

# Beats (Drive)

## Overview

Name	Design Status
Beats General	READY
Beat bars	READY
Light sparks	READY
Music	READY

---

## Beats General

*The beats of the music will provide the basic drive of the gameplay.*

✓ [Click here for more details](#)

## Beats

- The gameplay is based on rhythms throughout the entire game
- The beats will suggest the player actions and will decide on the player's performance
- Scoring for a beat requires the player to do input in the right time, "on the beat"
- The player will have to turn on and off and adjust lights in the club according to the beats

## Properties of beats (music theory)

- A single beat according to music theory can be analysed by the following
  - Pitch or **Frequency** (bass, mid-range, high, etc)
  - Dynamics (loudness)
  - Tone colour (bright, dark, rich, mellow, etc.)
  - Duration (very short effect, long effect, etc)

## Frequency

- The visualised beats of the music will be the main drive of the game
- The visuals will be run in real time with the music
  - There will be no time line to show past or future beats
  - Only the current beats will be presented
- The presentation will be created via the calculation of the frequency of each beats
  - Each beat's frequency can be detected to present a positive whole number
  - Measurement in hertz is not relevant for the game calculations
- Depending on the frequency of the a beat, there will be three frequency categories:
  - Low frequency or BASS :
  - Medium frequency or MID-RANGE :
  - High frequency of TREBLE :
- Every beat will fall into exactly one category only
- As the music plays, there are several beats heard at a time
  - In continuous sessions, this makes up the music as a whole
- The beats will be visualised as 3D bars that will have specific heights depending on the frequency
- Strong beats will have taller bars and weak beats will have shorter bars
- When a beat is strong enough and it's bar is tall enough it produces a light spark in a form of a bubble
- The player will interact with these sparks to enable the club lights

### Design Note

If we can differentiate the sounds in the music and combine them with the lights to provide a consistent design, then so will the player. This will give them a sensation of fairness as they have the ability to learn the game. They will then be able to connect the music with the visuals and can use both for decision making instead of only the visual representation.

## Beat bars

*The music will be visualised as 3D bars in the world*

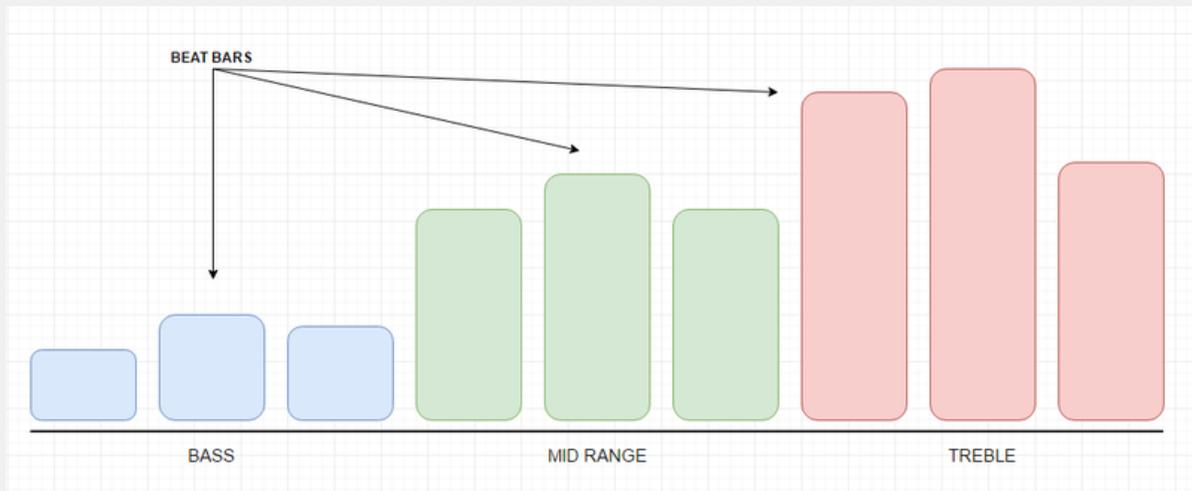
✓ [Click here for more details](#)

