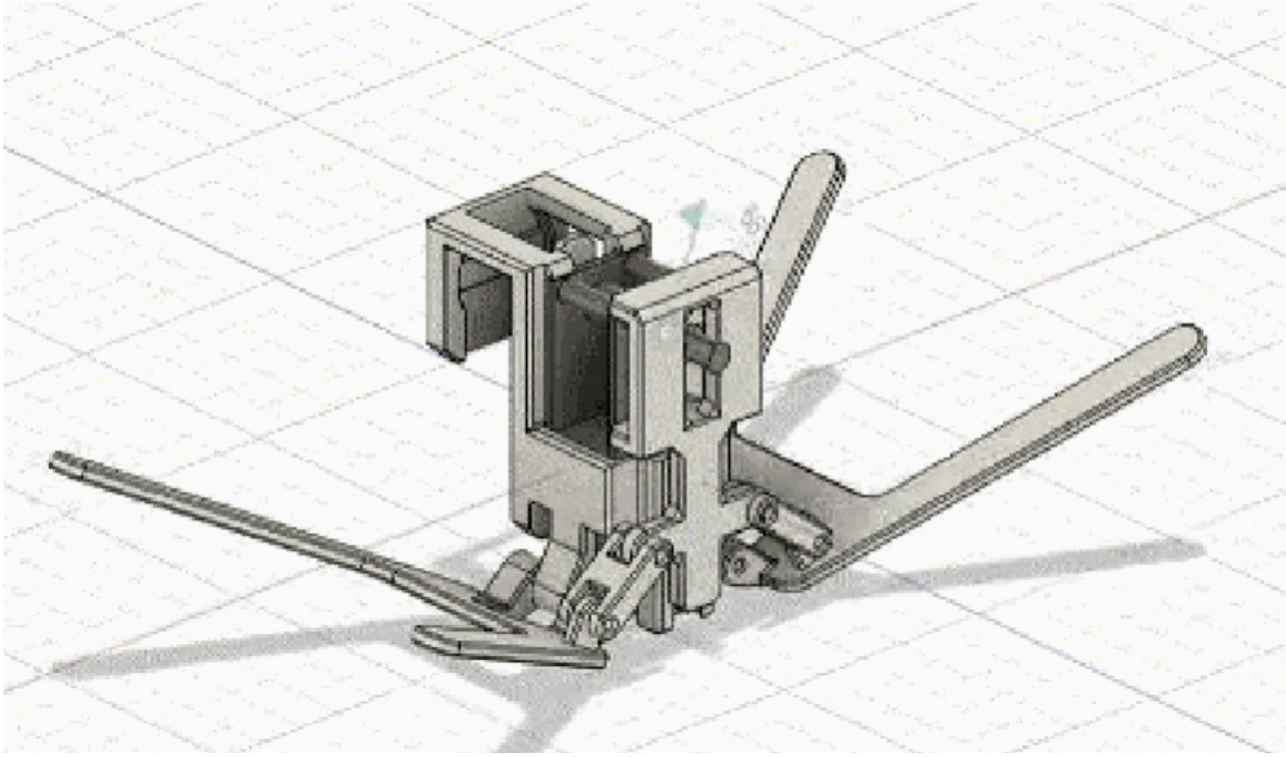
# P3: Generating physical interfaces Write-up

cmis-25SP(05899)-Team-3: Christine Mendoza (mendoza@cmu.edu), Flora Xiao (weierx@andrew.cmu.edu), Nivedhitha Dhanasekaran (ndhanase@cs.cmu.edu)

Github: https://github.com/Kiddo-Xiao/cmis-25SP-P3-PhysicalUI

## Section 1: Initial Designs and Evolution into Final Proposal

### Initial Design 1: Origami-Inspired Flapping Animals



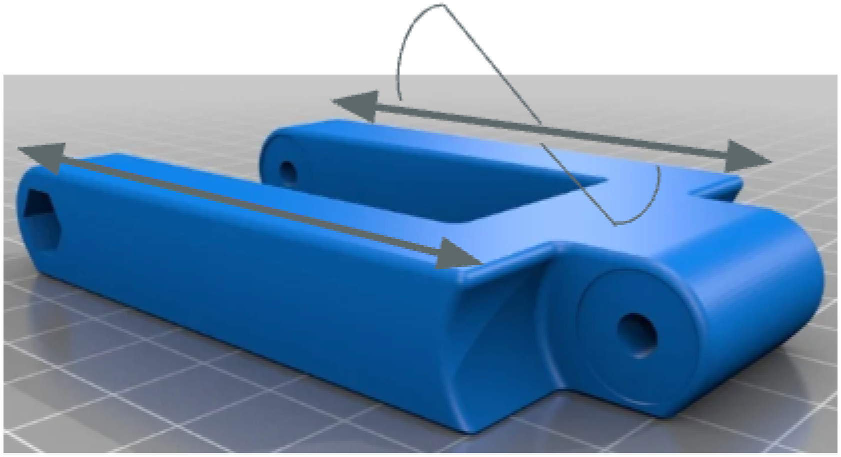
This idea proposed the creation of kinetic, origami-style animal figures with flapping wings. The primary motivation was to offer a calming and aesthetically pleasing interactive object suitable for desk use or stress relief. These models would target users interested in artistic, gentle interactions.

Personalization was a key feature. Wing flap amplitude, softness of movement, and hinge design could be customized through parameterized modeling. Users could select different animals and tune the motion style to their preference.

The optimization involved inputs like actuation type, flap frequency, and softness. Decision variables included hinge thickness, wing angle, and link length. The objective was to maximize flap performance within safe mechanical constraints, balancing motor torque, material stress, and overall symmetry.

Fabrication was planned via a combination of paper elements and 3D-printed hinges or soft joints. Three working examples would be printed and tested, each showing distinct behavior under motion.

## Initial Design 2: Modular Eye-Tracker Mount



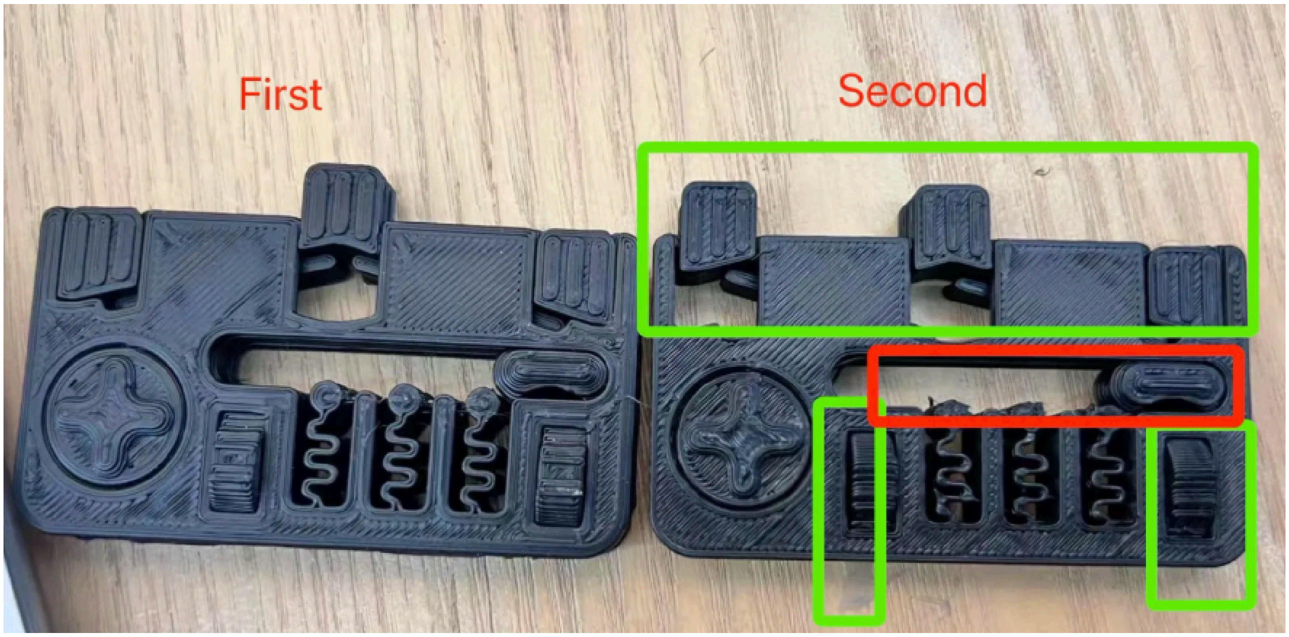
This design addressed users with accessibility needs. It aimed to help people who rely on eye-tracking devices by offering a modular, position-adjustable mount that could be configured to different environments: desk, wheelchair, or bed.

