

Super Mario Run

The Movement of Mario



Figure 1 - Artwork by Shigeru Miyamoto sourced from: <https://www.youtube.com/watch?v=zWyAbvTDx4c>

Version: 1.0.0

Overview

High Concept

Super Mario Run is a 2D side-scrolling platformer in which players help Mario navigate through a series of courses. This document details the movement and characteristics of Mario as applicable to the aforementioned game.

Super Mario Run is intended for players of all ages, genders, and nationalities who enjoy 2D side-scrolling platformers and are looking for such an experience on mobile devices.

Core Movement Mechanics

- Basic Movement
- Jumping
- Spin Jump
- Triple Jump
- Wall Kicks
- Ledge Grabbing
- Roll
- Slope Slide
- Death and Respawn



Figure 2 - Concept art of Mario's athletic move style sourced from: https://img.neoseeker.com/v_concept_art.php?caid=48865

Targeted Platforms

- iOS
- Android

Document Notices

For the purpose of demonstrating mechanics detailed for *Super Mario Run*, this document will be populated with reference images from previous Mario titles.

Core Design Objectives

Accessibility

Being the first Mario title for mobile devices, it is crucial that the game's controls feel intuitive to players regardless of their experience with other 2D platformers or lack thereof. All input asked from the player therefore needs to be easy to accomplish from a physical standpoint. In summary, this means that none of the actions required should necessitate fast reflexes, continual rapid input, precision timing, or other mechanisms that would instill frustration. Note that care is also required to ensure the game is enjoyable regardless of which hand a player prefers (left or right) to give input during gameplay.

Outside of the physical demand required to help guide Mario, the controls of the game must also be accessible regardless of the phone model a player is using. As mobile devices come in a wide array of sizes and resolutions, input required of the player should have a high level of leniency in terms of precision. This is doubly important as the input device and the screen are one and the same for this game – the phone's screen.

Challenge

While *Super Mario Run* is intended to have a generally inclining difficulty curve, the source of challenge should never come from controlling Mario. As outlined above, controlling Mario is designed to be easy and forgiving, rather than difficult and punishing. While this thought process is a pillar in designing the controls of Mario, there is still the likelihood that players less experienced with the genre will struggle in later courses. As such, an option to select difficulty should be considered to enable players a chance to practice a course to their comfort before requiring some level of mastery. On the flip side, any segments of gameplay that use the mechanics of Mario outlined in this document to create intentionally difficult gameplay segments should be optional to game completion as to not frustrate players unnecessarily.

Accomplishment

Alongside systems such as coin collection and kingdom building, mastery of controlling Mario should also be rewarded. While not vital to game completion, actions such as hitting the top of an end-of-level flag should be utilized to encourage a feeling of accomplishment as players master input timing alongside spatial analysis. Rewarding the player with coins for moments of mastery in controlling Mario is encouraged to connect this aspect of gameplay to pre-existing systems.

Research

Super Mario Bros

One of the first games to ever feature Mario, this title is the first game to look at when designing movesets and controls for the iconic character's first mobile game. Being the origin of many staples in the franchise, it is important to refer to this title to ensure brand consistency in the mechanics of *Super Mario Run*. While *Super Mario Run* is designed for a completely different platform, controlling Mario should feel familiar to players who have experience with other games in the franchise. Below is a list of considerations for designing the movement and characteristics of Mario in this title:

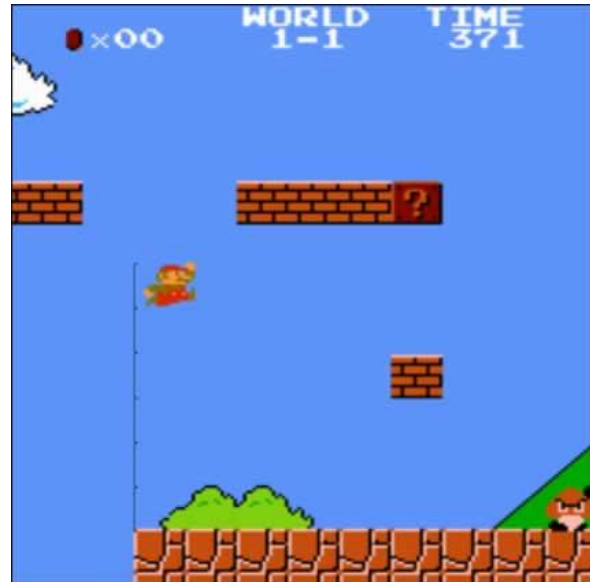


Figure 3 - Mario's maximum jump height sourced from: <https://hypertextbook.com/facts/2007/mariogravity.shtml>

- No checkpoints – if Mario is defeated, he restarts at the beginning of the level.
- The length of time the jump button is held affects the height of Mario's jump.
- Mario only has the ability to perform a standard jump – no extended moveset.
- Mario's jump has a peak height of five block-units (see Figure 3).
- Mario's speed directly affects Mario's jump height.
- Mushroom, Fire Flower, and Power Star - all cause Mario to grow except the Power Star.
- Landing on enemies allows Mario another jump before contacting the ground.
- Mario can speed up if he runs in the same direction for an extended period of time.
- Mario will run over small gaps at a higher speed – risk/reward mechanism.
- Mario will not fall off ledges until his entire hitbox is no longer colliding with ground.

Research Continued

New Super Mario Bros Wii

A modern approach to 2D Mario platforming building off its DS predecessors, *New Super Mario Bros. Wii* introduces mechanics seen in 3D adventures to a 2D play space. Designed for the Wii, many in-game mechanics and powerups utilize the system's motion control capabilities. As this game demonstrates, game mechanics and player input should tie in with platform-specific hardware in order to create unique experiences that stand out from other titles in the franchise. Below is a list of considerations based on the movement and characteristics of Mario in this title:



Figure 4 - Propeller Suit utilizes motion controls. Image sourced from: [http://mario.wikia.com/wiki/World_7-5_\(New_Super_Mario_Bros._Wii\)](http://mario.wikia.com/wiki/World_7-5_(New_Super_Mario_Bros._Wii))

- The length of time the jump button is held affects the height of Mario's jump.
- Utilizes motion controls for one-off level mechanics as well as base player controller.
- In-air spin allows minor increase in how far Mario can jump.
- Levels have visible checkpoints used when the player loses a life.
- Multiplayer mode utilizes bubble respawning mechanic – agency over respawn location.
- Game features wall kicking on vertical surfaces for gaining height.
- Mario is able to slide down slopes (defeats enemies on impact).
- Mario is able to ground pound at the cost of losing momentum – undesirable for *Run*.
- Mario can perform a triple jump with precisely timed jumps – never directly required.
- Landing on enemies allows Mario another jump before contacting the ground.
- Game expands upon pre-established powerups (Ice Flower, Propeller Suit, Penguin Suit).
- Player can influence how fast Mario moves by holding down the run button.

Research Continued

Super Mario Galaxy Series

Despite being a 3D experience, the *Super Mario Galaxy* games give insight into how a modern audience may perceive Mario from a mechanical and characteristic standpoint. As at the time motion controls were relatively new to general audiences, this game can also be examined in terms of teaching players how to handle new types of input devices. Likewise, the game's use of motion controls should be examined to see what types of physical gestures players find comfortable, alongside how frequently this demand on players should be placed. Below is a list of considerations based on the movement and characteristics of Mario in this title:



Figure 5 - Tutorial prompts use imagery rather than text. Image sourced from screen capture of *Super Mario Galaxy*.

- The length of time the jump button is held affects the height of Mario's jump.
- Utilizes motion controls for one-off level mechanics as well as base player controller.
- Spin jump allows increase in the length of Mario's jump & correcting overshoot jumps.
- While both games utilize level checkpoints, *Galaxy 2* makes these checkpoints visible.
- Tutorialization of new hardware controls are done through text and on-screen animations.
- Use of imagery instead of text can cut down on localization costs – to consider.
- Mario is capable of wall kicking to gain vertical height.
- Mario is able to ground pound to damage enemies/obstacles at the cost of momentum.
- Games make use of long-jump to allow players a way to clear long gaps.
- Mario is able to crouch to reduce hitbox size at the cost of speed.
- When crouched Mario can backflip – more vertical velocity than a normal jump.
- Mario can perform a triple jump with precisely timed jumps – never directly required.
- Advanced control: homing ground pound – never directly taught to player.
- Landing on enemies allows Mario another jump before contacting the ground.