### General

- The music will be visualised around the player in a 3D form
  - A half circular, semi transparent surface will home the beats
  - There will be 3 types of frequencies providing the main separation between the beats:
    - Bass
    - Mid-range
    - Treble
  - The different frequencies are visualised on the three parts of the half circle
    - One frequency type will have 3-5 bars that visualise one beat each
      - This number might change, please make it flexible
- The different types of beats should be visualised in different ways to give feedback to player about their nature
  - Colour differences
- The "beat bars" will grow and shrink on the beat, depending on their strength
  - The stronger the particular beat, the higher its bar
  - If a frequency type is missing from the current music, the bars have a height of 0.

More information about sound frequencies and their detection is [available here](#).

## Beat bar rules

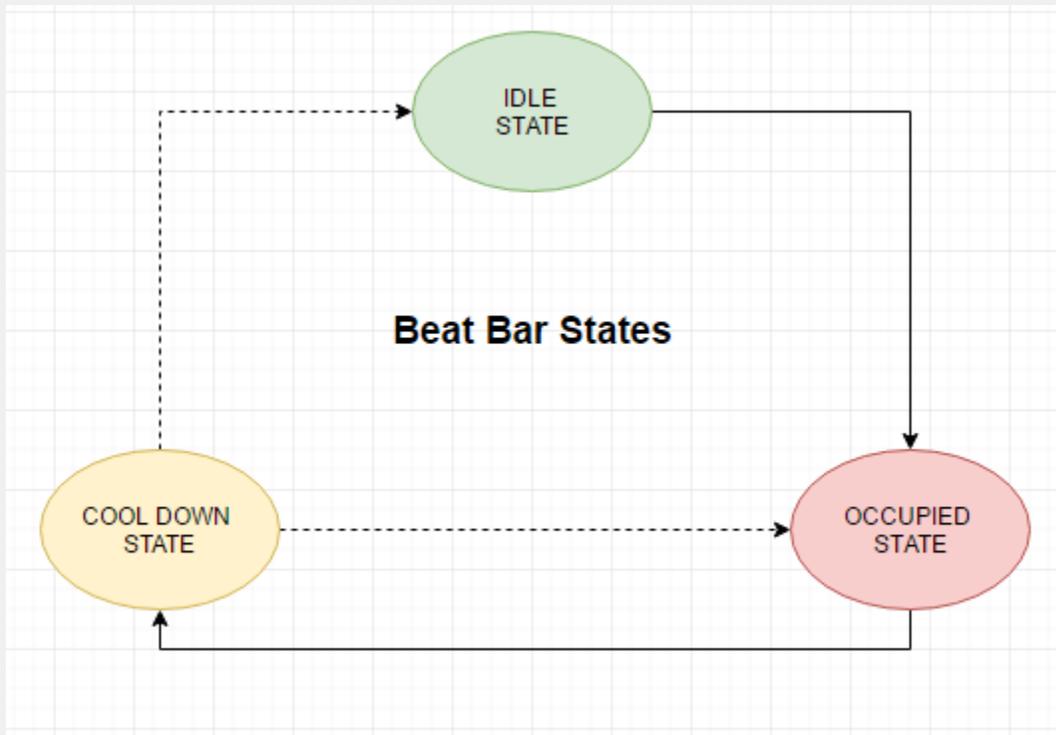


- There will be a visual representation of the continually playing beats (or the music)
  - This will be dynamically changing with the music to show all current beats
- A 120 degree circular 3D platform will be positioned in front of the player that will hold the beat bars
  - Every beat bar represents exactly one beat only
  - The beat bars will grow and shrink according to that beat's frequency
    - The height of the bars will be linear to the amount of frequency
    - While a lower frequency beat will be shorter a higher frequency beat will be taller
- The beat bars will be positioned next to each other
  - Since we don't know the exact number of beats at a time, there will be a set number of reserved bars for the beats as follows:
    - 3 reserved bars for bass
    - 3 reserved bars for mid-range
    - 3 reserved bars for treble
  - The reserved bars will be positioned in the following order:
    - 3 bass -> 3 mid-range -> 3 treble
- The bars will be present even when there is no beat, they will be just so short that they will be hidden from the player
  - The bars will start to appear when a beat is played
  - Since there is limited space for the beats, there will be specific rules on what beats to show and how to show them
- Every beats have the following states:
  - Evaluation state
  - Assignment state
  - Representation state
- The beats will be shown according to the following set of rules:
  - When a beat is played it is evaluated by the above mentioned frequency detector
    - Every beat is evaluated
  - Depending on the evaluation, it is categorised into one of the frequency categories
    - Bass, mid-range and treble
  - Once successfully categorised, the beat will be assigned to the next free beat bar in its category
    - A free beat bar is a bar that is not currently showing any beat
    - If there is no free beat bar, the beat will be compared to those taking up the space
