Trajectory Hub - Lista TO DO y Estado del Proyecto

1. CONTROLADOR INTERACTIVO - Estado de Implementación

1.1 Gestión de Macros ✓ (100%)

Función	Estado	Descripción
Crear macro	1 00%	Crea grupos de fuentes con comportamiento
Listar macros	1 00%	Muestra todos los macros existentes
Seleccionar macro	1 00%	Establece macro activo para operaciones
Eliminar macro	1 00%	Borra macro y libera fuentes
Duplicar macro	1 00%	Copia configuración completa de un macro

1.2 Control de Trayectorias ✓ (95%)

Función	Estado	Descripción	то ро
Establecer trayectoria	V	Define movimiento principal del	_
macro	100%	grupo	-
Trayectorias individuales	100%	Configura movimiento de cada fuente	-
Modos de movimiento	~	Stop, fix, random, vibration, spin,	
Modos de movimiento	100%	freeze	
Ajustar velocidades	100%	Control de velocidad por fuente	-
Aplicar desfases	100%	Offset temporal entre fuentes	-
Pausa/Resume	▼ 90%	Congela y reanuda movimientos	Mejorar persistencia de estados
Invertir dirección	100%	Cambia sentido de la trayectoria	-

1.3 Control de Distancias **☑** (100%)

Función	Estado	Descripción
Presets de distancia	1 00%	16 configuraciones predefinidas
Rango específico	1 00%	Define min/max personalizado
Distancia fija	1 00%	Establece distancia única
Ver presets	1 00%	Lista todas las opciones disponibles

1.4 Sistema de Deformación / (40%)

Función	Estado	Descripción	то ро
Aplicar respiración	80%	Expansión/contracción rítmica	Verificar integración con engine
Deformación por ondas	1 50%	Ondulaciones en el espacio	Implementar parámetros completos
Campo de fuerzas	<u> 1</u> 30%	Atracción/repulsión por zonas	Crear sistema de fuerzas
Deformación caótica	× 20%	Movimientos aleatorios controlados	Implementar generador caótico
Gestos manuales	× 0%	Control por movimientos del usuario	Sistema de captura de gestos
Combinar deformaciones	× 10%	Mezclar múltiples efectos	Sistema de composición

1.5 Comportamientos y Movimientos 1.5 (50%)

Función	Estado	Descripción	то до
Cambiar comportamiento	1 00%	Flock, rigid, elastic, swarm	-
Movimiento semántico	<u> 40%</u>	Estilos predefinidos (nervioso, flotante)	Conectar con sistema real
Parámetros comportamiento	1 30%	Ajustes específicos por tipo	Aplicar valores al engine
Toggle componentes	<u>1</u> 20%	Activar/desactivar partes	Implementar toggles reales
Reset movimientos	☑ 80%	Volver a estado inicial	Mejorar reset completo

1.6 Interacción entre Macros X (0%)

Función	Estado	Descripción	TO DO
Seguimiento	× 0%	Un macro sigue a otro	Implementar sistema de tracking
Fuerzas mutuas	× 0%	Atracción/repulsión entre macros	Crear física de interacción
Sincronizar movimientos	× 0%	Coordinar múltiples macros	Sistema de sincronización
Órbitas	× 0%	Macro orbita alrededor de otro	Implementar órbitas
Colisiones	× 0%	Detección y respuesta a choques	Sistema de colisiones
Ver interacciones	× 0%	Mostrar relaciones activas	UI de visualización

1.7 Presets y Composiciones 1.4 (60%)

Función	Estado	Descripción	то ро
Cargar preset artístico	1 00%	10+ presets completos	-
Guardar preset	4 0%	Guarda configuración actual	Implementar escritura a archivo
Composiciones temporales	4 60%	Timeline de eventos	Ejecutor automático de timeline
Generador aleatorio	1 00%	Crea configuraciones únicas	-
Exportar/Importar	× 0%	Intercambio de configuraciones	Sistema de serialización

1.8 Información del Sistema 🔽 (90%)

Función	Estado	Descripción	TO DO
Estado general	1 00%	Muestra recursos y configuración	-
Info de macros	1 00%	Detalles de cada macro	-
Estado OSC	☑ 80%	Conexión con Spat	Mejorar diagnósticos

2. RESTO DEL PROYECTO - TO DO

2.1 Core Engine

Componente	Estado	то до	Prioridad	
EnhancedTrajectoryEngine	V 90%	- Optimizar performance con 100+ fuentes -	Media	
Enhanced frajectoryEngine	90%	Implementar LOD (Level of Detail)	iviedia	
Motion Components	✓ 85%	- Completar modos de movimiento faltantes - Sistema	Modia	
Motion Components	0376	de prioridades	Media	
Trajectory Deformers	<u> </u> 60%	- Implementar todos los deformadores - Sistema de	Alta	
Trajectory Deformers	- 60%	composición	Aild	
OSC Bridge	V 95%	- Mejorar reconexión automática - Buffer para alta	Baja	
OSC Bridge	95%	latencia	Баја	
Distance Controller	~			
Distance Controller	100%	_	-	
Macro Behaviors	<u> </u>	- Implementar comportamientos faltantes - Sistema de	Modia	
IVIACIO DELIAVIOIS	- 70%	IA básica	Media	

