

# DRAGON BALL Z

ドラゴンボールゼット

# VR



Aid Harrison

# Dragonball VR

*Dragonball is an interesting candidate for VR given the amount of aspects which need to be taken into account, the following document is an overview of the steps which can be taken to create a virtual world based in the Dragonball universe.*

## Overview:

- > Engaging skill system
- > Mechanics over numerics
- > Improvement through practice as a first principal

Take full advantage of the VR space, choose movement over numerics and button inputs. Instead tie button inputs to motion, fighting in Dragonball is an exercise, so a game variant should reflect that.

## Input & Mechanics:

*This will be based off of the Meta Quest 2's input capabilities specifically for reference purposes*

Fighting power & Ki:

Fighting power dictates the total amount of energy you have to expend before you can no longer play, ki refers to your current useable energy. When charging ki you convert fighting power to ki and thus deplete your total energy in exchange for temporary energy.



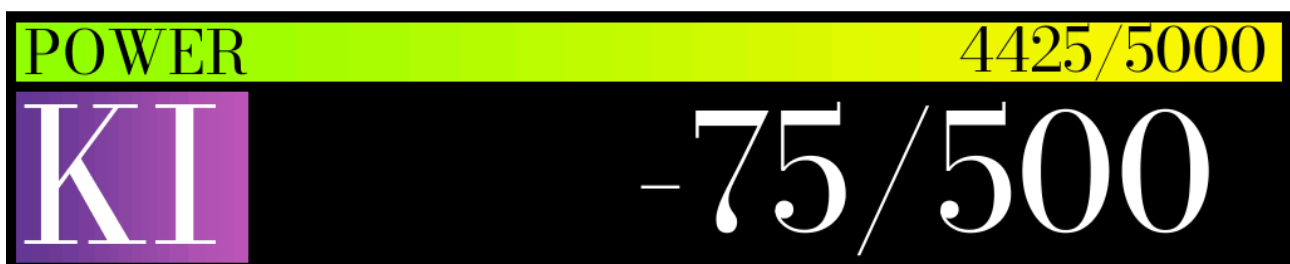
*Player starts with max Fighting Power and Ki*

Skills use ki, requiring a set amount. For example, a standard ki blast costs 25 however you don't need 25ki in order to use said skill, instead you can go into debt.

Debt:

**Both fighting power & ki** can go into a deficit/debt of 25% of their max respective values, when in a deficit you CANNOT use anything and the charge rate is reduced as a minor punishment. This incentivizes proper resource management however adds a small amount of leeway to provide a more frictionless experience.

In the previous example of a standard ki blast using 25 ki, while it uses this amount you can still use it even if you had 1 ki available, setting you at a debit of -24. You must then wait for the ki to **return to a value greater than 0** before being able to do any skill.



In short, good resource management will allow a player to be more potent in combat, however when in a pinch they can resort to going into debt, due to the debt system it is far more consistent to be able to use abilities instead of running into some “*not enough mana*” resource issues so commonly found in many games. However, the player is punished for going into debt thus incentivising proper resource management

#### Energy debt:

Energy debt is slightly different, when in this deficit the player takes more damage but is vastly more powerful in every other way. This last resort deficit gives the opportunity of finally defeating your enemy, the debt twist comes from the success of defeating an enemy. Assuming there are multiple enemies the player is brought out of debt and slightly “into the green” per defeat of an enemy when in this energy deficit.

#### Health and Ki shield:

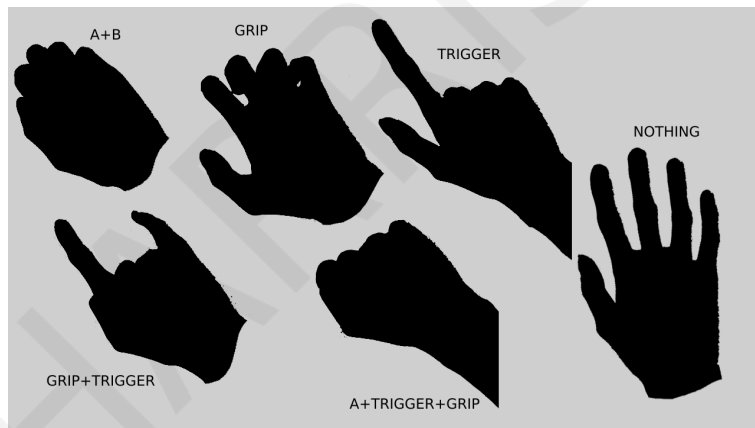
Ki shield provides damage mitigation. A portion of the damage is split between hp and the ki shield until the shield is depleted, at which point the hp takes full damage. Ki shield is only active when the player is blocking, encouraging more thought in combat. Much like Super Smash Bro's however the player can only block for a short period of time and must wait for a moment not blocking to regain the ability (Damage mitigation). Blocking while block is depleted serves no purpose, while it is an illogical, immersion breaking hinderance it paints a clear picture between offence and defence as well as forcing thoughtful combat as oppose to mindless swinging.

