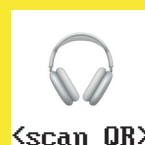
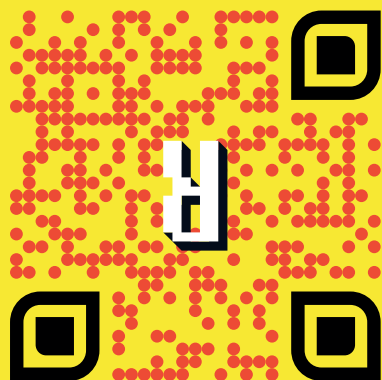


```
#include<polito.lv11>
int main(DISSERTATION){
printfTITLE("WhiteRabbit\n");
printfSUB("Progettazione di un
servizio di distribuzione musicale
digitale\n")
}
```



designed for
curious listeners and enterprising
artists\n");
}

/listen
start
listening
session

<scan QR>



<2RX35>



insert key upload
song

@followthewhiteR_bot

/manifesto
claim (Follow the white R)
#include <telegram.apk>
printf("whiterabbit is a music bot
designed for
curious listeners and enterprising
artists\n");
}

<<MANIFESTO>> send /manifesto
claim (Follow the white R)
#include <telegram.apk> int main(
printf("whiterabbit is a music bot
designed for
curious listeners and enterprising
artists\n");
}

insert key upload
song



<scan QR>

<2RX35>

@followthewhiteR_bot



<scan QR>

<2RX35>

insert key upload
song

@followthewhiteR_bot

<<MANIFESTO>>
send /manifesto
claim (Follow the white R)
#include <telegram.apk>
int main(void)

designed for
curious listeners and
enterprising
artists\n");
}

/listen
start
listening
session



**Politecnico
di Torino**

WHITERABBIT. PROGETTAZIONE DI UN SERVIZIO DI DISTRIBUZIONE MUSICALE DIGITALE

**DIPARTIMENTO DI ARCHITETTURA E DESIGN
DESIGN E COMUNICAZIONE VISIVA
TESI DI LAUREA DI PRIMO LIVELLO**

Candidato: Nicolò Lojero
Relatore: Luca Cattaneo


```
int main(INDEX){
printf INDEX("

    page7 <Premessa 🙌 >
    page9 <Abstract ? >
    page11 <Scenario 🖼️ >
    page19 <Casi studio 🖋️ >
    page25 <Personas 👤 >
    page29 <Concept 🎯 >
    page33 <Progetto 🐇 >
    page47 <Visual identity ID >
    page59 <Prototipo 📱 >

\n");
}
```



Un artista indipendente che oggi vuole diffondere la sua musica attraverso le piattaforme di streaming più note si deve affidare a un distributore online: un medium digitalizzato che mette in contatto diretto l'autore e gli ascoltatori.

Distrokid, CD Baby, TuneCore e molti altri offrono servizi di distribuzione e raccolta di royalties con il fine ultimo del guadagno dell'artista e/o del distributore stesso.

Ma cosa succede una volta che la musica viene caricata su Spotify o iTunes? Molto spesso niente!

I servizi di streaming che tutti utilizziamo ogni giorno privilegiano quegli artisti che vengono *posizionati* in una playlist editoriale - una playlist curata e promossa dalla piattaforma che genera ascolti sicuri - perché è attraverso queste che gli utenti scoprono nuova musica.

Difficilmente quindi, un artista emergente non posizionato genera ascolti consistenti e il risultato è tantissima musica che non verrà mai scoperta.

La creazione di una fanbase è la chiave del successo di un progetto musicale e una playlist editoriale, nel caso in cui la si raggiunga, non equivale comunque al consolidamento di un pubblico, ma più verosimilmente ad ascolti che passano inosservati: dei numeri, generati grazie a una coda di brani sconosciuti riprodotti automaticamente da migliaia di utenti.

Una playlist insomma, come si dice, genera ascolti... ma non ascoltatori.

È possibile quindi rielaborare alcuni dei meccanismi che hanno caratterizzato la fruizione di musica legata ai supporti fisici, i suoi ecosistemi e le sue connessioni, digitalizzarli e creare un sistema di distribuzione e riproduzione basato su interazioni reali che tutelino gli artisti indipendenti e al tempo stesso aiutino gli ascoltatori a scoprire sempre nuova musica?



WhiteRabbit è un sistema di **distribuzione musicale digitale** e interattivo sviluppato su un **bot** di **Telegram**.

Gli ambiti della **digitalizzazione** e della **scoperta** nel campo della fruizione di musica sono indagati a partire dal fenomeno atipico dei **bootleg**: con questo termine vengono identificate dagli anni sessanta tutte quelle produzioni musicali - demo, registrazioni live, cover, remix - diffuse senza l'autorizzazione del proprietario dei diritti. Questo meccanismo di aggiramento delle istituzioni, spesso **senza scopo di lucro** e dalla vena **anti-sistema**, costituisce a tutti gli effetti un sistema di distribuzione **indipendente** che non ha nulla in comune con la pirateria ma che rappresenta piuttosto un momento di **condivisione** di contenuti esclusivi.

La diffusione di queste produzioni ha subito, in maniera analoga a quella ufficiale, una **trasformazione digitale** che ha **dematerializzato** i supporti fisici tradizionali e ne ha delegato la distribuzione delocalizzandola attraverso la rete di **internet**. Il processo di **digitalizzazione** non ha però intaccato il ruolo di **scopritore** dell'utente: in passato costui era legato alla dimensione fisica dei supporti musicali e al collezionismo, oggi, in modo analogo, la scoperta passa attraverso l'accesso a delle **reti peer-to-peer**. L'esclusività data dal possesso di un disco in vinile a tiratura limitata viene meno in favore di un **download link** postato su un forum o condiviso sui social network.

"Follow the white R" è il claim del progetto WhiteRabbit che si colloca in continuità con lo spirito dei bootleg, il riferimento al **Bianconiglio di Carroll** infatti, introduce il tema della **scoperta** ed è su Telegram che il sistema di distribuzione assume la forma di un bot. WhiteRabbit offre un **servizio** completamente **digitalizzato** e **automatizzato** di distribuzione musicale indipendente con il quale l'**utente** può interagire nel ruolo di **ascoltatore** o di **artista** che desidera diffondere la sua musica attraverso di esso. I meccanismi di **interazione** veicolano l'ascoltatore a stabilire una **connessione diretta** con gli artisti che apprezza contribuendo così al consolidamento della **fanbase**.

La scoperta del sistema avviene non solo grazie al **passaparola** ma anche attraverso dei **QR code-stickers** che vengono recapitati ad ogni utente che intende supportare il progetto con una donazione. Una volta attaccati in giro per la città questi forniscono un punto di accesso diretto al bot per artisti e curiosi.

WhiteRabbit non è una soluzione pensata per scoprire musica di autori affermati: è una digitalizzazione di un **fenomeno underground** che vuole rimanere underground attraverso la promozione di musica indipendente. È un **mediatore digitale** fra artisti emergenti e utenti alla ricerca di novità.



CHE COS'È LA DISTRIBUZIONE MUSICALE? L'INTERMEDIARIO FRA GLI ARTISTI E GLI ASCOLTATORI

Quando si parla di **distributore** in ambito **musicale** oggi ci si riferisce a un particolare **servizio gratuito** o a **pagamento** a disposizione di chi sta **producendo musica** e vuole **diffonderla** attraverso la **piattaforme** più utilizzate per l'ascolto in **streaming**.

Attraverso un **sistema** di **caricamento online** della propria musica ogni artista è in grado in pochi minuti di avviare una **distribuzione digitale capillare** e iniziare a costruire un **catalogo** di opere che potranno generare delle **royalties** - compensi riconosciuti al proprietario di un'opera - sulla base del numero di ascolti totalizzati.

ARTISTI > DISTRIBUTORI > STREAMING > ASCOLTATORI

*"L'obiettivo della distribuzione digitale è quello di far arrivare la tua musica su iTunes, Spotify, Google Play ed altre piattaforme / negozi digitali di musica."*¹

Prima che i servizi di streaming capovolgessero gli equilibri di un sistema basato sui supporti fisici (CD, vinili e audiocassette) i **distributori** erano gli **intermediari** che gestivano una rete di **collegamento** con i **venditori** finali dei **supporti**: negozi di dischi, centri commerciali ecc.

The logo for 'amuse' is displayed in a bold, black, sans-serif font on a bright yellow rectangular background.

DISTROKID

The 'tunecore' logo is shown in a white, lowercase, sans-serif font on a solid black rectangular background.

Alcuni servizi online di distribuzione digitale

Ancora oggi **una parte** della distribuzione è basata su queste dinamiche, ma **include** soprattutto **artisti già affermati**: artisti che vendono.

¹ N. Donà, Tutto Ciò Che i Musicisti Devono Sapere Sulla Distribuzione Digitale della Musica, 30/06/17, blog. landr.com, <https://blog.landr.com/it/tutto-cito-che-musicisti-devono-sapere-sulla-distribuzione-digitale-della-musica/>



(3;4) 1969, Bob Dylan, GREAT WHITE WONDER



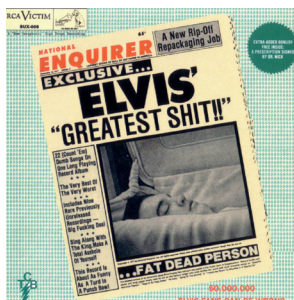
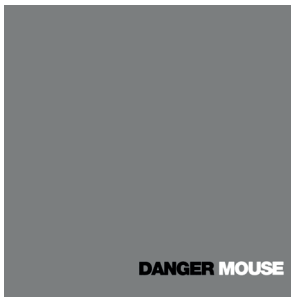
BOOTLEG, UNA PRODUZIONE FUORI DAL MERCATO

1969. Due losangelini, Dub **Taylor** e Ken **Douglas**, fondatori di una neonata etichetta discografica, **ottengono** una serie di **registrazioni demo** da **Bob Dylan** e le portano in una stamperia di vinili a Los Angeles, in California. Riescono a convincere lo stabilimento a **riversare** le registrazioni in **vinile** e a **stamparne** circa **duemila copie**, pagano in contanti e usano nomi falsi per mantenere l'anonimato. **Confezionano** i dischi dentro delle **semplici buste di carta** (3) e iniziano a **distribuirli** alle **radio** della costa occidentale.