      - It will take over the space of the smallest or the weakest beat in that category
      - A beat can only take over the beat bar of another beat within its category

*Example: there is a bass type beat under assignment. There is no space in the bass bar and the beat is weaker than any of the currently shown beats, there is a free space in the treble section. The bass type of beat under assignment cannot be assigned into the free space in the treble section because it is the wrong category*
    - If it is smaller than all the beats already shown, then the beat is not assigned anywhere
    - Re-assignment of beats will be not highly likely to occur in the game, but the logic should be created for edge cases
  - Once assigned, the visual "strength" of the beat is represented in a form of a grown beat bar
    - Since the height of the beats are linear to their frequency amount, bass type beats will be always smaller than mid-range types which will also be smaller than treble types
  - Due to the nature of the music and the simultaneously playing beats, the above mentioned evaluation and assignment logic of beats must be executed in a multi-threaded environment to enable continuous and smooth representation of the music
  - A beat is considered to be finished when it can no longer be detected by the system
    - Detecting and hearing a beat is a matter of seconds or even less than a second

- When a beat is "finished" the beat bars act in a specific way Beat bars have the following states:
  - Default idle state
  - Occupied state
  - Cool down state

## Beat bar states



- **Idle state - state 1:**
  - There is no beat represented on the bar
  - The height of the bar is 0
  - The bar is ready for a beat to be assigned
  - The bar goes into only occupied state when a beat is assigned
- **Occupied state - state 2:**
  - A beat is currently assigned to the bar
  - The height of the bar in its previous state changes to the calculated height of the bar depending on the beat's evaluation, regardless of what height it was before  
Example: the beat's evaluation tells that the beat should show a bar that is 58 tall. If the bar was 0 tall before, it instantly becomes 58 tall as well as if it was 35 tall when the beat was assigned
  - The bar is in this state until the beat is considered to be "finished"
  - In this stage the bar cannot be reassigned however this stage only last for a fraction of a second
  - The bar goes into cool down state when the beat is finished
  - From this state the bar can only go into cool down state
- **Cool down state - state 3:**
  - No beat is currently assigned to the bar
  - The bar is free to be occupied
  - The height of the bar is constantly shrinking over time, slowly until it hits 0
  - If the bar reaches 0, it becomes idle state
  - It is possible to assign a beat to this bar even when the bar is shrinking. When a new beat is assigned, the bar goes back to occupied state

∨ [Previous design versions and changes](#)