The design required personalization because each user's posture, reach, and visual alignment vary significantly. A fixed design would not meet the practical demands of a diverse user base.

User input would include preferred viewing angle, height, and available mounting position. Optimization variables were hinge lengths, rotation limits, and segment counts. The objective was to maximize the device's stable range of motion while preserving alignment and structural safety.

PLA or PETG would be used for hinge-based components. Devices would be printed in modular parts for assembly. Three variants would be fabricated for three usage contexts: a tabletop unit, a vertical post mount, and a reclining configuration.

### Initial Design 3: Customizable Stress-Relief Toy



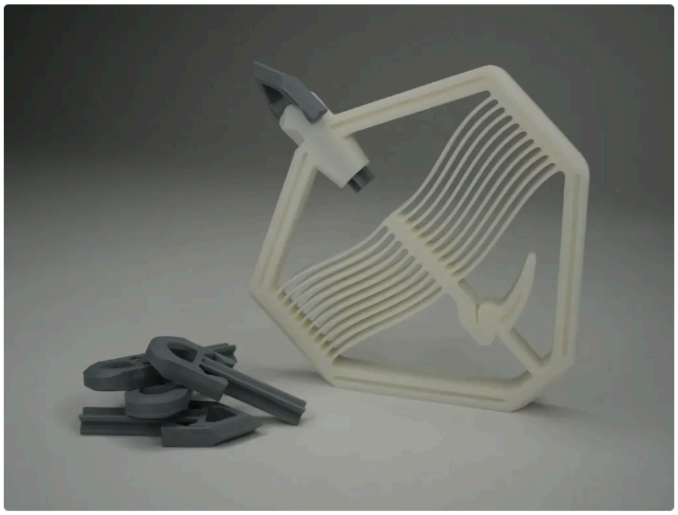
This proposal focused on a personalized hand toy optimized for different types of tactile interaction: squeezing, bending, or flicking. It was designed for three types of users: children, adults, and elderly individuals with physical therapy needs.

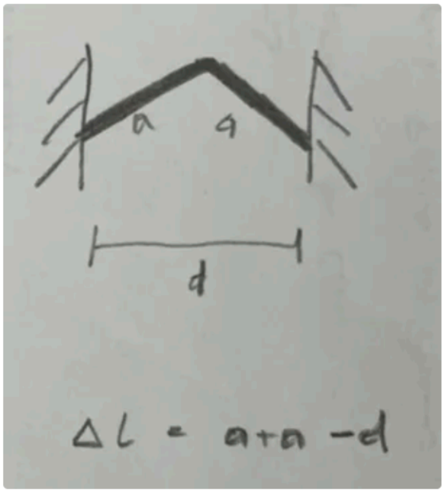
Customization was critical. Children would benefit from soft, oversized grips. Adults may prefer textured control areas or stiffer resistance. Elderly users may need ergonomically curved or lightly damped parts to support joint motion recovery.

User input would include hand size, preferred stiffness, and interaction style. Variables included curvature, thickness, and material distribution. The goal was to maximize comfort and engagement while remaining within printability limits.

Fabrication involved PLA or TPU with modular attachments and replaceable damping layers. Printed variants would be distributed across user groups for testing.

### Final Project Direction and Class Feedback Integration





During our class proposal discussion, the stress relief toy emerged as a conceptually rich idea, but with a need for better defined mechanical goals. The professor introduced the notion of bistable mechanisms, referencing the deformation and release behavior seen in spring-like materials.

The professor explained that computing properties like shooting speed would require full dynamic simulation, which could complicate implementation. Instead, the suggestion was to model elastic deformation geometrically, using the parameter delta L—the arc length difference between relaxed and tensioned states—as a proxy for stored force.

This idea reframed our thinking. We decided to shift toward a handheld bow mechanism that incorporates bistable geometry and supports geometric optimization without needing physics simulation. Rather than optimizing velocity directly, we target printable geometry that encourages different deformation profiles.

We preserved the customization ideas from the initial ideas but grounded them in a single, structured object. The bow's limbs, grip, and tension curve could be optimized per user profile. A fixed-size arrow slot ensures compatibility across designs. The professor also recommended printing and testing several versions early to empirically determine how geometric changes affect performance, and we adopted this approach for the rest of our workflow.

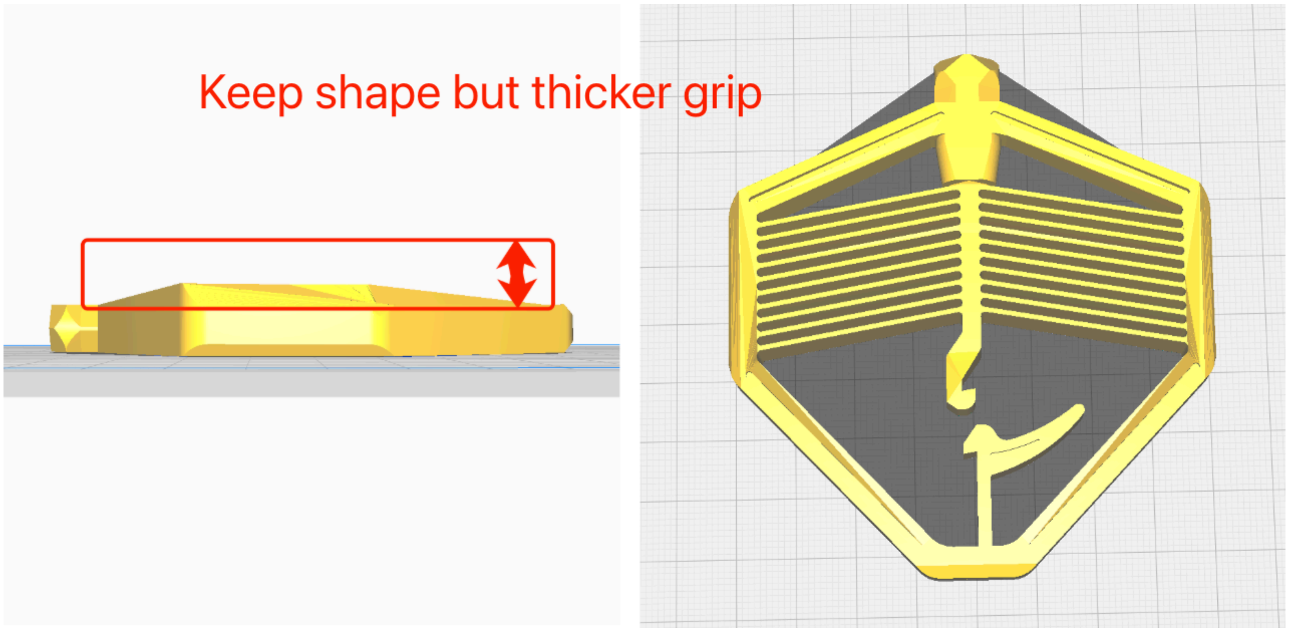
The final project combines the ergonomic intent of the toy, the mechanical precision of the mount, and the geometric tuning advised in class. It is also fabricable without assembly and customizable across different user profiles.

