

Computer Games Development

Project Report

Year IV

Caroline Percy

C00248791

24th of April 2023

**Contents**

[**Contents** 1](#_Toc133245838)

[Acknowledgements 2](#_Toc133245839)

[Project Abstract 2](#_Toc133245840)

[Project Introduction and/or Research Question 2](#_Toc133245841)

[Literature Review 4](#_Toc133245842)

[Ygor Rebouças Serpa et al 5](#_Toc133245843)

[Daniel Atorf et al 5](#_Toc133245844)

[Evaluation and Discussion 6](#_Toc133245845)

[Results 6](#_Toc133245846)

[Discussion 7](#_Toc133245847)

[Project Milestones 7](#_Toc133245848)

[Major Technical Achievements 7](#_Toc133245849)

[Project Review 8](#_Toc133245850)

[Conclusions 8](#_Toc133245851)

[Future Work 8](#_Toc133245852)

[References 10](#_Toc133245853)

# Acknowledgements

I would like to thank the following people who assisted in completing this project including;

Martin Harrigan, who provided me with guidance on the appropriate topic to cover and instructed me on how to manage my work on a weekly basis.

I would also like to thank SETU Carlow for use of their machines and knowledge of the topics I will discuss.

# Project Abstract

Video game creation started all the way back in 1958 and a new entertainment industry emerged. Hidden object games are a genre of video games, where you search for an object hidden in a busy photo.

Hidden object games were first defined in 2005, making it one of the later game genres to be realised. Hidden object games are some of the most intensive when it comes to 2D art assets – Not only does the background have to be fleshed out so objects can effortlessly blend into it, but each object must be drawn out separately as well. UI in hidden objects games are also often designed to match the environments it’s in as well. Overall, this requires a large budget for many assets used.

# Project Introduction and/or Research Question

I am going to attempt to make a hidden object game.

Hidden object games are video games that task the player to find an object or image within a busy landscape. They often have several objects that they ask you to find per level, as well as items called distractors that will naturally draw the eye but will not be what the game is asking for you to find.

Sometimes hidden object games implement a time limit to how long you can search for, to add some challenge, others ask you to replay levels, and they will often move objects around from places you originally found them in.

Hidden object games take a lot of principles from Wimmelbilderbücher, or Wimmelbooks, or hidden picture books as they’re all known as, Wimmelbilderbücher being “"teeming picture book" in German. Those books have a similar idea – a busy image where they ask you to find something. If you’ve ever opened a “Where’s Waldo” book, then you understand the fundamentals of hidden object games. Popular examples of hidden object games are Hidden Folks, Mystery Manor, and the Mystery Case Files series.

The genre has gained a lot of popularity due to the rise of casual gaming – these games can easily be downloaded onto your phone or tablet and played immediately, as well as how natural it is to tap or point at an object once it is found. Alternatively, they are often advertised through social medias like Facebook, where you can link up with your friends to share high scores and experiences playing these games.

The core of how to play these games is that your shown a large scene that is very busy with colour and objects, some you must find and some you don’t, and given a list of objects you must find in said scene. You would click on or tap the object when you find it in the scene, and the game would cross it off the list, continuing this loop until the list is empty of items to find, which means you completed the level.

A good hidden object game often spices up this formula. Some add other puzzle elements, so you are not stuck searching for objects over and over, others have expanded on this formula by adding different modes, like searching for an object by silhouette or only illuminating a part of the screen at a time. Some do both and more. A good hidden object game always tries to spice up the gameplay, so the player never feels like what they’re doing is repetitive or stale.

A difficult hidden object game would implement timers to give a sense of urgency, potentially decreasing the time if the player selects a wrong item. What makes a good hidden object game, and a difficult hidden object game is a fine line – both require challenge and constant change, so the players don’t get overly comfortable with their knowledge of the levels and objects. However, there is one distinctive way to make finding objects harder – purposely designing them to naturally blend into the background as much as possible. This creates the illusion that the object is something it’s not, and your eyes will glide over them if you aren’t paying close attention.

In terms of designing a level for a hidden object game, often you will need a theme. For example, you could be searching in a bathroom for a weighing scale or some soap. Then, you need distractors, maybe a bright red perfume bottle or a large laundry basket, neither of which are on your list. Anything added afterwards is optional, but a timer is always recommended to keep the pace up, and plenty of different kinds of objects on the list to keep the player engaged for around 5-10 minutes per level.