Research Continued

Doodle Jump

A popular 2D platformer for mobile devices, the controls of *Doodle Jump* have been considered when designing *Super Mario Run*. While the objectives of *Doodle Jump* and *Super Mario Run* are fundamentally different (vertical vs. horizontal forms of progression, to name one), there is still information to gain when looking at how the controls for this game were implemented from a UI/UX standpoint. Below is a list of observations regarding the player controller in *Doodle Jump*. The notes below should be considered when designing *Super Mario Run*'s controls:

- Intended phone orientation (portrait) meshes with the overall objective of vertical ascension.
- Player is unable to change orientation of screen based on physical position of device.
- Game utilizes phone's orientation to direct movement of on-screen character.
- One "button" controls most user input – simplistic design due to lack of buttons on most mobile devices.
- Player must tap on phone to shoot projectiles at enemy. Player can tap anywhere in an effort to discourage the player from blocking their view of player character.
- Game features continuous progression which the player cannot halt – forces rapid analysis of upcoming space.
- Powerups directly aid in goal of vertical ascension, however there is no clear indicator as to when these powerups will automatically become defunct.
- No life system present – high score driven experience.
- Player can bounce off an enemy to gain increased vertical momentum.
- Game lacks tutorialization on level mechanics as introduced – up to player to experiment with new platforms and powerups.
- Resuming play from pause screen instantly jumps the player back into the experience – no time to analyze space.

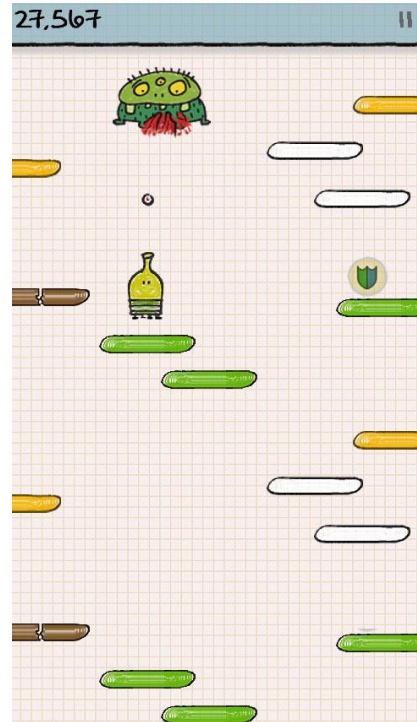


Figure 6 - Doodle Jump favors portrait orientation. Image sourced from: <http://eurodroid.com/2010/03/lima-sky-launches-its-quirky-doodle-jump-on-android/>

Mario Movement Mechanics

Basic Movement

With the intent of engaging players through creative use of level mechanics combined with Mario's different jumps, the pace and direction of Mario's movement will be outside the player's control. This should achieve two results: creating a constant pace for gameplay, alongside enabling players to focus on upcoming obstacles rather than prior level segments. Both elements are crucial to creating an experience that encourages gameplay that can be picked up and put down at a moment's notice, as is the intent for most mobile games.

As seen in small segments of previous Mario titles as well as *Doodle Jump*, one downfall of on-rails gameplay is the inability for a player to self-regulate when to have a moment to rest. As it is vital to ensure the player is not stressed during the experience, and especially during the initial courses, one way to enable moments of rest will occur when Mario comes into contact with a column of blocks, pipe, or other obstacle at least two block-units high. When this occurs, Mario will halt, waiting for player input to clear the obstacle in his path. For the purpose of creating stairs or confining enemies to a single space, single blocks may be placed about the environment for Mario to navigate his way over or on top of. In order to not demand a constant need of input from the player, Mario will automatically vault over single block-unit obstacles. Despite this, there should still be a slight pause in Mario's progression. An athletic bound or leap animation can serve two purposes in this case: conveying the more athletic nature of Mario, alongside creating a slight delay in forward progression. Concept art below illustrates a possible pose for Mario to take when clearing smaller obstacles in his path.