## Section 2: Refinement and Implementation

We refined the project over 3 iterations.

### 1. Initial Iteration: Bow Thickness Optimization

Our first iteration focused on a minimalist approach to bow toy optimization, targeting only the bow thickness parameter as the primary variable for performance adjustment. This simplified model allowed us to establish the fundamental physics relationships and optimization framework before expanding to a more comprehensive system.



### Mathematical Formulation

Decision Variable:

$x1 : Bowthickness(4.0 - 7.0mm)$

### Objective Function

We employed a simple single-objective function:

$$f(x_1) = w_p \cdot (v(x_1) - v_t)^2 + w_s \cdot \max(0, v(x_1) - v_{max})^2$$

Where:

- $v(x_1)$ is the launch speed as a function of bow thickness
- $v_t$ is the target launch speed (different for adults and children)
- $v_{max}$ is the maximum safe launch speed
- $w_p$ and $w_s$ are performance and safety weights

## Physics Model

We implemented a basic physics model for launch speed:

$$v(x_1) = k \cdot \sqrt{\frac{x_1 \cdot d^2}{m_a}}$$

Where:

- $k$ is a constant (calibrated to approximately 0.15)
- $d$ is the fixed draw length (150mm for adults, 100mm for children)
- $m_a$ is the arrow mass (20g)

## Implementation

The optimization used SciPy's minimize function with simple bounds:

```
from scipy.optimize import minimize_scalar
result = minimize_scalar(obj_func, bounds=(4.0, 7.0), method='bounded')
```

We implemented two basic user types (child and adult) with different safety thresholds and target speeds:

- Child: $v_t = 5$ m/s, $v_{max} = 8$ m/s
- Adult: $v_t = 10$ m/s, $v_{max} = 15$ m/s

## Libraries Used

- NumPy: For numerical computations
- SciPy: For optimization algorithm
- Trimesh: For basic STL export

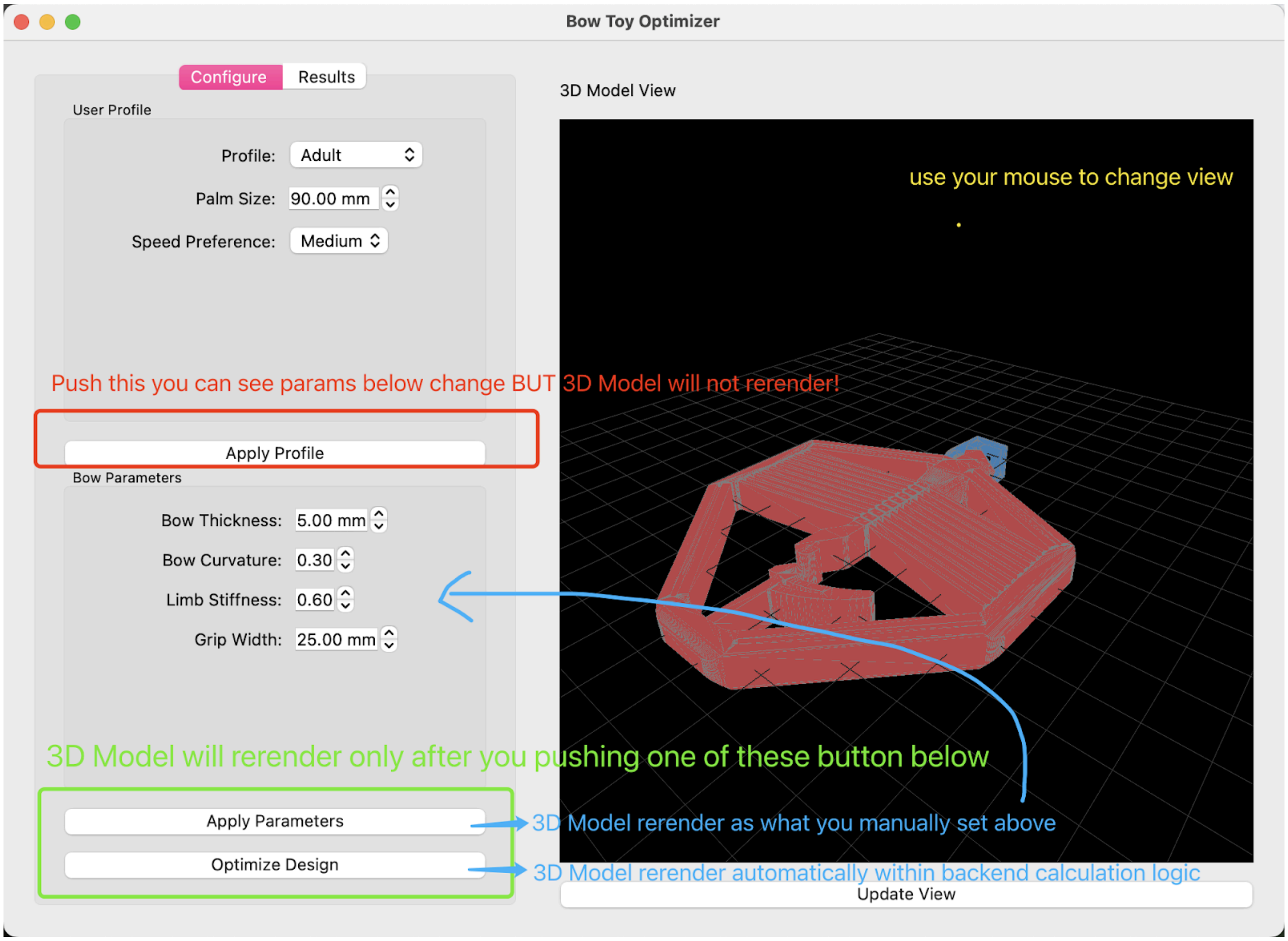
## Limitations

This initial version had several limitations:

1. Single parameter optimization only
2. Fixed arrow design regardless of bow parameters
3. No graphical user interface
4. Limited physics model with no consideration for bow curvature or material stiffness
5. No ergonomic adaptations for different users

Despite these limitations, this simplified approach provided us with valuable insights into the relationship between bow thickness and performance, serving as a foundation for our more sophisticated second iteration.

## 2. Middle Iteration: Advanced Optimization + UI



Bow Toy Optimizer

User Profile

Profile: Adult

Palm Size: 90.00 mm

Speed Preference: Medium

Push this you can see params below change BUT 3D Model will not rerender!

