FIRST TECH CHALLENGE ROMANIA

DECODE™ PRESENTED BY RTX

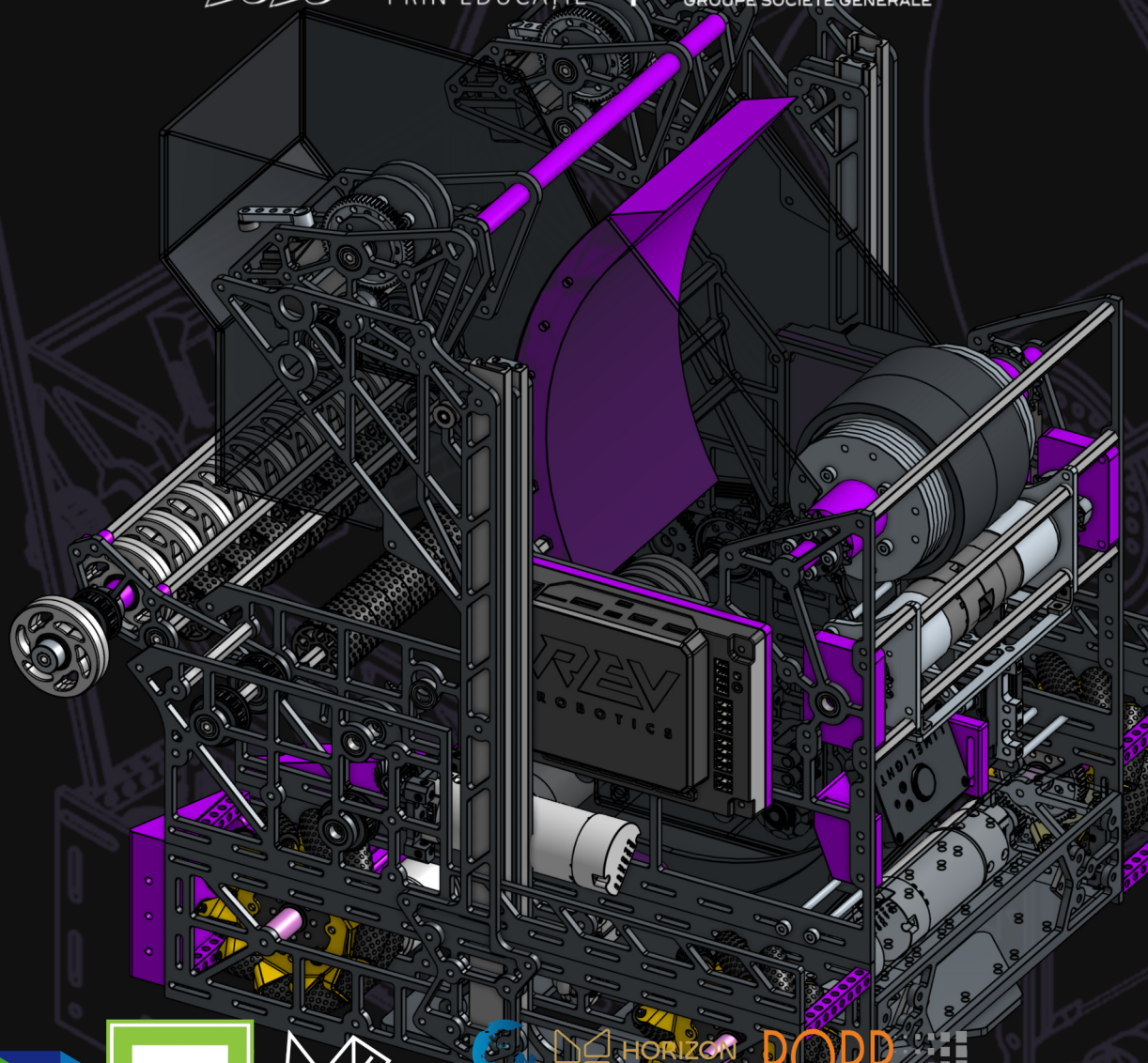
2025 SEASON#10 2026

ORGANIZATOR NAȚIE PRIN EDUCAȚIE

PARTENER FONDATOR BRD GROUPE SOCIETE GENERALE

ENGINEERING PORTOFOLIO

VICYBER #21476



orange™

PARKLAKE Meeting Nature

PLORIAN

ASUS

TINMAR energy

HORIZON CITY

POPP & ASOCIAȚII

F64 devoted to your passion

FormTex®

REZUMAT

Echipa de robotică **ViCyber #21476**, ce reprezintă **Colegiul Național de Informatică „Tudor Vianu”, din București**, a fost înființată în anul **2022** ca urmare a dorinței de a **dezvolta domeniul STEAM** a unor elevi **pasionați de tehnologie, inovație și progres.** Noi am avut de la început ca **scop crearea** unui **mediu de învățare,** atât în interiorul echipei, cât și în afara acesteia, și performanța în cadrul competițiilor de robotică. Am construit pas cu pas o **comunitate** bazată pe **colaborare, perseverență** și **dorința** de **autodepășire.** Fiecare proiect ne-a adus **mai aproape de visul** nostru de a misiunea noastră.

Dacă ar fi să alegem un cuvânt care să ne definească, acesta ar fi **evoluția.** De-a lungul acestui sezon, **am reușit să ne mărim echipa foarte mult.** Am reușit să aducem **2 noi mentori, convinși** de **misiunea** noastră. Am reușit să **comunicăm** mai bine, să **funcționăm** mai bine. Am reușit să-i facem pe sponsori **să creadă** în noi și să **investească**. Am devenit mai **uniți**. Totodată, am participat la numeroase evenimente și workshop-uri, unde am împărtășit din experiența noastră și **am învățat lucruri noi de la alte echipe.** În 2022, echipa avea câțiva membri. **Era „tânără”.** Cu fiecare an care a trecut, am devenit mai buni. Mai numeroși. Mai pasionați. Mai **„în spiritul FIRST Tech Challenge”.**

REALIZĂRI

Competițional

La etapa Regională Sud am obținut **Premiul I CONTROL** și am stabilit un **record european** de 317 puncte NP împreună cu Quantum Robotics #14270.

Impact

Am dedicat **peste 100 de ore** activităților de **outreach,** prin care am reușit să implicăm direct **peste 2300 de persoane** în **inițiative educaționale** și **tehnologice.** Am participat la și contribuit în mod activ la evenimente precum **ViCyber Echoes, Future Talks, DefCamp, Erasmus+, Snowy Circuits, Bloom Scrimmage și SciTech,** unde am promovat interesul pentru inovație, robotică și **formare continuă** în domeniul STEM. Prin **prezentări, ateliere interactive,** am încurajat **curiozitatea** tinerilor, colaborarea între participanți și dezvoltarea de competențe tehnice relevante pentru viitor.

Viitor

Nu ne putem atribui singuri calitatea de ambasadori FIRST, dar **considerăm cu tărie că suntem, cel puțin, pe traiectoria cea bună.** În primul rând, pentru a putea asigura perpetuarea echipei, trebuie **să asigurăm existența generațiilor viitoare.** Pe lângă modelul de recrutare menționat și cooptarea constantă de voluntari ce vor deveni membri în anii viitori, **mentorii noștri au conceput un proiect prin care “juniorii” să învețe:** un grup de lucru, asemănător unei echipe, premergător statutului de membru. Acesta ar avea **numeroase beneficii** – permite **învățarea prin experiență** și, în același timp, am avea parteneri de antrenament. **Voluntarii construiesc un robot** și îl codează, singura miză fiind acumularea de cunoștințe. Privind **dintr-o perspectivă financiară, grupul de lucru este sustenabil** întrucât ar folosi piesele noastre mai vechi, de care noi nu am mai avea nevoie, rezultând astfel **o folosire la maxim a resurselor** pe care la avem la dispoziție. Termenul de implementare este până sezonul viitor.

EVENIMENTE

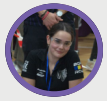
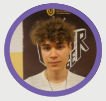
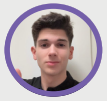
ERASMUS+ SCITECH VICYBER ECHOES

SNOWY CIRCUITS LEAGUE MEET ONE HOUR OF ROBOTICS BLOOM SCRIMMAGE

FUTURE TALKS

DEFCAMP

În **ViCyber Echoes,** am prezentat făcut laboratoare pline de provocări, workshop-uri cu programare și proiectare, iar la **Future Talks** am ascultat și am discutat cu speakeri din industrie și universități despre cariere în robotică, electronică, inteligență artificială și viitor lor. Am fost prezenți la **DefCamp,** unde am creat legături cu lumea securității cibernetice, consolidându-ne abilitățile în proiecte STEAM, iar **SciTech** a fost un treasure hunt captivant în care noi am avut o probă specială – să construiască un robot FLL după instrucțiuni. Prin Erasmus+, am prezentat elevilor din program competiția FIRST Tech Challenge, iar la **Snowy Circuits** ne-am antrenat intens pentru etapa regională. **Bloom Scrimmage** ne-a pus la antrenat direct pentru etapa națională, perfecționându-ne programarea și strategia de joc.


Bogdan

Alexia

Moise

Florian

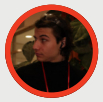
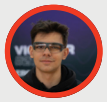
Dorothea

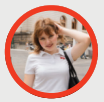
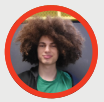
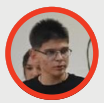
Teodor

Thomas

Ștefan

Alex


Ștefan

Dimitri

Alexandra

Bianca

Mihai

Mihnea

Isabela Coman

Corina Dobrescu

Cristiana Popescu

- Mentor
- Marketing
- Hardware
- Software

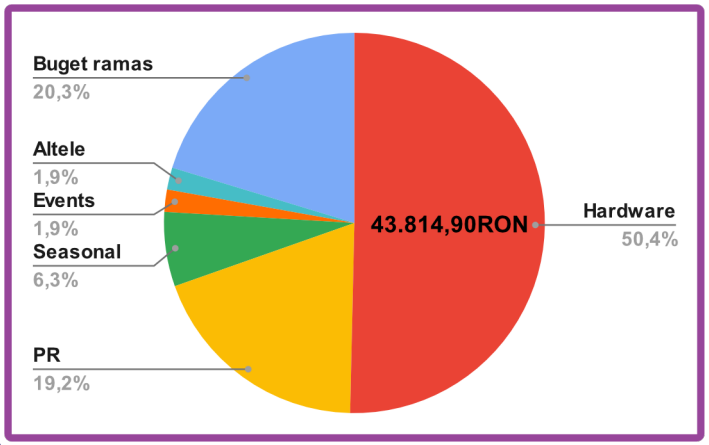
PROCESUL DE RECRUTARE

După cum am spus, anul acesta **ne-am mărit considerabil echipa.** În ceea ce privește recrutarea noilor membri, aceasta a reprezentat o **evaluare complexă a abilităților și dorinței de implicare.** De asemenea, o componentă importantă pentru noi a fost **„chimia”**. Degeaba avem fiecare individual abilități extraordinare, dacă nu avem o anumită compatibilitate.