"Quella del bootleg - il termine viene dai gambali dove i contrabbandieri nascondevano la merce, soprattutto alcol - è naturalmente anche una storia sociale. Nasce in epoca hippy con qualche vena anti-sistema. Ma il gesto di sfida e insurrezione verso l'industria musicale diviene quasi subito un mercato parallelo. All'inizio non ben compreso dalle major e dagli stessi artisti." ²



Il **successo immediato** di un disco stampato e distribuito **senza il permesso** dell'autore e in maniera **indipendente** mette in difficoltà le major e **innesc**a una **reazione** che avrebbe **cambiato** il **mercato discografico** in modo permanente: da quel momento una serie di album contenenti materiale inedito, registrazioni live e cover di diversi artisti vengono stampati e diffusi senza autorizzazione. È l'**inizio** dell'era dei **bootleg**.



(4) I 10 migliori bootleg di tutti i tempi secondo FAR OUT

2 K. Tedde, Bootleg: amori di contrabbando, 02/11/20, [tipomagazine.it](https://www.tipomagazine.it/bootleg-amori-di-contrabbando/), <https://www.tipomagazine.it/bootleg-amori-di-contrabbando/>

4 T. Golsen, From Prince to Bob Dylan: The 10 most essential bootleg albums of all time, 01/05/21, [faroutmagazine.co.uk](https://faroutmagazine.co.uk/the-10-best-bootleg-albums-prince-bob-dylan/), <https://faroutmagazine.co.uk/the-10-best-bootleg-albums-prince-bob-dylan/>



DISTRIBUZIONE E DIGITALIZZAZIONE: AUDIOCASSETTA VS FREEDOWNLOAD DA UN BAGAGLIAIO A UNA RETE PEER-TO-PEER

Il **progressivo sviluppo tecnologico** e i processi di **digitalizzazione** che investono la musica in tutti i suoi aspetti **influenzano** di riflesso anche gli ecosistemi paralleli, come quello dei **bootleg**.

I **vinili** si rimpiccioliscono, si trasformano prima in **audiocassette**, che possono essere **duplicate** in modo **amatoriale** con un semplice lettore stereo e contribuiscono esponenzialmente al **dilagare** del **fenomeno** dei bootleg; poi, si assottigliano nel **Compact Disc** e contemporaneamente vengono **introdotte** le prime forme di **tutela** del **diritto d'autore** che limitano legalmente i fenomeni legati alla duplicazione sempre più facilitata dei supporti; **infine**, attraversando un processo di **dematerializzazione** che segue di pari passo lo sviluppo di **internet**, la **musica** si **digitalizza** in un **file audio**.

Le trasformazioni che hanno portato alla musica liquida - così viene definita quando non necessita di supporti fisici per essere riprodotta - sono importanti anche per comprendere l'evoluzione dei bootleg: da fenomeno fortemente localizzato e dipendente dai supporti fisici a esperienza digitale globale.





Napster v2.0 BETA 7

File Actions Help

Home Chat Library Search Hot List Transfer Discover Help

Filename	File Size	User	Status	Speed	Progress	Rate	Time Left
Comedy - Maclean & Maclean - The Gross Manual (14 Sure Fire Jo...	299,200 of 2,220,560	ldhull	Downloading...	Unknown	13 %	5.06 k/s	00:06:19
(Comedy) Maclean & Maclean - Johnny Marijuana Seed .mp3	270,048 of 7,381,800	mrbrin	Downloading...	Cable	3 %	6.50 k/s	00:18:13
Comedy - Maclean & Maclean - The Gross Manual (14 Sure Fire Jo...	539,936 of 2,220,560	big_bang_	Downloading...	Cable	2.4 %	7.33 k/s	00:03:49
MacLean & Maclean - Bowser and Blue .mp3	5,606,005	adie_m	Getting Info...				
Andrews Sisters - Little Toot .mp3	375,392 of 8,528,129	F4phantom...	Downloading...	DSL	4 %	4.14 k/s	00:32:50
78's CHILDREN'S STORY RECORD -- LITTLE TOOT .mp3	89,512 of 5,840,561	dude_way	Downloading...	14.4	1 %	1.10 k/s	01:26:59
Disney Parks Toontown-Little Toot .mp3	64,608 of 2,354,597	hitemerbitter	Downloading...	Cable	2 %	1.28 k/s	00:29:47
Maclean & Maclean - Little Toot .MP3	12,288 of 602,112	KASDJFLK...	Downloading...	Cable	3 %	0.24 k/s	00:41:39
Disney Parks Toontown-Little Toot .mp3	185,648 of 2,354,597	Creano1	Downloading...	56K	8 %	5.82 k/s	00:06:12
Maclean & Maclean - Little Toot .mp3	210,944 of 882,688	stearch	Downloading...	56K	2.4 %	4.44 k/s	00:02:31
Big Tiny Little & Mickey Finn - Toot Toot Toots .mp3	40,128 of 2,888,844	albear69	Downloading...	Cable	1 %	1.89 k/s	00:25:08
Comedy - Maclean & Maclean - Little Toot .mp3	6,816 of 882,688	catalina2136	Downloading...	DSL	0 %	0.55 k/s	00:26:45
Brenda Lee - Toot Toot Tootsie Goodbye (Little Miss Dynamite, CD...	44,368 of 2,250,710	angel3339	Downloading...	Unknown	1 %	1.10 k/s	00:33:26
MacLean & MacLean - Little Toot (edit) .mp3	115,600 of 1,180,444	123412341...	Downloading...	14.4	9 %	4.68 k/s	00:03:47
Maclean & Maclean - Little Toot .mp3	882,688	raybauduc	Getting Info...				

Filename	File Size	User	Status	Speed	Progress	Rate	Time Left
Crosby, Stills, Nash And Young - Ohio.Mp3	2,725,888 of 3,709,496	cybermoose	Uploading...	56K	7.3 %	1.55 k/s	00:10:33
Lollipop lust kill - jesus chrysler .mp3	3,339,493	4StringMF	File Complete!				
Simpsons - The Life Of Homer Simpson .mp3	1,406,976 of 2,756,736	aolude67	Uploading...	56K	5.1 %	1.45 k/s	00:15:30
Fear Factory - Cars .mp3	3,921,141	tubbedbird	File Complete!				
Comedy - Richard Pryor - New Niggers .mp3	3,796,451	tubbedbird	File Complete!				
Unknown - Mopar (Chrysler) 440 6-pak V8 race-bait - real men don't...	869,049	TYLERSEX...	File Complete!				
Comedy - Jerky Boys - Fanning My Balls .mp3	675,840 of 1,784,264	tubbedbird	Uploading...	14.4	3.8 %	7.33 k/s	00:02:31
Ben Stiller & Andy Dick - Can We Get It Together Again .mp3	1,230,600	fluidpockets...	Waiting...				

Concurrent Downloads: 25 Concurrent Uploads: 1

Clear Finished Cancel

Online (tgfludtdbFvcv): Sharing 2703 files. Currently 526,589 files (2,246 gigabytes) available in 8,411 libraries.

Screenshot di Napster 2.0 Beta 7

Il processo di digitalizzazione e il conseguente **annullamento dei costi di stampa** hanno **ripulito i bootleg** dalla componente **collezionistica** senza però **alterarne le interazioni** alla base. La **fruizione** di questo materiale audio infatti rimane ancora oggi legato alla **ricerca** e alla **scoperta**. **Forum** e **social network** sono i luoghi digitali in cui non esiste moneta di scambio e la proprietà di un disco a tiratura limitata diventa un **download link condiviso**.

VINILE : STREAMING = BOOTLEG : FREE DOWNLOAD

5 A. N. Bonfranceschi, La storia di Napster, wired.it, 30/06/14, https://www.wired.it/play/musica/2014/06/03/la-storia-di-napster/?refresh_ce=

"Qualsiasi canzone da qualsiasi computer. Il diciottenne della Northeastern University di Boston, Shawn Fanning, nell'estate del 1999 aveva [...] creato Napster, il primo sistema peer-to-peer per la condivisione di file, mp3 musicali principalmente." ⁵



Uno degli **aspetti** più **interessanti** del fenomeno dei **bootleg** è la sua **natura sociale**, il suo **spirito di condivisione** intransigente che è rimasto **vivo** nonostante i processi digitali di dematerializzazione e il **ruolo** dell'**utente** veicolato dalla **scoperta** e dalla **ricerca**.

Telegram è oggi un **servizio** molto **apprezzato** per la **condivisione** di file: all'interno di **gruppi** e **canali** si creano delle vere e proprie **community** con lo scopo di **diffondere informazioni** e scambiare opinioni; inoltre, interagendo con i **bot** programmati dagli utenti si possono ottenere **informazioni**, **ascoltare musica** e **guardare video**.

L'**ambiente** di **condivisione** per certi versi fuori dagli schemi e spesso non compreso appieno posiziona Telegram in **continuità** con lo **spirito anti-sistema** dei **bootleg** al punto che l'app oggi costituisce uno degli **ambienti** più **dark** e, allo stesso tempo accessibili, del **web**.

