Five Nights at Cloney's: Technical Document

Introduction

This document serves as a detailed overview of the Technical aspects of Clone-Group-Two-7's game, and game development. This document contains prototype Objectives, description of technologies used, detailed documentation of features within the game and other testing and user documentation.

1. Overview

The team has cloned the popular indie-horror game: Five Nights At Freddy's 1. At its core, the game is mechanically a resource management game, and thematically a horror title. These two aspects of the game were the focus during the cloning process.

For more information on the original game and the clone game, please see the Game Design document [1].

2. Prototype Objectives and Requirements

We intended to clone the following aspects of FNAF 1:

- Office POV (doors and lights)
- Camera POV (navigation between cameras)
- Enemy Movement AI
- Enemy Attack AI
- Power Management System
- Sound design similar to original game (imperative for horror theme)
- All 7 Nights

Note: At first, we only intended to clone the custom night (night 7) of the original game, but upon further research, the team realised that all nights in the game use the custom night structure to determine the difficulty anyway.

3. Technologies and Methodologies Used

3.1. Hardware Used

Dell G15 5511 Laptops with an Intel® Core™ i7-11800H (11th Gen) processor and 16GB RAM

3.2. Software Frameworks Used

- Unity - Editor Version: 2020.3.30f1
- Github
- Discord
- Whatsapp
- Google Docs

3.3. Additional Templates and Resources Used

- Video on FNAF1 AI [2]
- Images sourced from Five Nights At Freddy's Wiki [3], NICEPNG [10], DeviantArt [11] and pngfind [12].
- Video on FNAF Sound Design [4]
- Sounds:
    - Jingles sourced from Sam Tesoro's Youtube video [5]
    - Many in-game sounds sourced from The Sounds Resource [9]
- Jumpscare videos sourced from various youtube videos [6,7]
- Text Font source from DaFont Free [8]