The state of the ki shield is tied to the ki meter AND available energy:

- Max ki shield is capped based on available energy
- Ki shield is drained when using skills along with the ki meter

Have X amount of hand poses

- Relaxed
- Open
- Fist
- Semi-fist
- Claw
- Palm/Slap
- Grip
- etc.



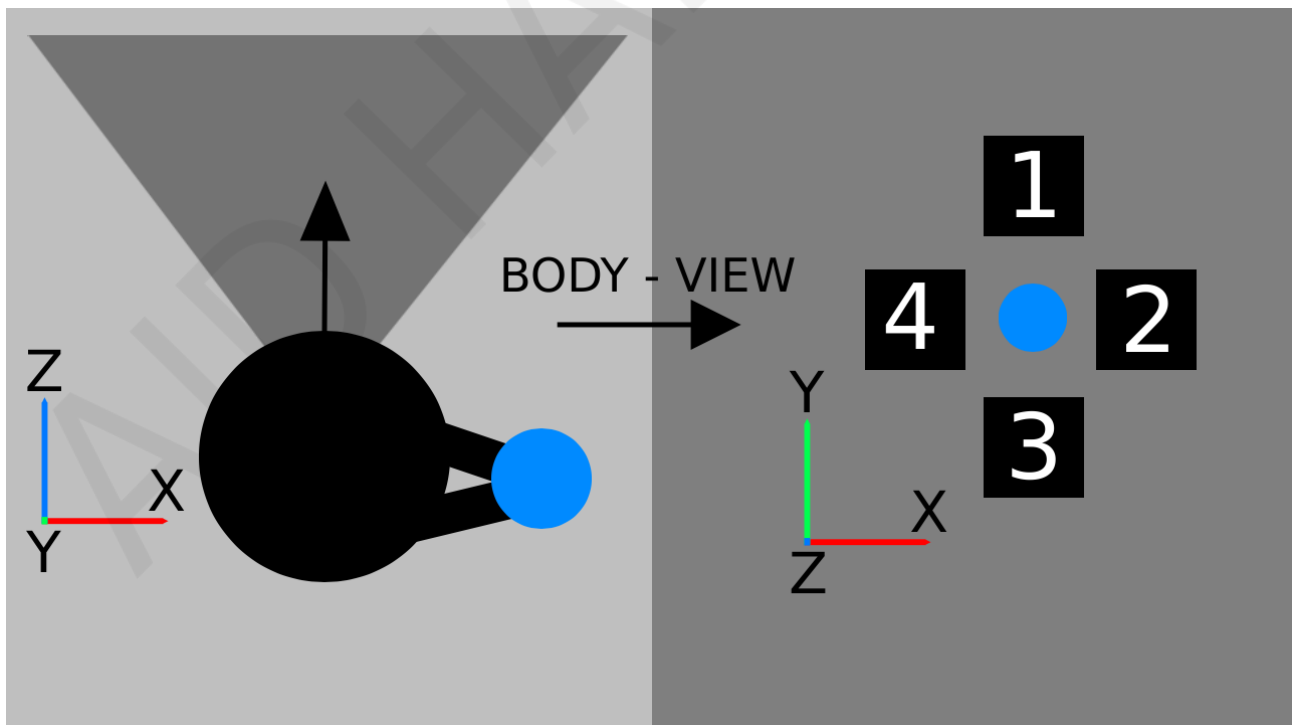
These poses are achieved with button combinations and capacitive touch. Hand tracking needs to be greatly refined via controllers (Index Knuckles) to be viable.