## Version 1

### General

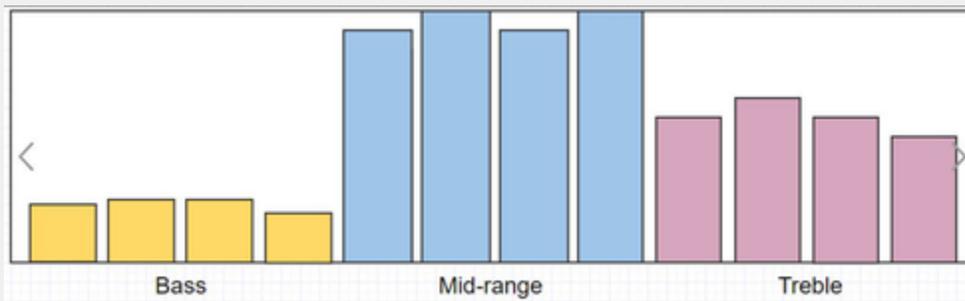
- The music will be visualised on an in-game virtual display in the VJ booth
- The different types of beats should be visualised in different ways to give feedback to player about their nature
- Although the visualisation will be programmed to show the music, it will not serve as the main scoring system

#### Design Note

Having a real-time visualisation where the player only sees the current beats will not allow us any foreshadowing. The player won't be able to learn to press the buttons on the right time if he doesn't know they are coming. In the real world, VJs do not know the music that will come and they just guess the lights. They also press buttons a little late but for most people it isn't noticeable. A clear visual instruction must be given to the player of what to press so he can master the game.

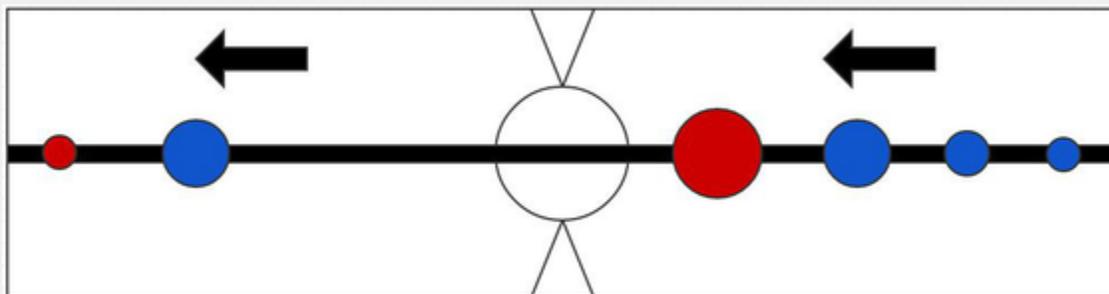
- Providing a "free" or "bonus" mode in the game where the player can press the buttons freely to create more lights beyond the minimum or general gameplay is possible.
- The clear feedback and the free mode can be combined into one experience

### Music visualiser



- Chosen style of 2D visualisation
  - Shows at least 3 types of beats in real time  
*Example: Bass, mid-range, treble*
- The 3 frequency types are extracted from the music in real time
- The bars symbolise the volume of the specific beat  
*Example: The louder a low frequency beat (bass), the higher the bar*
- The bars change with every beat
  - Change depends on design: can be linear or hold interpolation

### Timeline visualiser



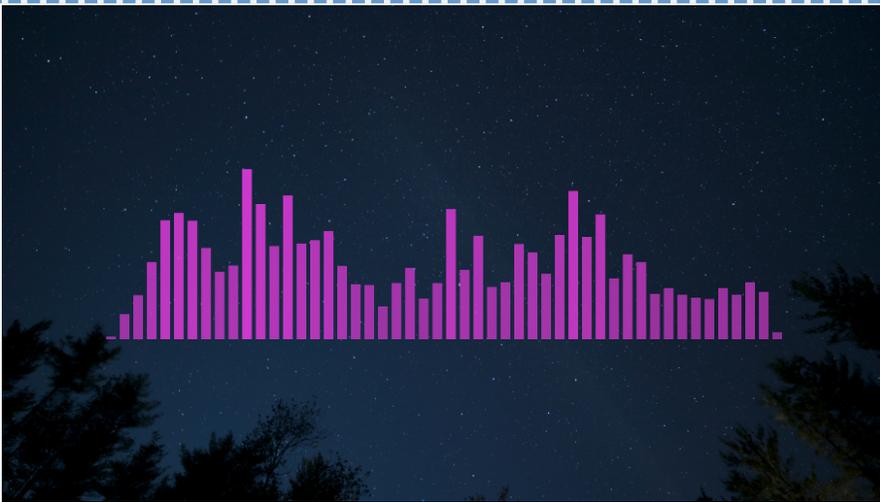
- Shows the required input on a timeline
- Scrolling right to left where left is the past beats and right is the future beats