Hidden object games are not carried on gameplay alone, however. They are usually supported by the game's atmosphere and story. Often, the theme used is mystery, as it ties in well with the hunting of unknown objects. There are outliers of course, Hidden Folks is an example of this as it is a rather calmer experience, more like opening a book of “Where’s Waldo”, but I have seen a lot of hidden object games where you’re a detective searching a crime scene for clues, or a person stuck in a supernatural area, where unknown forces move objects around for you to find them again. This mystery also drives the player to keep playing – what's behind all these items everywhere? Is there something darker going on underneath the surface? This heavier atmosphere also ties in well to the art style – oftentimes, hidden object games use a realistic, but appealing art style to make the image charming enough to not hurt the eyes after searching it thoroughly, as well as simple enough that each object is recognisable once you focus on it.



Figure 1: Example of Bathroom Level from Mystery Manor: HD

A third-party software will deliver game assets dynamically throughout the game, rather than crafted beforehand. A third-party is someone or something that less directly involved with the project than the main people that are – the main people being myself and SETU Carlow, who I am writing this for.

The third party in question I will be using is called Pixabay, a website where people upload and share royalty free images, videos and music. They have an in-built system that allows coders such as myself to pull images from the website.

The term 'dynamically deliver' refers to the process of retrieving art from Pixabay through keywords the user will have inputted. The key word is filtered and then used to searched the website, returning the images results to the program.

The question asked during this paper are if this can be implemented correctly, would the player notice the difference? If they couldn’t, this could streamline the process of creating assets for games, as well as save a lot on resources within the project.

So, to clarify what the questions are:

Research Question 1: Can we make a hidden-object game that relies on a third-party to deliver the game assets dynamically?

Research Question 2: Will such a game be seamless to the player? Can we make levels theme-based?

To try out both these questions, I will be creating a Python-based hidden object game, with the third-party being REST API.

# Literature Review

Others have done work similar to what I’m setting out to do.

## Ygor Rebouças Serpa et al

Ygor Rebouças Serpa and Maria Andréia Formico Rodrigues attempted to implement machine learning to generate assets, specifically line art for a pixelated fighting game called *Trajes Fatais: Suits of Fate*.(Serpa, 2019) The reason they did this is because the amount of work gone into the art of video game titles has increased, alongside audience expectations.

Serpa and Rodrigues studied how much work their artists had to go on tedious manual labour, such as drawing the next frame of an animation and making sure the colour and shading for each drawing was correct. They wished to see if they could use image translation techniques and computer-generated imagery to automate some of this process.

They reviewed previous works that have attempted to tackle this issue – such as Isola et al (Serpa, 2019), who used Pix2Pix architecture to generate images that are indistinguishable from their non-generated ones.

Other examples of this asset generation are from Horsley et al and Xue et al. Horsley et al attempted to generate sprites from random noise, and Xue aimed to generate the next-frame of a sequence of animation frames. Besides these, most art generation consists of background assets, where errors are less likely to be noticed.

Serpa et al calculated which step took the longest for the artist in their production line, which was the painting the completed grey sprite, which could take longer than the rest of the steps combined. They aimed to get this generated by itself.

Using the same Pix2Pix architecture used by Isola et al, they altered the process so that the decoder in the architecture is used twice, one for the gray sprites, and one for colours.

Their results in the end were mixed. The gray sprites came out well, having only minor issues with the shading, but the algorithm was unable to colour the sprites incorrectly from the grayscale alone.

Though testing the algorithm in different ways, Serpa and Rodrigues concluded that their generated sprites would’ve been cleaner and more usable if their dataset the algorithm was trained on was larger and had more pose variety.

## Daniel Atorf et al

Another group, Atorf et al, wished to create a dynamic difficulty system in a hidden object game.(Atorf, 2021) They attempted this because they wanted the missions in their hidden object games to evolve and improve each player’s immersion, irrespective to the player’s proficiency.

Atorf et al discuss how important immersion is for a positive player experience; A good game should create immersion and a state of flow for the player. Since each player is different, the game needs to include some kind of adaptivity.

