largely follow our intuitions. Doom is identified as the most angry game, Bloodborne as the most disgusting game, and Mobile Strike as the saddest game. Further the system equally ranked Super Mario 64, Super Mario Galaxy, and Candy Crush as all being the least disgusting.

Notably the system still has some unintuitive performance, for example rating Doom as the most joyful game. Over different input-output sequences, it can be said that the system tends to identify sharp contrast in pixel coloring with joy. Doom's overall color design exhibits sharp contrast between the objects themselves and the background, which thus explains the ranking. While we note that both of the Mario games appear next in the ranking, it is clear that there is the potential for improvement in this area of performance. However, we still take these initial results as promising.

Abstract Texture Generation

The second case study looks at whether our system, devoid of any stylistic parameters, can create abstract images that can be utilized as abstract textures for 3D games. We specify abstract textures in contrast to the majority of texture generation techniques for photo-realistic video games. To accomplish this, we set running Pollite through the complete generation process for each of the target emotions. As covered in the system overview section, this meant that the system first generated a 32-color palette based on greedily perturbing an initially random palette, then greedily make "brush strokes" with this palette on an initially blank canvas. We ran this complete process five times for each target emotion, and selected the output image that most highly activated the target emotion in our system. We note that we could have hand-specified a specific palette to use, but wished to present results that represented the entire system to understand it as a whole.

We present the results from this evaluation in Figure 3. From left to right the images present our system's attempts to visualize the target emotions of anger, disgust, fear, joy, sadness, and surprise. Informally, we find these results to be representative of the target emotions. Anger appears to have warm orange colors boiling underneath a selection of other colors. Disgust appears sewer-like with graffiti-esque highlights. Fear has similar markings and coloration as a poison dart frog. Joy is simple and bold. Sadness has cool colors weighing downwards. Lastly, surprise appears almost

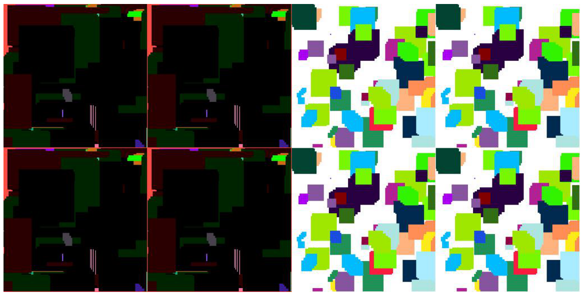

Figure 4: Tiled examples of the disgust and surprise images.

confetti-like. A formal evaluation is necessary to determine whether our subjective interpretations of these results hold true for others (see *Future Work*).

We highlight the disgust and surprise results as representative images that without any alteration can be treated as textures. We demonstrate this in Figure 4 with tiled versions of these two images. We note that the other final images appear more like abstract art than 3D game textures, besides perhaps fear. This is to be expected given our technique, and alterations could be made to encourage more texture-like output. However, we still claim that given that the system was based on a cognitive model of an abstract artist, it is a promising result that some of the output already resembles 3D game textures.

Adjusting Game Visuals

In this case study, we demonstrate the potential for our approach to be used to alter game visuals to better reflect a specific emotion or concept, in essence altering the game's mood or tone. Our approach was to take examples of existing game visuals (either in-game spritesheets or screenshots of an existing game) and pass them into our system along with a target emotion (either anger, disgust, fear, joy, sadness, or surprise).

Our system first determined the current palette of the image, or the set of colors present in the image. Given that our system can only represent palettes of thirty-two colors, in the case where there were too few colors, our system filled the remaining slots with noise. In the case where there were too many colors, our system made use of the thirty-two most common colors in the image. Then, our system greedily altered this palette to better match the target emotion, altering the original image colors as it went. This can be understood as automated palette-swapping, a technique used in video games to alter a character or environment to make it appear to be a different character or environment (e.g. the Luigi avatar in the original Super Mario Bros. is actually a palette-swapped Mario).

After this palette swapping phase the system then undertook the same process as in the prior case study. However, it made use of the input image instead of a blank canvas and the final palette from the palette-swapping phase. This lead to a greedy alteration process in order to better excite the target emotion in the trained CNN.

We present an illustrative example of this output in Figure 5. In this figure we present a comparison between an original image (left), an attempt by our system to make the original image seem "angry", and an art piece by an artist named Nathaniel Bart, who attempts to make "darker" versions of classic game visuals. We note a clear similarity between the system's output and Bart's image in terms of heightened contrast, particularly on the building and roots of the trees.

