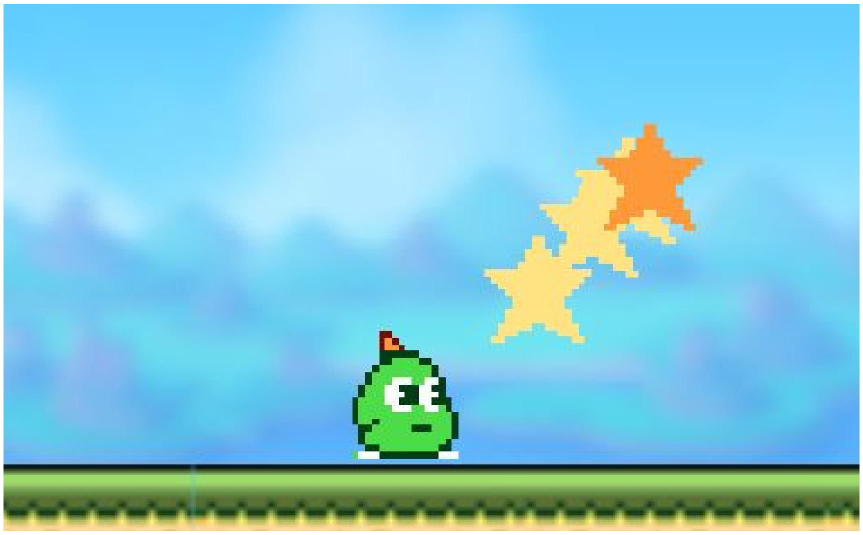
Gimmick 2.0

Robert McGuigan, Monica Pineda, Bilal Saleem

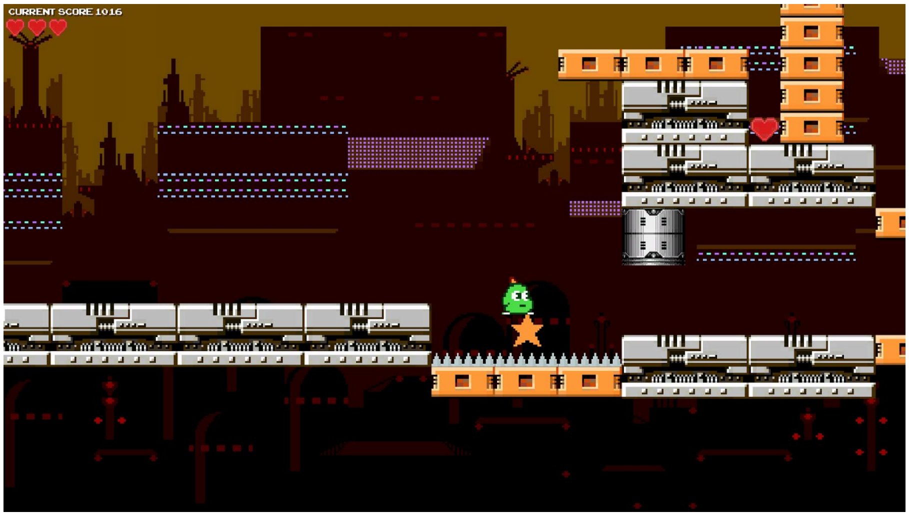
I. Introduction

Our Proposal

After around seven weeks of effort, our group project is complete. The project we chose to make was a 2D platforming game. The original idea was to make it a game similar to Mario. However, to make the project a bit more unique, we decided to make the game similar to 1990's NES game called Gimmick.



Gimmick is 2D platforming game with some added features. The main added feature is that the playable character, Gimmick, has the ability to throw a star at enemies. The star deals damage to the enemies. In addition to that, Gimmick has the ability to jump on top of the star in order to maneuver otherwise impossible obstacles.



One feature that was not in the original Gimmick that we added was that Gimmick could not only deal damage to enemies by throwing a star, but also by jumping on top of them.

Research

We worked on this project in the Unity 3D game engine. This game engine supports both 2D and 3D games. To develop in this platform, you have the option of using either Javascript or C#. Our group chose C# as our language of choice as that was what the tutorial we followed was using.

On the topic of the tutorial, we followed a Udemy course called “Learn To Code By Making a 2D Platformer in Unity”. This course had 4.7 rating and had around 13 hours of video content. This was a great starting point for us and really helped us understand some of the basics of game development in Unity. In addition to this we made use of a variety of other resources—the primary being Unity Docs.