Figure 7 – Concept art of Mario bounding over a small obstacle in his way sourced from:
https://img.neoseeker.com/v_concept_art.php?caid=48865

Mario Movement Mechanics Continued

Jumping

As noted in the research section of this document, all Mario titles examined have featured a system where the longer the jump button is tapped, the higher Mario's jump becomes. As such, one crucial element to making Mario feel right is to incorporate this mechanic into *Super Mario Run*. Having removed the requirement of a player to control Mario's direction and speed, the game can fundamentally become a single-button experience, ideal for casual gameplay. As mobile devices lack a universal button layout, the means of giving input must come from touching the device's screen. With this in mind, while a dedicated jump button could be placed somewhere on the UI of the game, a cleaner approach would be to allow the player to tap anywhere to inform Mario he should jump. Allowing freedom to select where to tap prevents a player from positioning their hands in such a way that vital elements of the screen are obscured. Likewise, leaving this input unanchored helps prevent against an issue where other apps are blocking the section of the screen dedicated to user input.

With tapping of the screen used as a means of player input, the way in which a player can affect the height of Mario's jump becomes clear: the length of time the player keeps their finger pressed to the screen. While not a universally common input action for mobile devices (tap, double tap, swipe being more common), the first course of the game can be used to teach this mechanic by creating scenarios where the player is required to hold down the jump button to progress. Being the first course of the game, learning this mechanic should be presented in a risk-free environment, such as being blocked by a tall pipe rather than by a huge bottomless pit.

For consistency with previous titles and brand imaging, Mario's jump should remain at a height of five block-units.

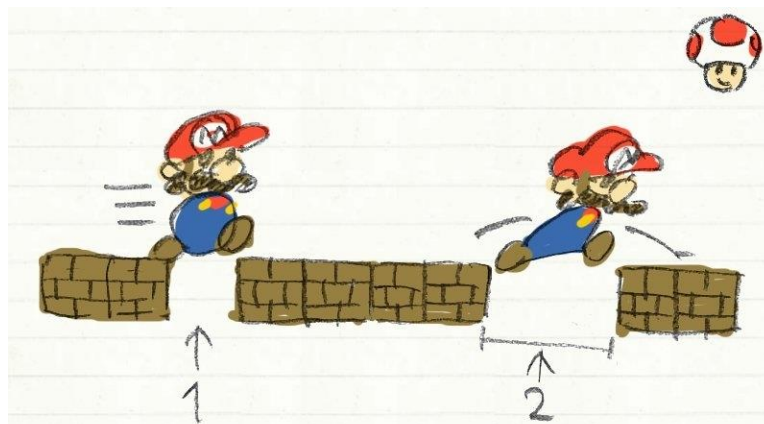


Figure 8 - Mario should automatically run/vault over small gaps. Concept art sourced from: https://img.neoseeker.com/v_concept_art.php?caid=48865

Mario Movement Mechanics Continued

Spin Jump

Players will be able to cause Mario to perform a spin jump by tapping the screen while he is midair. The spin jump's primary uses are to enable players to further the length of a jump should a slight miscalculation have been made, as well as allow for more precise movement for advanced players. Mario will be able to perform multiple spin jumps in a row as long as he has not hit the ground, with a slight delay preventing overuse of this mechanic (approximately 1 second).