We present further examples of this approach applied to a spritesheet from Super Mario World as visualized in levels in Figure 6. We made use of spritesheets, collections of all the visual elements in a game, for Super Mario Bros. with the the target emotions of anger, joy, and surprise. While we lack a human example to refer to, we find the results altogether to match our intuition. Specifically, anger mutes most colors while making the blocks more red, and giving an angry "eyebrow" to the goombas (the small, squat mushroom-like enemies in Mario). Joy lends a pleasing easter-like color scheme to the level elements. Surprise was the most interesting, adding pink polka dot splotches to the background and altering the arrow at the start of the level into a 7-shape, along with adding stripes to the arrow base. While it cannot be seen when not animating still, the target emotion of surprise also added a flashing green to the block animation, which is certainly surprising.

Future Work

A clear next step for this project would involve an evaluative study intended to formally assess how well the new textures generated by our system represent the target emotion. This could be accomplished by letting participants play a game several times, each time with a different texture applied. After each game-play session, the participant would be asked to match the tone of the game visuals with one of Ekman's six basic emotions (Ekman 1992). A study like this would provide a more quantitative measure of how well our system represents emotions.

Another area for future work is augmenting the system with the ability to analyze a game design document or other text forms and generate abstract textures from it. The challenge with this approach comes with not only correctly identifying what is useful to the generating mechanism, but also making sure that the system's understanding of those specifications matches the overall aesthetic intended for the game. That is where an evaluation mechanism would become increasingly important. Using an off-the-shelf text parser we have begun an initial exploration into this and present example output of this experimentation in Figure 7.

We note that while our system has shown some success generating abstract game visuals, the majority of game visuals are representative even if not photo-realistic. We identify the ability to represent game objects with a particular emotion as an open problem that we hope to engage with in future work. In particular, we think generative adversarial networks (GANs) might be helpful given the success they have shown in generating high quality representative images.

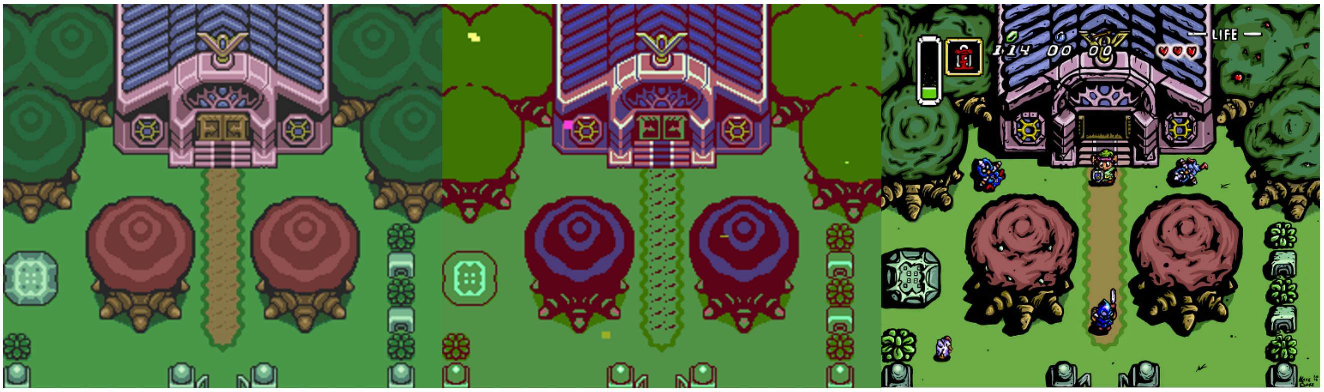

Figure 5: Comparison between the original background (left), our system's "anger" version of the background image (center), and a human interpretation of a "darker" version of the same scene.