TELEGRAM, CONDIVISIONE PRIMA DI TUTTO UN SERVIZIO DI MESSAGGISTICA OTTIMIZZATO PER LO SCAMBIO E LA RIPRODUZIONE DI CONTENUTI MULTIMEDIALI

Su Telegram è possibile **ascoltare** sia file **audio** precedentemente caricati nell'app che **canzoni** disponibili sulle principali piattaforme di condivisione audio e video, come YouTube (compreso YouTube Music), SoundCloud, Vimeo e Dailymotion. Il tutto tramite il **player integrato** di Telegram, che permette di ascoltare una canzone o guardare un video senza uscire dall'app. ⁶

"Italiani pazzi di Telegram. La tendenza del momento è sicuramente utilizzare Telegram sia per messaggiare con amici e parenti ma anche per iscriversi a canali, consultare contenuti, discutere, informarsi. Un mondo nuovo, iper-liberale e per questo anche criticato [...]" ⁷

⁶ S. Aranzulla, Come ascoltare musica su Telegram, aranzulla.it, <https://www.aranzulla.it/come-ascoltare-musica-su-telegram-1210972.html#:~:text=Bot%20per%20ascoltare%20musica%20su%20Telegram,-Se%20ritieni%20poco%20text=Tra%20questi%20c%3A8%20Classical,%2C%20quindi%2C%20di%20dominio%20pubblico.>

⁷ wolfagency.it, Boom di Telegram. La community italiana si chiama UpGogram, ansa.it, 30/06/21, https://www.ansa.it/pressrelease/lifestyle/2021/06/30/boom-di-telegram.-la-community-italiana-si-chiama-upgogram_f9b272ed-d0c4-4414-b582-a5ffd3875343.html



1969. BOB DYLAN, GREAT WHITE WONDER

IL PRIMO BOOTLEG DELLA STORIA DEL ROCK

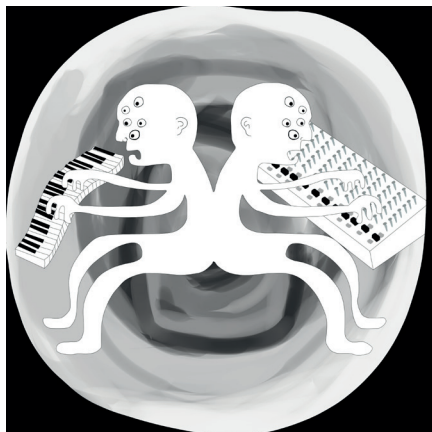


Soprannominato la “**grande meraviglia bianca**” dagli speaker radiofonici della Costa Occidentale che iniziarono a suonarlo per primi per via della copertina totalmente bianca della stampa originale, rappresenta l'**inizio dell'era bootleg**. Un disco **distribuito letteralmente porta a porta**, ristampato più volte, oggi un pezzo da collezione per gli appassionati.

1995. APHEX TWIN, MELODIES FROM MARS

DAL SUPPORTO FISCO A QUELLO DIGITALE

Melodies from Mars è una **raccolta di brani inediti** di Richard David James, meglio conosciuto con lo pseudonimo di Aphex Twin. Sebbene James non abbia ammesso la paternità fino al 2014, era **noto** ai fan come un **album non ufficiale** di Aphex Twin poiché almeno due dei brani in esso contenuti sono versioni alternative (probabilmente **versioni demo**) di tracce presenti in album ufficiali rilasciati successivamente. Si crede che Melodies from Mars sia stato dato ad alcuni amici di James su **cassette** che vennero poi **digitalizzate** e il loro contenuto **distribuito in rete**.

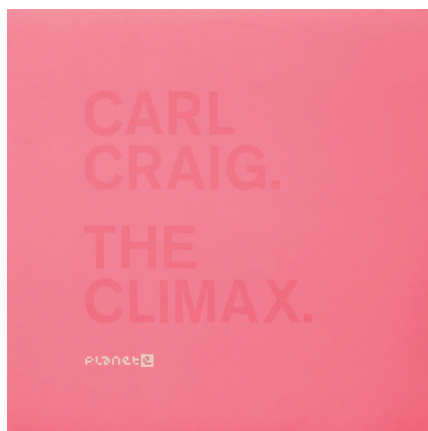




2011. CARL CRAIG, THE CLIMAX BOOTLEG EP

L'APPRODO SU SOUNDCLOUD

Quando nel 2011 Carl Craig scoprì che la traccia "The Climax" che aveva pubblicato nel 1995, era stata **utilizzata** in un **mashup** per una **stampa anonima** di un **vinile bootleg**, decise di **rispondere** al successo di questa produzione non autorizzata recuperando l'audio disponibile e **facendo** il proprio **re-editing** della traccia.



2012. CLUB DOGO, CHISSENEFREGA BOOTLEG REMIX EP

UN REGALO PER I FAN



Per festeggiare il successo dell'album Noi Siamo il Club, il gruppo decise di **regalare** ai **fan** il **download** di Chissenefrega Bootleg Remix EP. Al suo interno, i **remix selezionati** dalla band, che vennero prodotti a partire dalla traccia presente nell'album ufficiale Tu vs. Don Joe contenente la versione acappella del brano.



2016. MARRACASH, GUE PEQUENO, SANTERIA

CACCIA AL TESORO IN GIRO PER MILANO



"Abbiamo dato Santeria in anteprima alle strade di Milano [...] Vai nel link del primo commento e apri la mappa per scoprire tutte le location segrete dove abbiamo posizionato dei poster o degli adesivi contenenti dei QR CODE che vi permetteranno di ascoltare alcuni brani in anteprima. Potete aprirli usando un qualsiasi QR reader o scansionando il QR code con la fotocamera di Shazam.".
Marracash su Facebook

2021. GAZZELLE, OK

ASCOLTARE IL DISCO IN ANTEPRIMA

L'artista romano ha **disseminato** per la **città** di Milano il vinile della sua ultima opera **nascondendola** all'interno di **speciali cassette antincendio** permettendo così ad alcuni **fortunati** di poter **ascoltare** il **disco** 72 ore **prima dell'uscita ufficiale** sulle piattaforme di streaming.





SCDL TELEGRAM BOT

DOWNLOAD MP3S OF TRACKS/SETS FROM SOUNDCLOUD, BANDCAMP, YOUTUBE WITH TAGS.



Il bot **scarica** ed **invia canzoni** come **messaggi audio** in formato **MP3 128 kbps**. Una volta **inviato** il **link** relativo alla traccia da scaricare il bot **risponde automaticamente** con il **file audio** che è possibile **ascoltare** e **salvare** sulla memoria del dispositivo.

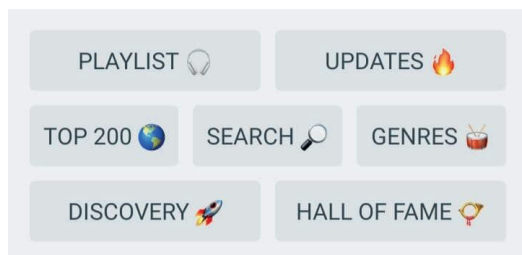
Lista comandi:

- In Download mode: download every link I support (or use /dl <links>)
- In Links mode: show direct download URLs for every link (or use /link <links>)
- In Ask mode: scan for supported links and ask you to choose download or links
- Use /settings to select working mode for messages without command.
- Use /settings to toggle Captions.

SHAZAM MUSIC CHARTS BOT

BOT TO LOOK FOR NEW MUSIC IN LOSSLESS QUALITY!

Permette di **ascoltare** i **brani** più **cercati** su **Shazam** attraverso una sezione di **ricerca libera** oppure consultando **classifiche** divise per **paesi**; ogni brano può essere **salvato** e **riascoltato** facilmente da una **playlist personale** creata dal bot.

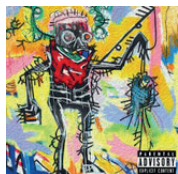


Il menù di interazione con il bot



ASCOLTATORI

SCOPRONO/ASCOLTANO/CONDIVIDONO



Ho tutti i numeri di RollingStone

Orsola Carli

Quella sul pezzo

ETÀ 26

LUOGO Torino

LAVORO Sviluppatrice videogiochi

INTERESSI

Leggere

Concerti

Conoscere nuove persone



Hit?! No grazie non fumo

Elia Barone

Il ribelle

ETÀ 23

LUOGO Palermo

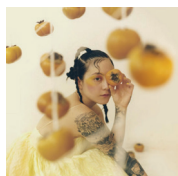
LAVORO Studente

INTERESSI

Musica

Arte

Blog e recensioni



Oramai tutti che fanno gli artisti!

Chiara Dato

La blogger

ETÀ 25

LUOGO Roma

LAVORO Bartender/Blogger

INTERESSI

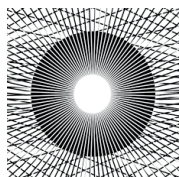
Scrivere e recensire

Musica



ARTISTI

CARICANO/CONDIVIDONO/DIFFONDONO



We're up all night till the sun

HEADACHE

IL DJ

ETÀ 25

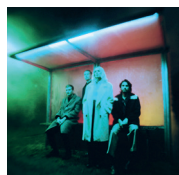
LUOGO Torino/Londra

LAVORO Dj

INTERESSI

Djing

Clubbing



Vorrei raggiungere più persone possibili con la mia musica

Erik Corvo

Il musicista

ETÀ 26

LUOGO Milano

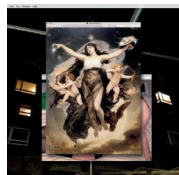
LAVORO Musicista

INTERESSI

Musica

Concerti

Blog e recensioni



Spotify? Nah!

