AvatarGAN

Final Year
Project Report

Submitted by

Bhooyas Kapadia(A074)
Soham Deshpande(A086)

Under The Guidance Of

Prof. Ashwini Rao

In fulfillment for the award of the degree of

B.TECH.

INFORMATION TECHNOLOGY

At

Department of Information Technology
Mukesh Patel School of Technology Management & Engineering
NMIMS (Deemed –to-be University)
JVPD Scheme Bhaktivedanta Swami Marg,
Ville Parle (W), Mumbai-400 056.

March, 2022

CERTIFICATE

This is to certify that the project entitled "**AvatarGAN**" is the bonafide work carried out by Bhooyas Kapadia and Soham Deshpande of B.Tech (IT), MPSTME, Mumbai, during the VIII Semester of the academic year 2021-2022, in partial fulfillment of the requirements for the award of the degree of Bachelors of Technology as per norms prescribed by NMIMS. The project work has been assessed and found to be satisfactory.

(Signature of Internal Mentor 1)
Name:
Designation:

(Signature of External Examiner)
Name:
Designation:

HOD (IT)
(Dr.Ketan Shah)

Dean
(Dr. Alka Mahajan)

DECLARATION

We, Bhooyas Kapadia, Soham Deshpande, roll numbers: A074, A086 respectively, understand that plagiarism is defined as any one or the combination of the following:

1. Uncredited verbatim copying of individual sentences, paragraphs or illustrations (such as graphs, diagrams, etc.) from any source, published or unpublished, including the Internet.
2. Uncredited improper paraphrasing of pages or paragraphs (changing a few words or phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did or wrote what. (Source: IEEE, The Institute, Dec. 2004)

We have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of our work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.

We affirm that no portion of our work can be considered as plagiarism and we take full responsibility if such a complaint occurs. We understand fully well that the guide of the seminar/project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature:

Name: Bhooyas Kapadia

Roll No.: A074

Date: 23/03/2022

Signature:

Name: Soham Deshpande

Roll No.: A086

Date: 23/03/2022

BA

GSTIN NO: 24CBQPB7020N1ZJ

Date: 15th October, 2021

Internship Letter

Dear Sir/Ma'am,

This internship letter has been issued to give a confirmation that **Soham Deshpande** has been working on the final year project for this term with **Being Ambitious**.

He worked with us as a **GAN Developer**. He has successfully completed his project within a span of 30 hours which started from **20th September, 2021** and ended on **26th September, 2021.**

Regards,

Akshay Banda

Founder & CEO

Being Ambitious

BA

GSTIN NO: 24CBQPB7020N1ZJ

Date: 15th October, 2021

Internship Letter

Dear Sir/Ma'am,

This internship letter has been issued to give a confirmation that **Bhooyas Atul Kapadia** has been working on the final year project for this term with **Being Ambitious**.

He worked with us as a **GAN Developer**. He has successfully completed his project within a span of 30 hours which started from **20th September, 2021** and ended on **26th September, 2021.**

Regards,

Akshay Banda

Founder & CEO

Being Ambitious

ACKNOWLEDGEMENTS

We take this opportunity to express our profound gratitude and deep regards to our mentor Prof. Ashwini Rao and our industry mentor Mr. Akshay Banda for their exemplary guidance, monitoring and constant encouragement throughout the course of this project. We would also like to express our gratitude towards the NMIMS University for giving us this opportunity and platform to come up with a praiseworthy project.

Lastly, we would like to thank each and every person who directly or indirectly helped us in the completion of the project especially our Parents and Peers who supported us throughout the project.

TABLE OF CONTENTS

List of Figures

List of Tables

Abbreviations

Abbreviation	Description
GAN	Generative Adversarial Network
DCGAN	Deep Convolution Generative Adversarial Network
SRGAN	Super Resolution Generative Adversarial Network

Abstract

Games play a very important role in a person's life. Playing regularly not only refreshes the player, but also increases self-confidence and self-esteem. It also helps develop logic skills and improves hand-eye coordination and fine motor skills. The gaming industry has constantly tried to improve the characters in the game and make them better and better. However, it is still one of the longest and most expensive processes in game development. In most cases, a human artist creates a sketch / drawing of the character. Which then serves as a basic reference to digitize the character using expensive software. With the exponential increase in the gaming industry, the demand for game characters is also increasing exponentially. The game character is the object that gives the player the feeling and interaction around the game world. The game character is the most important for the success and growth of the game.