4. Feature Documentation

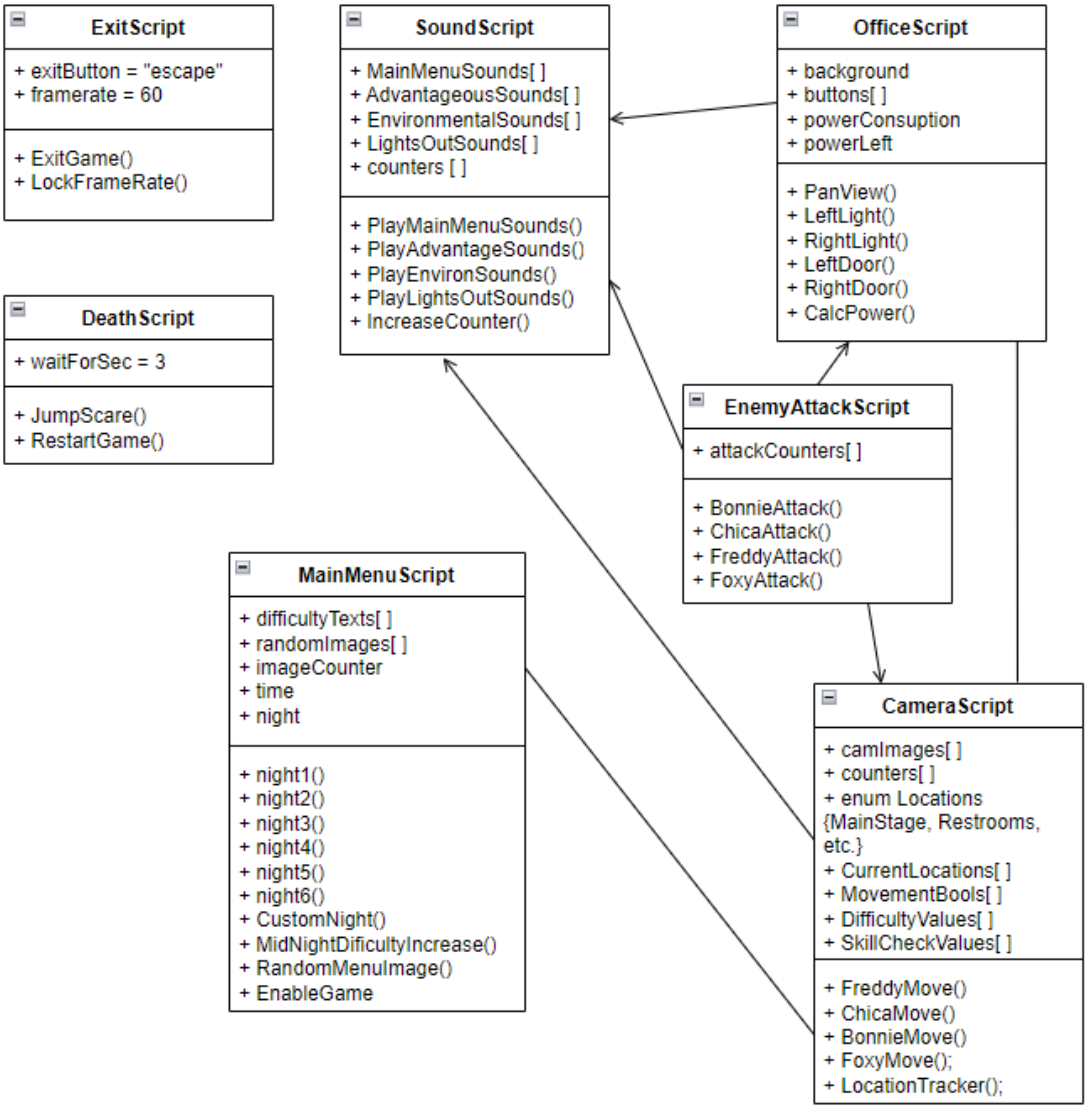


Figure 1: UML Class Diagram of Game Systems (based on scripts)

4.1. Feature: Office Gameplay

4.1.1. Office Point of View (Player POV)

Figure 2: Main Office View

In this view the player has the following options:

- Look left and right (panning the player's view)
- Turn on/off left or right Lights
- Close/Open left and/or right Doors
- Press Spacebar to enter or exit Camera POV

The player must keep an eye on the door lights and cameras to keep track of the animatronic's locations in order to **stay alive until 6AM**.

4.1.2. Doors

Figure 3: Closed Door

The player can open and close the two doors in the office, however this will increase the total power consumption (the power will drain faster).
The player should only close the doors when an animatronic is right outside the door, or is running towards the office.

4.1.3. Lights

Figure 4: Light on

The player needs to use the door lights to check the “camera system’s blindspots, that happen to be right outside the doors” for Bonnie and Chica. If they are there, the player should close the relevant door to stay alive. These lights also increase total power consumption.

4.1.4. Power System

Figure 5: Power Percentage indicator and Power Usage indicator

The power system consists of a single pool of “power” which starts at 100% and 5 drains that slowly decrease that power value. Each drain is indicated by a usage bar that is displayed to the player. The first drain is a constant drain throughout the night. Other drains are contextual, i.e. they only activate if the player **turns on lights, uses the cameras and/or closes the doors**. The more drains active at once, the larger the total power consumption, the faster the power value decreases.

4.1.5. Enemy Attack AI

4.1.5.1. Bonnie & Chica

After reaching the Office, Bonnie and Chica will wait outside the Left Door and Right Window respectively (visible in Office POV, not cameras) for about 15 seconds. If the doors are still open when their respective timers run out, they attack. If the correct doors are closed, they return to an earlier location on the map.

4.1.5.2. Freddy

Freddy will not go into the “Office location”, instead he will wait outside in the East Hall Corner (visible in Camera POV, not office) for about 20 seconds. If the right door is still open when his timer runs out, he attacks. If the right door is closed, he will return to the Main Stage. Freddy will also attempt to attack the player when the power goes out, see 4.1.5.4 Jumpscares for more on this.

4.1.5.3. Foxy

Foxy slowly creeps out of his stage at Pirate’s Cove, which is implemented as different states or “phases” in his code. When he reaches phase 4, he runs to the office, giving the player a little time to close the left door while they hear him running. If they close it before he reaches the office, he returns to phase 2. If the player was too slow, Foxy attacks.

In the original, there is an animation for Foxy running through the West Hall, but to save time during development for the clone, it was decided that an audio cue (Foxy’s running noise) was a sufficient warning to the player, since that is what most players tend to use anyway within the original (after they realise that the only camera the technically have to check, is the East Hall Corner for Freddy).