#### Zone based skill system

##### Details:

- Example | Ki-Blast
  - Pose: Open
  - Move hand close to body
  - Within Y amount of seconds, move hand forward until arm is close to being straight
- Example | Final-Flash
  - Pose: Claw (Both hands)
  - Move both hands together so their rotations are within a certain threshold
    - Rotation thresholds add a high level of depth, closer to perfect = more damage, faster charge, accuracy etc.
    - Rotation thresholds also make it so you are forced to do the move correctly
    - In final-flashes case, the base of both hands must be touching and about 50-60 degrees apart
  - Hold until charge passes threshold, drains ki meter while charging
    - Can continue for much much longer but ki drain rate increases slowly
  - Charge system:
    - To make charge moves more engaging and to add another level of skill there is a pattern based system involved

- In this final flash example, once the arms are extended and the charging has begun, 4 buttons will appear \*\*along with a reset zone\*\*
  - Think of it like the Halo intro where you have to look at the targets but the arms are the eyes
- By holding the pose and moving your entire arm to hit these buttons you can charge the move much faster AND subsequently overcharge while using the same amount of ki
- These would have set dynamic patterns
- Dynamic pattern system:
  - There are several different button arrangements and counts per move
  - Patterns are dynamic in the sense of their functionality, they are not set
  - For example:
    - Final flash is a lateral pattern, meaning you can start at ANY button BUT have to go to its opposite every time
    - So it could go:
      - RESET -> R -> L -> RESET -> U -> D -> RESET -> R -> L -> RESET -> L -> R.....
      - This can be done at any speed, until the moves power is capped (Based on base power) OR until ki is drained
- Solutions to certain moves:
  - A large majority of moves which use the charge system are outside of the players view OR may require a non conforming hand position, e.g. Special Beam Cannon
  - The solution to those moves out of view: Galick Gun, Destructo Disc, Kamehameha etc.
  - Is to cast the hands position in front of the player, like a targeting system for a fighter jet
  - The display which goes in front of the player is transparent and relays where the hand/s is/are surrounded by its associated pattern
  - Moving the hands will move the reticle and patterns can be engaged, thus the player can look at there target while still charging



- As for a move like Special Beam Cannon, the arm which is not in use will become the pattern arm and must fulfil a set of button inputs instead of moves, diversifying the types of charging methods, again all displayed in front of the player

## Hand-to-hand combat:

- Hand to hand functions with three poses:
  - Fist
  - Semi-fist
  - Palm/slap
- Similar system to ki-blasts requiring a set amount of time to execute but with a 100% execution rate
- For example:
  - Fist moving slowly but still hitting a target counts as a strike, it just doesn't do much damage
  - Moving too slow will deal no damage
  - Moving fast deals more
    - Moving fast and continuing the fist after collision (Like you would a actual punch)
    - Will increase the force applied and damage dealt (These are two separate systems)
      - Force = distance pushed and ki shield damage
      - Damage = ki shield and health
        - Ki shield does NOT have to be broken to deal hp, it only mitigates a %
- Example clash:
  - Two fists striking one another would have several states
    - If the player and enemy rapidly strike, the resulting shockwave and impact is intense
    - If both rapidly strike AND push forward, the resulting shockwave is even more impactful
- Specific moves:
  - Uppercut
    - Same system as basic skills like ki blast
    - Fist
    - Move arm to waist
    - After leaving the 'waist zone' fist must reach target in a given amount of time else it won't count
  - Axe/Downward spike:
    - Two fists
    - Put hands together above head
    - Rapidly move both downwards
- Blocking:
  - Cross both arms
- Specific interactions:
  - Uppercut
    - Striking stomach OR chest = wind
    - Striking jaw = knock up
  - Axe
    - If on ground = stun
    - If in air = send flying down to ground
  - One Inch Punch
    - Palm -> Fist with rapid movement
    - Distance to enemy should be tiny

**Environment:**

Technical implementation to world construction

-> Chunk segmentation (Unreal Engine 5 world partition grid for example)

Designing world interaction at such scale and for the purpose of Dragonball is an open ended, highly expandable section. For the sake of combat gameplay interacting with the human world would provide little benefit to the player, outside of a Monster Hunter style food consumption system for temporary benefits.