The goal of this project is to create a computer application that automatically creates different characters for the game. The characters in the game will be generated using Generative Adversarial Networks (GAN). A combination of Deep Convolutional Generative Adversarial Networks (DCGAN) and Super Resolution Generative Adversarial Networks (SRGAN) will be used by the users to generate different characters for the game.

1. Overview

1.1 Project specification

For this project, we will be making the use of Generative Adversarial Networks (GAN) which are a framework of Machine Learning. Provided with a training set, a GAN model learns to generate new data with the same or similar nature as the input training set, for example, a GAN trained on photographs can be used to generate new photos that look almost authentic to the observers and have many realistic characteristics, same as the original photograph. Though it was originally considered a form of generative model for unsupervised learning, it has also proven useful for supervised and reinforced learning. Adversarial models are also known as discriminators, they are simple classification models which can be used to classify data as original or fake. A GAN model typically consists of a generator and a discriminator. The generator generates data which is classified by the discriminator as “real” or “fake”. This concept is used to pass generated fake data as authentic fake data by refining it. GANs can be classified under different types based on the purpose they serve. We researched various GAN models to identify which best suit our project goals and identified DCGAN and SRGAN as the optimal choices. DCGAN, short for deep convolutional generative adversarial network, is an extension of the GAN architecture for using deep convolutional neural networks for both the generator and the discriminator models and training these in a way that results in the stable training of a generator model. SRGAN is a type of GAN which is used for increasing the resolution of a single image by four times.

To get a wider perspective on this project, we discussed the idea and the project with a few people in various fields of life, which included the CEO of “Being Ambitious”, the company for which the project is being developed, two players with heavy gaming experience in different genres of games, and a person with not much gaming experience. The result of this discussion essentially being that people have their own preferences of looks and features in a character that they like to use. Our project helps achieve this by giving the user an infinite number of choices that are randomized and can be picked based on the character that the user liked.

This project was given to us by “Being Ambitious” as an industrial project, which we were seeking to be in the Machine Learning Domain. It will be utilized by the company’s game development team. We aim to broaden our knowledge of this domain as we work on this project while also gaining experience of working with the industry and their requirements.

1.2 Literature Survey

The use of GAN in game development is so far limited to randomizing sprites / characters and generating new levels. Character randomization is performed using the Multi-Discriminator Generative Adversarial Network (MDGAN). It takes the character's color, shape, and animation data. The model then combines this data and generates a new animated image of the character. In doing so, the model even tries to eliminate the generated noise. For generating different levels, Procedural Content Generation via Machine Learning (PCGML) is used. Procedural Content Generation (PCG) is a method used to generate different levels in a game. To achieve this, a conditional integration generative adversarial network (CESAGAN) is used.

2. Analysis and Design

2.1 Requirement Analysis

This section will give a high-level overview into our application and will help us gain some insight into the project, it's features and it's working. The main purpose of this document is to describe the system's capabilities and functions that it will ultimately perform.

i. External Interface Requirements

A basic user interface with can help the user to interact with the models is required.

ii. System Features

The system should be able to generate new randomized game characters which can be implemented into games.

2.2 Feasibility Study

The feasibility of the project is defined as follows: -

i. Making the dataset of game characters
ii. Training of DCGAN and SRGAN.
iii. Integrating DCGAN and SRGAN.
iv. Deploying the app.