Development Process

We didn’t encounter too many issues working together. The only issue that really came up was how we were going to collaborate. Naturally, we chose Github as the means of working together on the project. However, due to the way Unity works, this made merge conflicts likely. In fact, due to our first merge conflict we decided to switch to Unity’s built-in collaboration tool. Using Unity’s Collaborate tool we work able to work seamlessly and didn’t encounter any further merge conflicts.

II. <u>Program Usage</u>

As mentioned our project was to make a game. Gimmick must traverse each level in order to beat the game. In order to beat the levels, Gimmick generally move from a left endpoint to a right endpoint; all the while encountering various obstacles and foes. There are four levels in the game, they are:

- Factory Level
- Seaside Level
- Cave Level
- Forest Level

In order to make grading easier, all four levels are unlocked and the user is free to skip around and select whichever they wish to play.

In each level, there are no lives or a checkpoint system. However, Gimmick does maintain a certain amount of health. The depletion of which will result in a menu popping up asking if the user wishes to retry the level (among other things).

III. Controls

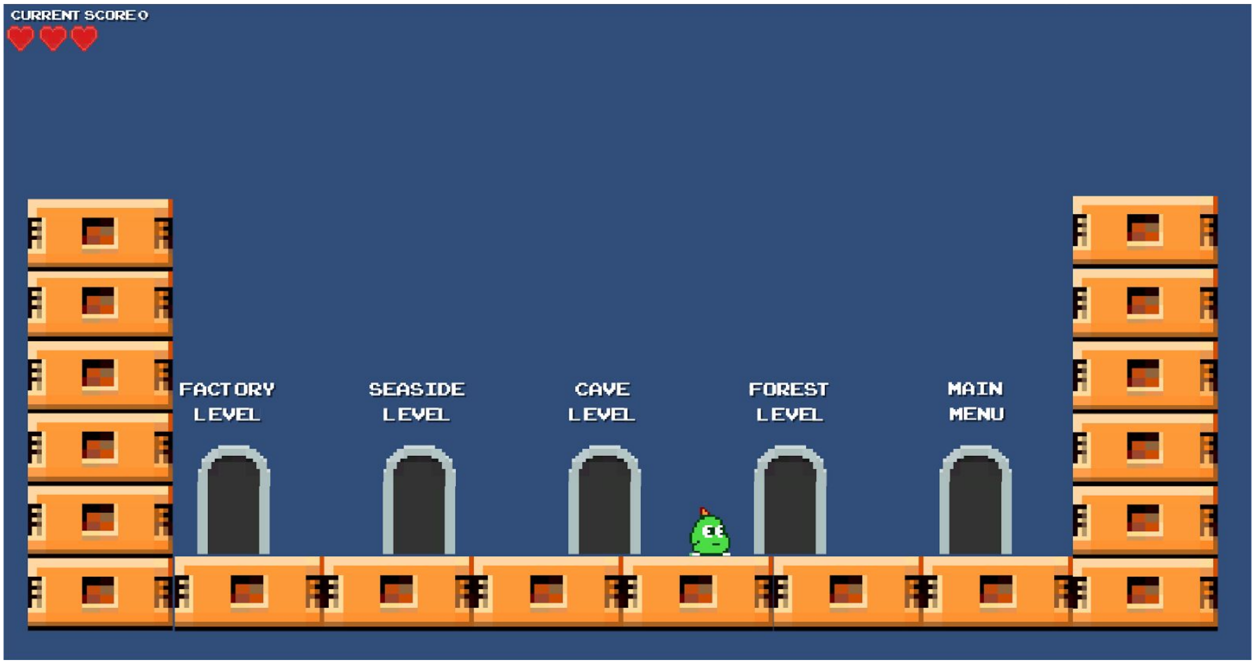
The user is able to make use of the keyboard and mouse to control various aspects of the game. The main thing controlled is the playable character, Gimmick.