Referitor la procesul de **recrutare a voluntarilor,** acesta s-a desfășurat la nivelul școlii noastre. Am promovat echipa, le-am prezentat-o „bobocilor”, am organizat workshop-uri, iar elevii s-au înscris la un interviu. De asemenea, aici, **de ajutor au fost mentorii noștri,** care au sfătuit elevii să se intereseze și să se înscrie. În prezent, avem **o comunitate de 25 de voluntari** implicați activ.

PLAN DE AFACERI

Anul acesta **bugetul nostru a crescut cu 380%** față de cel de anul trecut, dată fiind atragerea unor noi sponsori și susținerea celor deja existenți, ce au avut și au încredere în misiunea și obiectivele noastre. **Sezonul trecut bugetul nostru total a fost de 15.600 lei,** din care am cheltuit **100%:** 13.300 lei pe cheltuieli legate de robot și 2.300 lei pe cheltuieli legate de materiale promoționale. În acest sezon, bugetul total este de aproximativ 74.800 de lei, până acum, din care am cheltuit 79,7%, adică 59.600 de lei. Sezonul nu este totuși terminat...



Strategie de atragere a fondurilor

Crearea unor **materiale și oportunități atractive,** care să **convingă** sponsorii că merită să **investească** în echipa noastră. Profitând de resursele care ne sunt la îndemână, **am elaborat un plan complex de sponsorizare,** flexibil, adecvat pentru orice sponsor.

Bronze	Silver	Gold	Platinum	Titlu
1000	2500	5000	7500	10000

MISIUNEA ECHIPEI

Promovarea inovației, educației și în răspândirea valorilor comunității FIRST.

Ne propunem, pe lângă **stimularea interesului** față de știință și tehnologie, să convingem tinerii să aibă **inițiative.** Să **creeze proiecte,** în care să implice, la rândul lor, pe alții. Ne dorim **să mărim comunitatea STEAM,** iar pentru aceasta este nevoie de inițiative, la rădăcina cărora stau valorile de bază.

CUM NE RESPECTĂM ȘI CONTINUĂM MISIUNEA?

Răspunsul presupune 2 perspective. În primul rând, trebuie să avem **o activitate de outreach,** care să reflecte principiile noastre, implicit ale comunității. În al doilea rând, trebuie, pe **plan interior,** în echipă, să implementăm în orice activitate desfășurată setul de valori FIRST.

SCOPUL ECHIPEI

EXPLORARE

Mereu **căutăm să descoperim** noi viziuni, să dezvoltăm noi idei și să dobândim noi abilități.

COLABORARE

O **echipă unită** înseamnă succes. Încercăm să comunicăm eficient și să ne folosim de abilitățile fiecăruia dintre noi.

CREATIVITATE

Apelăm la **cunoștințele și modurile** de gândire proprii pentru a găsi noi soluții la probleme. Avem constant sesiuni de brainstorming, în cadrul fiecărui departament.

GRACIOUS PROFESSIONALISM

Diferențele de opinie duc la **idei noi și progres,** dacă sunt abordate corect, într-o manieră respectuoasă, lucru pe care îl aplicăm.

DISTRACȚIE

Ne bucurăm de orice moment împreună. După fiecare situație dificilă, vorbim, tragem concluzii și, eventual, râdem de greșelile noastre, într-o manieră constructivă.

OBIECTIVE VIITOARE

Calificarea la Campionatul Mondial

Mentoratul noilor generații și juniorilor

Creșterea outreach-ului și timpului petrecut în acest scop

FILOSOFIE DE DESIGN

Încă de la începutul sezonului, am adoptat principiul **KISS (Keep It Simple and Straightforward)** , cu obiectivul de a construi un robot ușor de gestionat din perspectiva hardware-ului, software-ului și a drive team-ului.

Un mecanism simplu oferă avantaje operaționale clare:

1. Calibrare rapidă și predictibilă
2. Reparații eficiente între meciuri
3. Reducerea erorilor în timpul meciurilor

În procesul de proiectare am prioritizat:

1. Scoruri ridicate și consistente
2. Reducerea complexității inutile

Abordarea noastră a fost orientată către **consistență înainte de eficiență maximă.** Fiecare sistem a fost evaluat după criterii clare:
1. Fiabilitate ridicată
2. Număr minim de puncte de posibilă defecțiune

INTAKE

Am optat pentru o transmisie 1:1 în sistemul de intake, cu obiectivul de a minimiza pierderile mecanice și de a menține un transfer de putere cât mai eficient.

Designul inițial s-a bazat pe un intake cu

ȘASIU

Am ales transmisia reductoare folosind roți dințate de 24T pe motor și 60T pe roata mecanum. Acest detaliu nu a adus doar ușurință în întreținerea sa, mai mult de atât, ne permite precizie mult mai mare prin intermediul unui transfer de putere eficient și constant.

SORTARE

Am ales să implementăm un sistem de sortare liniară deoarece designul simplu al acesteia ne-a permis să avem un robot compact care are un flow constant și rapid în teleop, dar care nu ne-a împiedicat din a acumula RP-urile pentru sortare

OUTTAKE

Am ales transmisia 1:1 pe sprocketuri de la 2 motoare bare, pentru a putea trage de oriunde de pe teren si un unghi static, intrucat reduce vibratiile si twichurile unuia cu unghi ajustabil, facandu-l mai rapid si mai reliable

ȘASIU

Concept

Am pornit de la obiectivul de a construi un **robot ușor, rapid și cu torque ridicat**, capabil să:

1. mențină accelerație și viteză ridicată
2. facă față roboților mai grei în situații de contact

Principiile de bază ale designului au fost:

1. Distribuția egală a greutății pe fiecare roată, pentru a optimiza tracțiunea și funcționarea mecanum wheels
2. Spațiu structural dedicat pentru rampă, shooter și intake, fără interferențe între subsisteme
3. Plasarea roților de odometrie pe axa centrului de rotație,

Brainstorming

Pornind de la concept, **am dezbatut** pe baza studiilor modalitățile de transmisie: pe curea , pe lanț și pe **gear-uri.**

Curea
- deformarea curelei
- funcționare lină

Gear
- cea mai eficientă
- fiabilitate

Lanț
- robustă
- pierderi energie
- solicitare crescută

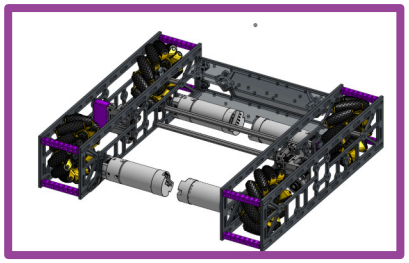
Testare

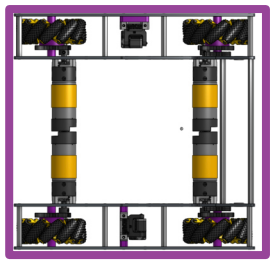
Perioada de testare a durat **2–3 zile**, timp în care **am evaluat fiecare subsistem individual.** Acest proces a fost posibil datorită **arhitecturii modulare** a robotului, care permite **demontarea și testarea rapidă a ansamblurilor.**

În urma testelor, am identificatv **o problemă l**a montarea roților mecanum:
1. Șaibele utilizate împiedicau strângerea uniformă a șuruburilor

Acest lucru genera aliniere imperfectă a roților. Rezultatul era afectarea mișcării șasiului

Parametru	V1–NOPAPI1	V2–Final
Distanțieri roți	printați	metalici de precizie
Joc mecanic angrenaj	prezent după utilizare îndelungată	absent
Șaibe poduri	prezente	eliminate
Uniformitate strângere roți	afectată	restabilită
Intervenții mentenanță	necesare periodic	absente pe durata sezonului
Fail points noi	—	Niciunul





SORTARE V1

Concept

Obiectivul sistemului de sortare a fost direcționarea consistentă a artefactelor colectate către shooter sau hopper, în funcție de culoarea detectată, fără a întrerupe fluxul continuu al intake-ului. În proiectare am prioritizat:

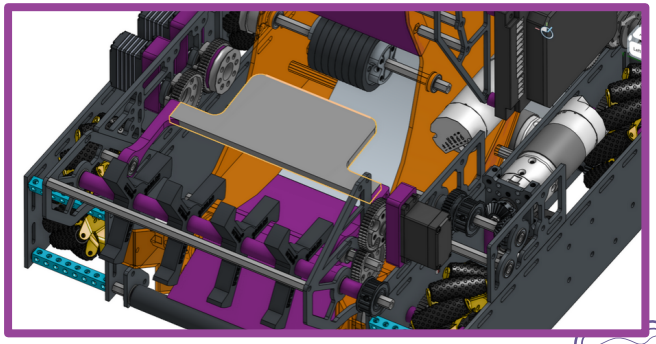
1. Flux de transfer neîntrerupt între intake și subsistemele de scoring
2. Simplicitate mecanică

Brainstorming

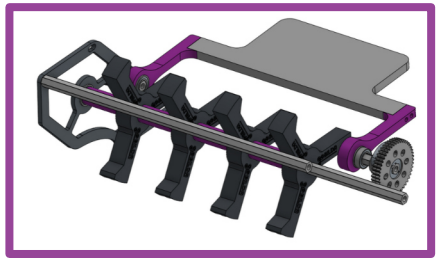
One-way door printat acționat de servo AXON: ușă printată montată pe traseul artefactului, blochează sau eliberează unul dintre cele două trasee:shooter,hopper, comutarea se realizează direct prin comandă servo

Deflector activ cu piesă metalică pe ax pasiv are structură mai rigidă și mai rezistentă la impact repetat, complexitate mecanică mai mare-mai multe componente în lanțul cinematic

Sortare pasivă prin geometrie **a fost analizată inițialv** pentru simplitate maximă, astfel respinsă, deoarece geometria artefactelor din Decode nu permite o separare pasivă fiabilă



ANSAMBLU INTAKE SORTARE V1



Fail points

Rezistență mecanică insuficientă a ușii printate

Impactul repetat al artefactelor la viteze mari de transfer provoca:

- deformarea piesei
- fracturarea în zonele de solicitare

Consecințe:

înlocuiri frecvente ale componentei, scăderea fiabilității mecanismului

Zonă de ambiguitate la bifurcație

Flexibilitatea materialului printat genera **deflecții** parțiale ale redirecționări **impredictibile** ale artefactelor.