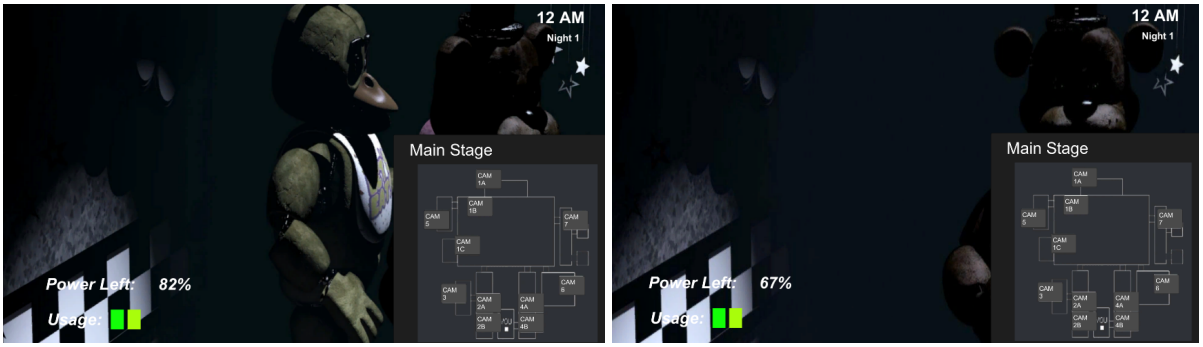
4.1.5.4. Jumpscares

Figure 6: Bonnie Jumpscare

When an animatronic successfully attacks the player, the game shifts to the relevant scene that houses the correct jumpscare video.

When the power goes out, Freddy will play music for a random amount of time, then walk in the darkness for a random amount of time, then the jumpscare will play.
The time in between gives the player a small chance to reach 6AM and win.

4.2. Feature: Cameras



Figures 7 & 8 : Images of the main stage camera view (with and without Chica)

The cameras are essential to the player to keep track of the animatronics locations, and were essential to facilitate the movement AI of the animatronics.

The cameras consist of 11 Raw Image GameObjects that are enabled/disabled by the navigation system. Each camera's image is changed based on the current location data of the animatronics (to display which animationics should appear on-screen when a certain camera is enabled).

4.2.1. Navigation

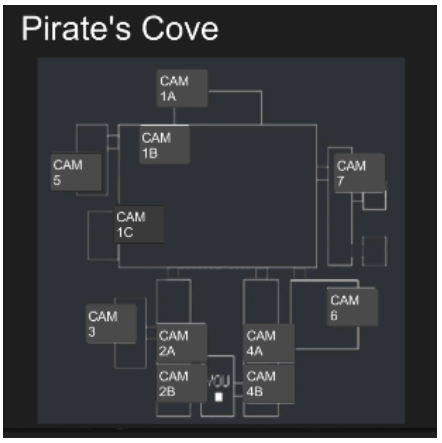


Figure 9: Image of Camera Navigation panel

The navigation panel in the camera view, gives the player a map of the pizzeria and the locations of every camera. Every camera is represented with a button that switches the view to that location (as well as the location text). This map, while functionally it is only a button-based navigation system between images, contributes to the horror tone of the game as well, since it makes the danger factor of the game (the distance between the animatronics and "YOU") more tangible to the player, since the navigation takes the form of a literal map.

4.2.2. Enemy Movement AI

The movement AI of the animatronics are intrinsically tied to the camera, since the images that display each location have to change with respect to the animatronics' locations.