Food vendors:

While mechanics are at the forefront it is important that there is some form of sub-game beneath the combat, this is one of many ways to implement that by bringing in the world and taking a step to truly fulfil the term “Dragonball **VR**” (Not “Dragonball combat VR experience”). Food can provide temporary benefits to the few stats which exist, for example:

- **Pad Thai:** +15% Fighting Power in your next fight
- **Dumplings:** +33% Ki recovery/charge speed in your next fight
- **Ramen:** Ki shield lasts 2x as long

Players may only purchase a single item of food prior to battle, encouraging deeper thought behind the bonus said player wishes to have for their next fight.

Social activities:

Taking regular activities which can be ported to VR can have separate modes to take advantage of the Dragonball theme, most of which can be ported from the combat system. An example of this is volleyball. Standard mode and DB mode.

Sight seeing:

The most fundamental aspect of VR is the visuals, thus providing a large open world (contained or not) can allow for an enormous amount of iconic locations to be brought to life in a new way. Dragonball's style also benefits VR from a technical standpoint, using standalone headsets as a basis a flat shading model with lower poly counts carried by art-style is easier on mobile hardware.

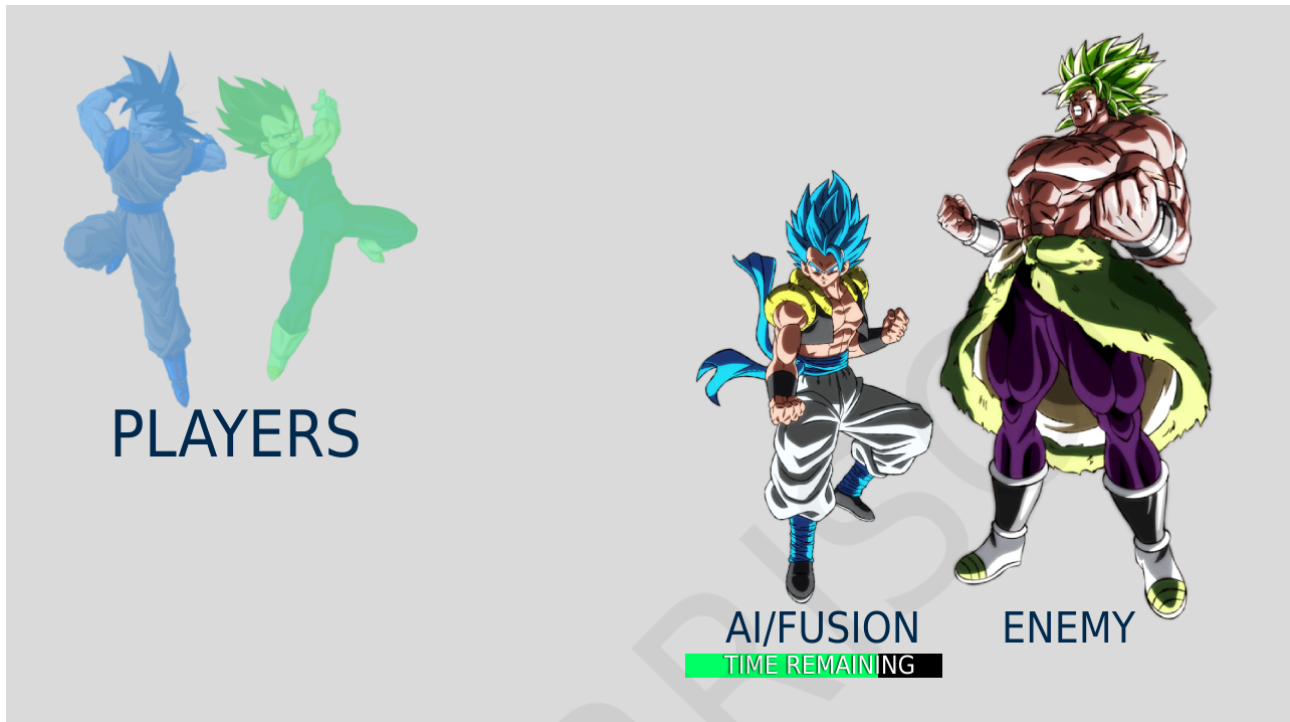
**Multiplayer:**

Players vs AI is a straight forward implementation and most aspects of combat don't have to be altered but rather expanded. For example, co-op moves are an extra addition which don't break the established mould. This also applies to enemies, little is needed for co-op to fundamentally work.