Optimizare

În urma testării V1, am concluzionat că **servoul AXON nu reprezenta limita sistemului.** Viteza și precizia de poziționare erau mai mult decât suficiente pentru ritmul de sortare necesar în meciuri. **Limitarea reală era exclusiv mecanică:**

- ușa printată nu putea absorbi impactul repetat al artefactelor

- piesa se degrada progresiv în urma ciclurilor de

Tranziție la V2

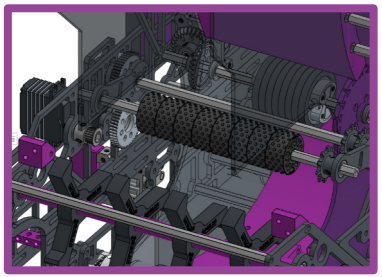
Am păstrat servoul AXON, care acționează o roată dințată printr-o **transmisie 1:1 pe un ax pasiv.** Pe roata dințată este fixată o piesă metalică, care funcționează ca **element activ de sortare.** La rotirea axului, piesa redirecționează artefactul:

element valid → shooter

element invalid → hopper

Între axul pasiv și axul roților Gecko de 32 mm am introdus **o transmisie pe lanț**, care **asigură avans continuu și controlat** al artefactelor prin zona de sortare.

SORTARE V2



Eliminarea failure point-urilor mecanice și creșterea vitezei și preciziei de sortare. Sistemul trebuia să redirecționeze consistent artefactele către shooter sau hopper, fără degradare mecanică și fără dependență critică de timing.

Parametru	V1 – One Way Door	V2 – Ax Pasiv + Gecko
Actuator	servo AXON	servo AXON (același)
Piesă activă	ușă printată	piesă metalică
Rezistență mecanică	scăzută	ridicată
Consistență la flux rapid	scăzută	ridicată
Degradare în timp	prezentă (deformare print)	minimă pe durata sezonului

INTAKE

Concept

Am pornit de la **obiectivul de a colecta artefactele cât mai consistent** și cu **efort minim** pentru driver, menținând compresie constantă în transferul către shooter.

Principii de bază:

1. Colectare stabilă a artefactelor

2. Compresie constantă pe traseul către shooter

Brainstorming

Pornind de la concept, am analizat mai multe soluții pentru preluarea artefactelor, nefiind clar dacă acestea sunt suficient de compliant pentru poly roller cu grip tape sau suficient de rigide încât să necesite flex wheels.

Am restrâns opțiunile la:

1. op-take cu rubber tube

2. compliant stars

3. boot wheels

La începutul sezonului, testarea mai multor variante de rubber tube ar fi consumat mult timp. În plus, boot wheels oferă inner

Motorul de intake (1150 RPM, goBILDA) transmite mișcarea prin **bevel gears 30T într-un raport 1:1,** menținând turația nominală la arborele bootwheel-urilor.

Alegerea unui **raport 1:1 a fost intenționată,** pentru a păstra **viteza maximă de preluare.** La un diametru efectiv al bootwheel-ului de aproximativ 50 mm, rezultă **o viteză tangențială ridicată la suprafață,** favorabilă pentru colectarea rapidă și consistentă a artefactelor. $v = \frac{1150\,rpm \cdot \pi \cdot 0.050m}{60} \approx 3.01\,m/s$

Data matching

Parametru	Teoretic	Măsurat
Viteză tangențială	**3.01 m/s**	**2.78 m/s**
Turație roller	**1437 RPM**	**1390 RPM**
Timp intake	**0.25 s**	**0.31 s**
Forță compresie	**4.2 N**	**3.8 N**

Rampa printată a introdus o **ușoară rezistență la avans** pe suprafețe abrazive**, reducând viteza efectivă de intake cu ~7%.**

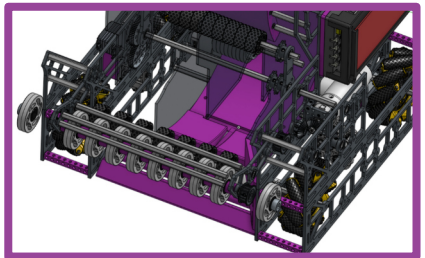
Optimizare pentru V2

1. **reproiectarea rampei** cu unghi variabil, pentru a acomoda elementele dezaliniate și a reduce rezistența la avans

2. **înlocuirea bevel gear-urilor cu bevel clamp-uri,** soluție deja implementată, care a eliminat uzura progresivă a dinților și a asigurat o transmisie mai silențioasă și mai rigidă torsional

3. **analiza unui raport ușor overdrive la bootwheel** (ex. 1:1.15), pentru a compensa pierderile de viteză tangențială identificate în data match

INTAKE V2



Criteriu	Intake static (regionale)	Intake varianta 2
Tip role	Boot wheels Ø96 mm	REV Ø50.63 mm + Gecko Ø32 mm
Tip compresie	Constantă, fixă	Progresivă, adaptabilă
Flow rate general	Foarte rapid	Rapid + consistent
Performanță în colțuri	Problematică	Îmbunătățită geometric
Dependență de poziționare	Mare	Redusă
Repetabilitate estimată	Medie	Ridicată

SHOOTER

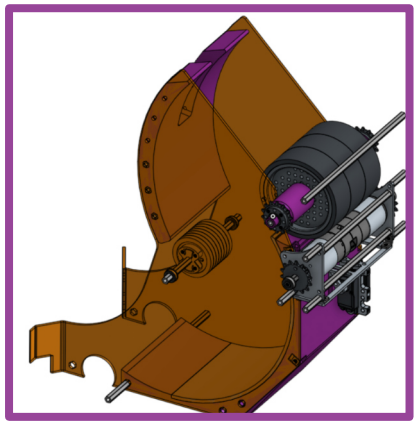
Concept

Obiectivul shooterului a fost **lansarea consistentă a artefactelor** în goal, cu traiectorie repetabilă, din **o singură poziție fixă pe teren.**

Filozofia de design a urmărit: mecanism **compact**, un **singur unghi de lansare optimizat** teoretic, transmisie de accelerare **eficientă energetic.**

Unghiul de lansare a fost stabilit **inițial la 45°** față de orizontală, ales ca punct de pornire, deoarece în cinematica clasică este maximizată bătaia orizontală.

SHOOTER V1



Brainstorming

Roți de accelerare — Rhino Wheels Ø96mm

Dublul avantaj al diametrului mare: masa distribuită radial **funcționează ca volant** de inerție, iar pentru același cuplu motor un diametru mai mare **produce viteză liniară mai mare la suprafață.**

Compresia — Foam 2mm

Testate două metode de fixare: **super glue** și **scotch dublu adeziv.** Super glue-ul rigidiza local foam-ul, creând puncte dure unde compresia activă dispărea. Scotch-ul menținea foam-ul elastic pe toată suprafața.

Unghiul de lansare — 45°

45° ales ca unghi inițial deoarece maximizează teoretic bătaia orizontală pentru o viteză de ieșire dată

Teorie vs. realitate

Parametru	Teoretic	Măsurat
Viteză bilă	24.13 m/s	sub estimare
Compresie glue	uniformă	puncte dure
Compresie scotch	uniformă	uniformă
One Way Door	4.2 N	prea scurt
Compresie mid roller	4.2 N	insuficientă
Mid roller viteză	4.2 N	lentă

Puncte slabe

One-way door prea scurt — nu acoperea complet secțiunea canalului, bilele treceau parțial pe lângă el.

Mid roller — round belt introducea slip și pierderi de tensionare, viteză la suprafață sub valoarea necesară.

Foam cu super glue — rigidizare locală, compresie neuniformă.

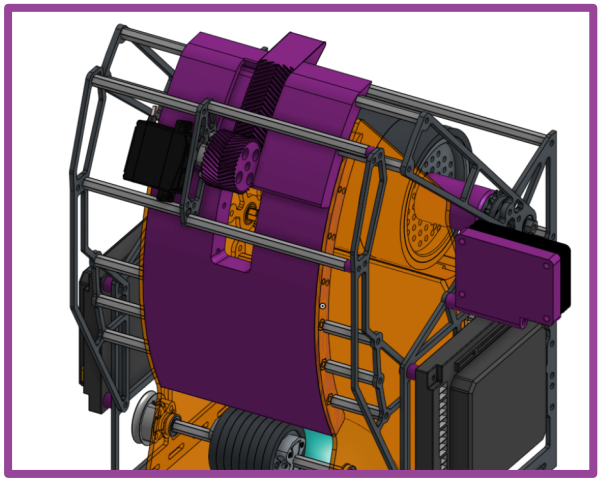
Optimizare

Beam custom CNC pentru one-way door.

Reproiectare transmisie mid roller. Scotch ca metodă standard de fixare foam.

Introducere mecanism unghi ajustabil pentru testarea eficientă a unghiurilor în sesiunile următoare.

SHOOTER V2



Brainstorming

Versiunea shooterului a fost **construită** în jurul unui **Rhino Wheel goBILDA de 96 mm,** cu placă de **backplate la 45° față de wheel**, compresie **constantă de 3 mm** asupra bilei

Sistemul **funcționa consistent la distanța nominală** de tragere, însă **analiza geometrică a evidențiat o limitare:** un unghi fix de lansare **nu poate compensa** variațiile de poziționare ale robotului în timpul meciului.

Testarea variației de unghi: pe baza calculelor balistice, am analizat o reducere a unghiului cu 20°. Această modificare **schimbă profilul traiectoriei:**

- curbă mai **plată**
- componentă **orizontală dominantă**
- **distanță mai mare la aceeași viteză periferică** a wheel-ului

Pentru a valida ipoteza, **am proiectat o variantă cu unghi ajustabil mecanic,** cu un singur increment de −20°, **utilizată exclusiv în mediu de testare** pentru a evalua relația dintre: unghi de lansare, range, precizie.

Testare

Testarea a acoperit **ambele configurații** — 45° și 25° — pentru fiecare măsurându-se distanța de aterizare, precizia față de target și consistența pe **serii de 10 lansări consecutive.** S-au monitorizat deviația laterală, rata de intrare în target și sensibilitatea față de variațiile de RPM, construind harta unghi–range utilizată pentru înțelegerea completă a comportamentului shooter-ului.

Optimizări

Testarea variantei cu mecanism de ajustare a unghiului (45° → 25°) a furnizat datele necesare pentru **o simplificare** intenționată a sistemului înainte de etapa regională.

Observațiile principale:

- **Repetabilitate mai scăzută** la 25° comparativ cu 45°
- **Sensibilitate crescută a traiectoriei** plate la variații de RPM și compresie
- **Risc mecanic suplimentar** introdus de mecanismul de comutare în condițiile dinamice ale competiției

Pe baza acestor rezultate, **am decis eliminarea mecanismului de ajustare și revenirea la unghiul fix de 45°,** pentru a maximiza consistența și fiabilitatea sistemului la V3.