In-game
a-d or **arrow keys** to move around
j to throw star
Space to jump
Esc to bring up pause menu

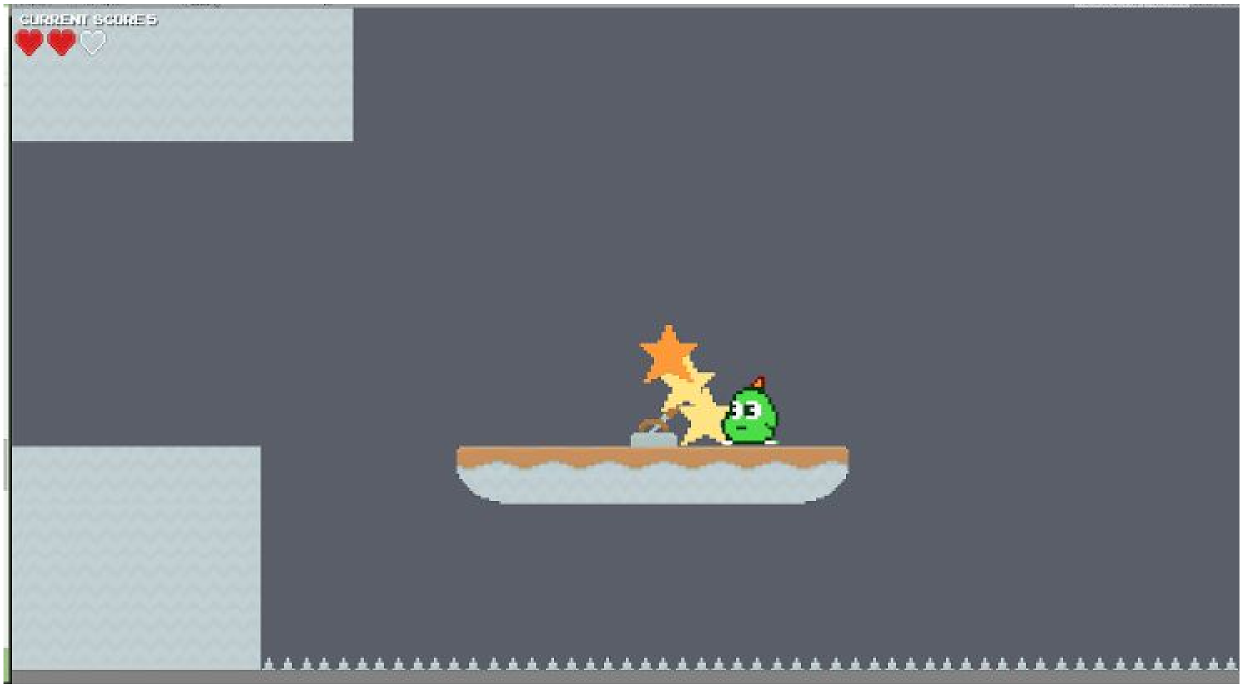
Pause Menu
Mouse click or **arrow keys** to select a button
Esc or r to resume the game
l to go to the level select
q to quit to the main menu

Main Menu
Mouse click to select a button
Arrow keys + Enter to browse and select from menu options

Other
W to enter a level from Level Select scene



J to throw star at lever to activate it and set moving platform in motion



IV. Play-through Instructions

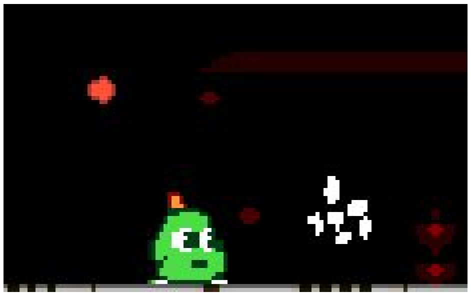
There are two ways to play the game:
(1) If you have Unity installed, you can play it through Unity with the provided project folder. To do this, open Unity and then select the provided GimmickV2 project folder. If you are not on the start menu, then go to the Scenes folder within Assets folder and select the Menu scene. Once you are on the start menu scene, click on the triangular play button to begin. Right-clicking on the play button stops the game.

(2) You can play the provided .exe file. You do not need to download anything for this option. Simply follow the directions below:

1. Extract the contents from the zip file and run the executable called Gimmick 2.0. It is found in the folder named Mr.Gimmick 2.0. Please do not remove the executable from its folder as it need to access the surrounding files. Press "Play" to begin the game.
2. Choose "New Game" from the main menu. You will be dropped into the Level Select scene. In this scene move in front of the first door from the left and press "w" to enter the level.
3. Begin moving generally towards the right. Combat enemies and traverse obstacles. Please note that some obstacles may seem impossible but can be traversed with the help of the star (which you can jump on and off of). If you die before completing the level, you have the option of retrying the level from the beginning. You can also choose to go the main menu or to the Level Select scene. You can also open the pause menu by hitting

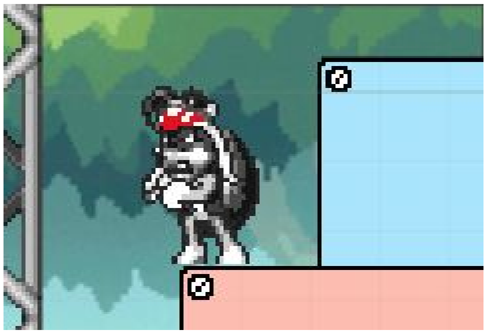
"Esc" during gameplay. During which you can return to the main menu, go to the level select or resume the game.

