

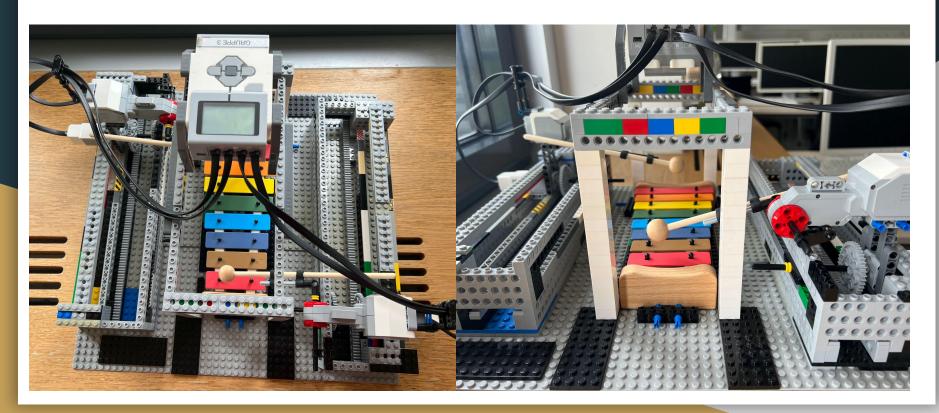
# Xylobot-2.0

Dila Su Celikkol Si-Hoon Kang Batuhan Semercioglu Aneta Větrovská

# Wie hat es angefangen?



# Wie sieht es jetzt aus?

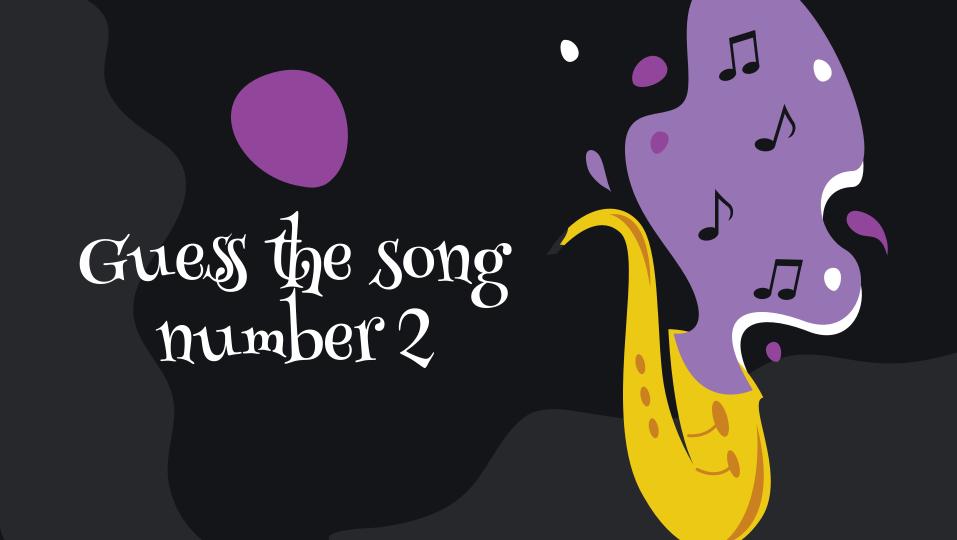


## Was gab es dazwischen?



Goals	
Ein Roboter, der Xylofone selbstständig spielen kann. Er ka	ann sich im Noten orientieren immer den richtigen Tone
richtigen Ton finden und ihn sauber spielen.	
	<b>Y</b>
Optional Goals (i.e. nice to have)	455
Noten selbst erkennen.	
Gleichzeitig bewegen und spielen.	
2 Klöppel benutzen zu können.	





# Ergebnisse:

#### **Hardware**

- spielende Hand, Bewegungsplattform (Zahnräder)
- Hand und Plattform zusammengebaut
- verschieden Klöppel (Holzklöppel, Stift, Lego
  Stück, Lippenstift -> am besten : Holzklöppel)



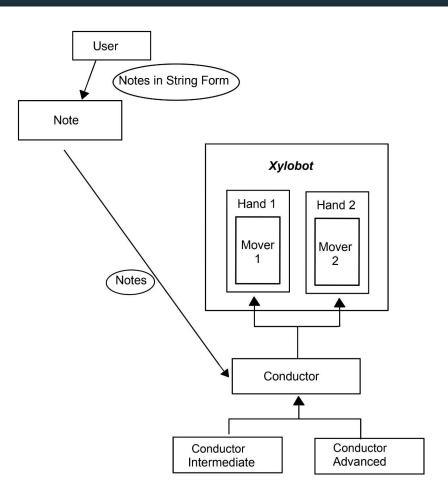
#### **Hardware**

- zweite Hand aber nur ein Plattform ->
  Xylophon in zwei Teilen geteilt (C1-F und G-C2)
- zweite Platform, erste Idee : zwei Stöcke -> unrealisierte Idee -> Plattformen von beiden Seiten
- zu chaotisch -> mehrere Konstruktionen gebaut
- Upgrade zum besseren Xylophon (25€!!!)



## **Software**

-



# **Optimierungsalgorithmus**

- Beispiel: notes = [C1, A, G] -> [\_, \_, \_, \_, \_] -> jede mögliche Kombination machen, wo die Reihenfolge so bleibt (also entweder i-te oder i+len(notes)-te Position):

- in zwei Teilen teilen: erste Array für erste Hand, zweite Array für zweite Hand und weil die erste Hand beim C1 und zweite beim C2 anfängt -> als erste Note immer darstellen: 1. [C1, C1, \_, G] 2. [C2, \_, A, \_]
- für jede Möglichkeit die Entfernungen berechnen, hier zB.: [C1, C1, \_, G]: 4, [C2, \_, A, \_]: 2
- Die Entfernungen addieren (6) und dann das Minimum auswählen
- hier: [C1, C1, \_, \_]: 0, [C2, \_, A, G]: 3 -> beste Lösung Hand 1: [C1, C1, \_, \_]; Hand 2: [C2, \_, A, G]
- nicht benutzt -> 16 Noten : cca 1 sec, 17 Noten -> 2 sec, 18 : 4 ... -> exponentielles Wachstum (schlecht !!)

#### WUNSCHSONG

Was würdet ihr am Ende gerne von unseren

Xylophone zuhören?



# Fragen?