Consecințe

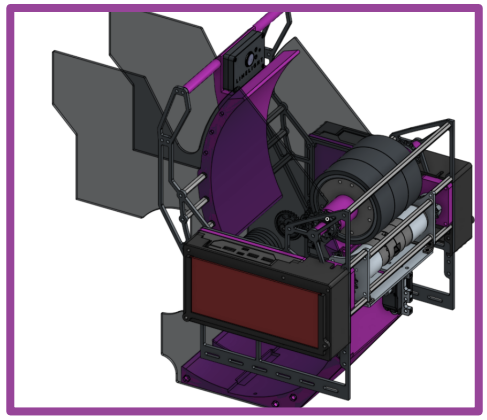
Eliminarea mecanismului de unghi ajustabil

Creșterea compresiei prin adăugarea foam-ului

Adăugarea unui strat de foam pe placa din spate, crescând compresia de la 3 mm la 6–7 mm asupra bilei

Înlocuirea ajustării de unghi cu controlul RPM din cod

SHOOTER V3



Concept

V3 **adresează ultima sursă de inconsistență** din V2: **absența compresiei active** pe latura superioară a bilei. Introducerea unui **counter roller** acționat printr-o transmisie în trepte menține **compresia activă bilateral pe bilă pe toată durata canalului**. Unghiul ajustabil din V2 este înlăturat.

Brainstorming

Counter roller — transmisie 60T→20T (3:1) + GT2 20T 1:1
↓ turație, ↑ cuplu, menține compresia constantă.
Rubber rollers Ø32 mm pentru fricțiune ridicată.
Mid roller — relocat pe transmisia intake (pulley) elimină transmisia dedicată și sincronizează direct alimentarea.
Elastice pe disc wheels Ø48 mm

Repetabilitate

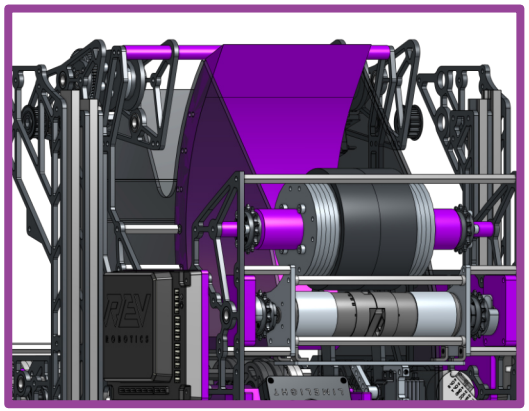
Cea mai ridicată dintre primele trei versiuni. Compresia activă bilaterală **elimina variațiile de contact** la bile ușor dezaliniate. Elasticele pe disc wheels **absorbeau micro-variațiile fără ajustări** între meciuri. **Carcasa rigidă unică a eliminat microvibrațiile,** iar codul maturizat compensa variațiile de distanță fără intervenție mecanică.

Fail points

Mecanismul de ajustare a unghiului — vibrațiile induse în carcasa printată de jocul rezidual din articulații afectau consistența lansărilor mai mult decât flexibilitatea unghiului o compensa în condiții de joc real.

Tensionarea curelei GT2 necesită verificare periodică.

SHOOTER V4



Concept

După **trei iterații cu unghi ajustabil,** am revenit la **filozofia KISS.**

Datele din V3 au arătat că **vibrațiile introduse de mecanismul de ajustare afectau consistența mai mult** decât compensa flexibilitatea unghiului.

Odată cu **dezvoltarea codului de tragere,** ajustarea mecanică a devenit nenecesară.

Am ales un unghi static de 45°, **confirmat în testele V2 și V3** ca fiind **optim** pentru distanțele specifice terenului.

Brainstorming

Unghi static 45°- maximizează bătaia orizontală pentru viteza de ieșire dată — confirmat teoretic și în testele V2–V3.

Eliminarea mecanismului de ajustare - reduce jocul mecanic și microvibrațiile.

Cod de tragere — calculează distanța față de goal în timp real și adaptează parametrii de tragere.

Astfel, nu mai este necesară ajustarea mecanică a unghiului.

Repetabilitate

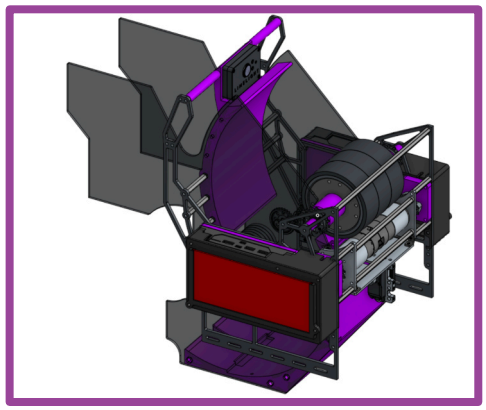
V4 Regio a demonstrat cea mai ridicată repetabilitate de până atunci.

Fail points

Backspin observat la trageri de la distanță mare. Cauză: fricțiune asimetrică între bilă și canalul inferior, generând backspin net.

Efect Magnus, care reduce bătaia efectivă față de valoarea calculată.

SHOOTER V5



Concept

V5 reprezintă rafinarea finală pe baza datelor colectate la regională. **Problema majoră izolată:** backspin la trageri de la distanță mare, **cauzat de fricțiunea asimetrică** dintre bilă și canalul inferior.
Îmbunătățiri implementate:
Rhino Wheel 30A
Flywheel de 1 kg
Eliminarea canalului printat inferior, înlocuit cu două tije de aluminiu cu bandă de cupru
Ajustarea unghiului static la 41°, pe baza datelor de la regională.

Brainstorming

Problemă: **backspin generat de fricțiunea asimetrică** între bilă și canalul inferior → Magnus effect negativ → bătaie redusă.
Soluție mecanică:
1. **eliminarea canalului printat inferior**
2. **două tije de aluminiu** → contact redus la două linii
3. **bandă de cupru pe tije** → μ mult redus, frecare minimă.
4. Rhino Wheels 30A
   - compliancy mai mare
   - arie de contact crescută
   - transfer energetic mai eficient către bilă.
5. **Flywheel ~1 kg**
   - moment de inerție crescut
   - decelerare minimă între lansări
6. **Unghi ajustat la 41°**

Repetabilitate

V5 a demonstrat **cea mai ridicată repetabilitate din întregul sezon.**

1. Backspin redus prin tije de aluminiu + bandă de cupru.
2. Flywheel ~1 kg → inerție crescută, viteză de ieșire mai stabilă.
3. Unghi 41°, calibrat pe datele reale de la regională.

Traiectoria efectivă a devenit aliniată cu modelul balistic.

Fail points

Banda de cupru prezintă **uzură la contact repetat** — μCu crește progresiv spre μ Al pe măsură ce suprafața se zgârie. Necesită inspecție și înlocuire periodică pe durata unui sezon extins.

Versiune	v_bilă	Ec	Unghi	Backspin	R efectiv
V1	24.13 m/s	78.7 J	45° fix	ridicat	sub teoretic
V2	27.14 m/s	99.4 J	20°–45° ajust.	ridicat	sub teoretic
V3	27.75 m/s	103.9 J	20°–45° ajust.	ridicat	sub teoretic
V4	27.75 m/s	103.9 J	45° static	moderat	moderat sub teoretic
V5	28.35 m/s	108.5 J	41° static	redus	≈ teoretic

Parametru	V4 Regio	V5 Natio
Unghi static	45°	41°
Rhino wheels	standard	30A
Masa flywheel	nemodificată	1 kg
E_rot flywheel	~181.9 J	~454.8 J (+150%)
Canal inferior	print solid	2 tije Al + bandă cupru
μ contact inferior	~0.40 (print)	~0.12 (Cu)
F_Magnus	~-3.64 N	~-1.31 N
Perturbație față de G	~137%	~49%
Coef. transfer	92%	94%
v_bilă	27.75 m/s	28.35 m/s
Ec	103.9 J	108.5 J
R efectiv	sub teoretic	≈ teoretic

PARCARE V1

Obiectivul mecanismului de parcare a fost înclinarea robotului suficient față de sol pentru validarea punctelor de parcare la finalul meciului, în cel mai scurt timp posibil și cu complexitate mecanică minimă.

Principii urmărite:

1. respectarea filozofiei KISS
2. fără impact asupra arhitecturii robotului
2. masă adăugată minimă

Brainstorming

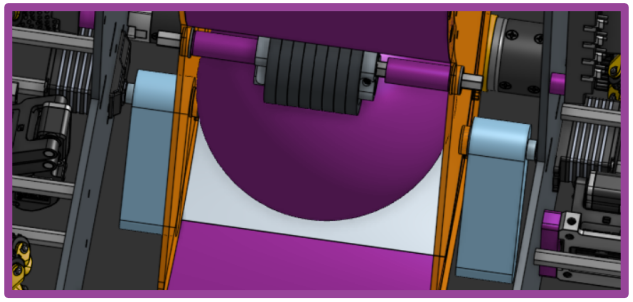
Brațe extensibile acționate liniar — respins, complexitate ridicată, volum mare, timp de implementare incompatibil cu deadline-urile.

Sistem cu scripete și cablu — respins, necesita motor dedicat și ancorări structurale pe șasiu, crescând masa și complexitatea electrică.

Două picioruțe rigide acționate de servouri AXON Max — adoptat, brațe printate PLA, rotite de AXON Max, montate pe add-on plate-ul șasiului, în poziție retrasă rămân pliate pe profilul robotului

Design

Două brațe printate din PLA sunt montate simetric pe șasiu, câte unul pe fiecare lateral, fixate pe placa add-on a șasiului. Fiecare braț este acționat direct de un servo **AXON Max — ales pentru cuplul ridicat disponibil la viteză mică**, esențial pentru a genera forța de reacție necesară înclinării robotului printr-un braț de pârghie relativ scurt.



Fail points

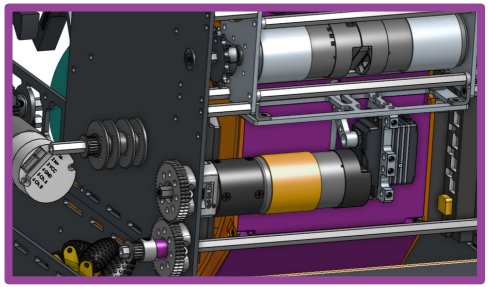
Fragilitatea brațelor din PLA

Stabilitate laterală

Compatibilitate cu roboții aliați

Dependență de sol plat

PARCARE V2



V2 introduce o schimbare fundamentală de arhitectură față de picioruțele din V1: ridicarea completă a robotului prin spool cu fir de pescuit, acționat de un motor dedicat goBILDA 312 RPM.