They go on to discuss their implementation – they use an asymmetric world, where vehicles and other objects are found on the map. Players are given of list of objects to find, and they gain points the more they find. To implement dynamic difficulty adjustment (DDA) into this world, they would change the map size, the number of vehicles as well as the types of vehicles, the number of distractor objects and the time available.

The participants of Atorf et al’s testing of their modified game went well, where they enjoyed the creativity and promise of the concept. They concluded with saying that further work of investigating what would make the game more fun would be in needed.

I aspire to blend Atorf et al’s and Serpa et al’s two findings– art personalisation through programming rather than traditional means.

# Evaluation and Discussion

The programme was generally received well by play testers, being impressed by the complexity of what I aimed to do.

## Results

The results of my project are a programme where a user can create a hidden object level in the level editor, and then load said level or any previously made levels in the level loader screen.



Figure 2: One of the Level Loader’s levels.



Figure 3: The Level Editor of the programme, after the word “test” is inputted.

## Discussion

This means that while I was not allocated the correct amount of time to cover everything I wished to in this project, I was able to go in the right direction with what I wished to achieve.

For example, the Nvidia instance insertion library would’ve been striking to get it implemented.

# Project Milestones

Figure 4: Work schedule by milestone.

Overall, I adhered to my established work schedule; however, a few tasks required more time than anticipated. I was able to catch up on my workload by effectively utilizing my available spare time.

# Major Technical Achievements

I put a lot of effort into the organisation and retrieval of the images from Pixabay – I aimed to make it as seamless as possible to get the assets and add them to the levels.

The retrieval of images is handled by Python’s request library, and I worked around it as the core. This included filtering the input into the requests, and organising the outputs once its retrieved, turning the json files to images that can be displayed. It seems simple on the surface, but there’s many checks you have to do;

Does the input have spaces? If so, remove them, because it will break the link. Does the output return no images? If so, throw up an error. Are the objects transparent, the right size, etc.

I also worked hard on the level editor’s save and load functionality, as I had to figure out a way to save the images gotten from Pixabay onto the local machine, and then be able to retrieve them dynamically. Getting the silhouette of the objects to find and display them in a list, many other issues that made themselves known over time.

# Project Review

The parts that went right are general concept of my programme – It generates art that can be used in a hidden object game. It can also generate a background image for a hidden object game level, and the user can make a hidden object level using only art from Pixabay.

However, there was a lot that I could have improved upon. For example, I not think of all the details of how I would generate a hidden object game level when I had begun this project, resulting in me struggling to find either libraries that could detect surfaces in an image, or an alternative solution.

Nvidia’s Instance Insertion library was the closest solution for the first option (Lee, 2019). However, its neural network being trained on people and cars, rather than objects, resulting in the library not being implemented.

I believe I did correctly choose the right technologies for this project, as python is very good with working with web development and data, as well as having many useful libraries that I called upon throughout my project.

# Conclusions

In response to the research questions put forward at the beginning of this study, a hidden-object game was developed that relies on third-party software to deliver dynamic game assets. The game was designed with levels, which facilitated by the use of a level editor.

However, further development is required to ensure a seamless experience for the player. Art generated by the game does not always blend seamlessly with the background, which can be noticeable to players. Therefore, additional work is necessary to achieve an indistinguishable experience.

# Future Work

If another student were to undertake this project in the future, I would recommend dedicating additional time to familiarizing themselves with the Nvidia Instance Insertion software. At the time of when I reviewed the software, I lacked a comprehensive understanding of neural networks, and therefore was unable to customize the program to align with my project's objectives in a timely manner. A more knowledgeable student may possess the requisite skills to undertake such a challenge.

Another aspect that the student could contemplate is saving the levels to a server, rather than locally, enabling the levels to load from any location and potentially expanding the game's accessibility as a website game.

# References

Atorf, D. (2021). *202108C040.pdf*. Retrieved from Web Archive: https://web.archive.org/web/20220116131039id\_/http://www.iadisportal.org/components/com\_booklibrary/ebooks/202108C040.pdf

Lee, D. (2019, April 11). *NVlabs/Instance Insertion*. Retrieved from Github: https://github.com/NVlabs/Instance\_Insertion

Serpa, Y. (2019, December 9). *Towards machine-learning assisted asset generation for games: A study on pixel art sprite sheets*. Retrieved from IEEE Xplore: https://ieeexplore.ieee.org/abstract/document/8924853