Bow Parameters

Bow Thickness: 5.00 mm

Bow Curvature: 0.30

Limb Stiffness: 0.60

Grip Width: 25.00 mm

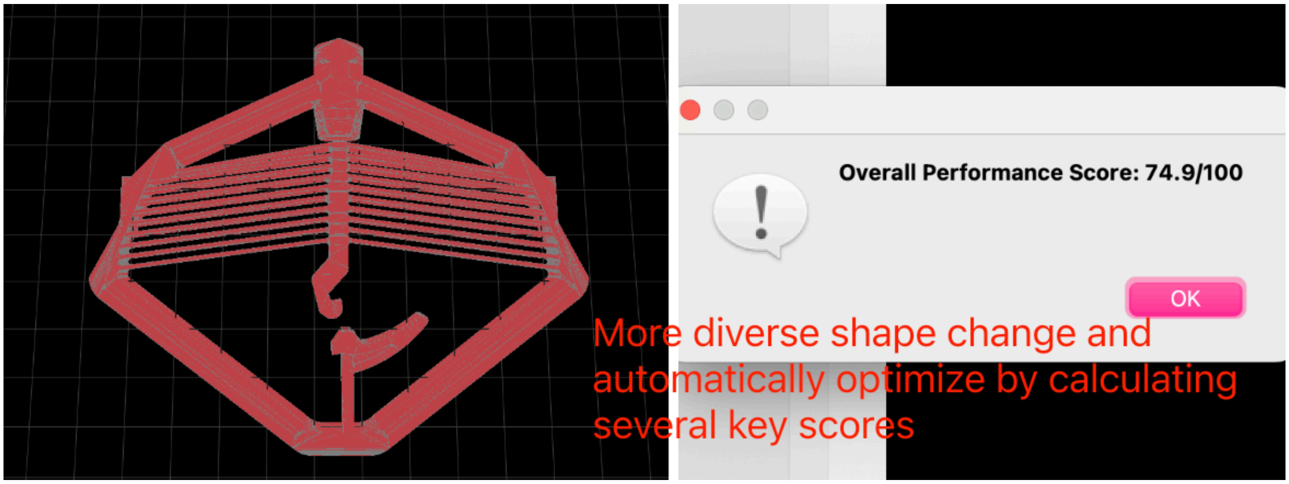
3D Model will rerender only after you pushing one of these button below

3D Model rerender as what you manually set above

3D Model rerender automatically within backend calculation logic

3D Model View

use your mouse to change view



Overall Performance Score: 74.9/100

More diverse shape change and automatically optimize by calculating several key scores

Our initial prototype focused primarily on a simplified bow design with basic grip thickness optimization for power adjustment. Based on in-class feedback, we recognized several limitations in our approach:

1. Limited parameter space: Users wanted more customization beyond just grip thickness

2. Lack of user profile adaptation: The one-size-fits-all approach didn't address different user needs

3. Simplified physics model: The performance metrics weren't accurately reflecting real-world behavior

4. Absence of ergonomic considerations: Hand size adaptation was missing from our initial design

To address these shortcomings, we significantly expanded our second iteration to include comprehensive parameter customization, user profiles, and a more sophisticated physics model.

## Mathematical Formulation of the Optimization Problem

## Decision Variables

Our optimization now operates on the following key parameters:

- $x_1$: Bow thickness (4.0-7.0mm)
- $x_2$: Bow curvature (0.2-0.4)
- $x_3$: Limb stiffness (0.3-0.9)
- $x_4$: Grip width (20-36mm)

## Objective Function

We formulated a multi-objective optimization problem with weighted priorities:

$$f(X) = \sum_{i=1}^{n} w_i \cdot (x_i - t_i)^2 + w_s \cdot S(X) + w_c \cdot C(X) + w_a \cdot A(X)$$

Where:

- $X = [x_1, x_2, x_3, x_4]$ is the parameter vector
- $t_i$ represents target values for each parameter
- $w_i$ are the weighting factors for each parameter

- $S(X)$, $C(X)$, and $A(X)$ are the safety, comfort, and accuracy penalty functions
- $w_s$, $w_c$, and $w_a$ are the weights for safety, comfort, and accuracy

## Constraints

We implemented both hard and soft constraints:

Hard Constraints (Parameter Bounds):

- $4.0 \leq x_1 \leq 7.0$ (Bow thickness)
- $0.2 \leq x_2 \leq 0.4$ (Bow curvature)
- $0.3 \leq x_3 \leq 0.9$ (Limb stiffness)
- $20 \leq x_4 \leq 36$ (Grip width)

These bounds vary by user profile, with narrower ranges for children and wider ranges for professionals.

Soft Constraints (Performance Penalties):

- Safety penalty: $S(X) = \max(0, v(X) - v_{max})^2 + \max(0, F(X) - F_{max})^2$
  - $v(X)$ is the calculated launch speed
  - $v_{max}$ is the maximum safe launch speed
  - $F(X)$ is the calculated draw force
  - $F_{max}$ is the maximum safe draw force
- Comfort penalty: $C(X) = \alpha \cdot |x_4 - p \cdot f|^2$
  - $p$ is the palm size of the user
  - $f$ is the grip size factor
  - $\alpha$ is a scaling coefficient

- Accuracy penalty: $A(X) = \beta \cdot (x_3^{-1} + x_1^{-1})$
  - $\beta$ is a scaling coefficient

## Physics-Based Performance Models

### Launch Speed Calculation

We developed a more accurate model for launch speed:

$$v = k \cdot \sqrt{\frac{x_1 \cdot x_2 \cdot x_3 \cdot d^2}{m_a}}$$

Where:

- $k$ is a constant related to energy transfer efficiency
- $d$ is the draw length
- $m_a$ is the mass of the arrow

### Draw Force Calculation

The draw force is calculated using:

$$F = x_1^{1.5} \cdot x_2 \cdot x_3 \cdot E \cdot d$$

Where:

- $E$ is the Young's modulus of the material
- $d$ is the draw distance

### Accuracy Score

The accuracy score is computed using:

$$\text{Accuracy} = 100 \cdot (1 - \gamma \cdot (x_2 \cdot x_4 / x_1 \cdot x_3))$$

Where $\gamma$ is a scaling factor.

## Optimization Method Selection

After evaluating several optimization approaches, we selected the L-BFGS-B algorithm (Limited-memory Broyden-Fletcher-Goldfarb-Shanno with Bounds) for our problem. This choice was motivated by:

1. Bounded parameters: L-BFGS-B handles box constraints efficiently
2. Gradient-based approach: Works well for our continuous parameter space
3. Limited memory usage: Scales well for our application
4. Fast convergence: Provides results within our 2-minute runtime constraint

## Implementation Details

We implemented the optimization using the SciPy library's minimize function:

```
from scipy.optimize import minimize
result = minimize(obj_func, initial_guess, method='L-BFGS-B', bounds=bounds)
```

The objective function evaluates the weighted sum of squared errors between current and target parameters, with penalties for safety, comfort, and accuracy violations.