Carlo Forte

IL JAZZISTA

ETÀ 29

LUOGO Milano

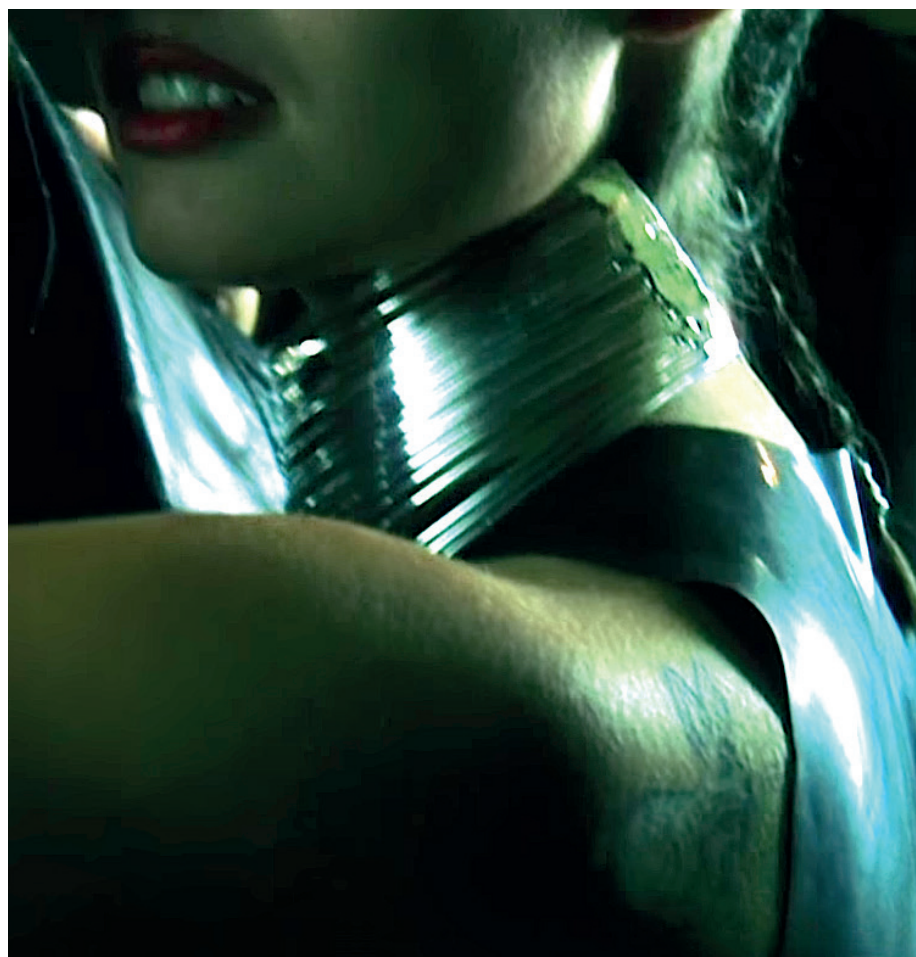
LAVORO Musicista

INTERESSI

Blog e recensioni

Concerti







FOLLOW THE WHITE RABBIT...

I BOOTLEG, IL BIANCONIGLIO E TELEGRAM

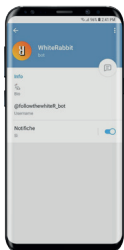
Il riferimento al Bianconiglio di Carroll introduce il tema della scoperta. Si è invitati all'inseguimento di qualcosa di sconosciuto, una ricerca che apre le porte di luoghi inesplorati. Un filo conduttore lega le dinamiche di compravendita dei bootleg - la ricerca, la fruizione e il possesso di un supporto fisico - ad un nuovo meccanismo di scoperta e accesso digitalizzati.

Progettazione di un sistema di distribuzione e riproduzione basato su un bot di Telegram.



DIGITALIZZAZIONE E SCOPERTA

BOT DI TELEGRAM



I temi della **digitalizzazione** e della **scoperta** assumono la **forma** di un **bot** di **Telegram** la cui **logica** di programmazione **veicola** l'**utente ascoltatore** alla **connessione** con gli **artisti** che apprezza e **offre** allo stesso tempo a questi ultimi la **possibilità** di **accrescere** la propria **fan base**.

I meccanismi della scoperta di nuova musica passano non solo attraverso le dinamiche digitali all'interno di Telegram, ma anche attraverso un sistema di adesivi-QR che contribuiscono alla diffusione del bot dal mondo reale.

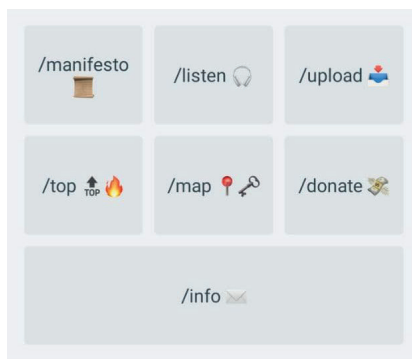
IL BOT

LE INTERAZIONI

L'**interfaccia** del bot è **completamente** in lingua **inglese**, questo non solo per essere **compreso** da **chiunque** ma anche perché quest'ultima è **utilizzata** negli **ambienti** di **sviluppo informatici** all'interno dei **codici di programmazione**. Delle **emoji** arricchiscono tutti i **comandi** e i **messaggi**.

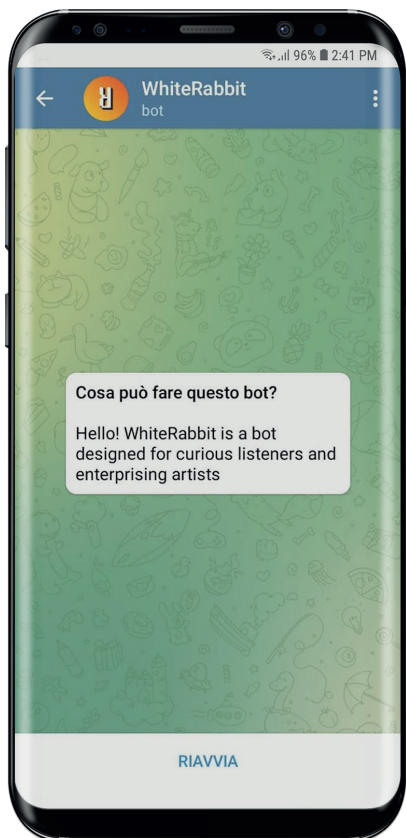
Gli utenti possono in qualunque momento richiamare i comandi del bot attraverso un menù che ne semplifica l'operazione oppure possono inviarli come messaggi.

/start **/manifesto** **/listen** **/upload**
/top **/map** **/donate** **/info**

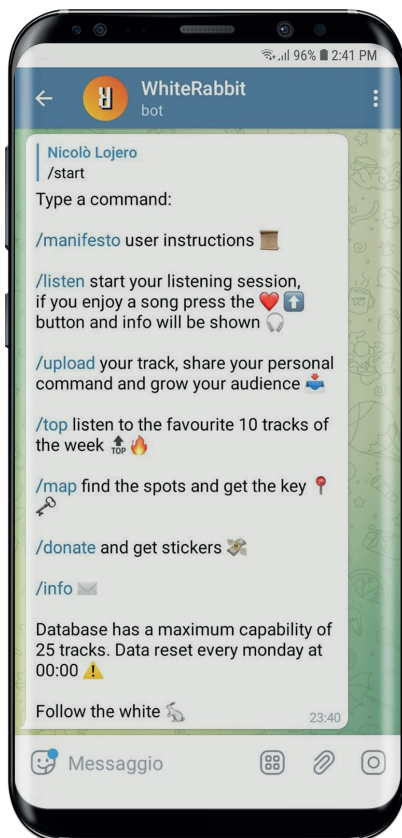




/START



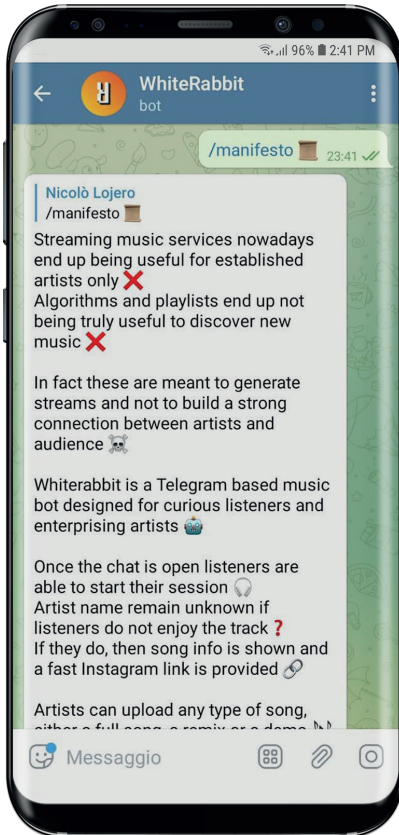
Una volta **aperta** la **chat** compare un **messaggio di benvenuto** che **introduce** sinteticamente le **funzionalità** del bot.



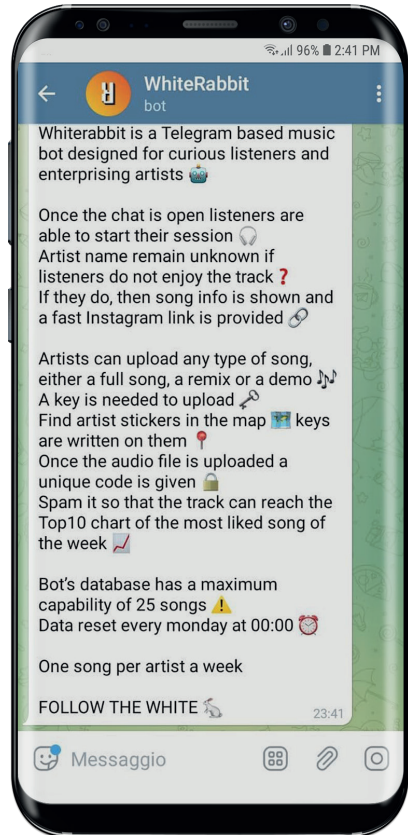
Una volta **avviato** il bot, il primo **messaggio** che l'utente riceve è un **riassunto** di tutti i **comandi** con i quali è possibile interagire, richiamabile in qualsiasi momento digitando **/start**.



/MANIFESTO



Il manifesto racchiude tutte le informazioni necessarie all'utilizzo del bot, il suo scopo e le modalità d'uso.



La **premessa** identifica i **meccanismi** propri della **distribuzione musicale** che **WhiteRabbit** vuole **ricostruire**. Vengono in seguito **illustrate** le modalità di **interazione** degli utenti **ascoltatori**: al fine di **stabilire** una **connessione** solo con gli **artisti apprezzati**, i nomi di questi rimangono sconosciuti nelle informazioni della traccia all'interno del player se l'utente



non **reagisce positivamente** all'ascolto, è soltanto una volta che viene espresso un giudizio preferenziale che il bot risponde fornendo il nome dell'artista e il link al suo **profilo Instagram** in modo che possa stabilirsi una connessione fra le due parti.

Il **manifesto** prosegue poi rivolgendosi agli **utenti artisti** e indica quali sono le **istruzioni** e i **passaggi** da seguire per **caricare** la propria **musica** sul portale: per completare l'operazione è necessario un **codice (key)** presente sugli **artist stickers** che ogni utente può **trovare** grazie all'utilizzo di una **mappa** richiamabile tramite apposito comando. Una volta **completata** la **procedura** viene fornito un **codice univoco** corrispondente alla traccia caricata, questo potrà essere **condiviso** dall'artista per **ottenere** più **reazioni** positive ed **entrare** nella **Top10** dei brani preferiti della settimana. Il **database** del bot può ospitare un **massimo di 25 tracce** ogni settimana così da garantire una user experience agile e musica sempre aggiornata.