2.3 Design Development

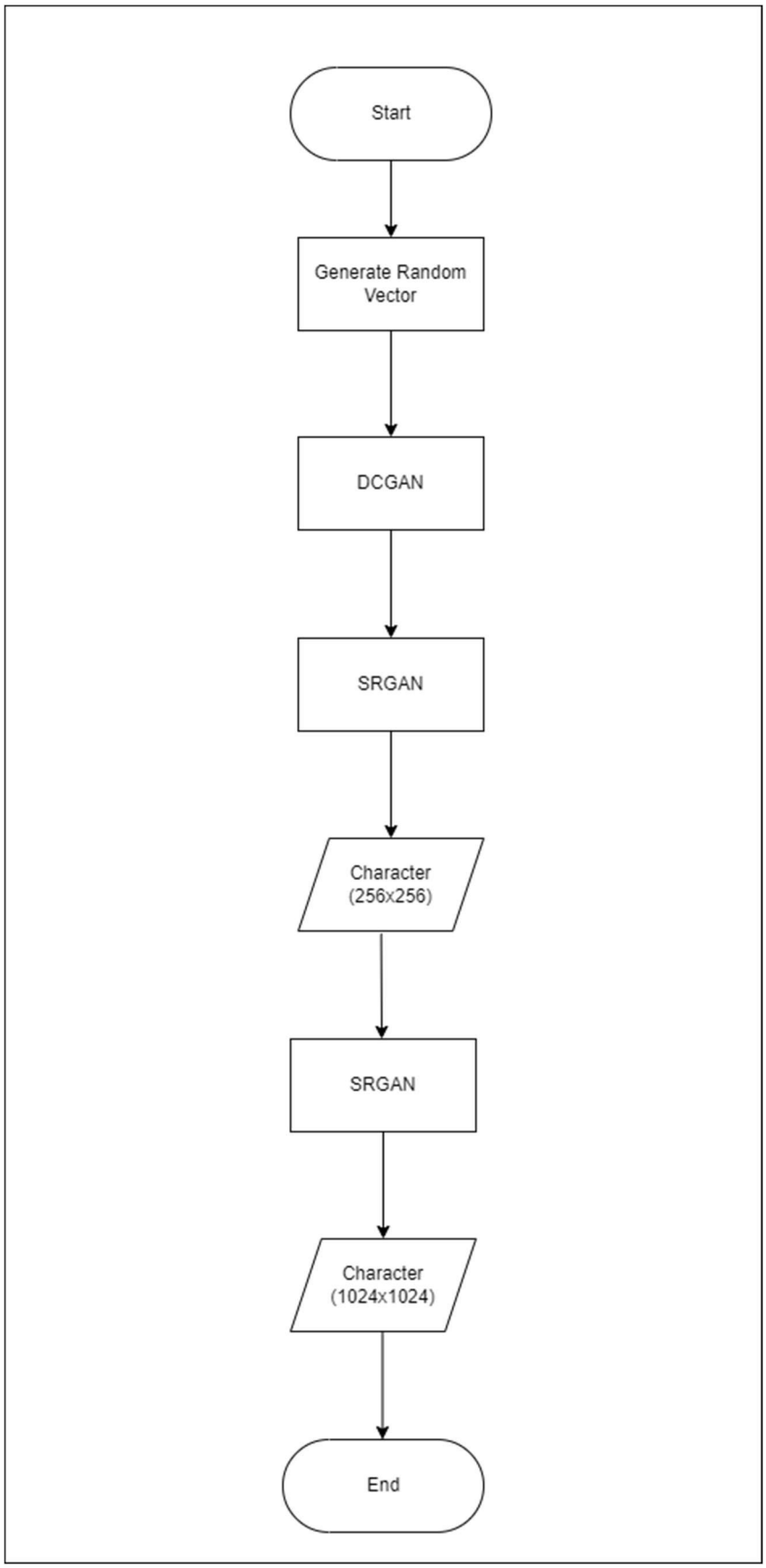

Fig 1. Flowchart of project.

The user will click on a button which generates a random 1x128 vector which is then be passed to DCGAN. The DCGAN processes and generates a 64x64 image. This 64x64 image is then passed on to SRGAN. SRGAN rescales the image to 256x256. The 256x256 imaged is again passed to SRGAN which is then enhanced to 1024x1024 image.

2.4 Technology and Software Details

The app is built completely in python. The models (DCGAN and SRGAN) are built using Keras, numpy and tensorflow libraries. The user interface of the application is also built-in python using kivymd library. Sublime text editor is used for writing the code. Windows 10 is the operating system on which the application is developed.

The dataset for training GAN's was generated manually. This was done to avoid copyright infringement. The GAN's were trained on Nvidia GeForce GTX 1650 Ti. The final application gives better performance when executed on a system with 8GB ram and dedicated GPU.

2.5 Project Planning

SN	Task	Sub Task	T_O	T_L	T_P	T_E	Start Date	End Date	Predecessor
1	Requirement Gathering	Research and Requirements gathering.	12	15	20	15.3	15/07	1/08	-
2	Literature Survey	Research on pre-existing apps, Survey Form	15	20	22	19.5	01/08	20/08	1

3	Algorithm Selection	Researching on ML Algorithms	8	13	18	13	21/08	03/09	2
4	Dataset Creation	Scrapping datasets from different websites and refining the dataset.	30	45	64	45.6	04/09	20/10	3
5	GAN Implementation	Creating GAN to generate the model	50	62	75	62.1	22/10	22/12	4
6	UI Development	Generating a basic screen to interact with model and connecting it to model	5	10	18	10.5	23/12	03/01	5
7	Testing	-	10	13	17	13.1	04/01	17/01	6
8	Feedback	Bug fixes, Implementing feedbacks from mentors	16	20	28	20.6	18/01	8/02	7
9	Deployment	-	10	15	22	15.3	08/02	23/02	8