Figure 9 – Concept art from Super Mario World. Note star effect and extra jump height. Image sourced from: https://www.ssbwiki.com/Mario_Tornado

Triple Jump

A type of jump first introduced in 3D Mario experiences, the triple jump allows a way to reward more experienced players for a series of well-timed jumps. This jump allows Mario to gain more height and distance than normal, useful for clearing obstacles or obtaining out of the way coins. Mario will perform this move if players tap on the screen three times in a row in sync with his landings. A grace period of +/- 0.25-0.5 seconds will allow some leeway, which may have the added side effect of causing players to have Mario perform this move by accident. This can serve as a hint that there are ways to use Mario's moves to further his capabilities. As mentioned in the Core Design Objectives section of this document, mastery of Mario's movement should be rewarded, and the triple jump affords this opportunity. Since this is a more advanced move, performing a triple jump should not be required for the core gameplay experience.

Mario Movement Mechanics Continued

Wall Kicks

Allowing a quick means of vertical ascension, wall kicks enable the player to reach otherwise inaccessible areas. Wall kicking can be used in *Super Mario Run* to encourage players to explore branching paths of levels, allowing individual-defined difficulty.

To execute a wall kick, players first need to come into contact with a solid portion of level geometry (pipes, walls, blocks, etc.). When players tap on the screen while Mario is colliding with a vertical surface, he will begin sliding down the object. While sliding, players can tap the screen again to cause Mario to propel himself away from the object. As this jump applies horizontal velocity to Mario opposite of his facing direction, this technique can be used to shoot Mario a short distance backwards to previous level segments. This allows for the collection of missed coins alongside the ability to alter which path a player wishes Mario to travel down.

Ledge Grabbing

Ledge grabbing allows leniency if a player's jump ended up a bit short. Just like progression over smaller obstacles, ledge grabbing will result in a small delay in movement while Mario pulls himself up to the top of the surface, which can be sped up should a player tap on the screen. This allows the player time to get their bearings and not be taken by surprise when they find Mario is still alive and preparing to progress further, while simultaneously allowing a brief moment to look ahead. This moment is critical to preventing a domino effect of miscalculated jumps that would ultimately lead to the player spending more time just to lose the level. Below is concept art displaying how this action may play out on screen to communicate Mario's athletic prowess and determination to reach his end objective.

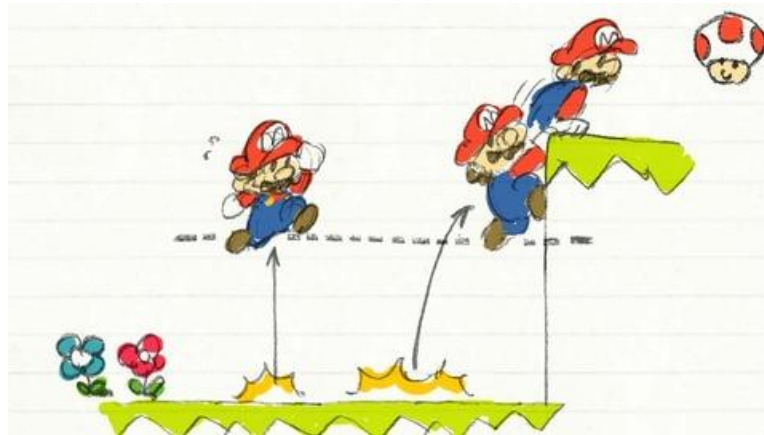


Figure 10 - Concept art of how ledge grabbing may look in game. Image sourced from: <http://super-mario-run.net/687>

Mario Movement Mechanics Continued

Roll

Inspired by *Super Mario 3D World*, rolling can be used as a form of offense against enemies that Mario drops in front of without breaking the game's pace by maintaining forward momentum. As control over Mario's movement direction and speed are taken out of the player's hands, rolling when falling from a height of at least two block-units prevents a player from taking damage from an enemy that could have been easily avoided in traditional Mario titles. An important note for this movement mechanic is that control is not taken away from players during the roll animation. Players may still tap on the screen while this action is taking place, breaking the roll and causing Mario to jump.