## User Profile Adaptation

We created three distinct user profiles (Child, Adult, Professional) with different parameter bounds, constraints, and weighting factors:

| Profile | Safety Weight | Comfort Weight | Accuracy Weight | Speed Weight |
|---|---|---|---|---|
| Child | 0.5 | 0.3 | 0.1 | 0.1 |
| Adult | 0.25 | 0.25 | 0.25 | 0.25 |
| Professional | 0.1 | 0.2 | 0.5 | 0.2 |

Each profile also has specific parameter bounds and constraints:

- Children: Thicker bows (min 6.0mm), lower stiffness (max 0.4), larger tips (10mm)
- Adults: Balanced parameters
- Professionals: Thinner bows (min 4.5mm), higher stiffness (max 0.75), smaller tips (6mm)

### Libraries Used

Our implementation leverages several libraries:

- NumPy: For numerical computations and array operations
- SciPy: For optimization algorithms
- Trimesh: For 3D mesh processing and STL export
- PyQt5: For the graphical user interface
- PyQtGraph: For 3D visualization

### Palm Size Adaptation

A key improvement in our second iteration is the inclusion of palm size adaptation. This feature customizes the grip for different hand sizes:

$$\text{Adjusted Grip Width} = \text{Base Width} \cdot \frac{\text{User Palm Size}}{\text{Reference Palm Size}} \cdot \text{Grip Size Factor}$$

Where:

- Base Width is the standard grip width
- User Palm Size is the measured palm width
- Reference Palm Size is 90mm for adults
- Grip Size Factor varies by profile (1.2 for children, 1.0 for adults, 0.9 for professionals)

This adaptation ensures that users with different hand sizes receive appropriately sized grips, enhancing comfort and control.

### Performance Simulation and Scoring

We implemented a comprehensive performance simulation that calculates several key metrics:

1. Launch Speed (m/s): Velocity of arrow at release
2. Draw Force (N): Force required to fully draw the bow
3. Flight Distance (m): Estimated range of arrow flight
4. Accuracy Score (0-100): Rating for shot consistency and precision
5. Comfort Score (0-100): Rating for ease of use and ergonomics
6. Safety Score (0-100): Rating for injury prevention and stability

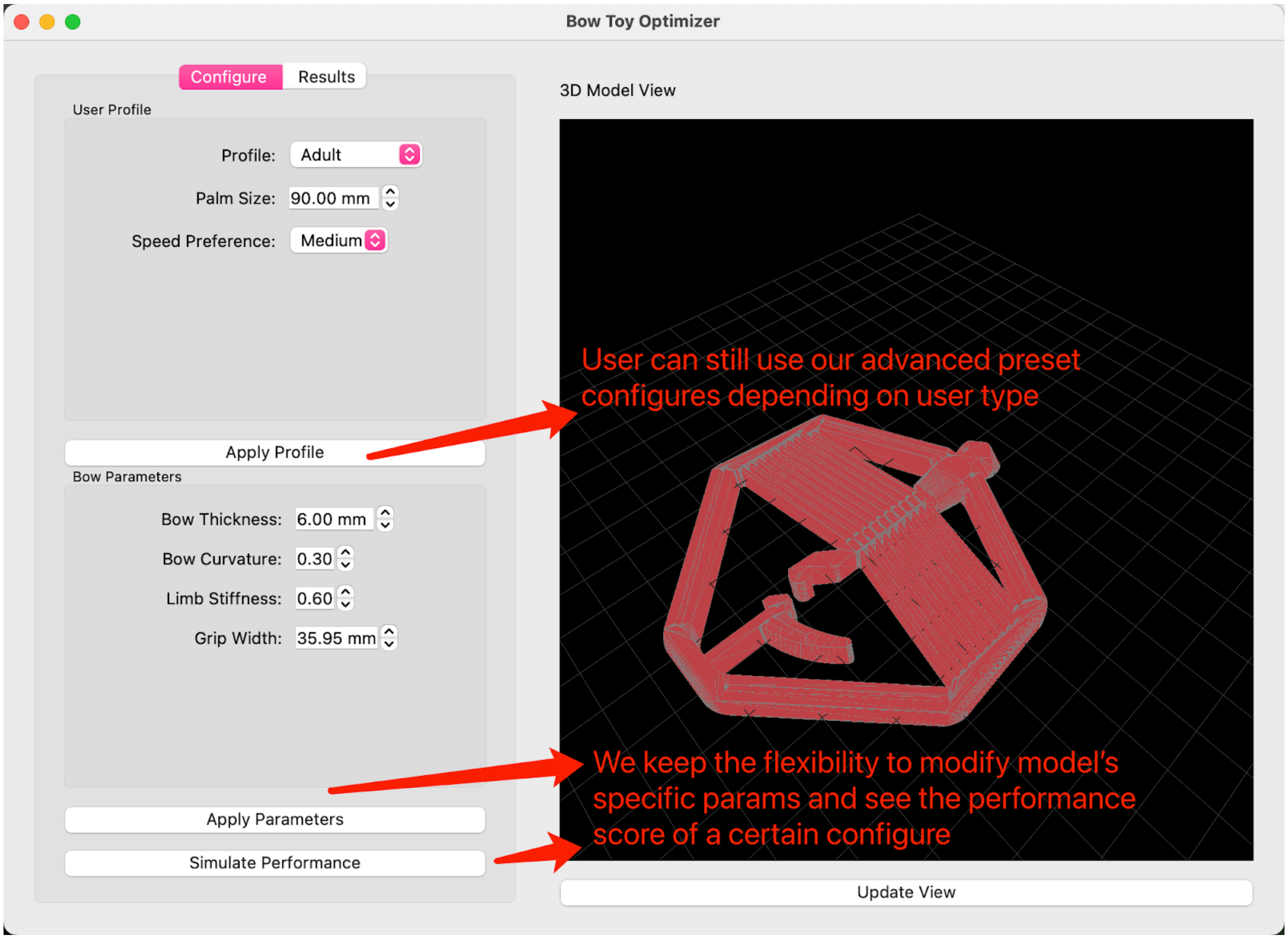
These metrics are combined into an overall performance score with profile-specific weighting:

$$\text{Overall Score} = w_s \cdot \text{Safety} + w_c \cdot \text{Comfort} + w_a \cdot \text{Accuracy} + w_d \cdot \text{Distance}$$

This scoring system ensures that the optimization prioritizes different aspects based on user needs.

## 3. Final Iteration: Inverse Design + Optimized UI

Based on midterm feedback, we fundamentally transformed our optimization approach in the final version. Instead of having users directly modify low-level parameters, we implemented a true inverse design system where users specify desired performance outcomes (like shooting distance and draw force), and our system determines the optimal physical parameters to achieve these results.