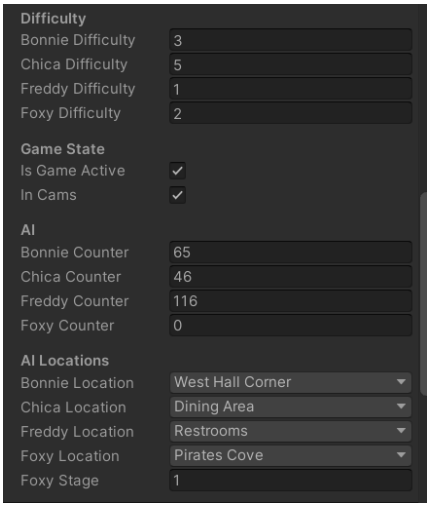


Figure 10: Inspector view of main movement AI variables during runtime

4.2.2.1. Bonnie & Chica

```
private void Bonnie()
{
    bonnieCounter++;
    if(bonnieCounter >= 298) //frame counter
    {
        bonnieCounter = 0; //reset counter

        //Movement Oppertunity
        Debug.Log("Bonnie Tries to Move");

        int bonnieSkillCheck = Random.Range(1,20);
        if(BonnieDifficulty >= bonnieSkillCheck)
        {
            //moves


            if(BonnieLocation == Location.MainStage)
            {
                BonnieLocation = Location.DiningArea;
                Debug.Log("Bonnie in Dining Area");
            }
            else if (BonnieLocation == Location.DiningArea)
            {
                BonnieLocation = Location.BackStage;
                Debug.Log("Bonnie in BackStage");
            }
            else if (BonnieLocation == Location.BackStage)
            {
```

```
private void Chica()
{
    chicaCounter++;
    if (chicaCounter >= 299) //frame counter
    {
        chicaCounter = 0; //reset counter

        //Movement Oppertunity
        Debug.Log("Chica Tries to Move");

        int chicaSkillCheck = Random.Range(1, 20);
        if (chicaDifficulty >= chicaSkillCheck)
        {
            Debug.Log("Chica Moved!!!");


            if ((ChicaLocation == Location.MainStage) && (BonnieLocation != Location.MainStage))
            {
                ChicaLocation = Location.DiningArea;
            }
            else if ((ChicaLocation == Location.DiningArea) && (FreddyLocation != Location.Restrooms))
            {
                ChicaLocation = Location.Restrooms;
            }
            else if (ChicaLocation == Location.Restrooms)
            {
```

Figures 11 & 12: Code Snippets from Bonnie & Chica Algorithms (Run in Update if isGameActive = true)

Like all the animatronics, Bonnie & Chica performs a skill-check (calls a random number from 1 to 20) on set intervals to see if they can move or not. Whether this skillcheck is successful depends on the respective difficulty-levels each animatronic is set to and a random number that they generate. The intervals are 4.97 seconds (298 frames)* and 4.98 seconds (299 frames)* for Bonnie and Chica respectively.

For example: Bonnie is set to level 3 (at the start of Night 2), during this time, every 4.97 seconds the Bonnie() algorithm generates a random number between 1 and 20. Let's assume the number generated is 2. Since Bonnie's Level is higher than (or equal to) the random skill-check number, Bonnie can move. Let's assume the number generated is 17. Since Bonnie's Level is lower than the random skill-check number, Bonnie can NOT move.

```
else if (BonnieLocation == Location.WestHallCorner)
{
    BonnieLocation = Location.OFFICE; //danger zone
    Debug.Log("Bonnie in OFFICE");
}
```

Figure 13: End of Bonnie's Path

While Bonnie moves in the left side of the map, and Chica on the right, they both start at the Main Stage and end outside the **office doors** (Location.OFFICE). Bonnie and Chica are the two animatronics that, before attacking the player, wait outside the doors. They are thus no longer viewable on the cameras, but are viewable in the office POV.

Bonnie moves from the main stage first always, then Chica, then Freddy. In the original version of the game, Chica is able to move first on Night 6, but this was not included in the clone, since it was a very rare occurrence in the original anyway. A consequence of this order of operations, is that if Bonnie's difficulty is set to 0, it effectively prevents Chica and Freddy from moving at all. This seems to be the case for the original game from night 1 to 5, since Bonny's difficulty value is always the highest of all the animatronics.

4.2.2.2. Freddy

Like Bonnie and Chica, Freddy also starts on the Main Stage, however Freddy follows a shorter path to get to the player and he does not, at any point, become visible in the office POV. He moves straight from the **East Hall Corner** to attacking the player, making him more dangerous to the player.

Freddy's movement opportunities are also governed by the skill-check-system that Bonnie and Chica use. The time interval between these skill-checks is 3.02 seconds (181 frames)* for Freddy.

4.2.2.4. Foxy

Foxy is the most different AI of the bunch. For the most part, he remains in Pirates Cove, but he has four states within that single camera location. The first three states is him creeping out of his stage, and the fourth stage is him running to the office (which gives the player a small amount of time to close the left door).

```
if (foxyCounter >= 301) //frame counter
{
    foxyCounter = 0; //reset counter

    //Movement Oppertunity
    Debug.Log("Foxy Tries to Move");

    int foxySkillCheck = Random.Range(1, 20);
    if (FoxyDifficulty >= foxySkillCheck)
    {
        Debug.Log("Foxy Moved!!!");

        FoxyStage++;
        if(FoxyStage == 4)
        {
            FoxyLocation = Location.OFFICE;
```

Figure 14: Foxy's skill-check

While the cameras are not on, Foxy uses his skill-check to advance stages, rather than to move, thus the same variable type and logic can be used to customise his difficulty. The time-interval between each skill-check is 5.01 seconds (301 frames)* for Foxy.