Figure 11 – *Super Mario 3D World* animation. Note: Arms are used to protect face for believability. Image sourced from: https://www.mariowiki.com/File:SM3DL-Mario_Roll_Artwork.png

Slope Slide

Another move used to give Mario offensive capabilities without costing forward momentum, sliding occurs if a player jumps onto a sloped surface, or walks/falls onto a steep incline. While in a slide, Mario will move faster than normal. As such, in early courses bottomless pits or other obstacles should not be placed at the edge of a steep incline. Just as with rolling, player agency will not be taken away while Mario is in this state. Should a player wish to interrupt the slide, they may tap on the screen to cause Mario to jump.



Figure 12 - Mario holds his balance during slope slide to make mechanic feel safe to use. Image sourced from: https://www.mariowiki.com/Slope_Slide

Mario Movement Mechanics Continued

Death and Respawn

In order to maintain the game's pacing, when Mario loses a life, the course will not restart. Instead, Mario will enter into a protective bubble that the player may pop at any time by clicking the screen. The bubble Mario finds himself in after losing a life will drift backwards through the level, allowing the player to choose where they think the best place for respawn is, rather than using an arbitrary checkpoint system to restart players at a particular segment. With the ability to go to previous level segments through the use of a bubble, some players may choose to lose lives or jump into bottomless pits on purpose in order to backtrack to collect a missed coin or powerup. While this is not the intended functionality of this mechanic, giving the player agency over where they feel is the best place to resume play, based on their intrinsic objectives and time remaining, should be encouraged.

Movement and behavior of bubbles should feel akin to the respawn system used in *New Super Mario Bros. Wii* during cooperative gameplay. Note that the Wii remote icon above the bubble, as seen below, represents the method of popping the character out of their confinement which could be replaced with a finger tap icon for *Super Mario Run*. One concern with this from a global representation standpoint, however, would be what color the finger icon would be.



Figure 13 – Reference of bubble respawn mechanic seen in *New Super Mario Bros. Wii*. Image sourced from: <https://ryan-burns.blogspot.com/2012/07/late-in-game-review-new-super-mario.html>

Supplemental Level Mechanics

Pause Blocks

The primary means for a level designer to enable the player more control over Mario's movement, Pause Blocks, freeze Mario in place when he stands on top of them. The primary function of these blocks, is to enable slight adjustments to a level's pacing as desired. To keep with the single "button" format, tapping on the screen while on a pause block will cause Mario to continue his journey to the end of the course. Below is an example icon associated with pausing for artists to take inspiration from when creating the visuals for this level mechanic.

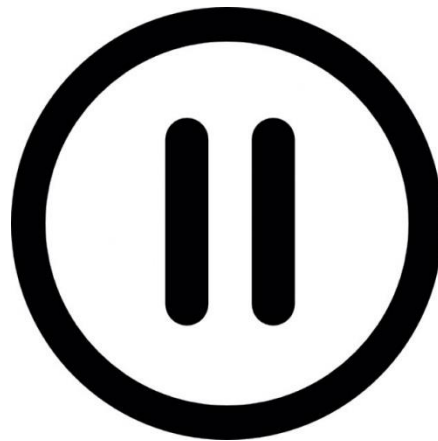


Figure 14 - Circle may prove redundant with exterior block shape. Possibly match with UI symbol? Image sourced from: https://www.freepik.com/free-icon/video-pause-button_750973.htm

Coin Spawners

While not a direct mechanism of Mario's movement, coin spawners indicate to the player which way to direct movement. Coin spawners will appear as unfinished blocks to indicate to players that they are non-solid objects. The interior of the block will house an arrow, indicating on contact with Mario which way its coins will appear. Coin spawners should never lead the player into a trap (pointing the player into a bottomless pit, lava, etc.). See below concepts.

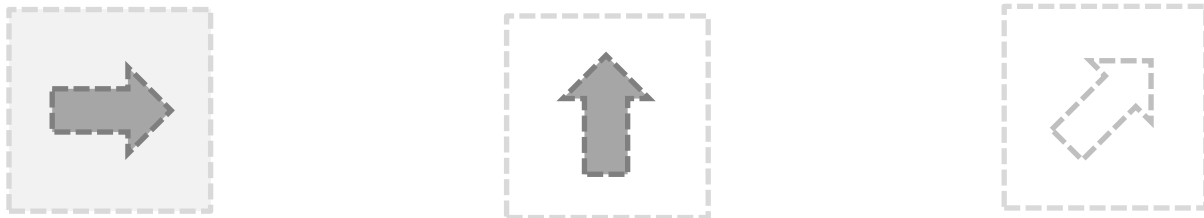


Figure 15 - Conceptualization of coin spawner blocks. Note blocks should convey that they are non-solid.