Streaming music services nowadays end up being useful for established artists only ❌
Algorithms and playlists end up not being truly useful to discover new music ❌

In fact these are meant to generate streams and not to build a strong connection between artists and audience 🐇

Whiterabbit is a Telegram based music bot designed for curious listeners and enterprising artists 🐇

Once the chat is open listeners are able to start their session 🎧
Artist name remain unknown if listeners do not enjoy the track ?
If they do, then song info is shown and a fast Instagram link is provided 🔗

Artists can upload any type of song, either a full song, a remix or a demo 🎵

A key is needed to upload 🔑

Find artist stickers in the map 🗺️ keys are written on them 📌

Once the audio file is uploaded a unique code is given 🗝️
and they can spread it through their fan base so that the track can reach the Top10 chart of the most liked song of the week 📈

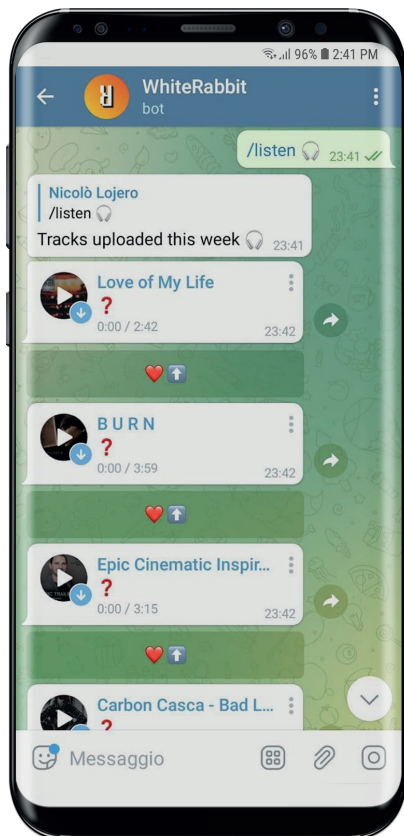
Bot's database has a maximum capability of 25 songs ⚠️
Data reset every monday at 00:00 🕒

One song per artist a week

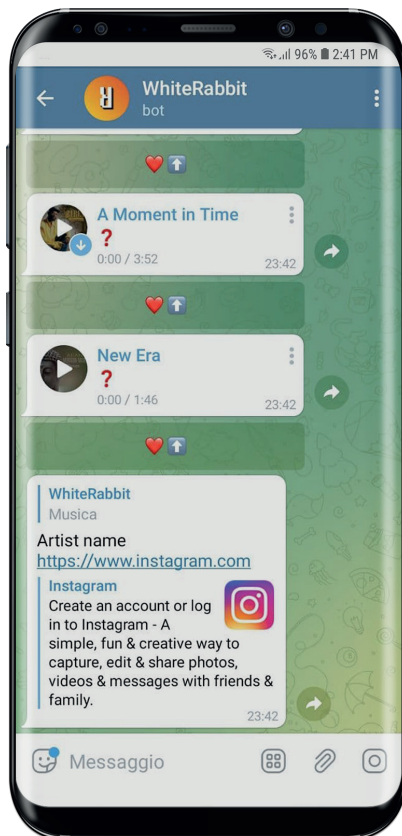
FOLLOW THE WHITE 🐇



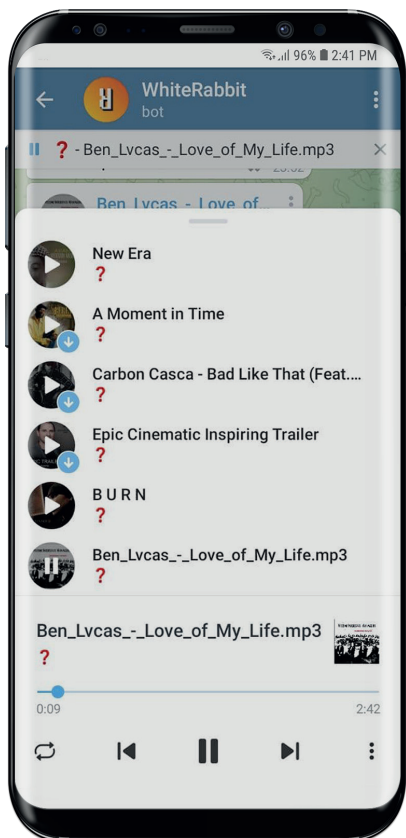
/LISTEN



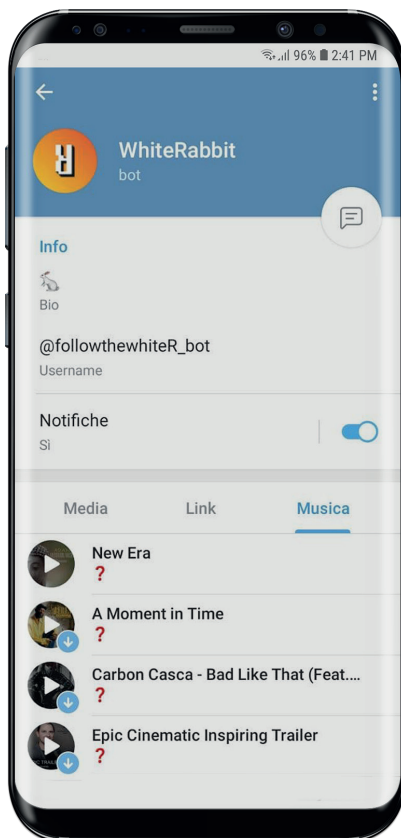
Questo comando avvia una sessione di ascolto dei brani caricati durante la settimana.



Ogni **traccia** viene inviata dal bot tramite **messaggio vocale**; sotto ogni traccia è presente un **pulsante like** che consente agli utenti **ascoltatori** di **esprimere**, se lo desiderano, un **apprezzamento**, una volta schiacciato **rende noto** attraverso un messaggio in risposta alla traccia il nome dell'artista fino a quel momento sconosciuto e il link al suo **profilo Instagram**.



Ogni reazione positiva contribuisce all'**accrescimento** del numero dei **like** ricevuti, ogni settimana una **classifica** dei **dieci brani preferiti** viene stilata e può essere ascoltata digitando l'apposito comando. Il **player integrato** di Telegram consente di **riprodurre** tutta la **coda di brani** anche una volta chiusa l'applicazione, garantisce la riproduzione



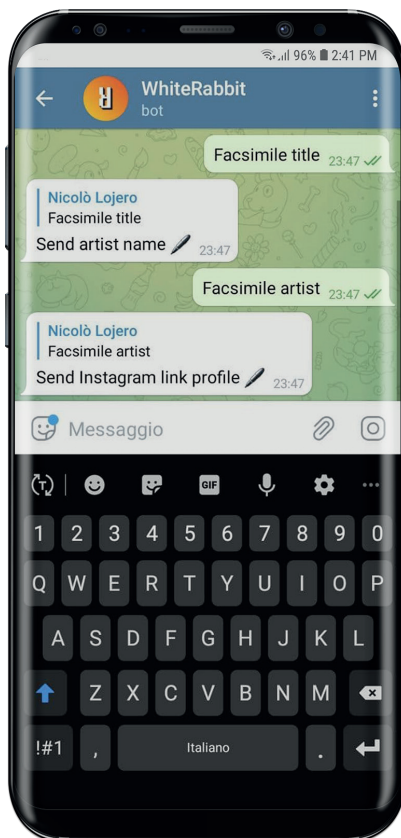
anche in **background** e a **schermo spento**, per mostrare la canzone in riproduzione nella chat, è presente una funzione di richiamo nella finestra del player. Tutti i **media** scaricati saranno **disponibili** nella **sezione musica** all'interno delle **info** del bot e le tracce rimarranno presenti nella chat fino a quando l'utente eliminerà i messaggi corrispondenti.



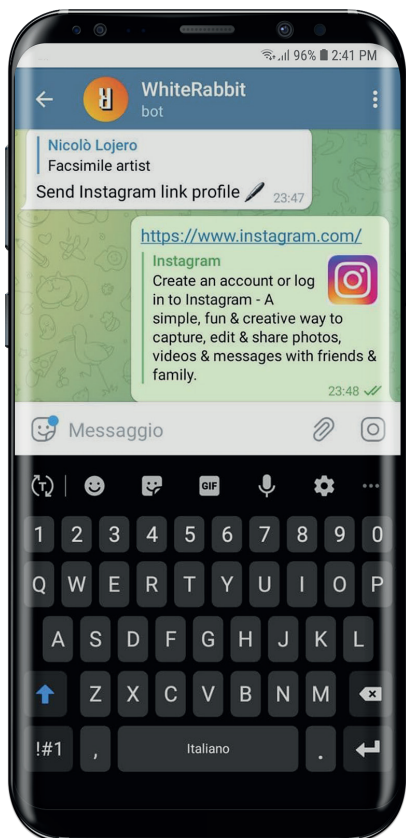
/UPLOAD



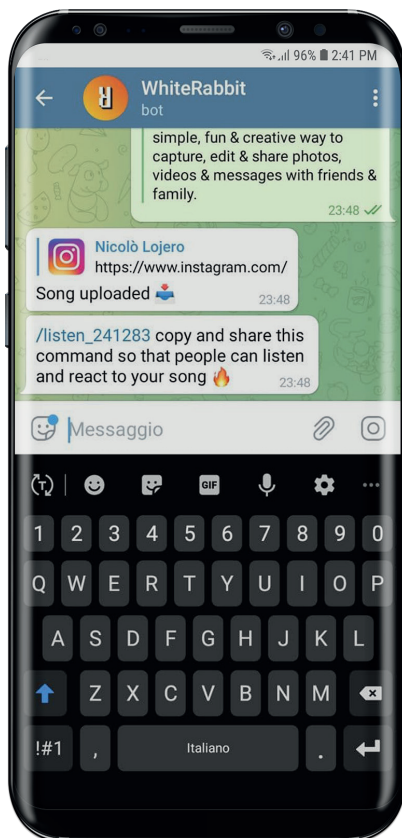
L'iter di caricamento di un brano è automatizzato e segue l'utente in ogni singolo step.