De ce motor dedicat și nu servo?

Experiența din V1 a arătat că servourile AXON Max, deși capabile să încline parțial robotul, **nu ofereau arhitectura potrivită** pentru o ridicare completă controlată. **Un servo lucrează optim pe un unghi definit de rotație** — nu pe o deplasare liniară continuă și predictibilă. Un motor **DC cu encoder oferă control precis al deplasării liniare prin numărul de rotații ale spoolului,** viteză constantă sub sarcină și posibilitatea de a opri exact la înălțimea dorită prin feedback software. Motorul goBILDA 312 RPM a fost ales ca motor nou dedicat exclusiv parkingului — decizie justificată de necesitatea unui control independent al mecanismului față de orice alt subsistem al robotului.

Repetabilitate

Ridicare stabilă și reproductibilă în teste de laborator.

Platformă rigidă din policarbonat, cu 4 puncte de prindere, distribuia uniform forța pe structură.

Eliminată instabilitatea laterală din V1, unde brațele independente nu se sincronizau.

Poziționare aproape de centrul de greutate → moment minim față de axa transversală.

Robotul se ridica vertical, fără tendință de basculare.

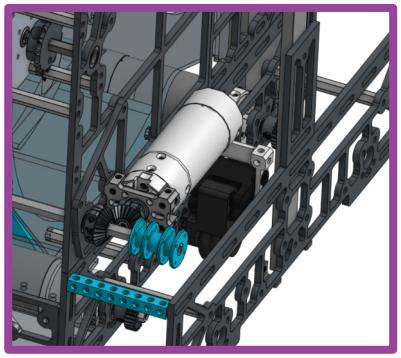
Optimizare V3

Eliminarea curelei GT2v și înlocuirea cu transmisie rigidă, capabilă să preia șocurile de pornire fără pierderi de sincronizare.

Elemente **păstrate** neschimbate (validate în testele de laborator):

- motor dedicat
- spool Ø55 mm
- platformă policarbonat cu 4 puncte de prindere
- poziționare aproape de centrul de greutate.

PARCARE V3



Concept

V3 **elimină complet cureaua GT2** — singurul fail point identificat în V2 — și **introduce transmisie rigidă** pe bevel clampuri goBILDA în raport 1:1.

Modificarea este **implementată prin integrarea transmisiei** noi într-o placă din aluminiu 6082 frezată CNC, ancorată în structura existentă a robotului.

Brainstorming

Lanț — respins, pierderi prin frecare, necesită tensionare, masă mai mare decât GT2.

Gearuri drepte — respins, backlash la pornire → risc de rupere fir Ø0.2 mm.

Bevel clampuri goBILDA — adoptate, fără backlash, fără element elastic, preiau șocurile prin suprafață conică.

Integrare modulară, placă Al 6082 ancorată direct în structura existentă → iterație rapidă V2→V3, fără modificări structurale.

Glisiere V-groove — ghidare strict verticală, elimină mișcările laterale ale platformei la ridicare.

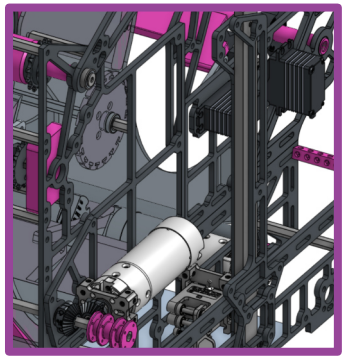
Repetabilitate

1. **Bevel clampuri** au eliminat complet fail point-ul din V2 — nicio săritură de dinți în testele de laborator, inclusiv la porniri repetate cu sarcina de ~12 kg.

2. **Glisierele V-groove** au introdus ghidare verticală rigidă, eliminând mișcările laterale parazite observate în V2.

3. **Ridicarea platformei** a devenit reproductibilă și consistentă în toate ciclurile de testare.

PARCARE V4



Concept

V4 adresează simultan cele două probleme rămase din V3: alinierea incorectă a V-groove-urilor generată de toleranțele insuficiente ale plăcilor de montaj, și dimensiunile reduse ale platformei care limitau compatibilitatea cu roboții aliați în parcarea cooperativă.

Brainstorming

Plăci noi vs. refrezare V3 — plăci complet noi, plăcile V3 nu aveau referințe geometrice, iar refrezarea ar fi compromis rigiditatea și timpul de execuție.

Referințe CNC integrate: plăcile V4 includ suprafețe de referință frezate care fixează poziția șinelor, distanța și perpendicularitatea față de axa de ridicare → alinierea devine proprietatea piesei.

Platformă policarbonat extinsă +40 mm crește suprafața utilă pentru parcare cooperativă fără schimbări de material sau arhitectură. Repoziționarea celor 4 puncte de prindere menține distribuția uniformă a forței de ridicare pe platforma extinsă.

Repetabilitate

Ridicare lină și predictibilă pe întreaga cursă verticală.
Referințele geometrice CNC ale plăcilor noi garantează aliniere identică la fiecare reasamblare, eliminând variabilitatea manuală din V3. **Astfel sunt eliminate blocajele** intermitente observate anterior. Platforma extinsă cu +40 mm a crescut numărul configurațiilor valide de parcare cooperativă în testele simulate.

Fail points

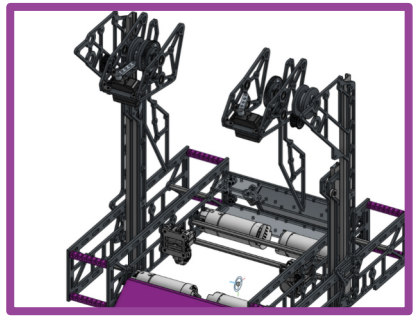
Platforma extinsă (+40 mm) a rămas insuficientă pentru toate configurațiile cooperative.
Limitare externă echipei, dependentă de arhitectura roboților aliați.
Motor dedicat pentru parcare — ineficient din perspectiva masei și complexității electrice.

PARCARE V5

V5 introduce **schimbarea arhitecturală finală a mecanismului de parcare.**
Implementat **dual PTO (Power Take-Off):** puterea este preluată direct de la motoarele de intake și mid roller în momentul parkingului.
Secvență: motoarele se opresc → servoul angrenează gear-ul de cuplare → arborii devin surse de cuplu pentru două spool-uri independente.
Eliminat un motor dedicat, împreună cu ESC-ul, cablajul și masa asociată. Înlocuit cu un servo + două angrenaje de cuplare.
Cele două ansambluri PTO sunt **montate simetric,** fiecare cu lanț cinematic și spool propriu.

PTO

ANSAMBLU



Argument PTO față de motor dedicat

Motor dedicat pentru parcare - funcționa ~0.13 s din 120 s de meci **(~0.11% utilizare).** În restul timpului era masă pasivă, consumând **spațiu și buget de masă** fără beneficiu în intake, sortare sau shooting.
Dual PTO elimină această ineficiență:
Folosește motoarele existente de intake și mid roller deja alimentate și integrate în control.
Costul suplimentar este exclusiv mecanic: un gear de cuplare care conectează axul motorului la lanțul cinematic al spoolului.

Parametru	V1	V2	V3	V4	V5
Actuator	2x AXON Max	motor dedicat	motor dedicat	motor dedicat	dual PTO
Transmisie	directă	GT2 20T	bevel clampuri	bevel clampuri	pulley 1:1 + gear 24T/60T
Spool	—	1x comun	1x comun	1x comun	2x independente
Cuplare PTO	—	—	—	—	servo → gear, motoare opri
Marjă cuplu	minimă	~3.6%	~3.6%	~3.6%	~49.6%
Glisiere	—	—	dezaligniate	CNC aliniate	CNC aliniate
Platformă	brațe laterale	spool compact	subdimensionată	+40mm	+40mm
Motor dedicat parcare	da	da	da	da	eliminat
Timp parcare total	~1.2s	~0.15s	~0.12s	~0.12s	~0.25s
Testat competițional	da	da	da	da	nu
Fiabilitate	scăzută	medie	ridicată	ridicată	nevalidată competițional

ARHITECTURĂ

Arhitectura codului a fost gândită pe **principiul modularizării**, astfel fiecare subsistem hardware dar și logic este **abstractizat într-o clasa independentă**, într-un fișier separat.

Principalele sisteme ale robotului sunt **drivetrain-ul, shooter-ul și intake-ul.** Dintre care primele două sunt **dependențe** de **sistemul de localizare.** Sistemul de viziune este folosit atât în sistemul de localizare, cât și în autonomii. **Sortarea este integrată în sistemul de intake.** În final, **toate sistemele sunt integrate în TeleOp**, autonomiile neavând implementat sistemul de localizare.

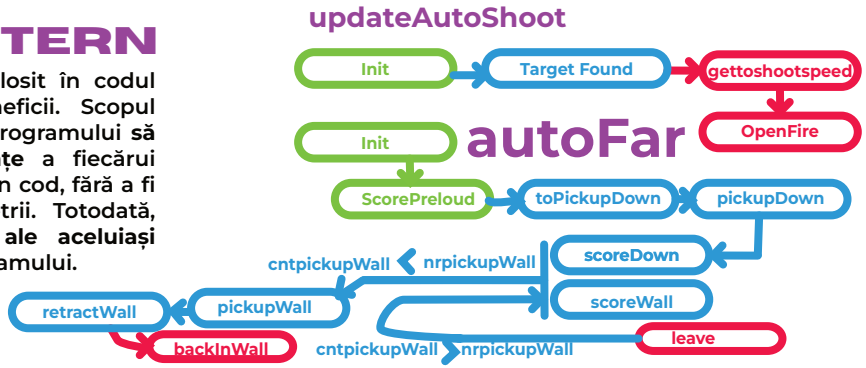
SINGLETON PATTERN

Pattern-ul Singleton este foarte des folosit în codul nostru, acesta având numeroase beneficii. Scopul acestuia este că pe durata desfășurării programului **să asigure existența unei singure instanțe** a fiecărui subsistem. Astfel, pot fi acesate oriunde în cod, fără a fi nevoie să fie transmise drept parametrii. Totodată, **previne prezența unor stări diferite ale aceluiași subsistem**, eliminând inconsistența programului.

FINITE STATE MACHINE

Finite State Machine-ul este un model de control în care **sistemul analizat se află mereu într-o singură stare** la un moment de timp. Acesta poate trece în alte stări prin tranziții bine definite, pe baza unor condiții. Noi folosim acest pattern pentru **shooter în autonomie, intake** și **sortare, lock-în drivetrain** și în fiecare clasa principala de autonomie (**PathStates**).