Supplemental Level Mechanics Continued

Springboards

Springboards are used to apply a massive amount of vertical velocity to Mario in a very limited space. Triggered by tapping on the screen while Mario is standing on top of them, springboards can be used to create branching paths or provide a quick change of elevation as courses may require.



Figure 16 - Note: Springboards have not had a consistent appearance but maintain accordion-style base. Images sourced from: <https://www.mariowiki.com/Springboard>

Long Jump Blocks

In an effort to prevent overwhelming players with too many moves at one time, alongside the constraint of a single button, case-specific jumps like the long jump will be inaccessible during most gameplay. To use the long jump, blocks will be placed by level designers in courses that light up while Mario stands on top of them. While in this lit up state, if the player chooses to tap the screen to make Mario jump, he will instead perform a long jump. Note that while Mario stands above these blocks, normal jumping will not be possible. This is to prevent players from becoming confused over what Mario can and cannot do, especially when they lack the ability to completely stop Mario over these objects.



Figure 17 - Consider use of ! block color scheme to associate new mechanics with preexisting brand entities. Image sourced from: https://www.mariowiki.com/!_Block

Supplemental Level Mechanics Continued

Powerups

Although not directly influencing the movement of Mario, the capabilities and behaviors of powerups must be changed to accommodate for his automated movement. Instead of hitting a block to see a powerup slowly pop out for the player to grab, Super Mushrooms and Power Stars will be thrown forward instantaneously when released from a block. Powerups will move at a slightly slower speed than Mario, to allow the player to easily pick up the object.

While Mushrooms simply grow Mario in size and allow him to take an additional hit before being defeated, the use of a Power Star should be considered. In previous titles, alongside becoming invincible to most damage, Mario is either seen to move faster while under the influence of a Power Star, or creates a trail of coins from behind. While increased speed would directly contribute to the overall objective of a course, spawning a trail of coins behind Mario makes little sense as Mario cannot be turned around. An alternative which would tie in with the secondary level objective of obtaining all specially colored coins would be to turn Mario into a magnet for coins. This effect would allow the player the ability to bolster their overall high score on a level, as well as more easily collect the specially colored coins when moving at a faster pace.



Figure 18 - Powerups seen in New Super Mario Bros. Wii. Powerups that change movement style (Propeller Suit, Penguin Suit) or are based in combat (Fire Flower, Ice Flower) should not be considered as they conflict with core objectives. Image sourced from: <https://www.youtube.com/watch?v=fOLE7IrxRIE>

Supplemental Level Mechanics Continued

Enemies

Even though they do not directly contribute to Mario's move set, enemies placed throughout stages have a new risk in *Super Mario Run*: being perceived as unfair obstacles. As players cannot control the speed of Mario as he runs to the end of a course, it is likely, if not inevitable, that players may find themselves in a situation where enemies have been clustered in such a way that complete avoidance is impossible. As such, a way to mitigate this is diagrammed below: allow Mario to automatically vault over enemies as he does smaller obstacles. While Mario should automatically avoid taking damage from enemies he will run right into, this behavior does not need to be utilized for every enemy. Enemies incapable of chasing after or running into Mario, such as Piranha Plants or Podoboos for example, should still cause Mario to take damage. This behavior prevents Mario from taking damage a player couldn't have avoided, while still creating a sense of tension when desired in later courses. Further, this helps convey to players that Mario's primary objective is to reach the end of the course (and by extension Princess Peach) rather than contend with all of Bowser's baddies.