Agli **utenti** che intendono **caricare** la propria **musica** sulla piattaforma è **richiesto** un **codice**, una **key**, presente su degli **stickers sparsi** in giro per varie **città** e sopra i quali è **scritta** una **parola d'ordine** che consente di avviare la procedura.



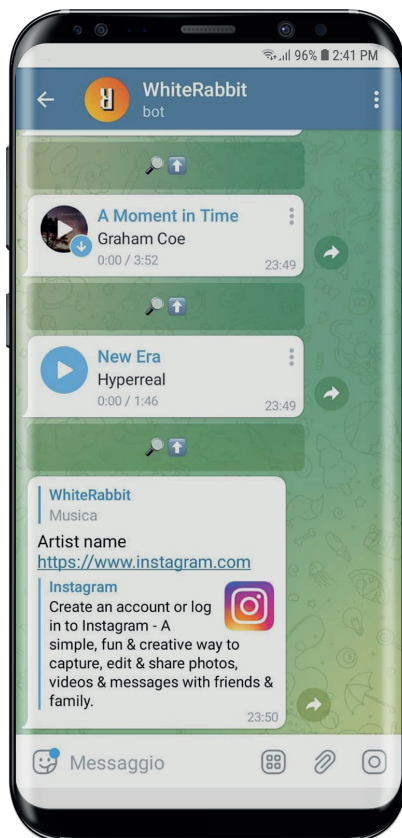
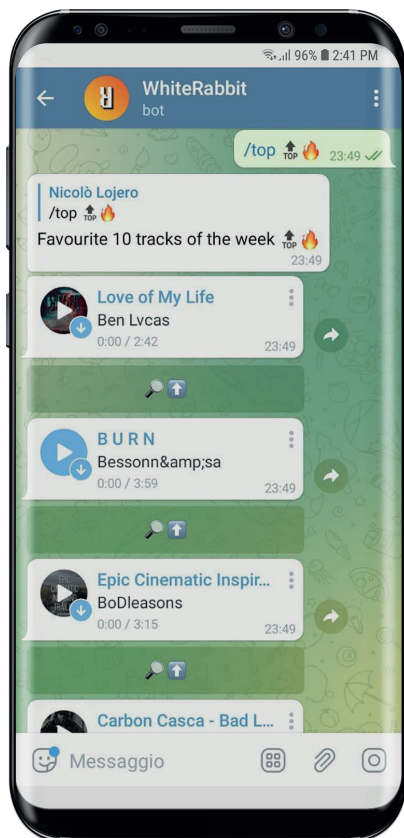
I **codici non** sono **univoci** e basta essere a conoscenza della parola d'ordine per poter completare il caricamento, in questo modo **viene promosso** il **passaparola** fra gli utenti e il **consolidamento** della **community**.



Al **termine** dei passaggi viene **consegnato** agli artisti un **codice-comando** identificativo della **traccia** appena caricata, **digitandolo** nella chat questa potrà essere **direttamente ascoltata** e i like ricevuti concorreranno insieme agli altri brani presenti nel bot in quella settimana per entrare nella Top10.



/TOP



La Top10 viene aggiornata in tempo reale e azzerata ogni settimana insieme al resto del database.

Il comando richiama i dieci brani preferiti dagli utenti durante la settimana. La Top10 rappresenta una vetrina importante per gli artisti che possono cercare di entrarvi proponendo contenuti di

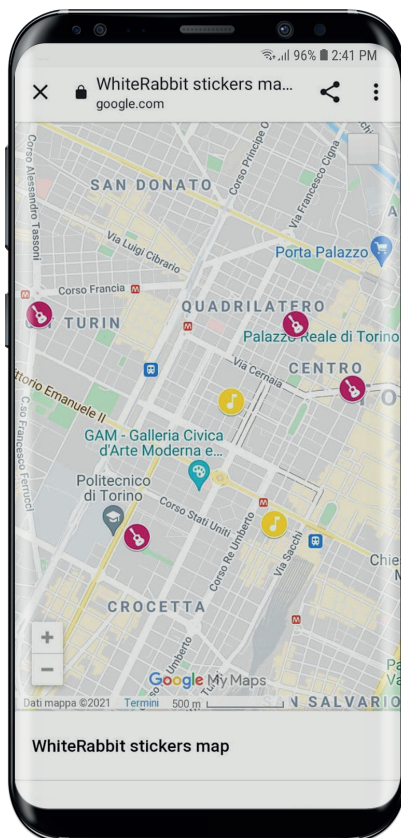
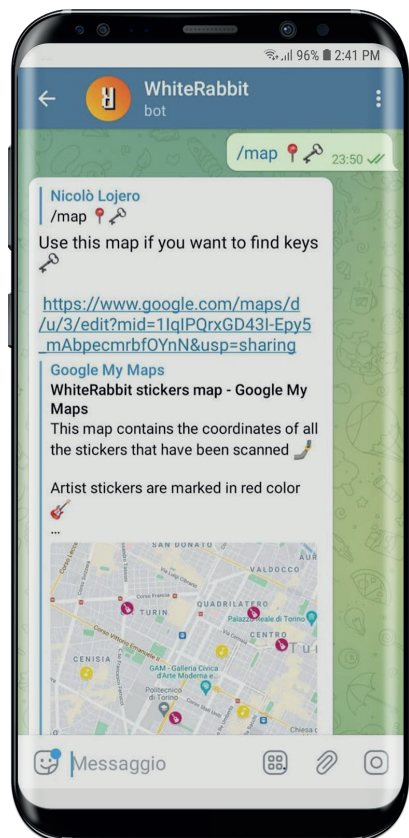
qualità e/o diffondendo in maniera capillare il codice univoco della loro traccia.

Con questa modalità non è possibile esprimere alcuna preferenza cosicché la classifica non possa essere influenzata.

Un collegamento fra l'artista e l'utente viene comunque garantito dal pulsante di ricerca che una volta pigiato invia il link al profilo Instagram dell'artista corrispondente.



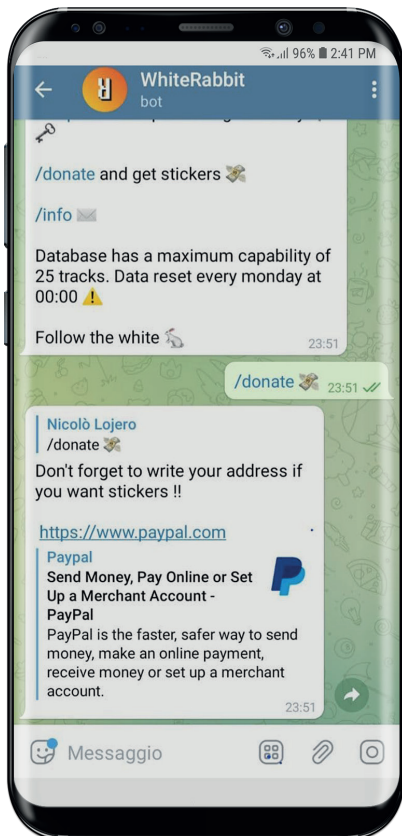
/MAP



Visualizza una mappa di Google Maps con i segnaposto che indicano dove si trovano gli **stickers** sparsi sul territorio. Questa si **aggiorna automaticamente** ogni qualvolta che uno sticker viene scannerizzato da qualcuno.



/DONATE



Ai **donatori** viene chiesto di **inserire** un **indirizzo** per poter **ricevere** a casa degli **stickers**.



/INFO



Viene inviato l'**indirizzo email**.

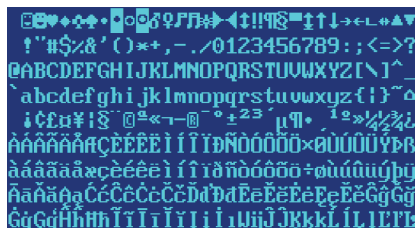


IMMAGINE COORDINATA

LOGO/FONT

Il **logo** del progetto ha come **esigenza** principale quella di essere **riconoscibile** chiaramente come **immagine del profilo** all'interno dell'app di Telegram.

La lettera R rovesciata assume le forme stilizzate di un coniglio. Il font utilizzato richiama il linguaggio informatico di programmazione e quello dei terminali.



Classic Console true type font

```

http://www.irssi.org | Latest version: 0.0.15 | SSL verification issues connecting to freenode? See h
093201 |Ignatius( muzone: see Irssi::set_timeout :) sounds like what you're after
093558 |muzone( thank you Ignatius!
093641 |Ignatius( cool, I actually helped someone \o/
101429 |notinik( hm how do one setup irssi so that it doesnt try to join a channel before the nick is
101757 |Ignatius( err, well I did it a lazy way - you know in the /server add thing you get to specify
101859 |Ignatius( I set the commands to identify and *then* join channels x y z instead of adding them
101926 |Ignatius( by the way, it's almost certainly the completely wrong way to do it
101947 |notinik( Ignatius: oki, like /autocmd ;wait xxx ?
102004 |notinik( but it's a workaround if you get completely stuck
102014 |Ignatius( yeah, like that
102117 |Ignatius( but I'd hang around for a more elegant answer, personally
102156 |Ignatius( I'm sure we're not the only people to have had this problem
102217 |Ignatius( I was simply feeling lazy and rebellious ;)
102302 |notinik( I tried ./autosendcmd "/msg nickserver identify pass;wait 2000;/join #python" but
102312 |BlastHardcheese( notinik: use sasl
102358 |notinik( BlastHardcheese: oki, does that work out of the box in ubuntu or do I need to
102442 |BlastHardcheese( http://freenode.net/faq.shtml
102450 |BlastHardcheese( notinik: for irssi you need a script
102458 |BlastHardcheese( see the link for one
103341 |Jewellyn( oh, notinik left. doh, oh well.
103730 |Ignatius( never mind, there's always a next time
111709 |muzone( Hey
111713 |muzone( Hey
111719 |BlastHardcheese( cccyyyyy
111737 |muzone( Dunno how to explain this
111748 |muzone( Is there a way to show only highlighted windows in the statusbar?
111804 |muzone( the pink ones in Act: 1,2,3 (depending on your theme)
111840 |BlastHardcheese( probably but I don't know it :)
111857 |muzone( hard cheese
111900 |muzone( haha funny name :)
111903 |muzone( okay man, thanks though
113048 |theledge( *11) [5:55Irssi (Pgnt)] [Act: 1]
[ 01:irssi!
2011-09-24 13:04:19 @ sedna bash 10 - $1 (1: 0.12 0.30 0.26) 0* bash 1 bash 2 bash 3 bash 4 bash

```

The "Classic Console" truetype font is free to download and use from webdraft.eu or webdraft.hu and can be redistributed either by other web sites or be included in your package, download, product or source repository *

* <https://webdraft.hu/fonts/classic-console/>



Il **pittogramma** è costruito a partire dalla **lettera maiuscola R** che viene **ruotata** di 180° gradi in senso orario e viene **semplificata** per una migliore lettura.