		Total Days	**15 6**	**21 3**	**38 4**	**216**			

Table 1. Activity Table

Where, To = Most Optimistic (Best Case Scenario)
T_L = Most Likely (Normal Case Scenario)
T_p = Most Pessimistic (Worst Case Scenario)
T_e = Expected Duration

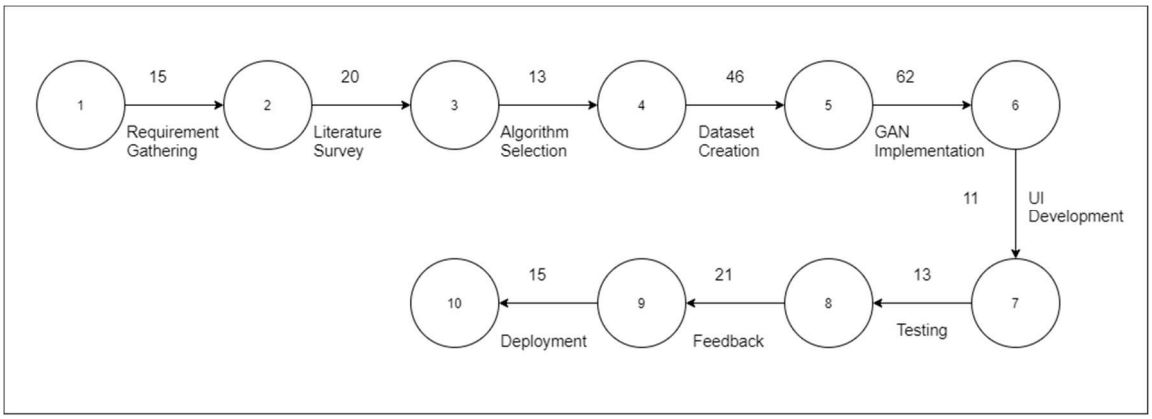

Fig 2. Pert Chart

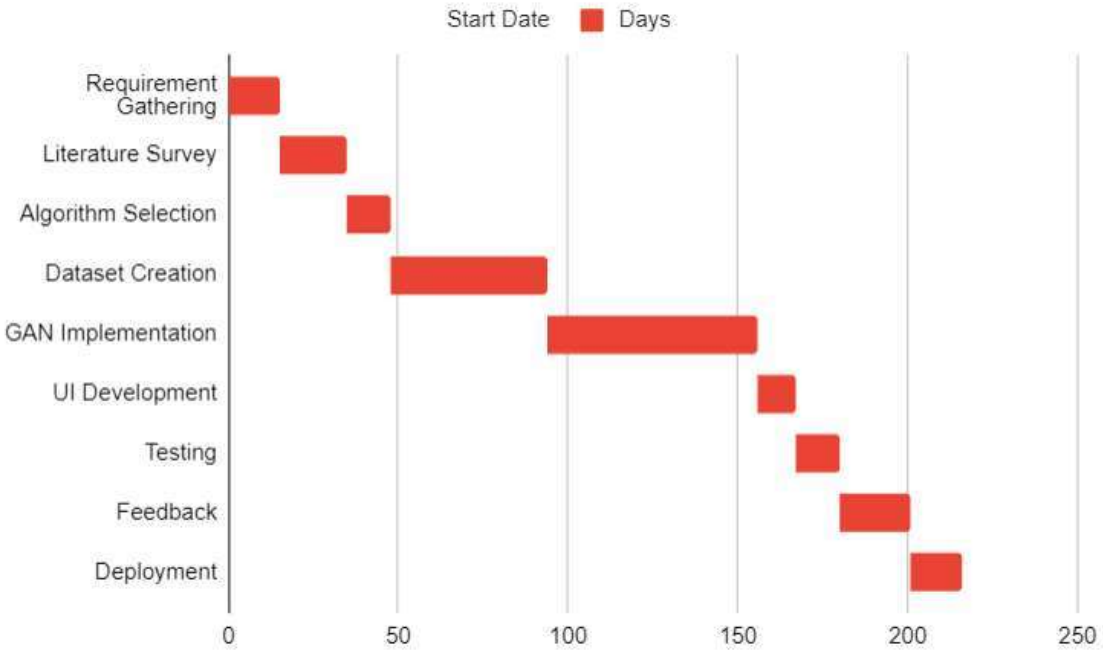

Fig 3. Gantt Chart

3. Project Description

3.1 Final Deliverable

The final deliverable uses a combination of 2 different Generative Adversarial Network models, namely Deep Convolutional Generative Adversarial Networks (DCGAN) and Super Resolution Generative Adversarial Networks (SRGAN). The main objective of DCGAN is to generate new images related or similar to the images that it has been trained on. The DCGAN receives a vector of fixed length containing random values between -1 and 1 as an input and returns a low-resolution image. The purpose of using DCGAN in this project is to take a random vector as input and return a low-resolution image of a character. SRGAN receives this low-resolution image as input and generates a 4 times higher resolution image. These two models of GAN therefore work in coordination to provide the final output image of a high resolution. DCGAN receives a random vector sized 1x128 and returns a 64x64 resolution image which is then be passed to an SRGAN which generates a 256x256 resolution image. This 256x256 image is again passed through SRGAN leading to 1024x1024 image.

3.2 Flow of Application

i. User opens the application.

ii. User clicks on the generate button.

iii. User generates characters using generate button until satisfied.

iv. Once satisfied the user clicks on the save button.

v. The generated character is saved as image in folder with current timestamp.

4. Project Implementation

4.1 DCGAN Training

The DCGAN was first trained on a dataset of 5000 game character images. These images were scrapped from different websites. After which we understood that these images cannot be used because of copyright infringement. We then had to manually create the dataset for training. DCGAN was then trained on this dataset.

4.2 SRGAN training

The SRGAN was trained on the same dataset as DCGAN. The SRGAN was trained in such a way that the input and the output shape of the GAN model is dynamic, i.e., if an image with resolution of 16x16 is given as input then the output image dimensions will be 64x64. This model is then used twice, first time to enhance the resolution of DCGAN generated image. Second time to enhance the image generated by SRGAN in the previous step.

4.3 User Interface

User interface is created using kivymd, a python library. Kivymd is used considering the future scope. Kivymd allows to generated cross-platform graphical user interface.