EXEMPLE



MecanumDrive

Drivetrain-ul, specializat în MecanumDrive, **se ocupă cu tot ceea ce implică mișcarea robotului** pe teren în timpul teleop-ului. Clasa MecanumDrive are drept feature-uri:

- **Inițializare**
- **Funcție Update**
- **Activare/Dezactivare LOCK-IN**
- **Speed multiplier**

LOCK-IN

LOCK-IN este funcția care **ne permite să țintim goal-ul cu ușurință în perioada de TeleOp** prin preluarea controlului LimeLight-ului asupra drivetrain-ului și folosind rotationPID.

```
private void driveLockedIn(double y, double x){
    x*=speedMultiplier;
    y*=speedMultiplier;

    double rx = rotationPID.run();
    if(Math.abs(rotationPID.getError())<=headingThreshold){

        rx=0;
        telemetry.addLine( lineCaption: "threshold reached");
    }
    telemetry.addData( caption: "pid rx", rx);
```

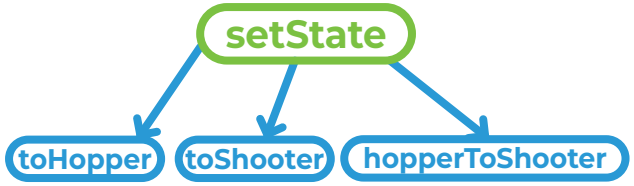
INTAKE

Intake-ul are rolul de a **gestiona flow-ul artefactelor** atât în timpul TeleOp-ului cât și în timpul autonomiei. Acesta constă **într-un finite state machine care controlează poziția one way door-ului și puterea intake-ului** propriu-zis respectiv a midroller-ului.

STATE-URI POSIBILE PENTRU INTAKE

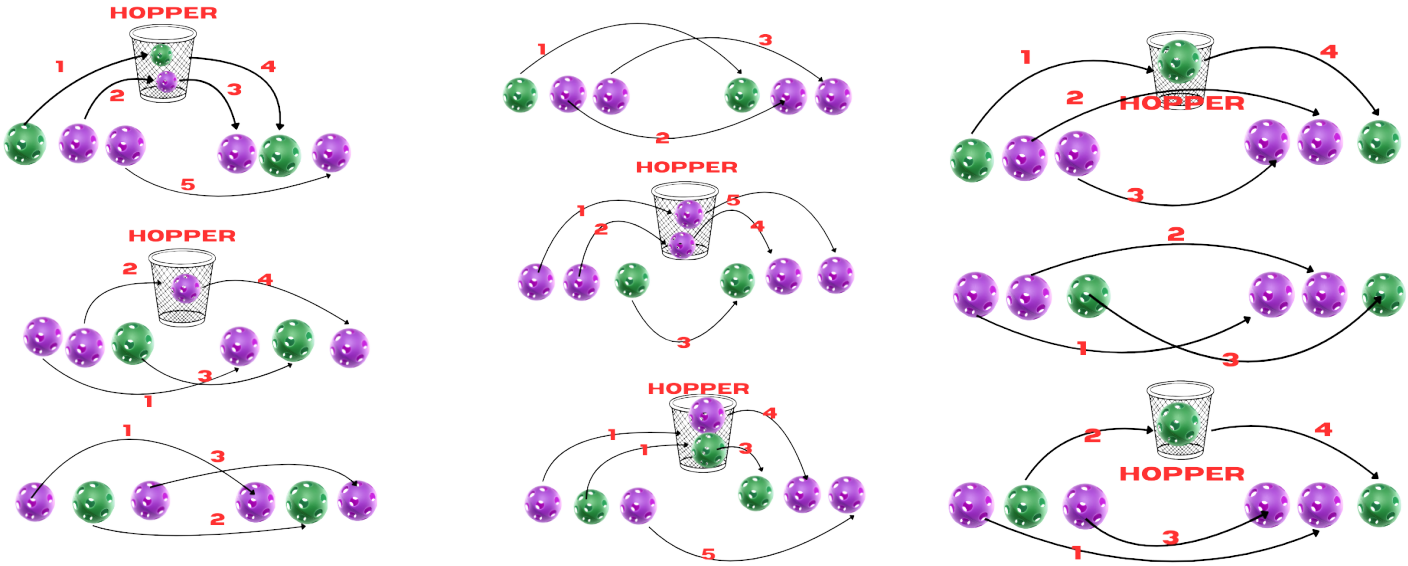
```
/// Intake states
public enum IntakeStates {
    5 usages
    NONE,
    TO_HOPPER,
    TO_SHOOTER,
    13 usages
    HOPPER_TO_SHOOTER,
}
```

Metodă setState primește ca parametrii **starea dorită pentru intake cât și un multiplier pentru a gestiona viteză și sensul de rotație în caz de nevoie.** Prin acest Finite State sunt controlate pozițiile bilei, realizându-se în cele din urmă sortarea.



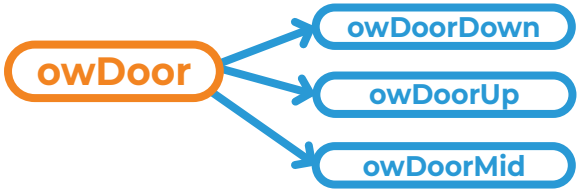

```
public void setState(IntakeStates state, double powerMultiplier){
    switch (state){
        case TO_HOPPER:
            owdoorDown();
            setIntakePower(defaultIntakePower*powerMultiplier);
            setMidrollerPower(-defaultMidRollerPower*powerMultiplier);
            break;
        case TO_SHOOTER:
            owdoorMid();
            setIntakePower(defaultIntakePower*powerMultiplier);
            setMidrollerPower(defaultMidRollerPower*powerMultiplier);
            break;
        case HOPPER_TO_SHOOTER:
            owdoorUp();
            setIntakePower(-defaultIntakePower*powerMultiplier);
            setMidrollerPower(defaultMidRollerPower*powerMultiplier);
            break;
        case NONE:
        default:
            owdoorMid();
            setIntakePower(0);
            setMidrollerPower(0);
            break;
    }
    currentState=state;
```

SORTAREA LINIARĂ



ONE WAY DOOR

Acestea sunt **metodele cele mai low level** care interacționează cu componeta hardware. **Ele sunt folosite mai departe adăugând un nivel de abstractizare.** Un lucru important de menționat este că sensitivitatea servoului a fost programată extern printr-un **Axon Servo Programmer** astfel că servo ul să aibă **un threshold în atingerea poziției** dorite pentru a elimina oscilațiile.




```
/// Owdoor
2 usages
private static double owdoorUpPos=0.11,owdoorMidPos=0.58,owdoorDownPos=0.97;
3 usages
private double lastOwdoorPos=-1;
1 usage
public void owdoorDown(){
    if(lastOwdoorPos==owdoorDownPos) return;
    owdoor.setPosition(owdoorDownPos);
}
1 usage
public void owdoorUp(){
    if(lastOwdoorPos==owdoorUpPos) return;
    owdoor.setPosition(owdoorUpPos);
}
2 usages
public void owdoorMid(){
    if(lastOwdoorPos==owdoorMidPos) return;
    owdoor.setPosition(owdoorMidPos);
}

/// Motor powers
4 usages
private void setIntakePower(double power){
    intakeMotor.setPower(power);
}
4 usages
private void setMidrollerPower(double power){
    midrollerMotor.setPower(power);
```

LOCALIZARE

Sistemul de localizare stă la baza bunei funcționari a shooter-ului. Funcțiile acestuia sunt determinarea **orientării necesare** și a **distanței** față de goal a robotului. În decursul sezonului, acesta a suferit **o gama largă de schimbări.**

Versiune finală (Odometrie + LimeLight)

Sistemul curent de localizare **se bazează atât pe LimeLight cât și pe odometrie.** Am ales acest sistem dublu întrucât cele două modalități de localizare au **caracteristici complementare.** Astfel, din punct de vedere al preciziei, **LimeLight-ul păstrează o acuratețe constanța** pe toată durata meciului, în timp ce **odometria este predispusă la pierderi.** Un exemplu des întâlnit în timpul mecirulor este **contactul dintre roboți care destabilizează IMU-ul pinpoint-ului.** De asemenea, LimeLight-ul **nu necesită resetare manuală**, spre deosebire de odometrie. Cea din urmă este dependență de inițializarea cu o poziție de pornire care încurcă flow-ul driver-ului.

ANALIZĂ COST/OPORTUNITATE

LIMELIGHT

- Precizie constantă
- Nu necesită resetare
- Are refresh rate scăzut
- Vulnerabilă la motion-blur
- Poate fi blocată de alți roboți

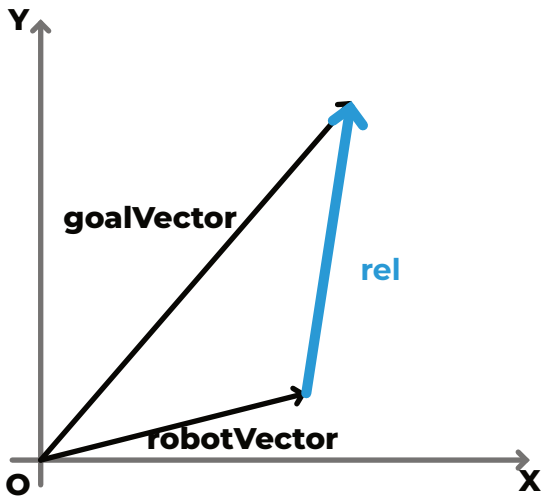
ODOMETRIE

- Are pierderi pe durata meciului
- Vulnerabilă la contactul cu alți roboți
- Necesită cunoașterea unei poziții în prealabil
- Vibrațiile shooter-ului generează drift în IMU-ul pinpoint-ului
- Refresh rate ridicat

Funcția de update

```
private Pose2D estimatedPose;
public void update() {

    odometry.update();

    Pose2D odometryPose = odometry.getPose(),llPose=null;
    LimelightHandler.get().updateRobotPose(odometryPose.getHeading(AngleUnit.DEGREES));

    if(LimelightHandler.get().getRobotPose()!=null) {
        llPose = new Pose2D(DistanceUnit.METER,
                LimelightHandler.get().getRobotPose().getPosition().x,
                LimelightHandler.get().getRobotPose().getPosition().y,
                AngleUnit.DEGREES,
                LimelightHandler.get().getRobotPose().getOrientation().getYaw(AngleUnit.DEGREES));

        if(!usePrimaryOdometry || !odometryActive){
            odometry.setPose(llPose);
            odometryPose=odometry.getPose();
            odometryActive=true;
        }

        estimatedPose=llPose;
    }
    if(odometryActive && (usePrimaryOdometry || llPose==null)){
        estimatedPose=odometryPose;
    }
}
```

Astfel, principiul pe care se bazează sistemul este **actualizarea în mod regulat a odometriei cu coordonatele obținute** de la LimeLight. În plus, odată ce LimeLight-ul actualizează pentru prima dată odometria, **se poate opta pentru folosirea unui anumit sistem drept sistem principal** în defavoarea celuilalt.