Foxy does not move as long as the cameras are on. While dialogue in the original game suggests that the player has to keep the camera on the “Pirate’s Cove” camera specifically, upon further inspection players quickly figured out that if any camera was open, Foxy would not move.

*Note on Movement Opportunities' Timing

The intervals between skill-checks for each animatronic are the same as the original game, and the game is locked at 60 frames per second (just like the original). The original game also used frames in many of its calculations, instead of real time. This was quite advantageous when creating the game in unity, since many of the calculations can be done in the Update() function.

4.3. Feature: Main Menu & Difficulty Settings

4.3.1. Night Selection

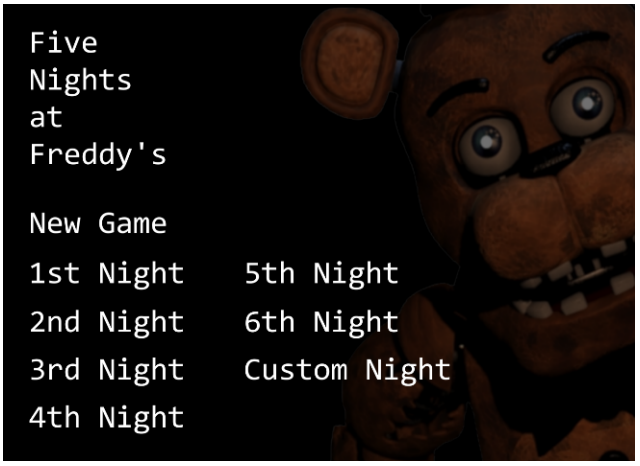


Figure 15: Night Selection (any night available for clone purposes)

The player can select any night that was available in the original game, all of which have similar difficulty scaling to the original game as well.

In the clone, all nights are available to the player from the start (rather than the linear progression offered by the original game’s “continue” button). This was done to make testing, reviewing and marking the game easier.

All other aspects of the main menu, such as the changing background image and the custom night menu, is the same among the clone and the original game.

Note: The Difficulty increases at set intervals during each night for nights 1 to 6, but NOT for the custom night. For example: Night 1 starts at difficulties: 0, 0, 0, 0 and ends at 0, 4, 2, 2 (as seen on the next page, those four values refer to Freddy, Bonnie, Chica and Foxy’s difficulty values respectively).

4.3.2. Custom Night

Figure 16: Difficulty customization

The custom night, as the name suggests, allows the player to customise each animatronic's AI difficulty settings. This mode helps with understanding the animatronics behaviour and patterns.
These are the difficulty values that the random number skill check values are compared to when the animatronics have a movement opportunity.

4.4. Feature: Sound

Sound is a very important part of games, but especially for HORROR games.

4.4.1. Ambient

There is ambient noise and music that contribute to the horror theme of the game. This sound is to make the player more paranoid than they really need to be, in order to increase the tension and stress that the player feels.

4.4.2. Warnings to player

There are noises, like animatronics' footsteps or Freddy's laugh that audibly indicate inflation to the player.

- Freddy's laugh indicates that he is on the move.
- Foxy's run warns the player to close the left door as quickly as possible.

Note: there are some sounds that, to new players, seem like they are warnings or contextual, like Foxy's laugh, but upon further play, players realise these sounds are random ambient sounds. These "fake" warnings contribute to the horror tone of the game, since it will trick new players into thinking they are missing something important since they do not know what certain sounds signify.

4.4.3. Jumpscares

One of the most iconic horror elements of the FNAF franchise is the constant threat of being jumpscared by one of the creepy animatronics. This is a startling and sudden scream that can surprise and frighten the player. These, along with the 6AM win screen, are the only

parts of the game that is contained in different scenes. This was done, simply because it lightened the coding workload for the team, since no extra code was needed to prevent other operations from happening during a jumpscare (which could lead to overlapping sounds or a second jumpscare playing).

