Sprite Generation

CS 534 - Fall 2018 -Term Project Proposal

Instructor: Mohit Gupta - University of Wisconsin Madison
Date: September 26, 2018
Authors:

- Curt Henrichs - cdhenrichs@wisc.edu - 9070390126
- Sayem Wani - swani@wisc.edu - 9080301972
- Saheen Feroz - sferoz@wisc.edu - 9080316251

Problem

Developing visual content for games, even simplified 2D games, proves to be a hurdle for indie studios and solo game developers who have neither the time nor capital to develop captivating environments, engaging character designs, and remarkable particle effects. So these developers turn to procedural algorithms to assist them in this endeavour. Current, general-use algorithms either iterate over the equivalent of building blocks to construct sprites that lack the detail of a hand-crafted image or explore a crafted algorithm using a pseudo-random seed [1]. Alternatively a developer can develop a deep neural network, such as a generative-adversarial network, to produce novel sprites for their game [2]. However, a neural network is not a panacea; it requires a large dataset of images. Additionally, the developer must rely on the network converging to a optimal weight space while simultaneously not overfitting.

Solution Proposed

Our solution takes significant inspiration from Lewis Horsley's and Diego Perez-Liebana's paper titled *Building an Automatic Sprite Generator with Deep Convolutional Generative Adversarial Networks* [2]. Specifically we will use a similar DCGAN model including their use of strided convolution in lieu of maxpooling and fractional strided convolution instead of upsampling and we will implement batch normalization. Additionally, we are making the following changes,

1. Our network will generate images of either 16x16 or 32x32 pixel images depending on the prevalence of sprites closer to one or the other. Note only one size will be implemented.
2. We will adjust the hyperparameters such as learning rate and network structure to get the best results from our dataset.
3. We will perform pretraining of the generator model through auto-encoders. Our network will use successive decoders from result back trained through the normal unsupervised process. This should populate the weight with a better chance of generalizing from input. A potential drawback will be lack of novelty in output images if the auto-encoder overfits during training.

Several other features that may be tried to improve performance,

1. Generative model will first develop a larger image in its network before downsampling. This larger space is analogous to a scratch pad for details that can then be downsampled to the final size through a convolutional network. Our hypothesis is that the network may have a hard time dealing with noise if the result space is small so by increasing it we hope the noise only affects details that get lost during downsampling.
2. Labeled inputs into generator and labeled outputs from discriminator to provide more control than merely a random seed as was the case in the paper. The challenge will be in constructing a dataset with appropriate labels.

Regarding the dataset, we will be collecting sprites from various open source resources. This should provide enough data for an initial system. If necessary we can perform simple manipulations such as changing the color palette and single-pixel manipulations. Finally, if we need more data than what can be collected online and with time permitting, we will employ neural style transfer on our current sprite. Our team recognizes the largest risk to the success of the project is collecting enough quality data for our network to generalize.

Several sources of data that we are currently investigating,

1. Oryx Design Lab - https://www.oryxdesignlab.com
2. Itch.io - https://itch.io/game-assets/free
3. Game Art 2D - https://www.gameart2d.com/freebies.html
4. OpenGameArt.org - https://opengameart.org/

Our models will be trained on constrained topics such as humanoid-character, items, faces, and non-humanoid enemies similar to the paper we took inspiration from.

Implementation Plan

An initial time table is provided below to guide our development process.

Date	**Goal**
Sept. 23	Website started, Project infrastructure started.
Sept. 26	Proposal submitted. Website is updated.
Sept. 30	Keras tutorials complete, basic mockup of toy DCGAN.
Oct. 21	Initial dataset complete without image manipulation. Labeling is not required. First DCGAN model designed to follow paper, statistics recorded.

Oct. 28	Improvements to DCGAN model such as autoencoder, statistics recorded. Dataset is extended with minor image manipulation. Team decides whether to pursue neural style transfer to increase dataset or pursue data labeling. Team decides whether to attempt scratch-pad network structure.
Oct. 31	Mid-Term Report submitted. Website is updated.
Nov. 18	Improvements as decided upon by team are implemented. Results recorded.
Nov. 25	Last minute changes are made. Final analysis is performed. Presentation write up started. Website results write up started.
Nov. 30	Website updated with all content. Presentation is complete.
Dec. 3-12	Final Presentations.
Dec. 12	Project Website Submitted.

Regarding technologies, we intend to use Keras as our neural network library due to its ease of use when leveraging Tensorflow along with it being a Python library. Our website will be hosted on github.io which provides free static hosting. The site will be generated using Jekyll with a standard predeveloped theme.

References

1. Hendrickx M., Meijer S., Velden J. V. D., Iosup A. *Procedural Content Generation for Games: A Survey*. ACM Transactions on Multimedia Computing, Communications, and Applications. v.9, n.1, p.1-22, February 2013. Online Access: https://dl.acm.org/citation.cfm?id=2422957
2. Horsley L., Perez-Liebana D. *Building an Automatic Sprite Generator with Deep Convolutional Generative Adversarial Networks*. IEEE Conference on Computational Intelligence and Games, p.134-141, August 2017. Online Access: https://ieeexplore.ieee.org/document/8080426/?part=1

Partner Contribution Breakdown

Curt Henrichs: Discussed goal, scope, research. Wrote initial rough draft. Reviewed final draft.
Sayem Wani: Discussed goal, scope, research. Reviewed and edited rough draft. Reviewed final draft.
Saheen Feroz: Discussed goal, scope, research. Reviewed and edited rough draft. Reviewed final draft.