Determinarea rotației

Cu ajutorul calculului vectorial **se determina vectorul „rel”** a cărui **magnitudine este distanță căutată** iar unghiul acestuia reprezintă orientarea necesară a robotului. **Eroarea de orientare** este transmisă mai apoi drivetrain-ului pentru a o corectă și a îndeplini funcția de aliniere automată.

Pentru ca artefactul să nu sară din depot, **terenul este împărțit în două regiuni,** iar în funcție din ce zona trage robotul, **este ales goal pose-ul** pe unul dintre pereții depotului. Astfel, precizia de tragere este îmbunătățită, **scăzând riscul de a ieși bilele din depot.**

Versiune inițială

Inițial, sistemul de localizare **era conceput să folosească Megatag2,** iar yaw-ul robotului să fie dat de un PoseTracker care prelucrează IMU-ul Pinpoint-urilor Gobilda. În cazul în care odometria era activată, **poziția rezultată avea drept coordonate interpolarea coordonatelor date de LimeLight** cu cele ale odometriei. In urma multor teste am identificat **probleme legate de trecerea din cele doua sisteme de coordonate, FTCCoordinates si Pedropathing.**

Versiune următoare

Următoarea versiune a acestui sistem era **bazată tot pe localizare cu Megatag2, headingul poziției fiind dat de odometrie.** Acest sistem era conceput mult mai simplu, poziția estimată fiind egală cu cea dată de limelight. Pentru a putea utiliza **orientarea dată de Pinpoint-uri**, odometria necesită resetare. Astfel, este concepută funcția de **setHeading() pentru a putea fi implementată în TeleOp.**

```
public void setHeading(double heading){
    odometryTracker.setPose(new Pose( x: 0,  y: 0, heading));
    LimelightHandler.get().setUseMT2(true);
}
```

Versiunea premergătoare celei finale a constat **în adăugarea unei funcții de transmitere a poziției receptate de LimeLight** către odometrie. Astfel, după resetarea odometriei este apelata această funcție pentru că odometria să fie actualizată cu valorile preluate de limelight.

SHOOTER

Sistemul de aruncare a bilelor este conceput, astfel încât **viteză optimă de tragere** să fie calculată pe baza **distanței** dintre poziția robotului și cea goal-ului, fiind evitată utilizarea unor valori statice.

$$v_0^2 = \frac{gd^2(1+\tan^2\alpha)}{2(d\tan\alpha+\Delta h)}$$

α - unghiul shooter-ului

g - accelerația gravitațională

d - distanța dintre robot și goal

Pentru a spori acuratețea sistemului, viteză este menținută constanța în timpul aruncării artefactelor cu ajutorului unui controller dublu de PID-uri.

CONTROLLER PIDF

Acesta este implementat prin **două seturi de coeficienți PIDF.** Cel primar este **mai agresiv**, iar cel **secundar** este **cel fin.** Astfel roulul primului este de a aduce **cât de rapid posibil eroarea sub un anumit prag,** iar de acolo minimizarea erorii este efectuată de al doilea set.

Sistemul este conceput **să folosească integrală numai când este cu adevărat utilă,** cazurile propice fiind acelea când motorul nu își atinge saturatia sau când și-o atinge, însă eroarea ajută la ieșirea din saturație.

Varianta 2 PIDF

Varianta precedentă avea în ambele seturi de coeficienți PIDF integrală egală cu 0. De asemenea, controller-ul de PID-uri era resetat după fiecare iterație. După o serie de teste **am sesizat că integrală trebuie să fie mai mare ca 0** pentru a elimina eroarea statică de viteză, iar această nu trebuie resetată pentru că își pierde din „memorie" , fiind recursivă $I_k = I_{k-1} + e_k \Delta t$

```
if(speedPid.getTargetPosition()==speed) return;
speedPid.reset();
speedPid.setTargetPosition(speed);
```

```
private static FilteredPIDFCoefficients speedPidCoeff = new FilteredPIDFCoefficients( p: 0.1,  i: 0,  d: 0,  t: 0,  f: 0);/// f=0
private static FilteredPIDFCoefficients speedPidCoeffSecondary = new FilteredPIDFCoefficients( p: 0.003,  i: 0,  d: 0.00001,  t: 0,  f: 0
private FilteredPIDFController speedPid = new FilteredPIDFController(speedPidCoeff);
```

```
public void updateError(double error) {
    long now = System.nanoTime();
    deltaTimeNano = now - previousUpdateTimeNano;
    previousUpdateTimeNano = now;

    double dt = deltaTimeNano / 1e9;
    if (dt <= 0) return;

    previousError = this.error;
    this.error = error;

    // ----- Derivative -----
    errorDerivative = (error - previousError) / dt;
    previousDerivative = filteredDerivative;
    filteredDerivative = T() * previousDerivative + (1 - T()) * errorDerivative;

    // ----- Anti-Windup (Conditional Integration) -----
    double candidateIntegral = errorIntegral + error * dt;

    double predictedRaw = error * P() + filteredDerivative * D() + candidateIntegral * I() + F() * feedForwardInput;

    double predictedClamped = MathFunctions.clamp(predictedRaw, LowerLimit(), UpperLimit());

    // Only integrate if not saturating further
    if (!(predictedRaw != predictedClamped && Math.signum(error)==Math.signum(predictedRaw))) {
        errorIntegral = candidateIntegral;
    }
```

Randament și viteză

În funcție de launch zone-ul din care trage robotul, am **determinat experimental două randamente.** În scopul de a eficientiza timpul lansării bilelor, **am optat să implementăm două viteze intermediare**, una pentru apropiere și una pentru îndepărtare. **Aceste pot fi păstrate constant** pe tot parcursul meciului și schimbate în funcție de locul de tragere, astfel viteză dorită putând fi atinsă mult mai repede.

```
if(Shooter.get().getRestSpeed()==0){
    Shooter.get().setRestSpeed(restShortSpeed);
}
else if(Shooter.get().getRestSpeed()==restShortSpeed){
    Shooter.get().setRestSpeed(restLongSpeed);
}
else {
    Shooter.get().setRestSpeed(0);
}
if(!shooting)
    Shooter.get().toRest();
```

Autonomie

În timpul perioadei de autonomie, **shooter-ul este gândit să utilizeze o viteză statică,** iar stările acestuia sunt modelate printr-un **Finite State Machine** care este apelat în funcția de loop.

LIMELIGHT HANDLER

Viziunea robotului este datorată camerei integrată cu AI limelight 3A. Această este implementată în cod într-o clasa numită LimelightHandler. Pentru a schimbă pipeline-urile între ele, în funcție de ce apriltag trebuie să recepteze sunt utilizate metodele setShooterPipeline() și setObeliskPipeLine().

```
public void setShooterPipeline(){

    if(currentPipeLine!=1)
        ll.pipelineSwitch( index: 1);
    currentPipeLine=1;
}
no usages
public void setPipeline(int index){
    if(currentPipeLine!=index)
        ll.pipelineSwitch(index);
    currentPipeLine=index;
}
3 usages
public void setObeliskPipeline(){
    if(currentPipeLine!=0)
        ll.pipelineSwitch( index: 0);
    currentPipeLine=0;
}
```

În cazul sortării, **funcția updateObeliskID()** este actualizată în loop() pentru a putea identifica fiducial-ul corespunzător motifului din meciul respectiv.

Funcția getGreenPosition() prelucrează id-ul obținut de la obelisk și determina astfel poziția bilei verzi din pattern.

```
public int getGreenPosition() {
    if (obeliskID == 0) {
        return 0;
    }
    return obeliskID-20;
}
```

Versiune inițială

Sistemul inițial de localizare era integrat în clasa LimelightHandler, unde reprezenta unul dintre task-urile esențiale asociate camerei Limelight. Rolul principal al acestui modul era identificarea reperelor vizuale din teren și determinarea poziției acestora în raport cu robotul. Prin intermediul metodei identifyTarget, sistemul analiza cele mai recente date furnizate de cameră, verifica validitatea rezultatului obținut și căuta fiducialul corespunzător identificatorului cerut. În situația în care tag-ul era detectat corect, metoda returna poziția depoului în spațiul robotului, sub forma unei transformări 3D, informație necesară pentru etapele ulterioare de navigație și aliniere.

```
public Pose3D identifyTarget(int targetId){

    if(ll == null) return null;

    LLResult result = ll.getLatestResult();
    if(result == null || !result.isValid()) return null;

    LLResultTypes.FiducialResult depotTag = findFiducialById(result, targetId);

    if(depotTag == null) return null;

    telemetry.addData( caption: "apriltag pose", format: "x %f y %f z %f, pitch %f, yaw %f,

    return depotTag.getTargetPoseRobotSpace();
}
```

Metoda UpdateDistanceAndAngle

Metoda **UpdateDistanceAndAngle** era concepută pentru a calcula ultimul unghi și ultima distanță necesare poziționării robotului față de un punct de interes aflat în spatele apriltag-ului, **la o distanță prestabilită prin parametrul distanceTargetZ.** Practic, această metodă **nu urmărea doar poziția brută** a tag-ului, ci estima **o poziție țintă offsetată,** astfel încât robotul să nu se oprească exact în dreptul markerului, ci **într-un punct optim pentru interacțiune.** Pe baza coordonatelor furnizate de sistemul de localizare, se determinau atât distanța liniară până la acel punct, cât și unghiul de orientare necesar pentru ca robotul să ajungă corect aliniat.

Versiune finală

Versiunea curentă a LimelightHandler-ului se bazează pe o metodă de actualizare a poziției robotului în funcție de heading. Dacă **utilizarea Megatag2 este activată, LimeLight-ul** preia orientarea transmisă prin parametrii. In plus, am decis sa realizam calibrarea ChArUco pentru o mai buna precizie a camerei.

