

# Bounce: A 2D Platforming Collectathon Video Game

Jesse F. Kolb

CIS 4914 Senior Project

Advisor: Joshua Fox, *email:* [joshuafox@ufl.edu](mailto:joshuafox@ufl.edu)

Department of CISE

University of Florida, Gainesville, FL 32611

Date of Talk: 3 Dec 2018

## **Abstract**

The platforming collectathon genre in video games, characterized by jumping from platform to platform to collect items, its ease of entry for casual players, responsive controls, and emphasis on exploration, has gone from reaching peak popularity in the 1990s to near obscurity in modern day. Due to the gaming industry's focus shift towards realistic visuals running on top-of-the-line hardware, many games have lost core design elements that give them lasting satisfaction. The introduction of distributable cross-platform game engines, such as Unity, pioneered a new wave of independent game developers, and with it, the revival of 2D platform games. However, aforementioned games still lack defining characteristics of a platforming collectathon.

In response to this situation, we have developed an accessible game based on easy-to-learn, difficult-to-master platforming sections, exploration-and-puzzle-driven collectables, and a diverse moveset. Scattered across platforming challenges, the main collectables consist of keys and power-ups, both of which act as tools, in different ways, to access certain areas. It is up to the player to decide which keys to collect, making it accessible for casual players, while providing lasting challenge for seasoned players. The power-ups diversify the moveset, encouraging players to learn by progressing and exploring. By providing increasingly difficult platforming challenges, and encouraging player-driven discovery, *Bounce* revives key characteristics of classic game design.

## **1. Introduction**

The expanding disconnect of platforming video games from their original roots results in a significant number of players yearning for classic level design. As game development becomes more accessible throughout the years, due to streamlined engines such as Unity, the possibilities to explore the best old game design principles become endless. This report summarizes the development of a game that draws bits of inspiration from some of the best platforming collectathon titles throughout the ages, with the goal of combining the genre's best execution of ideas with novel mechanics.

## **1.1. Problem Statement**

As the video game industry expands with the rapid rate of growing technology, major developers compete with each other to create new and innovative titles, with increasing emphasis being placed on pushing system hardware to its limits, in an ever-decreasing time frame. The market quickly becomes saturated with novel ideas executed poorly, optimized to run smoothly and look beautiful, but missing one of the crucial pillars of a game: fun. One of the genres that has arguably suffered the most from this shift in focus is the platforming collectathon. Popularized in the early 1990s by 2D games such as *Super Metroid* and *Yoshi's Island*, then again in the late 1990s by 3D games such as *Banjo-Kazooie* and *Donkey Kong 64*, games in the platforming collectathon genre generally follow certain key design principles. They provide accessibility to new and casual players through easy-to-learn, challenging-to-master mechanics. They encourage and reward exploration, require awareness of how the world fits together, and often boast tight and precise mechanics. As the game industry shifted almost entirely to 3D by the early 2000s, the genre slowly faded into obscurity to make way for the more popular and realistic open-world sandbox genre. This trend has continued throughout modern game development, with visually stunning, yet undeniably empty worlds missing key foundational elements of game design. Due to the finite number of game developers, the oversaturation of one genre inevitably leads to the undersaturation of another. While modern platformers do exist, they are few and far between. The ones that do exist often have simple “point A to point B” levels, cutting the persistent collectables and power ups, and the exploration that goes hand in hand, at the expense of the player's desire to discover helpful or hidden items. While visuals have evolved tremendously over the past decade, and the quantity of games being produced is higher than ever, this too often comes at the cost of level design, fluid mechanics, and satisfying achievements.

## **1.2. Related Work**

Thanks to the recent streamlining of game development due to engines centered around object-oriented programming, and the ability to export games to nearly any popular console,

independent game developers are in the best position in history to fix the previously stated problem. Because the nature of independent game development almost always means low budget, well-executed game design becomes top priority to make up for the lack of state-of-the-art visuals and generally lower frame rates. Another surprising result has been the acclaimed return of 2D platformers, including *Hollow Knight*, released in 2017. The game's directors, Ari Gibson and William Pellen, created an intricate, interconnected map, and chipped away at linearity until the world started to become more open [1]. They utilized every area to house something unique and rewarding to discover, whether it be a helpful power-up or interesting boss. They made backtracking fun instead of tedious by teasing the player with an interesting area that they could reach if they could jump a few inches higher, then letting the player mark it on their map, and finally letting the player return and access the new area when a double jump power up is attained. *Hollow Knight* came with the return of one of the most missed and beloved game design elements: player-driven exploration and discovery.