2.2 Interfaz y UX

Componente	Estado	TO DO	Prioridad
Interactive Controller	<u>↑</u> 75%	- Completar funciones pendientes (ver arriba)	Alta
Visualización	× 0%	- Crear visualizador 3D opcional opcional opcional opcional opcional opcional opcional opcional opcional opcional opcional 	Baja
GUI	X 0%	- Interfaz gráfica (Qt/Web) - Control táctil	Futura
API REST	× 0%	- Servidor HTTP para control remoto - WebSocket para tiempo real	Futura

2.3 Persistencia y Datos

Componente	Estado	TO DO	Prioridad	
Cuandar/Carrar		- Sistema completo de serialización >- Formato estándar	A 14 -	
Guardar/Cargar 30'	30%	(JSON/YAML)	Alta	
Historial	× 0%	- Undo/Redo de operaciones >- Grabación de sesiones		
Presets	1	Castión de procete personalizados chr. Dibliatase compartible	Media	
Usuario	40%	- Gestión de presets personalizados - Biblioteca compartible	Media	

2.4 Integración MCP (Model Context Protocol) 🖈 CRÍTICO

Componente	Estado	TO DO	Prioridad	
MCP Server	× 0%	- Implementar servidor MCP completo - Exponer todas las	CRÍTICA	
MCP Server	7 0 78	funciones del sistema	CRITICA	
Herramientas	× 0%	- Crear macros - Control de trayectorias - Gestión de	CRÍTICA	
МСР	0%	distancias - Sistema de deformación	CRITICA	
Prompts	× 0%	Drompte entimizados pero central des. Fiemples de use con Claude	CRÍTICA	
Sistema	0%	- Prompts optimizados para control - Ejemplos de uso con Claude	CRITICA	
Recursos MCP	× 0%	- Estado del sistema - Configuración actual - Presets	CRÍTICA	
Recursos MCP	0%	disponibles	CRITICA	
Control	× 0%	- "Crea una bandada nerviosa" - "Haz que orbiten lentamente"	CRÍTICA	
Semántico	0%	- "Dispersa las fuentes"	CRITICA	
Feedback	× 0%	- Notificaciones de cambios - Confirmación de acciones -	Alto	
Estado	V 0%	Estado en tiempo real	Alta	

2.5 Integración y Compatibilidad

Componente	Estado	то ро	Prioridad
DAW Integration	× 0%	- Plugin VST/AU - Sincronización con timeline DAW Futur	
MIDI Control	× 0%	- Mapeo MIDI de parámetros - Control por hardware Media	
Automación	× 0%	- Curvas de automación - Scripting avanzado	Futura

2.5 Performance y Optimización

Componente	Estado	TO DO	Prioridad
Multiprocessing	× 0%	- Paralelizar cálculos pesados - Distribución de carga	Media
GPU Acceleration	× 0%	- Cálculos en GPU (CUDA/OpenCL) - Para 200+ fuentes	
Profiling	<u> 1</u> 20%	- Sistema de medición performance - Optimización automática	Baja

3. INTEGRACIÓN MCP - OBJETIVO PRINCIPAL DEL PROYECTO **

3.1 Arquitectura MCP

Componente	Descripción	Estado	то ро
MCP Server	Servidor que expone el sistema a lAs	× 0%	- Crear (trajectory_mcp_server.py) - Implementar protocolo MCP completo - Gestión de sesiones
Tool Registry	Registro de herramientas disponibles	× 0%	- Mapear todas las funciones a tools MCP br>- Documentación de parámetros br>- Validación de inputs
Resource Provider	Proveedor de recursos/estado	× 0%	- Estado actual del sistema - Lista de presets - Configuraciones activas
Prompt System	Sistema de prompts optimizados	× 0%	- Prompts para cada operación - Ejemplos de uso natural - Guías de contexto

3.2 Herramientas MCP Necesarias

Herramienta	Función	Ejemplo de Uso	Estado
<pre>create_sound_group</pre>	Crear macro	"Crea un grupo de 20 pájaros"	X 0%
<pre>set_movement_pattern</pre>	Establecer trayectoria	"Haz que vuelen en círculo"	X 0%
control_behavior	Cambiar comportamiento	"Que se muevan como bandada nerviosa"	X 0%
(adjust_distances)	Control de distancias	"Acércalos más, que estén íntimos"	X 0%
(apply_effects)	Deformaciones	"Añade respiración oceánica"	X 0%
(manage_interactions)	Interacciones	"Haz que el grupo A siga al B"	× 0%
(load_preset)	Cargar configuración	"Carga el preset Galaxia"	× 0%
<pre>get_system_state</pre>	Estado actual	"¿Qué está pasando ahora?"	× 0%