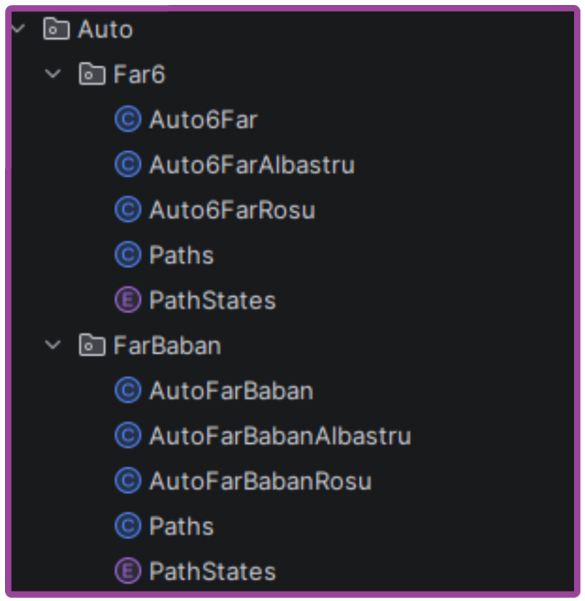
```
public void setUseMT2(boolean active) { useMT2=active; }
2 usages
public void updateRobotPose(double yawDegrees){

    if(ll == null){
        robotPose=null;
        return;
    }
    if(useMT2)
    {
        ll.updateRobotOrientation(yawDegrees);
    }


    LLResult result = ll.getLatestResult();
    if(result == null || !result.isValid()){
        robotPose=null;
        return;
    }
    if(!useMT2)
        robotPose = result.getBotpose();
    else robotPose = result.getBotpose_MT2();
}
5 usages
public Pose3D getRobotPose() { return robotPose; }
```

Calibrarea ChArUco

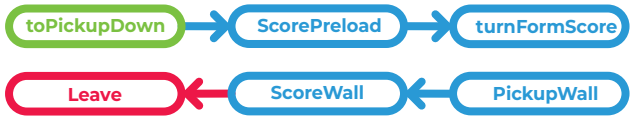
Calibrarea ChArUco a camerei Limelight este procesul prin care **se determină parametrii optici reali** ai camerei și coeficienții de distorsiune ai lentilei folosind o placă specială cu markere ArUco. Această procedură **permite corectarea imaginii** și îmbunătățește semnificativ precizia estimărilor 3D și a poziției țintelor, fiind esențială pentru localizare în robotică.

AUTONOMOUS

Autonomiile sunt **structurate pe 5 clase:** o **clasa** specifică pentru path-uri, un **enum** care conține stările, **clasa** principala unde este implementat **Finite state machine**-ul și altele **două** în funcție de culoarea alianței.



Exemplu Auto6Far



12 bile scorate, dintre care 7 sortate

loop()

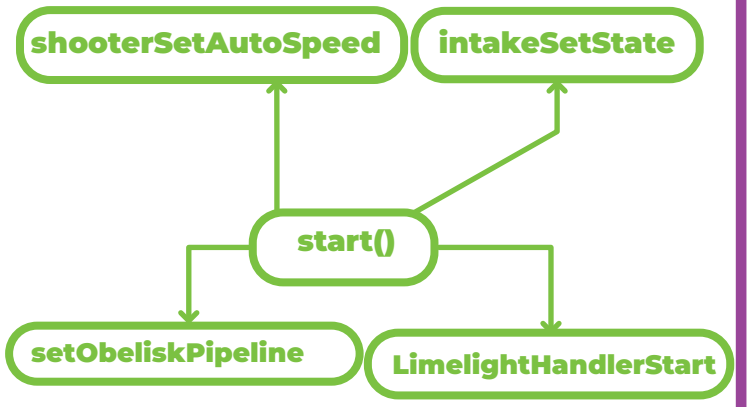
În loop() este **actualizat follower-ul** din pedropathing care urmărește traiectoriile. De asemenea, este apelata funcția de update a funcției de shoot specifică autonomiei, care trebuie să mențină o viteză statică (shooterSpeed).

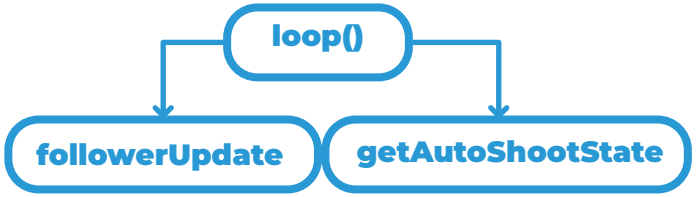
init()

În init() sunt **inițializate celelalte sisteme** folosite: LimelightHandler, shooter-ul, și Intake-ul. De asemenea, este atribuită follower-ului poziția de început.

start()

În start() **este selectat pipeline-ul necesar** pentru obleisk, iar shooter-ului îi este setată viteză de tragere. În plus, sistemul de captare a bielelor este setat în poziția lui default, în care artefactul merge spre shooter.





Sortare în autonomie

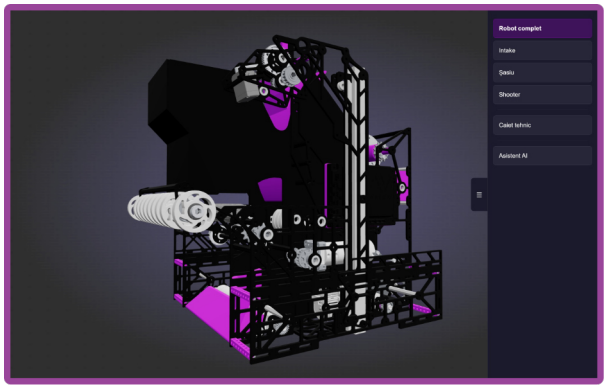
Pentru **a putea sorta cele 3 bile** indiferent de motif, am decis să implementăm câte **o clasa de path-uri pentru fiecare pattern**. Sortarea este efectuată **pe poziții**, ci nu pe timere. Pentru **a putea controla utilizarea sistemelor pe parcursul traiectoriilor** este utilizată funcția de **addParametricalCallback(),** alegând astfel de la ce procent din path să pornească intake-ul.

TeleOP

În perioada de TeleOp, am ales **o strategie de tip flow. Sortarea liniară** a robotului este **utilizată doar în autonomous,** astfel încât driverul să se poată concentra pe **scorarea unui număr cât mai mare de elemente.** Automatizările implementate pe robot îi permit acestuia să mențină o precizie ridicată, **putând trage atât de la far, cât și de la close,** cu o acuratețe foarte bună. Pentru a preveni ca artefactul să sară din depot **din cauza efectului de backspin,** terenul este împărțit în două regiuni. În funcție de zona din care robotul execută tragerea, **este ales goal-ul corespunzător** de pe unul dintre pereții depotului, **reducând astfel riscul de a nu puncta întregul ciclu.**

FTC ROBOT VISUALISER

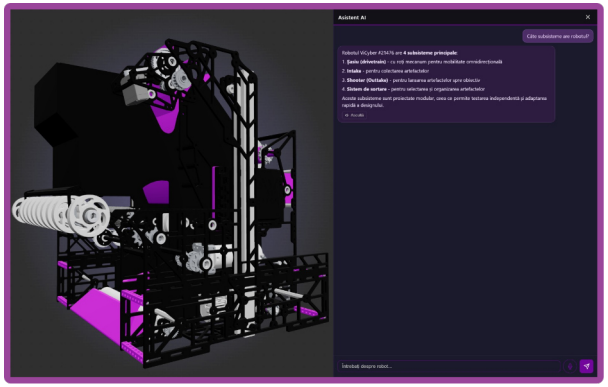
FTC Robot Visualizer este o aplicație desktop și web interactivă dezvoltată pentru **a ajuta judecătorii de competiție să exploreze și să înțeleagă designul robotului**, echipei în mod vizual și intuitiv, fără a necesita explicații verbale extinse. Aplicația afișează modelul 3D complet al robotului și permite utilizatorilor să navigheze prin subsisteme individuale — Intake, Șasiu, Shooter — **cu tranziții animate de cameră, izolare vizuală a componentelor, etichete informative direct în scenă 3D și un viewer integrat pentru caietul tehnic.**



Stack tehnologic

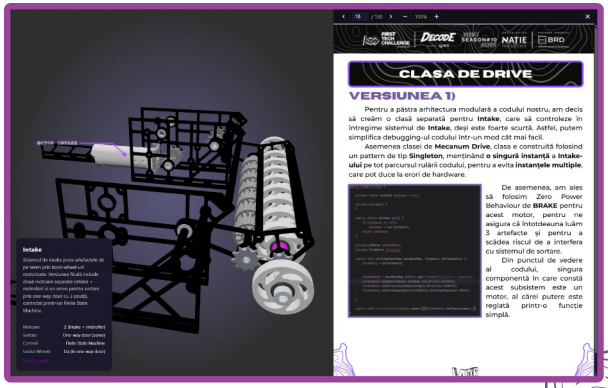
Three.js — rendering 3D WebGL cu pipeline ACES filmic, materiale PBR și un sistem de iluminare studiou cu 6 surse
camera-controls — orbita, zoom și animații de cameră fluide cu preset system
pdfjs-dist — viewer PDF embedded cu zoom și navigare pagini
Ollama — chat AI contextual bazat pe cunoștințe din caietul de inginerie
Vite — build tool cu dev server și plugin custom
Tauri 2 — wrapping desktop nativ pentru Windows, fullscreen kiosk, acces la sistemul de fișiere
TypeScript — ~8000 linii de cod în strict mode, fără erori de compilare

three.js



Funcționalități principale

- **Vizualizare 3D** full-robot cu auto-rotate la pornire, orbita liberă și zoom cu mouse și touch
- **Selectare subsisteme** — click dintr-un sidebar navighează camera animat la fiecare subsistem, dimming-ul evidențiază modelul selectat, panoul lateral arată specificații și descriere
- **Callout HUD** — etichete SVG cu linii de referință ancorate în spațiul 3D, stilizate în culorile echipei
- **Viewer engineering notebook** — PDF integrat sincronizat cu subsistemul selectat, direct la pagina relevantă
- **AI Chat** — panou Ctrl+Shift+A cu Claude Haiku sau Ollama local ca fallback offline, context complet din knowledge base
- Funcționează 100% offline — toate resursele bundle-uite local, zero CDN-uri externe



Probleme întâmpinate în procesul de dezvoltare

Performanță slabă rendering pe hardware modest
Interacțiune lentă pe High DPI display
Calibrarea pozițiilor callout-urilor în 3D
Sistem de auto-rotate inteligent
Overlay-uri 2D ancorate în spațiul 3D
Lazy loading și caching modele GLB
Integrare AI dual-provider (cloud + offline)

Soluții

Pentru fiecare caz, problema **a fost analizată tehnic,** identificând cauza principală: materiale PBR prea **costisitoare,** rezoluție **prea mare** în timpul interacțiunii, lipsa punctelor de referință în model sau **comportament rigid** al auto-rotate-ului. Apoi au fost dezvoltate **soluții țintite** și **testabile,** de exemplu **simplificarea materialelor, reducerea dinamică** a pixel ratio-ului, introducerea unui debug mode pentru capturarea coordonatelor și **definirea unui state machine** pentru rotație. În final, arhitectura a fost organizată astfel încât soluțiile **să rămână modulare, eficiente** și ușor de extins, fără să afecteze restul aplicației.