5. Testing Documentation

System Testing facilitated by: Unity Editor
The team often tested for bugs during development.
Playtesting:
The first playtesting session on 18 August 2023, before sound or jumpscares were added to the game, presented the following feedback:

- Bug where power ran out too soon (before 3AM on one bar)
- Early nights more uneventful than expected, thus the animatronics must be forced to move at certain intervals
- Ambient Noise for a spooky atmosphere just as important as contextual or helpful sound for a horror game.

The second playtesting session on 23 August 2023 (from fnaf expert playtester), presented the following feedback:

- Running noise playing to often (only applicable to foxy)
- Chica kitchen noise not working correctly
- Foxy running noise not playing at correct times
- Add: Static sound, play it when bonnie is near (in office view)
- Power out sound playing at wrong times

6. User Guide / README

- Alternate checking the cameras and the door lights as often as possible. (Press SPACEBAR to enter/exit cameras)
- If Bonnie (the purple/blue bunny) shows up outside the left door, close it until he leaves.
- If Chica (the yellow chicken) shows up outside the right window, close the right door until she leaves.
- If Freddy (the brown bear) shows up in the “East Hall Corner” camera, close the right door immediately and keep it closed until he leaves.
- Keep an eye on Foxy in Pirates cove, if he is no longer there or if you hear him running, close the left door immediately and keep it closed until he returns to pirate cove.
- If the cameras are on, foxy wont move.
- The cameras, doors and lights all consume power.
- If you run out of power, the doors will open, and Freddy will come for you.
- SURVIVE UNTIL 6AM.
-

References

[1] M. Govind, J-F. Retief, M. Braam, 25 Aug 2023, “Five Nights at Cloney’s: Game Design Document”, Unpublished Internal Development Document

[2] Tech Rules, 17 Jan 2020, “Ruining FNaF by Dissecting the Animatronics' AI”, Youtube, Available Online at: https://www.youtube.com/watch?v=ujg0Y5IziiY&t=630s (last accessed: 15/08/2023)

[3] Fandom.com, 2014-2023 “Five Nights at Freddy's Wiki”, Available Online at: https://freddy-fazbears-pizza.fandom.com/wiki/Five_Nights_at_Freddy%27s_Wiki (last accessed: 23/08/2023)

[4] Scruffy, 18 Oct 2020, “How Audio Enhances the Horror of Five Nights at Freddy’s”, YouTube, Available Online at: https://www.youtube.com/watch?v=1yTIhtfgDwY (last accessed: 23/08/2023)

[5] Sam Tesoro, Oct 15 2016, “FNAF All 6AM Jingles (FNaF1-Sister Location)”, YouTube, Available Online at: https://www.youtube.com/watch?v=s3qkvfH2Egg (last accessed: 23/08/2023)

[6] gwn_mathe, 10 Sep 2016, “FNAF 1 Freddy Jumpscare”, YouTube, Available Online at: https://www.youtube.com/watch?v=NigLg8fQHF0 (last accessed: 22/08/2023)

[7] Gemininds, 31 May 2021, “FNAF 1: Foxy Jumpscare”, YouTube, Available Online at: https://www.youtube.com/watch?v=m0DQft9j3SY (last accessed: 22/08/2023)

[8] DaFont Free, 7 Feb 2021, “Consolas”, Available Online at: https://www.dafontfree.io/download/consolas/#google_vignette (last accessed: 14/08/2023)

[9] The Sounds Resource, “FNAF1 sound”, Available Online at: https://www.sounds-resource.com/pc_computer/fivenightsatfreddys/sound/4561/ (last accessed: 18/08/2023)

[10] NICEPNG, “Withered Freddy- Five Nights at Freddy’s Withered Freddy”, Available Online at: https://www.nicepng.com/downpng/u2q8o0o0w7u2o0q8_withered-freddy-five-nights-at-freddys-withered-freddy/ (last accessed: 18/08/2023)

[11] DeviantArt, 23 Nov 2017, “FNaF 1 Chica Icon”, Available Online at: https://www.deviantart.com/the-structure/art/FNaF-1-Chica-Icon-716558546 (last accessed: 18/08/2023)

[12] pngfind, “Bonnie Fnaf png”, Available Online at: https://www.pngfind.com/download/oRwbhx_bonnie-fnaf-png-bonnie-fnaf-1-png-transparent/ (last accessed: 18/08/2023)