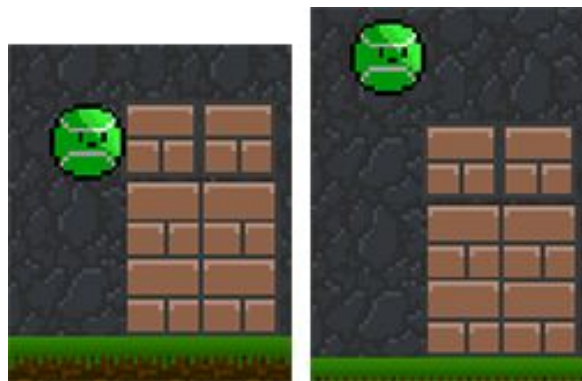
Although *Hollow Knight* received critical acclaim, it lacks some defining features of a platforming collectathon. The true challenge in the game comes from defeating tough enemies, sometimes to the point of frustration, leaving it inaccessible to many casual players. While the controls are tight, there aren't many platform-centric challenges in the game. Games that follow the *New Super Mario Bros.* formula are much more accessible to all types of players [2]. These games place nearly all of their challenge on platforming, with the additional challenge of collecting items such as star coins to fully progress, but at the expense of discovery and exploration. The level design is completely linear, and the player must finish the level within an allotted amount of time. This punishes the player for dawdling, and therefore thoroughly exploring. While both *Hollow Knight* and *New Super Mario Bros.* are widely considered good games on their own merits, they both lack defining elements of the platforming collectathon genre.

## **2. Solution**

In response to the undersaturated genre explained above, we have developed *Bounce*, a 2D platforming collectathon. Developed in Unity and scripted in C#, *Bounce* is built to run on

PC, featuring both standard keyboard/mouse and gamepad support. Bounce has been developed using an entity-component-system architectural pattern. Many objects exist in each level, each with a set of conditions and algorithms, dictated in C#, that will execute in real time given certain player input and in-game conditions. Our game challenges players with difficult-to-master platforming sections, exploration-and-puzzle-driven collectables, and a diverse moveset, featuring the bounce mechanic.

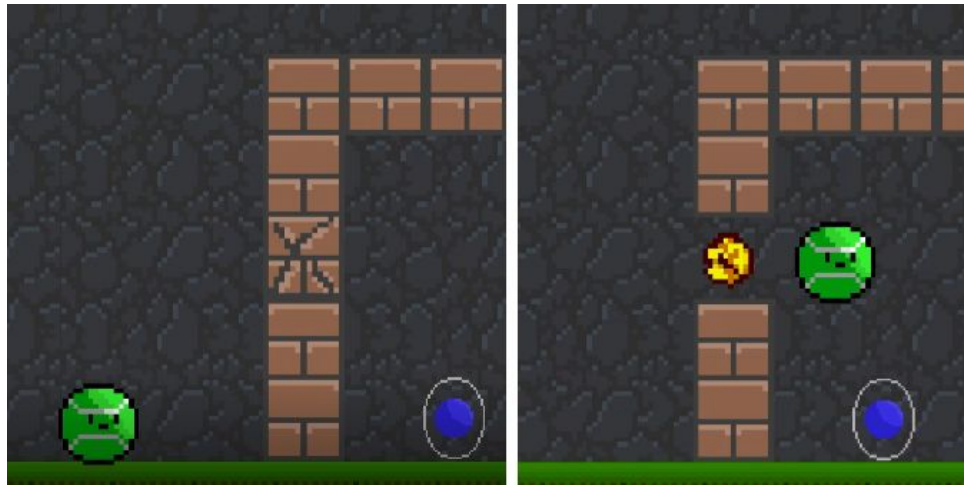
The bounce mechanic allows the player to vertically move 1.3 times higher than if they were to simply jump, as demonstrated in Figure 1. It is activated by pressing the “bounce” button when already in the air, causing the player to accelerate downward until the frame they collide with the ground, at which point the player will bounce upward. This mechanic shines in platforming sections, where maintaining momentum is incentivized by quickly clearing high jumps, and often times grabbing a collectable.



**Figure 1.** Demonstrating (a) the player’s height after jumping is not enough to clear a high platform, but (b) the player’s height after bouncing is enough to clear the same high platform.