4. Reach the point in the game where you see a flag. Go and touch that flag and wait to be dropped in the Level Select scene again.



5. Go to the next level's door and press "w" to start it. Repeat steps 3-5 for levels 2 and 3.

6. Once you reach level 4 (the Forest Level), you will follow step 3 as you have for the other levels. However, there won't be a flag at the end of the level, rather there will be a boss enemy at the end. Fight and defeat the boss enemy by throwing him into the spikes three times. Once you have done so, you have beaten the game! Enjoy the credits!



All C# code our team wrote for this project can be found in the "/Gimmickv2/Assets/Scripts" directory

V. Software Systems - Unity Overview

Our project is completed entirely through the Unity game engine. Here is how you can install Unity and setup our project's development environment:

1. Download and install the latest version of Unity at https://unity3d.com/
2. If asked to make an account, just make a free one for personal use
3. Extract submitted zip files. Move project files in folder Mr.Gimmick 2.0 to desired location. This folder contains our project files for the game with all of the levels unlocked.
4. Open folder from previous step in Unity. You may now see the development environment that we had before we built the game into an executable.

Understanding the Layout and functions of Unity

- **Hierarchy** This shows you what game objects are in the current scene.
- **Inspector** This displays the details of a highlighted game object. It will show attached scripts, components used, etc.
- **Scene** This is where the front-end work like manipulating position and size is done.
- **Game** This is where you can run the current build of the game without having to create an executable.
- **Animator** This allows you to create animation relationships between the various animations on an object.
- **Animation** This is where you can look at and modify an individual animation.
- **Project** This is the entire set of files for the game. Inside of it our files are further split into folders. For example we have a Sprites and Scenes folders.
- **Console** This is used primarily for debugging and any output to here doesn't make into the game once it's built.

Most of what is done graphically in Unity is done so with sprites. The sprite will be used to form various animations such as walking, jumping and so on. The relationships between the various animations is then controlled via an Animator.

Which animation is played is then determined by variables that defined those relationships. For example, if there is a variable called "isGrounded" in one of the relationships that is true when Gimmick is on the ground, and it is set false at runtime, then Gimmick's jump animation will be played.

VI. Additional Tools, APIs, Libraries, etc.

Unity Collaborate
As mentioned this was the tool we used to work together on this game. It allows some basic version control functionality. You are able to create a new commit as well as go back to previous commits. You are able to pull in changes from your fellow team members. Also, each commit comes with a description (should it be provided by the developer) and a commit number. We had around 500 commits during the lifetime of our project.

Photoshop and Gimp
In order to recognize an individual sprite, Unity requires the background of the sprite be transparent. So these tools were primarily to get a transparent background for our sprite sheets. We used online tutorials to learn how to use these softwares.

C# programming language
At the start of this project only one of our group members had prior experience with C#. For the other two, we learned through our Udemy course as well as online tutorials.

VII. Teammate Contributions

Robert McGuigan

Robert contributed to the following global and level-specific game objects:

- Gimmick: The playable character. Robert designed the physics governing Gimmick's movement, and resolved player input to affect Gimmick's actions on screen.
- Star: The star that Gimmick throws at enemies. Robert designed the star's physics, how it collides with other objects, and made it possible for Gimmick to ride the star.
- Camera: Robert implemented the way in which the camera follows Gimmick throughout gameplay.
- Flag: The flag that is the goal of each stage, marking that stage as "completed" and returning the player to the level select screen.
- Blob: A small, black enemy that always attempts to run toward Gimmick.
- Bullet: Can be fired by various enemies, doing damage to Gimmick.
- Robot: A blue robot that searches for Gimmick, and, upon finding him, chases him around the stage while firing bullets at him.
- Turret: Regularly fires bullets at Gimmick.
- Conveyor Belt: Moves Gimmick right or left while he is standing on it.
- Crusher: Moves up and down, and kills Gimmick if he is caught between the crusher and the ground.
- Boss (and Blue Shells): The boss runs and jumps around its enclosure, occasionally throwing explosive blue shells at Gimmick. The star causes the boss to ricochet away from it, and Gimmick must use this to maneuver the boss into the spikes on the right.