Una volta **estruso** per un ottenere un **effetto tridimensionale** viene colorato di **bianco** per rispettare le linee guida di comunicazione.

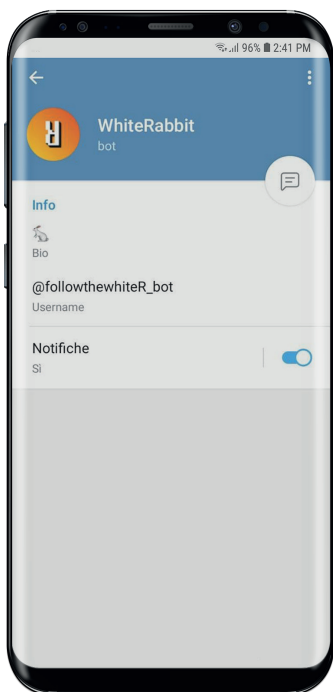


ffe519

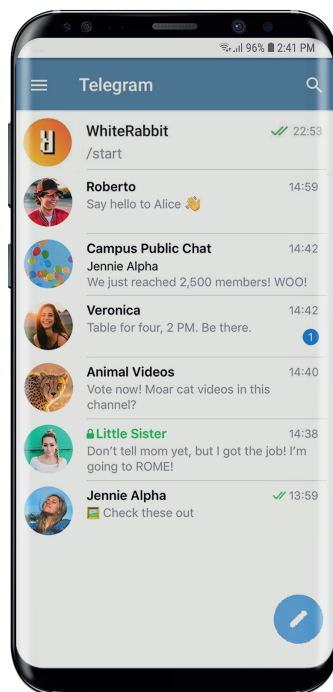


e53b29

Per essere **utilizzato** comodamente come **immagine profilo** nell'app può essere **sovrapposto** ad un **gradiente** dai colori sopraindicati.



Il logo all'interno della chat di Telegram



CLAIM

MATRIX E CARROLL

Il **claim** è progettato a partire dalla **frase** iconica "**follow the white rabbit...**" che viene **rielaborata** sostituendo la parola **rabbit** con la **lettera R** rovesciata del logo.

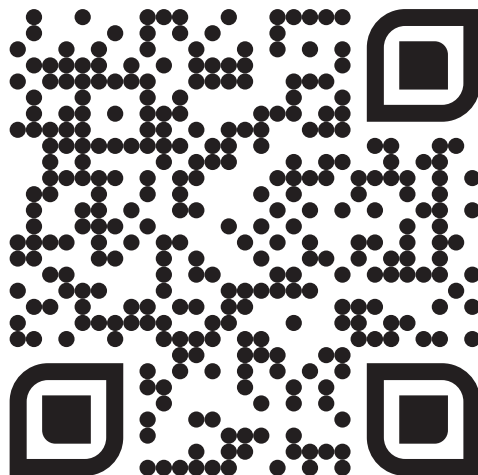
Follow the white R



STICKERS

PUNTO D'ACCESSO DAL MONDO REALE

Una volta **scannerizzato** il **QR code** si viene **reindirizzati** direttamente alla chat con il **bot** su Telegram.



t.me/followthewhiteR_bot

STICKERS LISTENER E STICKERS ARTIST

Sono state progettate due versioni dedicate alle due categorie di utenti



Sticker listener

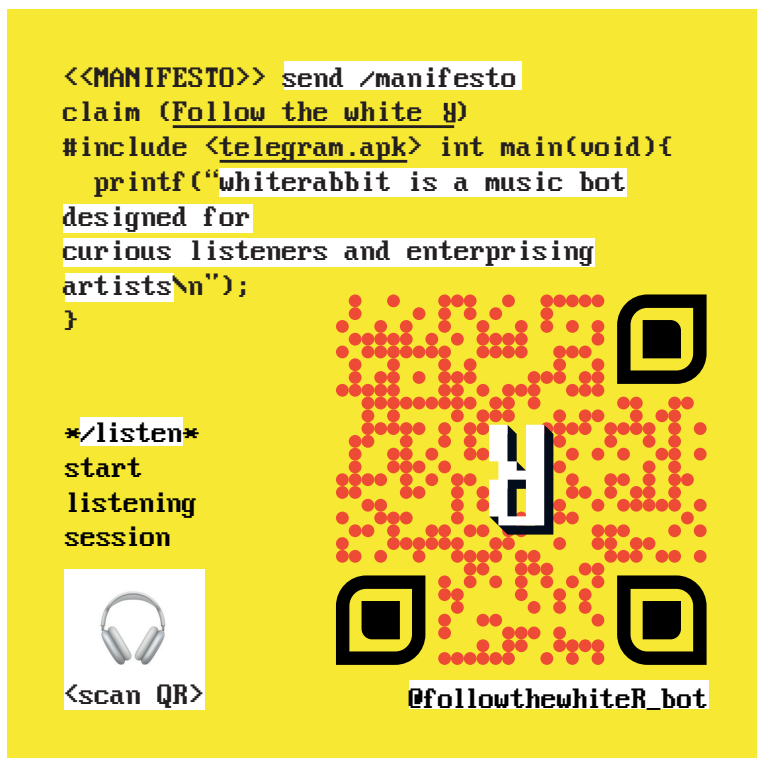


Sticker artist

Hanno la funzione di **pubblicizzare** il bot e al tempo stesso lo **alimentano**, ogni volta che vengono scannerizzati la loro **posizione** viene **aggiornata** sulla mappa.

Gli stickers artist e quelli listener si **differenziano cromaticamente** e **contribuiscono** in maniera diversa alla **diffusione** del bot. I primi sono **necessari** agli **artisti** per poter ottenere il **codice** che permette loro di caricare i propri brani mentre i secondi rappresentano un punto di **accesso diretto** al **bot** per ogni persona che li scannerizza.

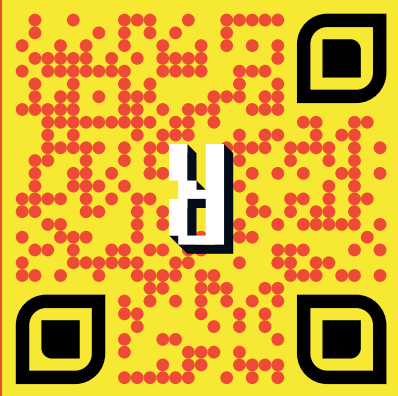
100x100mm



Scala 1:1




100x100mm




```
printf
("whiterabbit
is a music bot
designed for
curious
listeners and
enterprising
artists\n");

<<MANIFESTO>>
send
/manifesto
claim (Follow
the white R)
#include
<telegram.apk>
int main(void)
```

insert key upload
song





<scan QR> <2RX35>

@followthewhiteR_bot

Scala 1:1


```
<<MANIFESTO>> send /manifesto  
claim (Follow the white y)  
#include <telegram.apk> int main(void){  
    printf("whiterabbit is a music bot  
    assigned for  
    curious listeners and enterprising  
    artists\n");  
}
```

```
*/listen*  
start  
listening  
session
```



<scan QR>



@followthewhiteRabbit



insert key upload
song



<scan QR>



<2RX35>

```
printf  
("whiterabbit  
is a music bot  
designed for  
curious  
listeners and  
enterprising  
artists\n");
```

```
<<MANIFESTO>>
```

```
send
```

```
/manifesto  
claim (Follow  
the white H)
```

```
#include
```

```
<telegram.apk>  
int main(void)
```

```
@followthewhiteR_bot
```



```
<<MANIFESTO>> send /manifesto  
claim (Follow the white Y)  
#include <telegram.apk> int main(void){  
    printf("whiterabbit is a music bot  
designed for  
curious listeners and enterprising  
artists\n");  
}
```

* /listen*
start
listening
session



<scan QR>



@followthewhiteR_bot



insert key upload
song



QR>



<2RX35>

```
printf  
("whiterabbit  
is a music bot  
designed for  
curious  
listeners and  
enterprising  
artists\n");
```

<<MANIFESTO>>

```
send  
/manifesto  
claim (Follow  
the white Y)  
#include  
<telegram.apk>  
int main(void)
```

@followthewhiteR



PROGRAMMAZIONE

IL CODICE

Il codice **python** scritto per la realizzazione del prototipo **non possiede** una vera e propria **logica** ma risulta comunque efficace per **dimostrare** le **dinamiche** e le **interazioni** degli utenti con il bot.

```
import telebot
from telebot import types
```

```
API_TOKEN = '1949372920:AAFHNPHJZxOaEwro7cq0HfZU7SYmQzEdKk0'
```

```
bot = telebot.TeleBot(API_TOKEN)
```

```
@bot.message_handler(commands=['help', 'start'])
def send_welcome(message):
```

```
    bot.reply_to(message, """\
    Type a command:
```

```
    /manifesto user instructions 📖
```

```
    /listen start your listening session, if you enjoy a song press the ❤️ 📶 button
    and info will be shown 🎧
```

```
    /upload your track, share your personal command and grow your audience 📁
```

```
    /top listen to the favourite 10 tracks of the week 🔝 🔥
```

```
    /map find the spots and get the key 📍 🔑
```

```
    /donate and get stickers 🎁
```

```
    /info 📧
```