The collectables in this game consist of keys and power-ups. Both are obtained simply by entering their trigger area, at which point its instance will disappear from the overworld, and a script attached to the player will update and maintain this data until the player purposefully deletes all saved data by starting a new game. Keys are used to advance through the game; the player must collect a certain amount of keys, in any level, to unlock subsequent levels. Power-ups, in a similar vein, are used to advance through the game by giving the player a new move to clear a certain area. Oftentimes, keys are placed in early level areas that are inaccessible

until the player obtains a power-up in a later level, as shown in Figure 2. The player makes a mental note so that once they obtain the required power-up, the earlier area can be explored with the new move. This design strategy aims to immerse the player by requiring them to have persistent awareness of their surroundings.

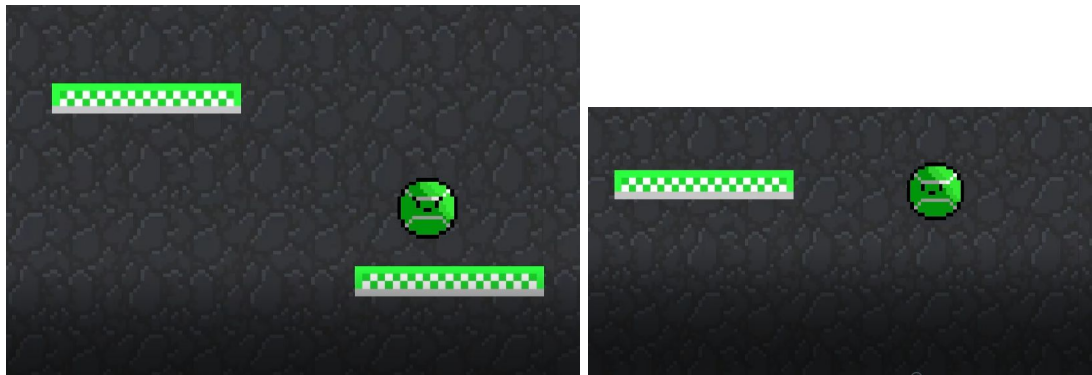


**Figure 2.** Example of the player using a power-up (dash bombing) to break through a wall and access a previously inaccessible area to obtain a collectable.

Power-up obtained moves include air dashing, dash bombing, double jumping, and gun shooting. The air dash accelerates the player horizontally while negating vertical movement for a brief stint of time. This is achieved by setting a timer for 0.1 seconds that is decremented every frame. Then, for the extent of that timer, disabling the player's local gravity scale, disabling horizontal input, and setting the velocity to a static value about 3 times greater than the player's normal horizontal movespeed. When the timer reaches zero, the player's settings are reset. Additionally, the dash bomb builds off the air dash, simply adding an extra animation and set of rules allowing the player to break through weak walls to collect power ups, and kill enemies when dashing. The double jump works to give the player extra vertical momentum once already in the air by allowing the player to use the jump button once more while in the air. The gun mechanic allows the player to shoot and kill enemies. All of the mentioned power ups, minus the gun, are essential for clearing certain areas and collecting every key, as well as advancing to the final point in some levels. This design strategy incentivizes the player to explore, discover, and

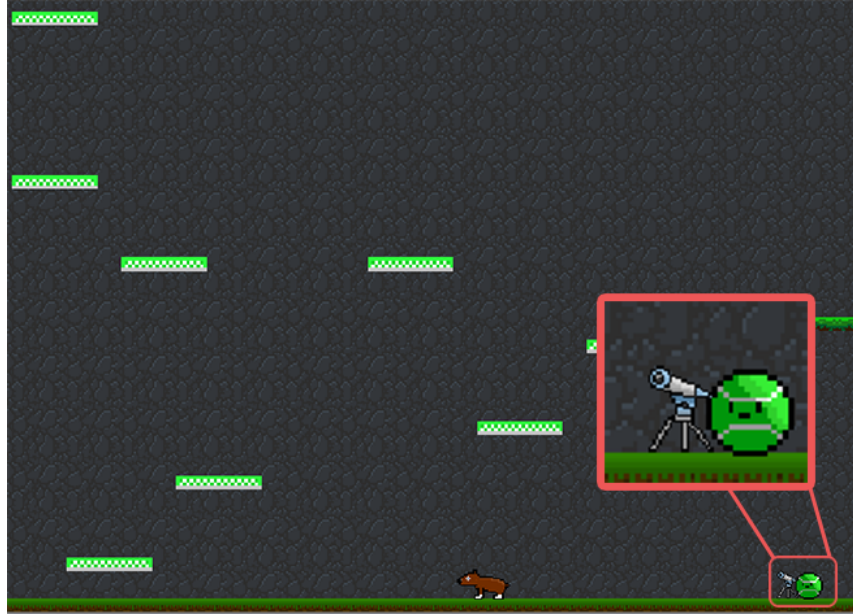
complete every platforming challenge they can find, even the optional ones, in order to receive the full experience the game has to offer.