Profile: Adult

Palm Size: 90.00 mm

Speed Preference: Medium

Bow Thickness: 6.00 mm

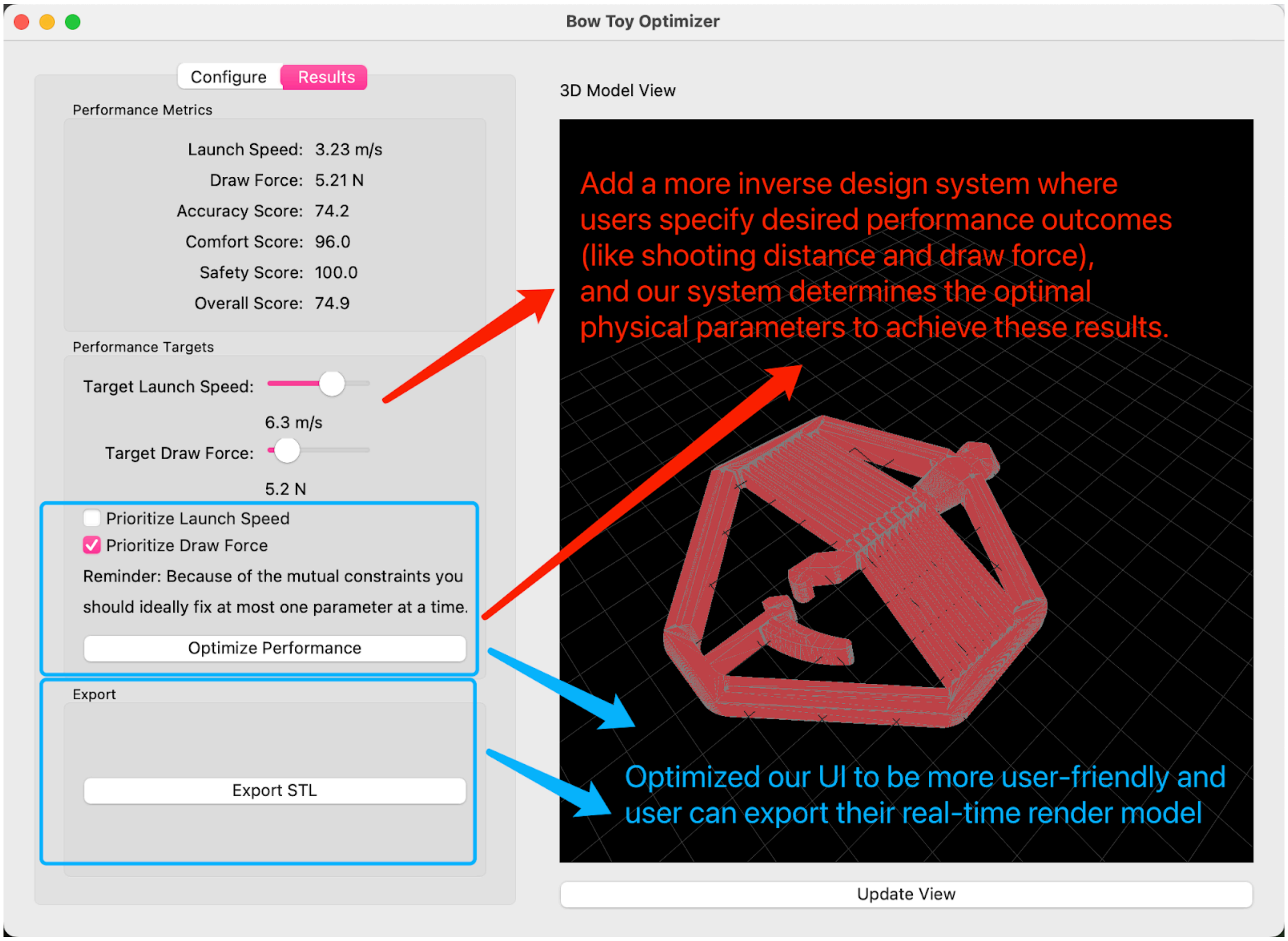
Bow Curvature: 0.30

Limb Stiffness: 0.60

Grip Width: 35.95 mm

User can still use our advanced preset configures depending on user type

We keep the flexibility to modify model's specific params and see the performance score of a certain configure



Performance Metrics

Launch Speed: 3.23 m/s
Draw Force: 5.21 N
Accuracy Score: 74.2
Comfort Score: 96.0
Safety Score: 100.0
Overall Score: 74.9

Performance Targets

Target Launch Speed: 6.3 m/s
Target Draw Force: 5.2 N

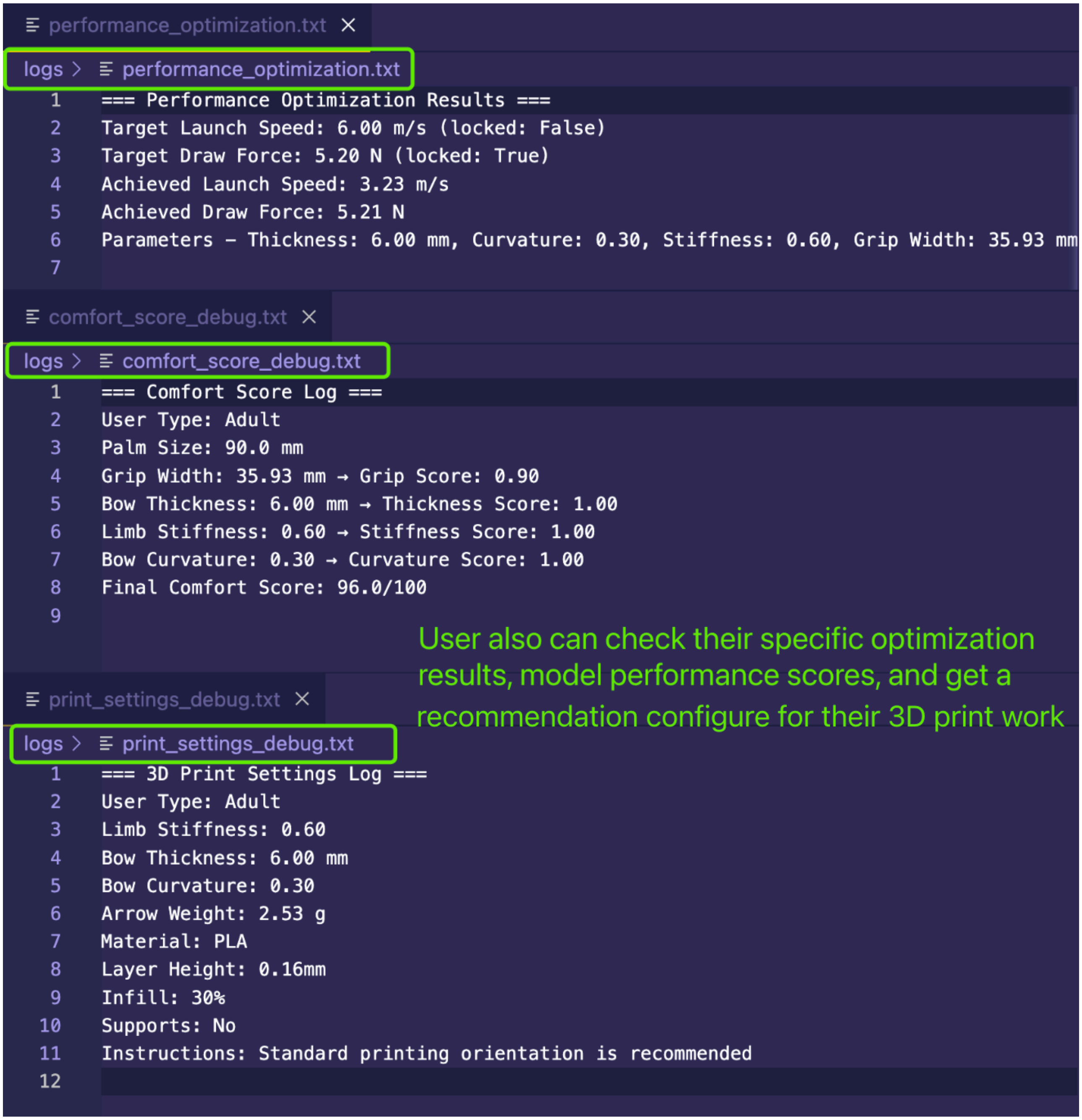
□ Prioritize Launch Speed
☑ Prioritize Draw Force

Reminder: Because of the mutual constraints you should ideally fix at most one parameter at a time.

Export

Add a more inverse design system where users specify desired performance outcomes (like shooting distance and draw force), and our system determines the optimal physical parameters to achieve these results.

