Todo:

Use Matter.Engine.update for user to select t =

Before “play,” you’re just drawing the items to canvas, but when press “play,” then you add all of the objects to the engine/world and apply everything (forces, velocity, etc)… I think

use push and pop to contain changes to an object and save the previous state (undo option for user) — you would save this state by doing like this.state or something, saving the state to the object

Use composites and connecting bodies for momentum! This might actually be genius

Javascript

MVP on Github Pages

Matter.js (Math, collisions, etc), p5.js (visuals/drawing)

Plugins to look at: matter-attractors by liabru

For springs: matter-springs by momentumworks

Feel like u can do this urself but check: matter-collision-events by dxu

Cool examples (like airFriction, constraints, events, gravity, etc): <https://github.com/liabru/matter-js/tree/master/examples>

<http://palmerpaul.com/p5-matter/>

On looks:

Some aesthetic sky background with sun, smooth grass(fricCoeff = 0) & ground (fricCoeff > 0);

Worry about trivial things like color scheme later XD Kipp is not an art teacher

Logistics:

Max force values able to be inputted; max objects in the world; etc

Difficult things you might want but don’t know how to do:

Different moments of inertias depending on object (more specifically that whacky double ring cylinder circle inner outer radius nonsense that was on the test

rolling down hills to see potential/kinetic energy transfers

Different screen sizes? Can this be a % responsive width/height?

Objects: Box (Cube; Rectangle); Car (simple), Disk (different sizes, radii, lengths, and hollow vs dense), Sphere, pendulum, 2 masses w light cord on frictionless pulley, pointed object (to place, let an object collide against and show collision against that one sharp point)

* Max 2 objects allowed (for collisions)
* Static/unmoving objects (For collisions) versus dynamic objects (move/react)

Free body diagram: Always displaying forces that will act on the object on “Play”; FBD is ON the object in the screen; how to display Fforce on FBD?

On play: Constant forces stay on there (color coded), instant forces disappear, but velocity vectors are shown (magnitude, direction)

Forces: Can select where on object to apply force (exactly what point, direction, and magnitude)

Torque: torque = rFsinθ so forces up top determine the torque being applied

Gravity: Can toggle on and off as a force; can change what g constant is equal to

Friction: Can adjust friction coefficient of surface; checkbox for static friction or not, and what that static friction coefficient should be (used in coef \* fN)

Air resistance: Checkbox to enable it (to program it, prob just slow down the velocity of an object based on what the air resistance is)

Surface: Smooth or friction; flat, incline (adjustable angle), downward circular hill, flat cliff with edge to fall off of

General: x-axis and y-axis labeled for measurements

Strings and tension forces need to be available as well!!! (obviously pendulum strings will exist, but I’m referring to like two blocks connected by a string, but the heavier one is pulled down by gravity and hanging on a pulley, etc etc

**Dashboard:**

Options: Trace object motion (highlights motion, ex for projectile motion) Pause, play, reset (to customizations b4 pressing play), clear (empties whole board/window)

On play: can edit forces incrementally or enter a force to change to at any time (this can help show static friction for example); can drag objects manually around (will prob have to base force on mouse speed?)

Values to display on screen/dashboard (can toggle on/off these values from dashboard): ΣF (magnitude, direction); velocity (dynamically changing), angVelocity, momentum, work, etc

After world functions:

Note collisions = perfectly elastic

Contact form for feature requests, bugs, etc

Some type of interactive tutorial

Way for Kipp to know you did it? (name/ID submit? etc)

Interactive problems (Ask Kipp on this — personally believe it’s a great idea + makes his lesson planning easier…):

What force is necessary to get object from pt A to pt B (person does math on paper, then puts in force # into program)? Visual dotted object at point b to show destination kinda thing (and can use the detector or whatever in p5 or matterjs to indicate success)

What torque is necessary to get circle object from pt a to pt b? (so person would need to know where to put the force vector on the screen — perpendicular and some magnitude).

What coefficient of friction necessary to stop object exactly 5 m away from bottom of ramp?

Extra features:

Dynamic graph drawing of velocity and position

Formula sheet (organized by units)

? help icons to explain physics concepts perhaps (ex on inclines w a square, ability to show component forces, how to calculate, etc in ? bubble, things like that)

Create your own objects (PhysicsEditor by Andreas Löw)

Create preset dashboards for forces, rotation, static equilibrium, etc

Add this as cool demonstration of forces (<https://www.youtube.com/watch?v=jsYwFizhncE>) (ask user to input how many digits of pi wants to approximate)

Use Three.js to make a 3d version (basically, option to select 2D or 3D, and to easily transfer objects, you’ll need to figure out how to rnder 3d based on the 2d objects matter.js already creates for u)

Websocket for ppl to interact on same pg

NOTES:

if ur gonna be a greedy capitalist… ads only and a donation button (which will literally get u 1 cent). THIS IS FREE!!!!! 無料!!!!