Bounce's most recurring platform challenges are the disappearing platform sections. Demonstrated in Figure 3, once the player exits the trigger area of the green-checkered platform, it disappears. Disappearing platforms can only be reset when the player is on the ground. These platforms are placed thoughtfully around the stage at specific distances to push the player to use a diverse moveset. Maintaining vertical momentum by bouncing is essential, and because of this, the player will need to plan their moves out beforehand in order to move quickly. To aid with this, we introduced a telescope object, shown in Figure 4, that allows the player to zoom out and see particularly difficult platforming sections.



**Figure 3.** After the player bounces off the green-checkered platform, it disappears, and can only be reset once the player is on the ground.

In addition to the features mentioned above, we have also added many other scripts and objects to make the game flow smoothly and feel more complete. This includes, but is not limited to, a UI system including menus, help signs, and a game over screen, a health system along with hazards and enemies, diverse platforms with attributes such as rotating, falling, and slippery, a final boss battle, and a single-player battle mode.



**Figure 4.** The player using a telescope to zoom out and plan for a difficult platforming section.

### 3. Results

When combined together using strategic game design, Bounce's individual objects interact with each other as intended. Because there are so many objects, each with their own individual rules and algorithms, all running in real-time, the simulation varying depending on the player's input down to the frame, it is impossible to examine every scenario, or create an objective definition of success. However, it is possible to determine how closely we followed the key game design principles of platforming collectathons, and how much of an impact that made on the game's overall enjoyability.

Each level was meticulously designed around the player's movement and abilities. Since tight controls were a top priority, it became essential that the player has a jungle gym to fully utilize its own movement. While the levels take a more linear approach opposed to the open-world approach many collectathons take, collectables are still scattered throughout the level, and can be found out of order. Additionally, some levels have multiple paths and skippable sections to promote the player having autonomy over their gameplay. Backtracking isn't necessary to simply get to the ending of the game, but it is necessary if the player wants to



complete every challenge. For this reason, Bounce is accessible to both casual and seasoned gamers, with the difficulty option embedded into the player's own choice of path.



**Figure 5.** A zoomed out view of Level 4, demonstrating the use of a level container.

Due to the nature of the levels being held within a container, as seen in Figure 5, it is impossible to fall off the map, and encouraged to explore every accessible part of the container. The inaccessible parts of the level due to lacking a power-up entice the player to explore further for the power-up, then return to discover what they had missed. Seasoned players may take advantage of this, while casual players may opt out and only explore their immediate surroundings. Regardless of the player's choice, exploration is always encouraged as there are plenty of collectables, platforming challenges, and various equally entertaining activities to be found in each level.

## 5. Acknowledgements

The author would like to thank his advisor, Professor Joshua Fox, for his encouragement, knowledge, and support throughout this project, and for the dedication he has to his students.

## 6. References

- [1] Fenlon, Wes. "How to Design a Great Metroidvania Map," *PC Gamer*, 3 Aug. 2017, [www.pcgamer.com/how-to-design-a-great-metroidvania-map/](http://www.pcgamer.com/how-to-design-a-great-metroidvania-map/)
- [2] Iwata, Satoru. "Iwata Asks: New Super Mario Bros. U." Nintendo, 2012, [iwataasks.nintendo.com/interviews/#/wiiu/nsmbu/0/2](http://iwataasks.nintendo.com/interviews/#/wiiu/nsmbu/0/2)
- [3] Schell, Jesse. *The Art of Game Design: A Book of Lenses*. A.K. Peters Ltd, 2018.
- [4] Smith, Gillian, et al. "A Framework for Analysis of 2D Platformer Levels." Proceedings of the 2008 ACM SIGGRAPH Symposium on Video Games - Sandbox 08, 2008
- [5] Koster, Raph. *A Theory of Fun for Game Design*. O'Reilly Media, 2014.

## 7. Biography

Jesse F. Kolb is completing his baccalaureate degree in Computer Science with a minor in Innovation at the University of Florida in Gainesville, Florida, where he expects to graduate on December 16th, 2018. He enjoys creatively using computer programming wherever he can, whether that be through designing a video game or innovating an experimental software application. He has recently completed two years of working as a technical consultant and supervisor at the University of Florida's Computing Help Desk.