Issues regarding multiplayer arise for two major aspects, fusions (Vegito, Gogeta, Gotenks, etc.) and PvP.

Multiplayer: Fusions

Merging two players requires an entirely new set of mechanics to best suite the system. In order for this to work the fusion body is an AI driven entity where both players actions drive the AI. Enemies can only target this AI with the players being ghosted out. Both players move along with the fused AI and are glued to the target they are attacking if within range for melee combat.



While in a fusion form players must cooperatively take part in mini games to get the AI to do moves, note that the fused AI's moves are significantly more powerful than either players. Mini-games are associated with the move they are going to perform, as an example, a fusion ki blast flurry is done by both players engaging in a mini game where they must shoot their target/s via key blasts.

As stated previously for melee strikes both players glue to the target, there are a series of inputs which can occur to define moves:

- General punches, unsynchronised, cause a generic attack
- General punches, synchronised, cause a generic but more damage attack
- Special moves, unsynchronised, cause a special attack but its effect won't apply. E.g. Double axe can't slam opponent down to the ground
- Special moves, synchronised, cause a special but more damaging attack, with the specials effect being activated and more potent. E.g. Double axe slams opponent down faster

Time takes place of health for a fusion ticking down constantly as well as also being reduced when taking damage. Solid cooperation between players makes a fusion a more guaranteed time of safety while also providing maximum damage, poor cooperation makes fusions short lived (ever more so than if it was just health) and thus relatively ineffective.

#### Multiplayer: PvP

PvP in VR is vastly more dependant on genre than flat-screen, the reasoning is due to interaction. VR interaction is fundamentally different to flat-screen, causing many aspects which work fine in flat-screen to not work in VR. Chief among these issues is melee combat and movement.

#### Multiplayer: PvP | Melee combat

Melee combat in VR is a technical challenge and for true immersion requires a design which alleviates ALL issues. For an open-ended, robust, dynamic combat system such as what we have created so far there are a lot of elements to take into consideration.



### Step 1) No collisions (physics)

Given the nature of the game collisions don't add much value or help solidify combat. Strikes are performed rapidly so any form of collision which might help convey impact isn't actually going to do much in the grand scheme. Many effects or physics based effects can be depicted through other means

### Step 2) Oversized hitboxes

With as fast paced as the game can get as well as the open worlds combat takes place in, adding precise collisions will be a significant detriment to consistency, quickly turning into a frustration. Skill floor and skill ceilings should not be too high to harm entry level play and high level frustration via minute movements causing a failure in combat.

Hitboxes can be generalised to an area around the player for the most part which should be scaled based on distance (Larger when further away). Scaling at a distance assists in skills being easier to hit, along with previously mentioned homing targeting abilities. **MOVE LAST SENTENCE UP TO CORE COMBAT SECTION.** Upon being close to a player, hitboxes should be swapped for a higher precision ones, however still generalised to core areas; limbs, head, torso, feet and hands. AI based opponents do NOT require this higher level of precision for the following reasons:

- > Animation system and animations used makes the chance not intersecting precise hitboxes small
- > With mass amounts, performance is improved when sticking to generalised bounds
- > General improved consistency irregardless of situation, more rewarding for player, suites PvE combat

### Step 3) Indicators, VFX, SFX

Several effects can be applied to several aspects such as where hits are coming from, making player arms more visible to improve visual clarity, VFX and indicators to explicitly state what is going on and SFX cues to tie audio and PvP together.

Some examples:

- Make enemy body slightly darker and arms slightly lighter when close to player to improve visual clarity
- Do screen based hit indicators to clearly define where strikes hit and the damage it dealt
- Add VFX and SFX to blocking strikes to make blocking an explicitly clear action
- Add distinct SFX to large melee attacks such as 'Double Axe' to clearly define a successful attack
- When at a distance from one another default to no targeting system to improve immersion, targeting system should only apply in the following scenarios:
  - Target is moving at high speeds
  - Target is extremely far away