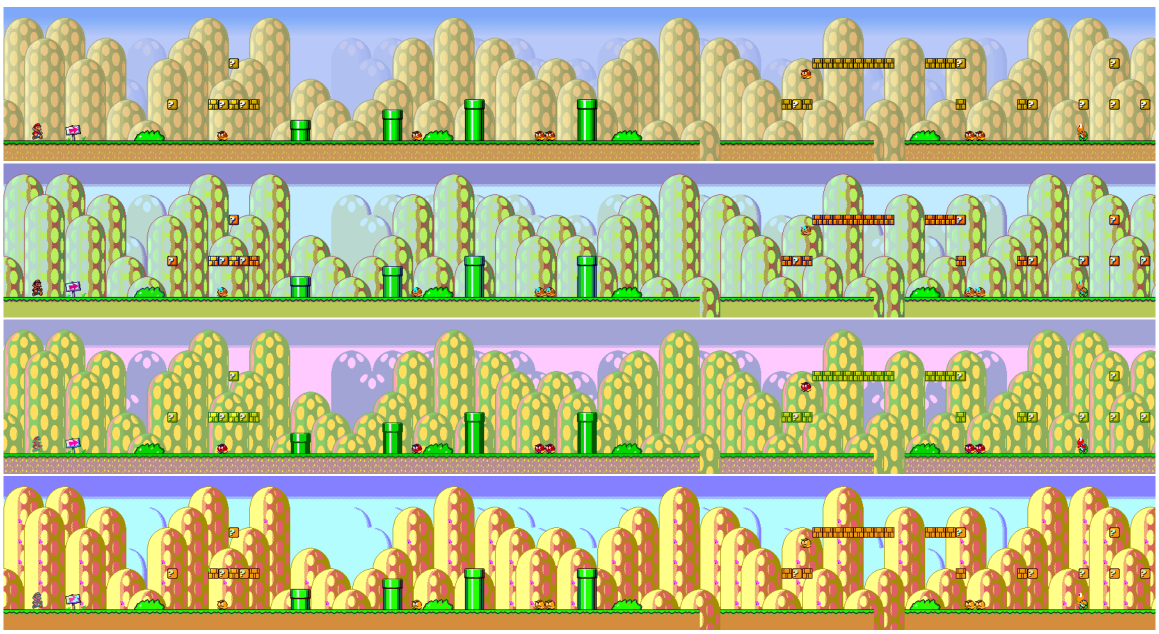

Figure 6: From top to bottom, Super Mario World graphics, then altered to express anger, joy, and surprise.

Conclusion

In this paper we present Pollite, an artificial abstract artist capable of evaluating, generating, and adjusting game visuals to better match target natural language emotions. Our technique makes use of a cognitive model of an abstract artist, with the majority of its intelligence made up of deep neural network architectures trained from thousands of images tagged with the names of emotions. Through a series of three case studies we demonstrate the breadth of tasks that the system can accomplish and some promising initial results. Taken together this represents an initial iteration for a system that could become a visual designer partner for game development.

Acknowledgments

Many thanks to Dr. Ashok Goel, Joshua Killingsworth, and Jose Delgado for their support and advice.

References

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35(8):1798–1828.

Cohen, M. F.; Shade, J.; Hiller, S.; and Deussen, O. 2003. *Wang tiles for image and texture generation*, volume 22. ACM.

Colton, S.; Nelson, M.; Saunders, R.; Powley, E.; Gaudl, S.; and Cook, M. 2016. Towards a Computational Reading of Emergence in Experimental Game Design. In *Computa-*

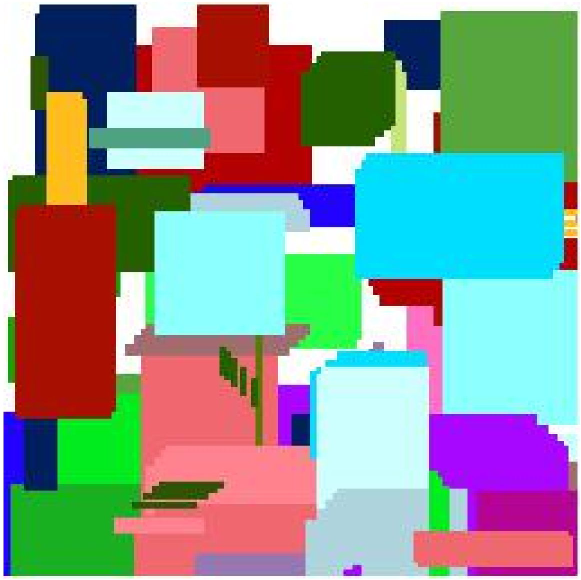

Figure 7: Initial output from experimenting with visualizing poetry, specifically the line "Two Roads Diverged in a Yellow Wood"

tional Creativity and Games Workshop at the 2016 International Conference on Computational Creativity.

Colton, S. 2011. The painting fool in new dimensions. In *Proceedings of the 2nd International Conference on Computational Creativity*, volume 112.