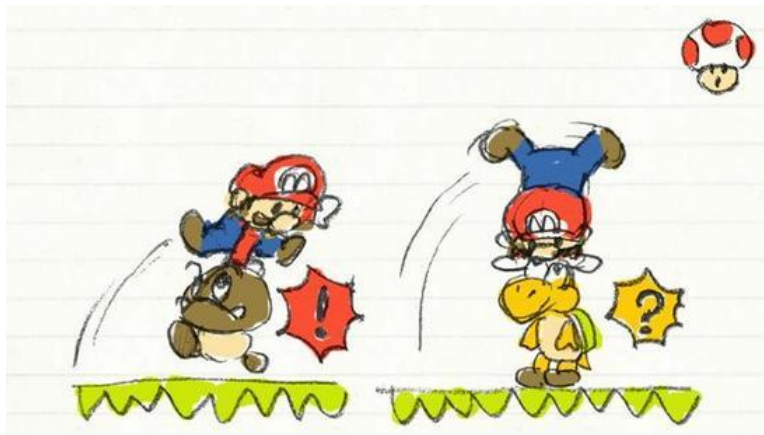


Figure 19 - Concept art of how Mario would vault over enemies directly in front of him. Image sourced from: <http://games.sina.com.cn/tw/qt/2016-11-25/doc-ifxyawmm3372941.shtml>

Visual Design Brief

Art Style

While Mario is meant to seem more athletic, there should be little if any change to his character's visual design. For brand consistency, Mario should be designed to maintain the same silhouette and shape language from previous titles. Use of rounder edges is preferred to give the character a more appealing design for audiences of all ages and backgrounds.

Animations

To make Mario's new movement style feel like it fits his pre-established character in the franchise, care must be taken in how he is to be animated. Below are some specific examples of how certain events should be animated to convey the character's mood during gameplay:

Note: Reference animation from previous Mario titles alongside gymnasts to create basic keyframes for each action described in this document.

- Mario should have several animations per action to prevent repetitive visuals.
- Mario should never become static. For example, when waiting for player input to jump over a pipe or wall, he should begin pacing, showing he is eager and ready to continue.
- Mario's standard walk cycle should be primarily composed of jogging actions/movement. This helps convey the pace of gameplay better than a walk or run.



Figure 20 - Concept art for Mario. Note facial expression conveys lighthearted tone. Image sourced from: https://img.neoseeker.com/v_concept_art.php?caid=48865

Audio Design Brief

Voice Acting

The primary way to convey Mario's character, the limited speech Mario has, should be utilized to convey the mood of the red-capped hero. In order to make the game more welcoming to players and to encourage course completion despite occasional failure, all of Mario's speech should be optimistic, regardless of the situation. Moments where voice acting is used should be saved for important moments during gameplay. As players will associate themselves with the character they are controlling, this approach will prevent players from feeling disjointed from gameplay. Examples of dialogue are listed below for reference:

- "Woohoo, letsa go!" – Starting the game.
- "Yes!" - Hitting top of the flag.
- "Oh yeah, Mario time!" – End of level catchphrase.
- "Wowie!" – Collecting a Power Star
- "Oh no." – Time runs out.

Note: Minor vocalizations from actions such as jumping should be used to keep the player in Mario's headspace.

Sound Effects

To ground Mario in the world, and keep a player's attention on his position in the world, most of the in-game sound effects should be sourced from Mario. In other words, sound effects that are heard during gameplay should be coming directly from Mario, or should have been caused by an action of Mario. For example, the sound effect of a Piranha Plant snapping in the air should remain absent from this title, however the sound of a Koopa shell getting kicked by Mario should be utilized.

While the approach to the game's audio listed above limits the ability to create immersive level soundscapes, it helps ground Mario as the most important object on screen. As players lack direct control over the speed at which Mario travels, and the direction he moves in, keeping attention to these aspects is critical. Decluttering the ambient soundscape allows for sound effects such as Mario's footsteps, sliding, and rolling to all become much more noticeable to the player. The only sound effects that level objects should create are those that hint toward movement. For example, if Mario is on a moving platform or conveyor belt, then a sound effect from either of these objects would be appropriate, as they directly convey movement to the player.