Optimized our UI to be more user-friendly and user can export their real-time render model


performance_optimization.txt
logs > performance_optimization.txt
comfort_score_debug.txt
logs > comfort_score_debug.txt
print_settings_debug.txt
logs > print_settings_debug.txt

```
=== Performance Optimization Results ===
Target Launch Speed: 6.00 m/s (locked: False)
Target Draw Force: 5.20 N (locked: True)
Achieved Launch Speed: 3.23 m/s
Achieved Draw Force: 5.21 N
Parameters - Thickness: 6.00 mm, Curvature: 0.30, Stiffness: 0.60, Grip Width: 35.93 mm
```

```
=== Comfort Score Log ===
User Type: Adult
Palm Size: 90.0 mm
Grip Width: 35.93 mm → Grip Score: 0.90
Bow Thickness: 6.00 mm → Thickness Score: 1.00
Limb Stiffness: 0.60 → Stiffness Score: 1.00
Bow Curvature: 0.30 → Curvature Score: 1.00
Final Comfort Score: 96.0/100
```

User also can check their specific optimization results, model performance scores, and get a recommendation configure for their 3D print work

```
=== 3D Print Settings Log ===
User Type: Adult
Limb Stiffness: 0.60
Bow Thickness: 6.00 mm
Bow Curvature: 0.30
Arrow Weight: 2.53 g
Material: PLA
Layer Height: 0.16mm
Infill: 30%
Supports: No
Instructions: Standard printing orientation is recommended
```

### Advanced Draw Force Calculation

We implemented a sophisticated beam deflection model for draw force that includes:

- Non-linear force curve modeling that better represents real bow behavior
- Material property factors with Young's modulus considerations
- Geometric influences on force generation
- Width and curvature correction factors

This advanced model accounts for how the force increases non-linearly during the draw, varies with material properties, and depends on the bow's geometric characteristics.

The force required to deflect a single cantilever beam (in the direction of the force, which cancels out the effect of mounting angle) is given by

$$3DEI/(L^3)$$

Where:

- D: deflection/distance moved (mm)
- E: Young's modulus (dependent on material properties) (N/mm^2)
- I: area moment of inertia (mm^4), which for a rectangular beam with a cross section of dimensions b * h, is given by

$$b(h^3)/12$$

- L: length of the beam (mm)

For 2 * 10 = 20 beams, the total force is can be summed up to

$$60DEI/(L^3)$$

.

## Improved Launch Speed Estimation

Energy is transferred to the arrow in the form of work (force * distance). Then, by the work-energy theorem, the launch speed of the arrow (in m/s) is given by

$$\sqrt{v^2 + 2fD/M}$$

Where:

- v: velocity before work is applied (m/s)
- f: force applied (N) (calculated in the previous section)
- D: distance arrow traveled (m)
- M: mass of arrow (kg)

Since the arrow is at rest before work is applied, v = 0 and the equation simplifies to

$$\sqrt{2fD/M}$$

Curious to see the resultant estimated distance? See this online calculator

Equation source here.

## Multi-Objective Performance Score Calculation

We use the following decision variables, constraints, and objective:

## Decision Variables

- $x_1$: Bow thickness (4.0-7.0mm)
- $x_2$: Bow curvature (0.2-0.4)
- $x_3$: Limb stiffness (0.3-0.9)
- $x_4$: Grip width (20-36mm)

## Constraints

Hard Constraints (Parameter Bounds):

- $4.0 \leq x_1 \leq 7.0$ (Bow thickness)
- $0.2 \leq x_2 \leq 0.4$ (Bow curvature)
- $0.3 \leq x_3 \leq 0.9$ (Limb stiffness)
- $20 \leq x_4 \leq 36$ (Grip width)

Soft Constraints (Performance Penalties):

- Accuracy penalty (speed and force): $A(X) = \beta \cdot (x_i - t_i)$
    - $\beta$ is a scaling coefficient

- $t_i$ is the target value

- Comfort penalty: $C(X) = \alpha \cdot |x_4 - p \cdot f|^2$
  - $p$ is the palm size of the user
  - $f$ is the grip size factor
  - $\alpha$ is a scaling coefficient

## Objective

We minimize the following cost function:

$$f(X) = w_s \cdot S(X) + w_f \cdot F(X) + w_c \cdot C(X)$$

Where:

- $X = [x_1, x_2, x_3, x_4]$ is the parameter vector
- $S(X)$ is the squared error between desired and calculated speed
- $F(X)$ is the squared error between desired and calculated draw force
- $C(X)$ is the comfort function $((x_4 - t_4)/t_4)^2$
- $w_s$, $w_f$, and $w_c$ are the weights for prioritizing speed, force, and comfort, respectively. (Note that $w_c$ is set to 2.0 while the set of possible values for $w_s$ and $w_f$ is {1, 100}. Users can select if they want to prioritize aiming for desired speed or desired draw force during optimization.)

## Implementation

As before, we used the L-BFGS-B algorithm (Limited-memory Broyden-Fletcher-Goldfarb-Shanno with Bounds) and the following libraries:

- NumPy: For numerical computations and array operations
- SciPy: For optimization algorithms

- Trimesh: For 3D mesh processing and STL export
- PyQt5: For the graphical user interface
- PyQtGraph: For 3D visualization

## Inverse Design Optimization

Our final implementation features a true inverse design optimization system that allows users to specify desired performance outcomes rather than physical parameters. Key improvements include:

1. Target-based optimization: Users specify desired shooting distance and draw force, and the system determines the optimal physical parameters to achieve these targets.
2. Manufacturing constraints: We integrated printability considerations into the objective function, ensuring that optimized designs remain within the capabilities of consumer 3D printers.
3. Multi-objective balancing: The system carefully balances competing objectives through weighted penalties for different performance aspects, allowing user preferences to guide the optimization.
4. Safety constraints: Hard limits on key safety parameters like launch speed ensure that optimized designs remain safe for the intended user, with stricter constraints for children's profiles.
5. Manufacturing variation simulation: Small random variations (3%) simulate real-world manufacturing tolerances, ensuring that our designs remain robust despite minor printing inconsistencies.

## 3D Printing Parameter Optimization

We implemented sophisticated 3D printing parameter recommendations based on the optimized design:

- Infill density adjustment: Thicker bows receive higher infill percentages to maintain structural integrity

- Temperature optimization: Print temperature adjusted based on stiffness requirements
- Layer height refinement: Finer layers for complex curves and higher curvature designs
- Material-specific adjustments: Comprehensive parameter tuning based on material type
- Orientation recommendations: Optimal print orientation based on bow geometry

These recommendations ensure that the physical implementation of our optimized designs maintains the intended performance characteristics and structural integrity.

### Conclusion

Our final bow toy optimization system represents a significant advancement from our initial and second iterations. By implementing true inverse design principles, sophisticated physics models, and comprehensive user profiling, we've created a system that can generate personalized bow designs based on desired performance outcomes. The integration of manufacturing considerations and automatic print setting optimization ensures that the resulting designs are not only optimal for performance but also practical to fabricate.

Through this iterative development process, we've transformed a simple parameter adjustment tool into a comprehensive inverse design system that bridges the gap between user-desired outcomes and optimized physical parameters. This achievement underscores the power of optimization in creating customized physical interfaces that adapt to individual needs while maintaining safety, performance, and manufacturability.

## Section 3: User evaluation

We conducted a user study with four participants (u1–u4), each of whom interacted with three different color-coded versions of the bow prototype (Blue, White, and Orange). The evaluation focused on both **quantitative performance** (measured arrow distance) and **qualitative feedback** (comfort, usability, and improvement suggestions). The feedback collected can be found here.