### **Changes and decisions**

#### Visualiser design!

- The 2D visualiser connected to a mixer platform does not take advantage of the principles of VR. The game in this design could also work on PC or even mobile.
  - Adding a 3D visualiser where the scoring requires broad movements of the player will have a stronger emphasis on the VR capabilities
  - The dynamics of the new design will provide a more active interaction system with a lot more dynamic body movements

▼ [Click here for references](#)

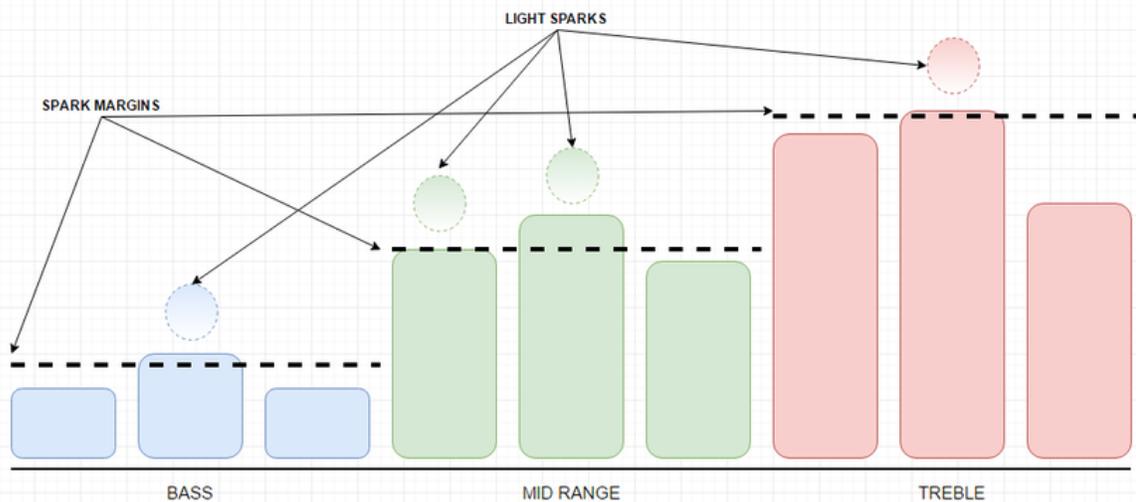


## Light Sparks

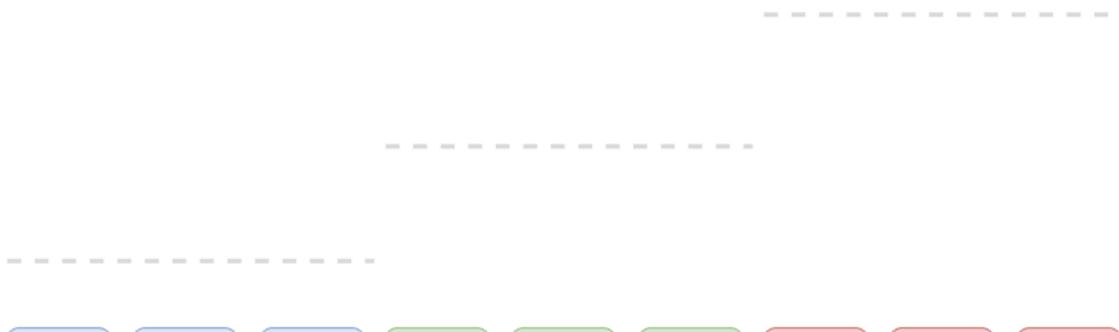
Strong beats generate light sparks that the player can interact with to trigger lights and score.