In addition, Robert designed and created the factory level and the boss battle area of the forest level, as well as the credits screen. Robert also implemented the slowly-scrolling, parallax-inducing backgrounds in each level.

Monica Pineda

Monica contributed to both Global and level-specific game prefabs. She created the following prefabs:

- Health Mushroom: these were a collectable item that increases the player's health.
- Crab: these are crabs that are either stationary or patrolling that can cause damage to the player.
- Throwing Crabs: these are crabs that are either stationary or patrolling between two points that throw shells at the player when the player is in range to cause damage to the player.
- Cloud Elevators: these are cloud platforms that can move between two points to carry the player from one location to another.
- Tentacles: these are octopus tentacles that will damage the player when they get two close.
- Spinning wands: these are swinging arms of water droplets or fireballs that damage the player if they touch him.

- Flying seagull: these are seagulls that fly at the player when they come in range to cause damage.

In addition, Monica created the Main Menu, Pause, Game over, options menus, and designed and implemented the seaside and forest levels.

Bilal Saleem

Bilal contributed to both Global and level-specific game prefabs. He created the following prefabs:

- Coins: these were a collectible item that increases your score for a level.
- Moving lever platform: this was a platform that initially started off still. However, once you activate the lever on the platform (by throwing a star at it) the platform is set in motion.
- Spikes: This was an item where if a player lands on it he will take damage.
- Stalagmite/Stalactite: This is very similar to spikes. However, these have a different sprite.
- Jumping fish: These are fish that jump out of the water and damage Gimmick if he makes contact with one. These were created as more of an obstacle as opposed to an enemy.
- Launcher: This item was created to help Gimmick overcome obstacles. It launches Gimmick into the air once Gimmick jumps on it.
- Bear: This was an enemy created that patrols an area moving left and right. Once the bear sees Gimmick and he is in striking range, the bear takes a swipe at Gimmick, dealing damage and pushing Gimmick back.
- Bonus Heart: This item, once collected, increases Gimmick's health and the total damage that he can take.

In addition to this, Bilal created saving and displaying of the high scores. Also, he designed and implemented the third level in the game (Cave Level). Finally, Bilal also wrote most of the final report you are reading.

VIII. <u>Conclusion</u>

We have all learned a lot during the course of this project. We have seen how far game development has come. In the past, it would take many developers and perhaps years of development to put out a single 2D game. However, with the progress of time, it is now possible to create such games with much less effort. A 2D game, for someone who is an experienced game dev, could take a matter of a couple of long weeks to build.

Of course for someone brand new to the industry, it takes a little longer than that. You have to learn the basics first and then you have to learn what your particular project requires of you. For most of our team members we were new to both the development environment and the programming language used. So the learning curve was slightly steeper.

Despite this, I think we all enjoyed working on the project and trying something we had never done before. Up to this point, most of our development experience has been working as individuals, but with this project we got to work with multiple people on the same thing. So, it was not only a new experience in terms of what we created but also in how we created it.

IX. Credits

Art:

- Gimmick: *Mr. Gimmick* (Sunsoft)
- Blob: *Mr. Gimmick* (Sunsoft)
- Factory Ground: *Mega Man 6* (Capcom)
- Factory Platform: *Mega Man 6* (Capcom)
- Conveyor Belt: *Mega Man 2* (Capcom)
- Robot: *Battletoads and Double Dragon* (NES) (Rare)
- Bullet: *StarTropics* (Nintendo)
- Flag: *R.C. Pro-Am* (Rare)
- Crusher: *Sonic CD* (Sega)
- Turret: *Xevious* (Namco)
- Factory Background: *Sonic The Hedgehog* (Sega)
- Squirrel: *Zero the Kamikaze Squirrel* (Sunsoft)
- Blue Shell: *Mario Kart Super Circuit* (Nintendo)
- Blue Shell Explosion: *Bionic Commando* (Capcom)
- Boss Spikes: *Super Mario Bros 2* (Nintendo)
- Boss Platforms: *Super Mario Bros 3* (Nintendo)
- Boss Wall: *Sonic the Hedgehog* (Sega)
- Credits Ground: *Mike Tyson's Punch-Out!!* (Nintendo)
- Credits Background: *Final Fight* (SNES) (Capcom)
- Skateboard: *Town & Country Surf Design: Wood & Water Rage* (LJN)
- Crab:*Mario & Luigi: Dream Team (AlphaDream, Good-Feel)*
- Mushroom: *Super Mario All-Stars (Nintendo)*
- Flying Fish: *Trickster Online Revolution (Privatia Co.,Ltd. & Pumpchi studio)*
- Clouds: *Kirby Super Star / Kirby's Fun Pak (HAL)*
- Forest background: *Spellsword (Everplay Interactive)*
- Seaside Background: *Sonic Lost World* (sega)
- Forest Ground: *Plee the Bear(*OpenGameArt.ORG)
- Palm Trees: *Fairy Farm* (Game Garden)
- Jumping Seagull: *MFO Wiki* (http://pl.my-fantasy-online.wikia.com/wiki/Plik:Mewa.png)
- Flying Seagull:Siput Scuba (https://www.behance.net/gallery/15084287/Siput-Scuba-Website-Character-Illustrations)
- Tentacle: *Angevon's RO Sprites Archive* (http://rosprites.blogspot.com/2010/12/monsters-octopus-kraken-and-much-more.html)
- Water droplets for the wands: *Deviant* Art (https://ryky.deviantart.com/art/Crystal-ball-428289208)
- Fire Bar Concept: *Super Mario Bros* (Nintendo)

- Fire Balls for the Fire Bars: *GhostBusters* (sega)
- Forest Level Tree: *Zombie Island* (Vizor Interactive) (https://www.spriters-resource.com/pc_computer/zombieisland/sheet/80317/)
- Mr.Gimmick : *NewGrounds* (http://www.newgrounds.com/art/view/emrox/mr-gimmick)
- Mr.Gimmick title art: *Mr. Gimmick* (Sunsoft)
- Turtle: *Super Mario Bros* (Nintendo)
- Cave Water: *OpenGameArt*
- Bear: *Curses N' Chaos* (Tribute Games)
- Forest in Cave Level: *Ben Bakker*
- Launcher: Learn To Code By Making a 2D Platformer (Udemy)
- Moving Lever Platform: Learn To Code By Making a 2D Platformer (Udemy)
- Stalagmite: Learn To Code By Making a 2D Platformer (Udemy)
- Stalactite: Learn To Code By Making a 2D Platformer (Udemy)
- Coins: Learn To Code By Making a 2D Platformer (Udemy)
- Cave Level Floors and Ceiling: Learn To Code By Making a 2D Platformer (Udemy)
- Bonus Heart: Learn To Code By Making a 2D Platformer (Udemy)
- Health Indicator: Learn To Code By Making a 2D Platformer (Udemy)
- Jumping Fish: Learn To Code By Making a 2D Platformer (Udemy)

Sound:

- Factory Stage Music: *Journey to Silius* (Sunsoft)
- Boss Music: Batman: *The Video Game* (Sunsoft)
- Forest Stage Music: *Ufouria: The Saga* (Sunsoft)
- Seaside Stage Music: *Guinea Pig Hero* (Trevor Lentz)
- Cave Stage Music: *Battletoads* (Arctic Cavern)
- Main Menu Stage Music: *Mr. Gimmick* (Sunsoft)
- Level Select Stage Music: *Mr. Gimmick* (Sunsoft)
- Game Over: *Super Mario World* (Nintendo)
- Squirrel Death: *Contra* (Konami)
- Blue Shell Explosion: *Gun-Nac* (Compile)
- Victory Theme: *Battletoads* (Rare)
- Robot Shoot: *Gun-Nac* (Compile)
- Robot Alert: *Mr. Gimmick* (Sunsoft)
- Star Hit: *Ufouria: The Saga* (Sunsoft)
- Credits Music: *Rad Racer II* (Square)
- Hurt sound: Learn To Code By Making a 2D Platformer (Udemy)
- Jump Sound: Learn To Code By Making a 2D Platformer (Udemy, from Super Mario World)
- Explosion: Learn To Code By Making a 2D Platformer (Udemy)
- Coin Sound : Learn To Code By Making a 2D Platformer (Udemy)
- 1-up sound: Learn To Code By Making a 2D Platformer (Udemy, from Super Mario World)

Other/Special Thanks:

- Udemy: https://www.udemy.com/unity2dplatformer/
- NES Stereo Remixes: *WiiGuy's 8BitStereo* (https://www.youtube.com/user/Wiiguy309)