Cook, M., and Smith, G. 2015. Formalizing non-formalism: Breaking the rules of automated game design. In *Proceedings of the 2015 Conference on the Foundations of Digital Games, Monterey, CA*.

Cook, M.; Colton, S.; and Gow, J. 2016a. The angelina videogame design system, part i. *IEEE Transactions on Computational Intelligence and AI in Games*.

Cook, M.; Colton, S.; and Gow, J. 2016b. The angelina videogame design system, part ii. *IEEE Transactions on Computational Intelligence and AI in Games*.

Cook, M. 2015. Would you look at that! vision-driven procedural level design. In *Third Experimental AI in Games Workshop*.

Davis, N.; Hsiao, C.-P.; Yashraj Singh, K.; Li, L.; and Magerko, B. 2016. Empirically studying participatory sense-making in abstract drawing with a co-creative cognitive agent. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 196–207. ACM.

Davis, N. 2015. An Enactive Model of Creativity for Computational Collaboration and Co-creation. In *Creativity in the Digital Age*. Springer. 109–133.

de Silva Garza, A. G., and Lores, A. Z. 2004. *Automating Evolutionary Art in the Style of Mondrian*. Berlin, Heidelberg: Springer Berlin Heidelberg. 394–395.

Ekman, P. 1992. An argument for basic emotions. *Cognition and Emotion* 6:169–200.

Fernandes, C. M.; Mora, A.; and Merelo, J. Kants–a stigmergic algorithm for data clustering and swarm art.

Fleischer, K. W.; Laidlaw, D. H.; Currin, B. L.; and Barr, A. H. 1995. Cellular texture generation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 239–248. ACM.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2414–2423.

Hastings, E. J.; Guha, R. K.; and Stanley, K. O. 2009. Automatic content generation in the galactic arms race video game. *IEEE Transactions on Computational Intelligence and AI in Games* 1(4):245–263.

Hendrikx, M.; Meijer, S.; Van Der Velden, J.; and Iosup, A. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9(1):1.

Horn, B.; Smith, G.; Masri, R.; and Stone, J. 2015. Visual information vases: Towards a framework for transmedia creative inspiration. In *ICCC*, 182–188.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 694–711. Springer.

Johnson, M. R. 2016. Towards Qualitative Procedural Generation. In *Computational Creativity and Games Workshop at the 2016 International Conference on Computational Creativity*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*.

Lim, C.-U., and Harrell, D. F. 2015. The marginal: A game for modeling players perceptions of gradient membership in avatar categories. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*.

Machado, P., and Amaro, H. 2013. Fitness functions for ant colony paintings. In *Proceedings of the 4th International Conference on Computational Creativity*, 32–39.

McCorduck, P. 1991. *Aaron's code: meta-art, artificial intelligence, and the work of Harold Cohen*. Macmillan.

Neumann, L., and Neumann, A. 2005. Color style transfer techniques using hue, lightness and saturation histogram matching. In *Computational Aesthetics*, 111–122. Citeseer.

Norton, D.; Heath, D.; and Ventura, D. 2010. Establishing appreciation in a creative system. In *ICCC*, 26–35.

Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *The Journal of Creative Behavior* 47(2):106–124.

Plowshares Media. 2010a. The Painting Techniques of Franz Kline: Chief.

Plowshares Media. 2010b. The Painting Techniques of Jackson Pollock: One: Number 31, 1950.

Plowshares Media. 2010c. The Painting Techniques of Mark Rothko: No. 16 (Red, Brown, and Black).

Pollock, J. 1950. Jackson Pollock: An Interview.

Pollock, J. 1951. Jackson Pollock: Paintings have a life of their own.

Summerville, A.; Snodgrass, S.; Guzdial, M.; Holmgård, C.; Hoover, A. K.; Isaksen, A.; Nealen, A.; and Togelius, J. 2017. Procedural content generation via machine learning (pcgml). *arXiv preprint arXiv:1702.00539*.

Wei, L.-Y., and Levoy, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 479–488. ACM Press/Addison-Wesley Publishing Co.

Wilson, M. 2002. Six views of embodied cognition. *Psychonomic bulletin & review* 9(4):625–636.

Zhou, B.; Khosla, A.; Lapedriza, À.; Oliva, A.; and Torralba, A. 2014. Object detectors emerge in deep scene cnns. *CoRR* abs/1412.6856.