✓ [Click here for more details](#)

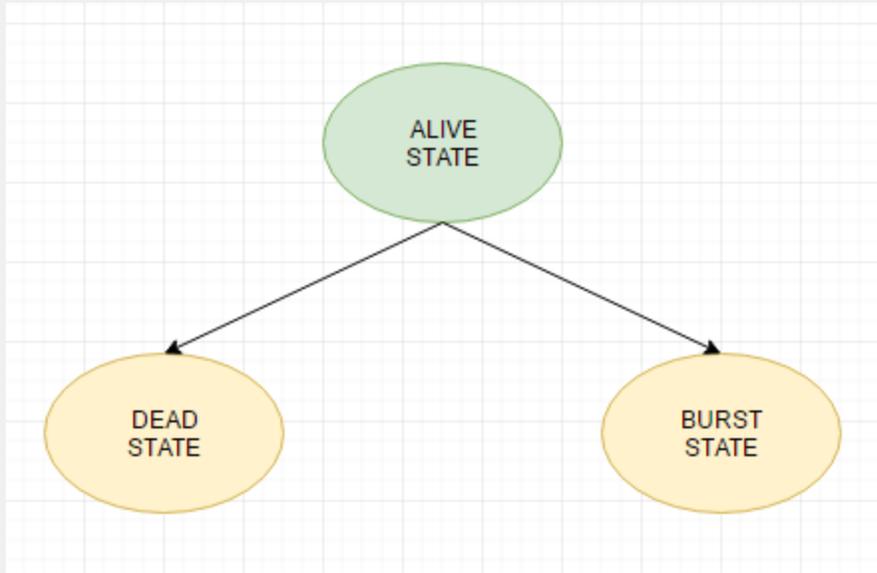




- Light sparks are generated by strong beats
  - Strength is relative within the categories
- Above a specific amount of frequency in each category, the beat is considered to be a strong beat. This amount of frequency is called the spark margin
- There are three spark margin numbers, all positive numbers:
  - Bass spark margin
  - Mid-range spark margin
  - Treble spark margin
- The margins are not static as the music isn't static either
  - The margins are calculated based on an average of the most recent beats in the category
  - **Exact rules for the calculations to be designed!**
- The spark margin can be comprehended as a horizontal line
  - When a beat goes beyond the spark margin, it produces a light spark
- Light sparks last only for a very short time, like for a fraction of a second or so, then they "die"
  - While the sparks are alive, they can be interacted with, by the player
  - When the player interacts with a spark while it's alive, it enables the assigned club light(s) on the scene
    - For more information on club light assignment, please see the Lights page
    - For more information on input assignments, please see the Player page
- Interacting with the sparks provide the players point
  - For more information about scoring, please see the Scoring page
- The player's goal is to interact with as many spark as possible as quickly as possible, as they have very short life.



## Light spark states



- The sparks have the following 3 states:
  - **Alive state - state 1:**
    - This is the first stage of the lifetime of the light spark
    - The spark remains in this state for its "lifetime" (example: 1.5 seconds) until
      - It is burst by the player and therefore goes into burst state or
      - The lifetime expires and the spark dies and therefore goes into dead state
    - During its lifetime the spark slowly floating upwards
    - The spark has a visual effect representing the assigned club light inside the bubble shimmering, twirling, etc
  - **Burst state - state 2:**
    - The spark becomes burst when the player interacts with it
      - This provides the player score
    - When burst, the assigned disco lights are triggered and enables
    - The light effect inside the bubble also bursts or explodes and rapidly fades
  - **Dead state - state 3:**
    - If there is no player interaction before the spark's life time is over, it dies
    - A dead spark provides no score and triggers no club lights
    - The light effect inside the bubble turns to smoke or ash as rapidly vanishes like a ghost or a shadow

## Music

*The music will gradually build up with more sounds and effects.*

✓ [Click here for more details](#)

### General

- Dynamic build-up, electronic music
- Should have 5 major parts just like levels
- The music will be broken down into beats in a written format here

#### Design Note

The music will be designed by the music department of this project

- The above mentioned points serve purely as guidelines

✓ [Click here for references](#)