3.3 Control por Lenguaje Natural

Capacidad	Ejemplos	Implementación Necesaria
Creación Semántica	"Crea una bandada de pájaros nerviosos" "Genera un enjambre caótico de 30 fuentes"	Parser de intenciones br>Mapeo semántico a parámetros
Control	"Hazlos más lentos" "Sepáralos un poco	Sistema de modificadores
Dinámico	más" "Que vibren sutilmente"	relativos Contexto de estado actual
Composiciones	"Crea una escena de tormenta" "Ambiente de catedral con ecos"	Templates semánticos Combinación de elementos
Queries Estado	"¿Cuántas fuentes hay?" "¿Qué preset está activo?"	Sistema de consultas Formateo de respuestas

3.4 Implementación MCP Propuesta

```
# trajectory_mcp_server.py
class TrajectoryMCPServer:
    """Servidor MCP para control de travectorias por IA"""
    tools = {
        "create_sound_group": {
            "description": "Crear un grupo de fuentes sonoras",
            "parameters": {
                "name": "Nombre descriptivo del grupo",
                "count": "Número de fuentes (1-100)",
                "behavior": "Comportamiento: bandada|rígido|elástico|enjambre",
                "shape": "Formación inicial: círculo|línea|grid|espiral"
        },
        # ... más herramientas
    }-
    async def handle_tool_call(self, tool_name, parameters):
        """Procesar llamadas de herramientas desde la IA"""
       # Mapear a funciones del sistema
```

4. SIGUIENTES PASOS RECOMENDADOS (ACTUALIZADO)

Fase 0: Integración MCP AAAA (MÁXIMA PRIORIDAD - 2 semanas)

1. Crear Servidor MCP Básico

- Implementar protocolo MCP
- Exponer herramientas principales
- Sistema de recursos/estado

2. Mapear Funciones Core

- create_sound_group → create_macro
- set_movement_pattern → set_trajectory
- control_behavior → change_behavior
- adjust_distances → distance_control

3. Parser de Lenguaje Natural

- Interpretar comandos semánticos
- Mapear a parámetros técnicos
- Manejo de contexto

4. Testing con Claude

- Configurar Claude Desktop con el servidor
- Probar comandos naturales
- Ajustar prompts y respuestas

5. **Terminar Sistema de Deformación** Alta prioridad

- Implementar deformadores faltantes
- Sistema de composición funcional

6. Interacciones entre Macros 🗭 Alta prioridad

- Al menos implementar seguimiento y fuerzas básicas
- Fundamental para composiciones complejas

7. Guardar/Cargar Completo 🖈 Alta prioridad

- Serialización de toda la configuración
- Formato estándar para intercambio

Fase 2: Estabilización (1 semana)

1. Testing Exhaustivo

- Pruebas con 100+ fuentes
- Casos edge y manejo de errores

2. Optimización Performance

- Profiling de puntos críticos
- Optimizaciones básicas

3. Documentación

- API completa
- Guías de usuario
- Ejemplos avanzados

Fase 3: Mejoras UX (2-3 semanas)

1. Timeline Automático

- Ejecutor de composiciones temporales
- Editor de timeline básico

2. Visualización Básica

- Vista 2D simple de posiciones
- Ayuda para debugging

3. MIDI Básico

Control de parámetros principales

• Mapeo configurable

Fase 4: Expansión (Futuro)

- 1. GUI Completa
- 2. API REST
- 3. Plugin DAW
- 4. GPU Acceleration

4. MÉTRICAS DE COMPLETITUD

Por Módulo Principal:

• Integración MCP: 0% X 🙀 CRÍTICO

Gestión de Macros: 100%

• Control de Trayectorias: 95% V

• Control de Distancias: 100%

• Sistema de Deformación: 40% 4

Comportamientos: 50%

Interacciones: 0% X

Presets: 60%

• Core Engine: 85% 🔽

TOTAL PROYECTO: ~55% Completo (considerando MCP como objetivo principal)

Funciones Críticas Faltantes:

- 1. Interacciones entre macros (impacto alto)
- 2. Sistema de deformación completo (impacto medio-alto)
- 3. Guardar/Cargar real (impacto alto)
- 4. Timeline automático (impacto medio)

Tiempo Estimado para v1.0 Completa:

Con 1 desarrollador: 4-6 semanas

• Con 2 desarrolladores: 2-3 semanas

Incluyendo testing y documentación

5. RECURSOS NECESARIOS

Desarrollo:

- Python developer con experiencia en:
 - Sistemas en tiempo real
 - Procesamiento de audio/MIDI
 - Protocolos de red (OSC)

Testing:

- Acceso a Spat Revolution
- Sistema con 100+ canales
- Beta testers con experiencia en audio 3D

Documentación:

- Technical writer (opcional)
- Diseñador para diagramas (opcional)