5. Screenshots

5.1 Training

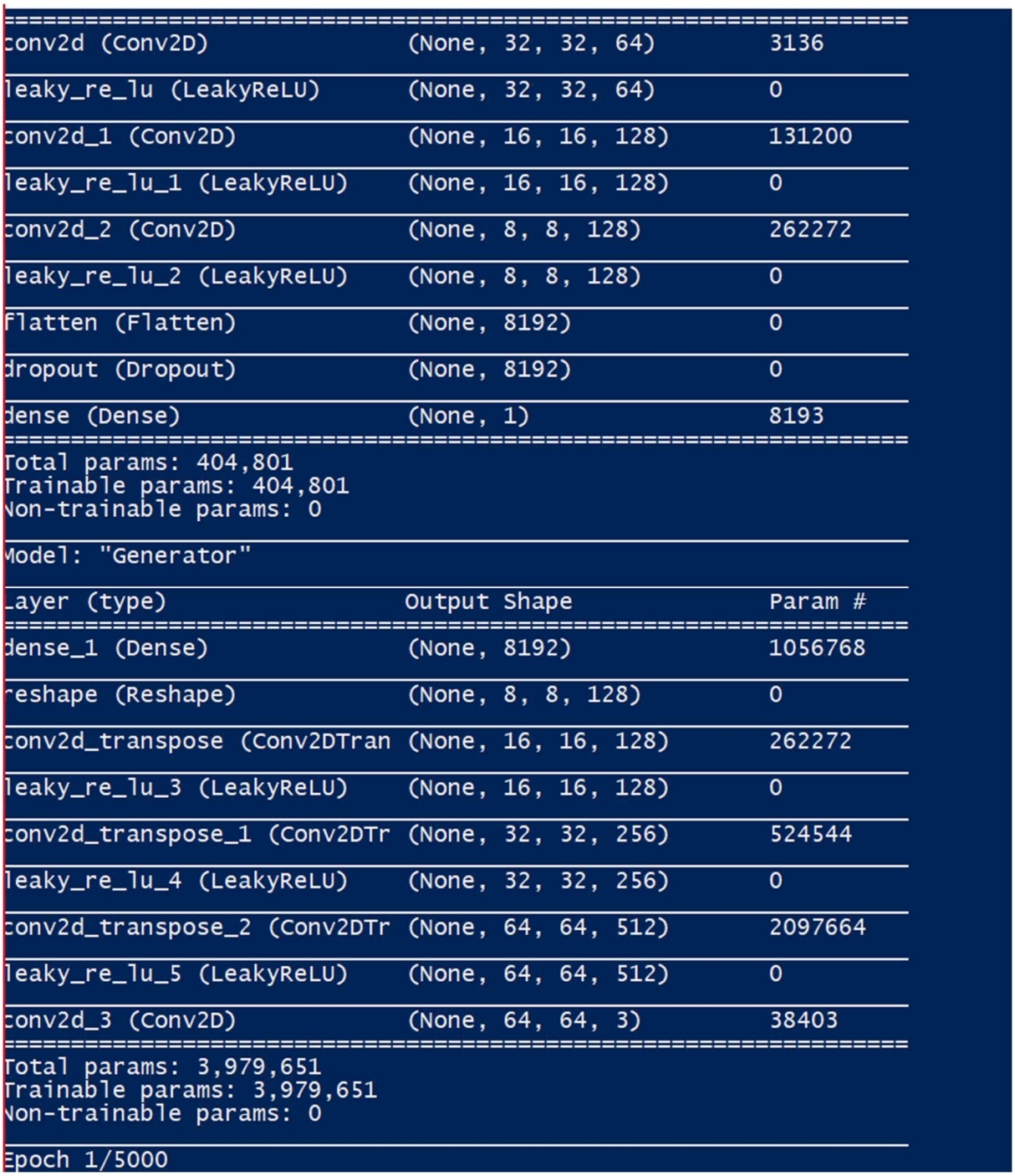

Fig 4. Training of Models

The above given screenshot is of DCGAN Training process. The image also shows the architecture of the Generator in DCGAN.

5.2 User Interface

Fig 5. User Interface Before Generation

The above given image is of user interface before game character is generated by the user. The save button is disabled at this point of time.

Fig 6. User Interface After Generation

The above image is of User interface after generation of characters. The character is displayed in 3 resolutions. Save button is now enabled.

Fig 7. User Interface of Regenerating Character

The above image is generating another character till the user is satisfied and saving the image.

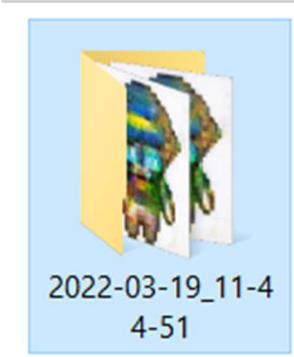

Fig 8. Saved Folder

The above image is of the saved folder with characters inside. The name of the folder is the timestamp.

Fig 9. Saved Characters

The above image depicts characters stored inside folder in png format.

6. Results and Discussions

After successful deployment of the app, it was found that all the modules were performing at with their expected capability. The user interface was consistent across multiple screen sizes (laptops, desktops). The animations were clear and working properly.

User interface was shown to a few users. The users were quite satisfied and happy with simple user interface. The main part they liked about the interface was font used since it gave a nostalgic feel. They were easily able to generate, save and locate the saved characters. Users were also quite satisfied with the animation used to display the characters.

7. Conclusion

7.1 Conclusion

With the exponential increase in games, there is a need for more and more game characters as well. The process of generating a game character is quite time consuming and expensive as well. AvatarGAN randomly generates a game character and makes this process easier.

7.2 Future Avenues

There is a lot of future work to be done in this field. Some of the aspects to work on in the near future are: -

i. Generating diversified characters.

ii. Ability to select a particular feature.

iii. Generate induvial features.

iv. Generate characters for specific game.

v. Improving the user interface.

vi. Making the application cross platform.

vii. Creating better and more extensive dataset.

viii. Adding Image filters and manipulation tools.

References

[1] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
[2] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681-4690).
[3] Torrado, R. R., Khalifa, A., Green, M. C., Justesen, N., Risi, S., & Togelius, J. (2020, August). Bootstrapping conditional gans for video game level generation. In *2020 IEEE Conference on Games (CoG)* (pp. 41-48). IEEE.
[4] Zhang, H., Fontaine, M., Hoover, A., Togelius, J., Dilkina, B., & Nikolaidis, S. (2020, October). Video game level repair via mixed integer linear programming. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (Vol. 16, No. 1, pp. 151-158).