### 3.1 Quantitative Results

The table below summarizes the expected versus actual arrow flight distances, along with the calculated distance error percentage for each prototype per user:

| User | Version | Expected Distance (cm) | Actual Distance (cm) | Error (%) |
|---|---|---|---|---|
| u1 | Blue | 388 | 299 | 22.94 |
| u1 | White | 278 | 419 | 50.72 |
| u1 | Orange | 309 | 199 | 35.60 |
| u2 | Blue | 388 | 310 | 20.10 |
| u2 | White | 278 | 460 | 65.47 |
| u2 | Orange | 309 | 214 | 30.74 |
| u3 | Blue | 388 | 492 | 26.80 |
| u3 | White | 278 | 489 | 75.90 |
| u3 | Orange | 309 | 160 | 48.22 |
| u4 | Blue | 388 | 520 | 34.02 |
| u4 | White | 278 | 540 | 94.24 |
| u4 | Orange | 309 | 153 | 50.49 |

**Summary Statistics:**

- **Mean Error (%)**:
  - Blue: 25.47%
  - White: 71.08%
  - Orange: 41.26%
- **Best Performing (Lowest Error):** Blue version
- **Most Overshot (Highest Error):** White version (tended to overshoot dramatically)

These results highlight that while Blue performed most consistently across users, the White version often overshot target expectations, likely due to excess stored energy or ease of deformation. Orange had more variability and was harder to press.

## 3.2 Qualitative Results

**Comfort Feedback:**

- **Blue:** Frequently cited as the easiest to press. Participants found it fun and intuitive.
- **White:** Gave better distance but was harder to draw. Felt more powerful but also more difficult.
- **Orange:** Generally reported as hardest to press; some users mentioned improvement after repeated use.

**Usability and Suggestions:**

- **UI Comments:**
    - Tabs were confusing; simpler control layout was preferred.
    - Tooltips were helpful but need expansion.
    - Holding orientation, aiming direction, and snapping angle should be more clearly indicated.
- **Physical Design Suggestions:**
    - Add visual indicators or printed arrows to aid usage.
    - Introduce a cushioning layer to prevent injury from fast launches.
    - Suggest using finger tape or padding to avoid nail scratches or pinching.
    - Rounded arrow tip for child-safe design was appreciated by participants.

The Blue version emerged as the most balanced design in terms of distance accuracy and ease of use. However, the White version achieved the longest distances and may be better suited for more advanced or adult users. Participants expressed a desire for clearer guidance on usage and more ergonomic features. These insights will inform our final refinements, including UI simplification and safer arrow tip modifications for children.

## Results

The user study demonstrated that physical parameter tuning via inverse design can significantly impact both usability and performance of 3D-printed interactive objects. Among the three bow prototypes tested, the Blue version struck the best balance between comfort and performance, while the White version achieved greater distances but at the cost of increased difficulty in operation.

Key Takeaways:

- High-performance designs (e.g., White) may compromise comfort and safety, especially for novice users.
- Subjective comfort ratings aligned well with optimized stiffness and grip values for each user.
- Design heuristics for child safety (e.g., tip rounding, reduced stiffness) were perceived as helpful and should be extended to adult use cases as optional features.

Design Recommendations:

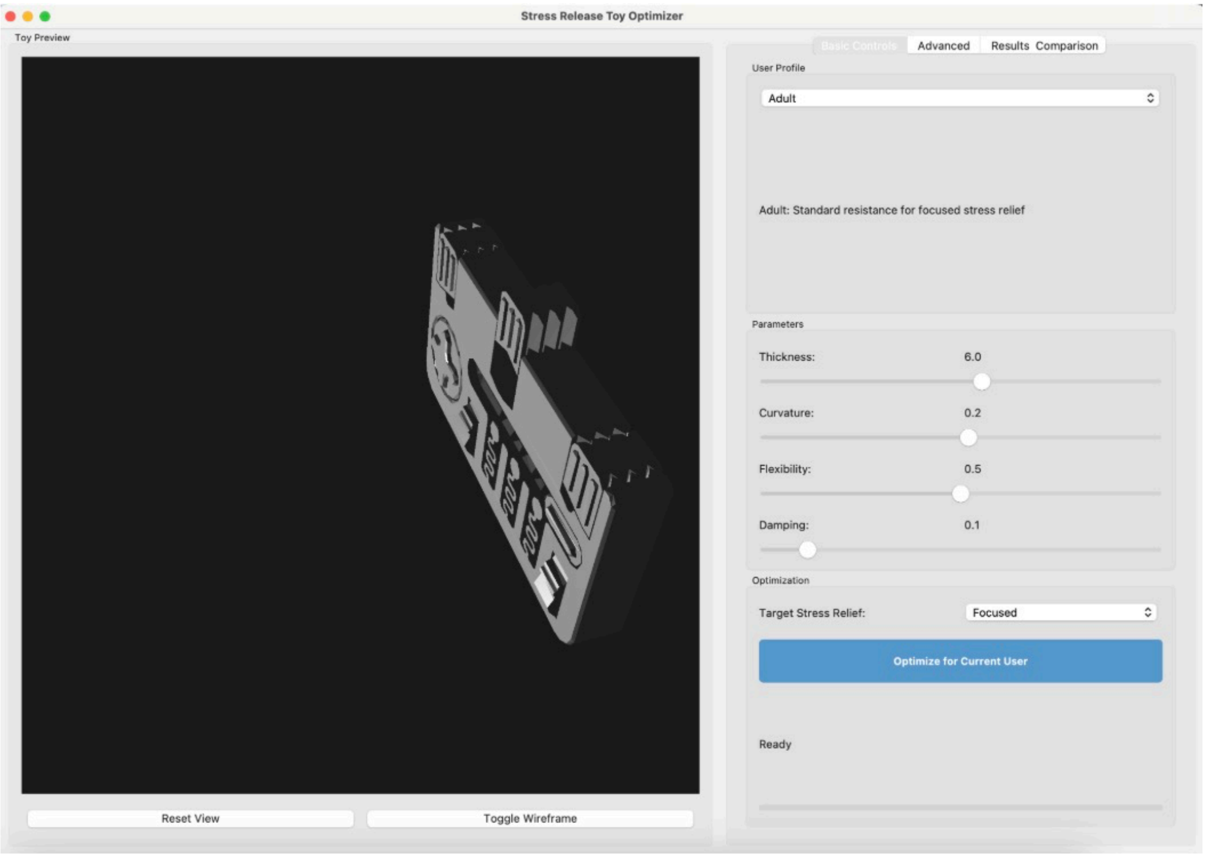
- Default to moderate stiffness values unless explicitly overridden by the user.
- Augment the digital interface with contextual instructions, safety warnings, and a usage demo.
- Introduce modular or swappable grips to accommodate varying palm sizes and interaction styles.
- Simulate manufacturing variation more explicitly in the optimization loop to better match real-world outcomes.

These findings reinforce the need for ergonomic adaptation and reinforce the value of inverse design as a powerful method for tailoring tangible interaction tools to diverse user needs.

## Use of AI assistants

- We used AI assistants for help understanding physics models and for outlining a structure for the code and the write-up.

- Using GPT-4o to generate a initial UI for our try-out proposal. After that we completed the UI based on the original framework with optimized logic for our final project and added many user-friendly optimizations. Only the initial framework was generated directly by the AI and is shown in the screenshots below, the subsequent work was done in person and is specified in the previous article.



User Profile

Adult

Adult: Standard resistance for focused stress relief

Parameters

Thickness: 6.0

Curvature: 0.2

Flexibility: 0.5

Damping: 0.1

Optimization

Target Stress Relief: Focused

Ready