Database has a maximum capability of 25 tracks. Data reset every monday at 00:00 ⚠️

```
    Follow the white 🐇 \
    """, reply_markup=set_buttons())
```



```
@bot.message_handler(commands=['listen_241283'])
def send_audio(message):

    audio1 = open('tracks/Hyperreal_-_New_Era.mp3', 'rb')
    bot.send_audio(message.chat.id, audio1, reply_markup=makeKeyboard(),
parse_mode='HTML')
```

```
@bot.message_handler(commands=['listen'])
def send_audio(message):
```

```
    try:
```

```
        bot.reply_to(message, "Tracks uploaded this week 🎧")
        audio1 = open('tracks/Ben_Lvcas_-_Love_of_My_Life.mp3', 'rb')
        audio2 = open('tracks/Bessonn_amp_sa_-_B_U_R_N.mp3', 'rb')
        audio3 = open('tracks/BoDleasons_-_
_Epic_Cinematic_Inspiring_Trailer.mp3', 'rb')
        audio4 = open('tracks/Carbon_Casca_-_Carbon_Casca_-_
_Bad_Like_That_Feat._King_Whiz_.mp3', 'rb')
        audio5 = open('tracks/Graham_Coe_-_A_Moment_in_Time.mp3', 'rb')
        audio6 = open('tracks/Hyperreal_-_New_Era.mp3', 'rb')
```

```
        bot.send_audio(message.chat.id, audio1, performer=" ? ",
reply_markup=makeKeyboard())
        bot.send_audio(message.chat.id, audio2, performer=" ? ",
reply_markup=makeKeyboard())
        bot.send_audio(message.chat.id, audio3, performer=" ? ",
reply_markup=makeKeyboard())
        bot.send_audio(message.chat.id, audio4, performer=" ? ",
reply_markup=makeKeyboard())
        bot.send_audio(message.chat.id, audio5, performer=" ? ",
reply_markup=makeKeyboard())
        bot.send_audio(message.chat.id, audio6, performer=" ? ",
reply_markup=makeKeyboard())
```

```
    except Exception as e:
        print(e)
```

```
@bot.message_handler(commands=['top'])
def send_audio(message):
```




try:

```

    bot.reply_to(message, "Favourite 10 tracks of the week TOP 🔥")
    audio1 = open('tracks/Ben_Lvcas_-_Love_of_My_Life.mp3', 'rb')
    audio2 = open('tracks/Bessonn_amp_sa_-_B_U_R_N.mp3', 'rb')
    audio3 = open('tracks/BoDleasons_-_
_Epic_Cinematic_Inspiring_Trailer.mp3', 'rb')
    audio4 = open('tracks/Carbon_Casca_-_Carbon_Casca_-_
_Bad_Like_That__Feat._King_Whiz_.mp3', 'rb')
    audio5 = open('tracks/Graham_Coe_-_A_Moment_in_Time.mp3', 'rb')
    audio6 = open('tracks/Hyperreal_-_New_Era.mp3', 'rb')

    bot.send_audio(message.chat.id, audio1,
reply_markup=makeKeyboard2())

    bot.send_audio(message.chat.id, audio2,
reply_markup=makeKeyboard2())
    bot.send_audio(message.chat.id, audio3,
reply_markup=makeKeyboard2())
    bot.send_audio(message.chat.id, audio4,
reply_markup=makeKeyboard2())
    bot.send_audio(message.chat.id, audio5,
reply_markup=makeKeyboard2())
    bot.send_audio(message.chat.id, audio6,
reply_markup=makeKeyboard2())

except Exception as e:
    print(e)

```

@bot.message_handler(commands=['manifesto'])

def send_manifesto(message):

bot.reply_to(message, """)

Streaming music services nowadays end up being useful for established artists
only ❌

Algorithms and playlists end up not being truly useful to discover new music

❌

In fact these are meant to generate streams and not to build a strong
connection between artists and audience 💀



Whiterabbit is a Telegram based music bot designed for curious listeners and enterprising artists 🤖

Once the chat is open listeners are able to start their session 🎧

Artist name remain unknown if listeners do not enjoy the track ?

If they do, then song info is shown and a fast Instagram link is provided 🔗

Artists can upload any type of song, either a full song, a remix or a demo 🎵

A key is needed to upload 🔑

Find artist stickers in the map 🗺️ keys are written on them 📌

Once the audio file is uploaded a unique code is given 🗝️

Spam it so that the track can reach the Top10 chart of the most liked song of the week 📈

Bot's database has a maximum capability of 25 songs ⚠️

Data reset every monday at 00:00 🕒

One song per artist a week

FOLLOW THE WHITE 🐇\

```
""", reply_markup=set_buttons())
```

```
@bot.message_handler(commands=['donate'])
```

```
def send_donate(message):
```

```
    bot.reply_to(message, "Don't forget to write your address if you want  
stickers !!\n\n https://www.paypal.com")
```

```
@bot.message_handler(commands=['full'])
```

```
def send_full(message):
```

```
    bot.reply_to(message, "Database full 💀\n\n Try again next monday at  
00:00 🕒")
```

```
@bot.message_handler(commands=['map'])
```

```
def send_map(message):
```

```
    bot.reply_to(message, "Use this map if you want to find keys 🔑\n\n https://  
www.google.com/maps/d/u/3/edit?mid=1lqIPQrxGD43I-  
Epy5\_mAbpecmrbfOYnN&usp=sharing")
```



```
@bot.message_handler(commands=['info'])
def send_info(message):
    bot.reply_to(message, "📧\n\n info@whiterabbit.page")

@bot.message_handler(commands=['upload'])
def send_upload(message):
    msg = bot.reply_to(message, "Send key 🔑")
    bot.register_next_step_handler(msg, send_mp3)

def send_mp3(message):
    msg = bot.reply_to(message, "Send your .mp3 file 🎵")
    bot.register_next_step_handler(msg, send_title)

def send_title(message):
    msg = bot.reply_to(message, "Send title ✍️")
    bot.register_next_step_handler(msg, send_artist)

def send_artist(message):
    msg = bot.reply_to(message, "Send artist name ✍️")
    bot.register_next_step_handler(msg, send_link)

def send_link(message):
    msg = bot.reply_to(message, "Send Instagram link profile ✍️")
    bot.register_next_step_handler(msg, end)

def end(message):
    bot.reply_to(message, "Song uploaded 📶")
    bot.send_message(message.chat.id, "/listen_241283 copy and share this  
command so that people can listen and react to your song 🔥")

def makeKeyboard():
    markup = types.InlineKeyboardMarkup()
    markup.add(types.InlineKeyboardButton(
        text="❤️👆",
        callback_data=["'value', '" + "❤️" + "'", '" + "❤️" + "'"]
    ))

    return markup
```




```
def makeKeyboard2():
    markup = types.InlineKeyboardMarkup()
    markup.add(types.InlineKeyboardButton(
        text="🔍📶",
        callback_data="['value', '' + '🔍' + '', '' + '🔍' + '']"))

    return markup

@bot.callback_query_handler(func=lambda call: True)
def handle_query(call):
    bot.reply_to(call.message, "Artist name \nhttps://www.instagram.com")

def set_buttons():
    markup = types.ReplyKeyboardMarkup(row_width=3)
    itembtn1 = types.KeyboardButton('/manifesto 📜')
    itembtn2 = types.KeyboardButton('/listen 🎧')
    itembtn3 = types.KeyboardButton('/upload 📶')
    itembtn4 = types.KeyboardButton('/top 📶🔥')
    itembtn5 = types.KeyboardButton('/map 📍🔑')
    itembtn6 = types.KeyboardButton('/donate 📦')
    itembtn7 = types.KeyboardButton('/info ⓘ')
    markup.add(itembtn1, itembtn2, itembtn3, itembtn4, itembtn5, itembtn6,
    itembtn7)
    return markup

bot.polling()
```

SITOGRAFIA

www.rollingstone.com
www.wikipedia.com
www.vulture.com
www.rarerecords.com
www.landr.com
www.britannica.com
www.louderthanwar.com
www.nme.com
www.billboard.com

www.hotpress.com
www.louder.com
www.faroutmagazine.com
www.vice.com
www.highsnobiety.com
www.theindependent.com
www.vogue.com
www.thenewyorktimes.com

BIBLIOGRAFIA

Cummings, Alex Sayf (2013). Democracy of Sound: Music Piracy and the Remaking of American Copyright in the Twentieth Century. Oxford University Press

Heylin, Clinton (1994). The Great White Wonders - A History of Rock Bootlegs. Penguin Books

Heylin, Clinton (2010). Bootleg! The Rise And Fall Of The Secret Recording Industry. Omnibus Press

Hilderbrand, Lucas (2009). Inherent Vice: Bootleg Histories of Videotape and Copyright. Duke University Press

Shuker, Roy (2013). Wax Trash and Vinyl Treasures: Record Collecting as a Social Practice. Ashgate Publishing



(A-Z) Ghira, Greta, Luca, Ma, Matti, Naso, Pa

```
designed for  
curious listeners and enterprising  
artists\n");  
}
```

```
*/listen*  
start  
listening  
session
```



<scan QR>

@followthewhiteR_bot



<2RX35>

key* upload

@followthewhiteR_bot

```
<<MANIFESTO>>  
send  
/manifesto  
claim (Follow  
the white R)  
#include  
<telegram.apk>  
int main(void)
```



insert key upload
song



<scan QR>



<2RX35>

@followthewhiteR_bot

```
printf  
("whiterabbit  
is a music bot  
designed for  
curious  
listeners and  
enterprising  
artists\n");
```

```
<<MANIFESTO>>  
send  
/manifesto  
claim (Follow  
the white R)  
#include  
<telegram.apk>  
int main(void)
```

```
*/listen*  
start  
listening  
session
```



low the white R)
<telegram.apk> int ma
whiterabbit is a mus
or
steners and enterpris
);

@followthewhiteR_bot

```
*/listen*  
start  
listening  
session
```



<2RX35>

key* upload

