Extending Modeling Instruction with Computational Modeling: A Pilot Study

John M. Aiken, Marcos D. Caballero, Scott S. Douglas, John B. Burk,

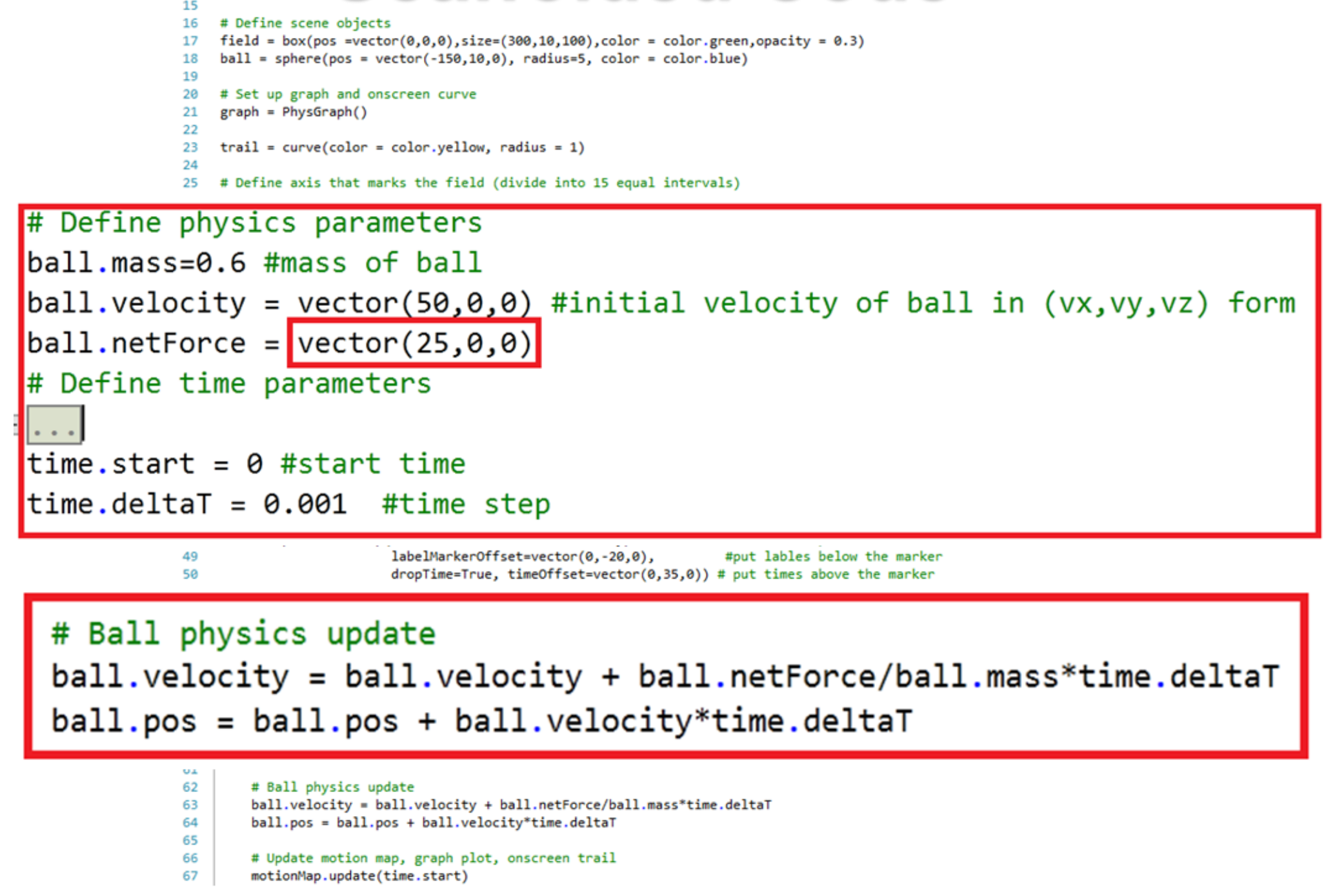
Erin M. Scanlon, Brian D. Thoms, Michael F. Schatz

School of Physics, Georgia Institute of Technology, Atlanta, GA 30332

Abstract

Recently, the National Research Council's framework for next generation science standards highlighted "computational thinking" as one of its "fundamental practices". Students taking a physics course that employed the Arizona State University's Modeling Instruction curriculum were taught to construct computational models of physical systems. Student computational thinking was assessed using a proctored programming assignment, written essay, and a series of think-aloud interviews, where the students produced and discussed a computational model of a baseball in motion via a high-level programming environment (VPython). Roughly a third of the students in the study were successful in completing the programming assignment. Student success on this assessment was tied to how students synthesized their knowledge of physics and computation. On the essay and interview assessments, students displayed unique views of the relationship between force and motion; those who spoke of this relationship in causal (rather than observational) terms tended to have more success in the programming exercise.

Scaffolded Code



```
# Define scene objects
field = box(pos =vector(0,0,0),size=(300,10,100),color = color.green,opacity = 0.3)
ball = sphere(pos = vector(-150,10,0), radius=5, color = color.blue)

# Set up graph and onscreen curve
graph = PhysGraph()

trail = curve(color = color.yellow, radius = 1)

# Define axis that marks the field (divide into 15 equal intervals)
```

```
# Define physics parameters
ball.mass=0.6 #mass of ball
ball.velocity = vector(50,0,0) #initial velocity of ball in (vx,vy,vz) form
ball.netForce = vector(25,0,0)
# Define time parameters
...
time.start = 0 #start time
time.deltaT = 0.001  #time step
```

```
                 labelMarkerOffset=vector(0,-20,0),        #put lables below the marker
                 dropTime=True, timeOffset=vector(0,35,0)) # put times above the marker
```

```
# Ball physics update
ball.velocity = ball.velocity + ball.netForce/ball.mass*time.deltaT
ball.pos = ball.pos + ball.velocity*time.deltaT
```

```
    # Ball physics update
    ball.velocity = ball.velocity + ball.netForce/ball.mass*time.deltaT
    ball.pos = ball.pos + ball.velocity*time.deltaT

    # Update motion map, graph plot, onscreen trail
    motionMap.update(time.start)
```

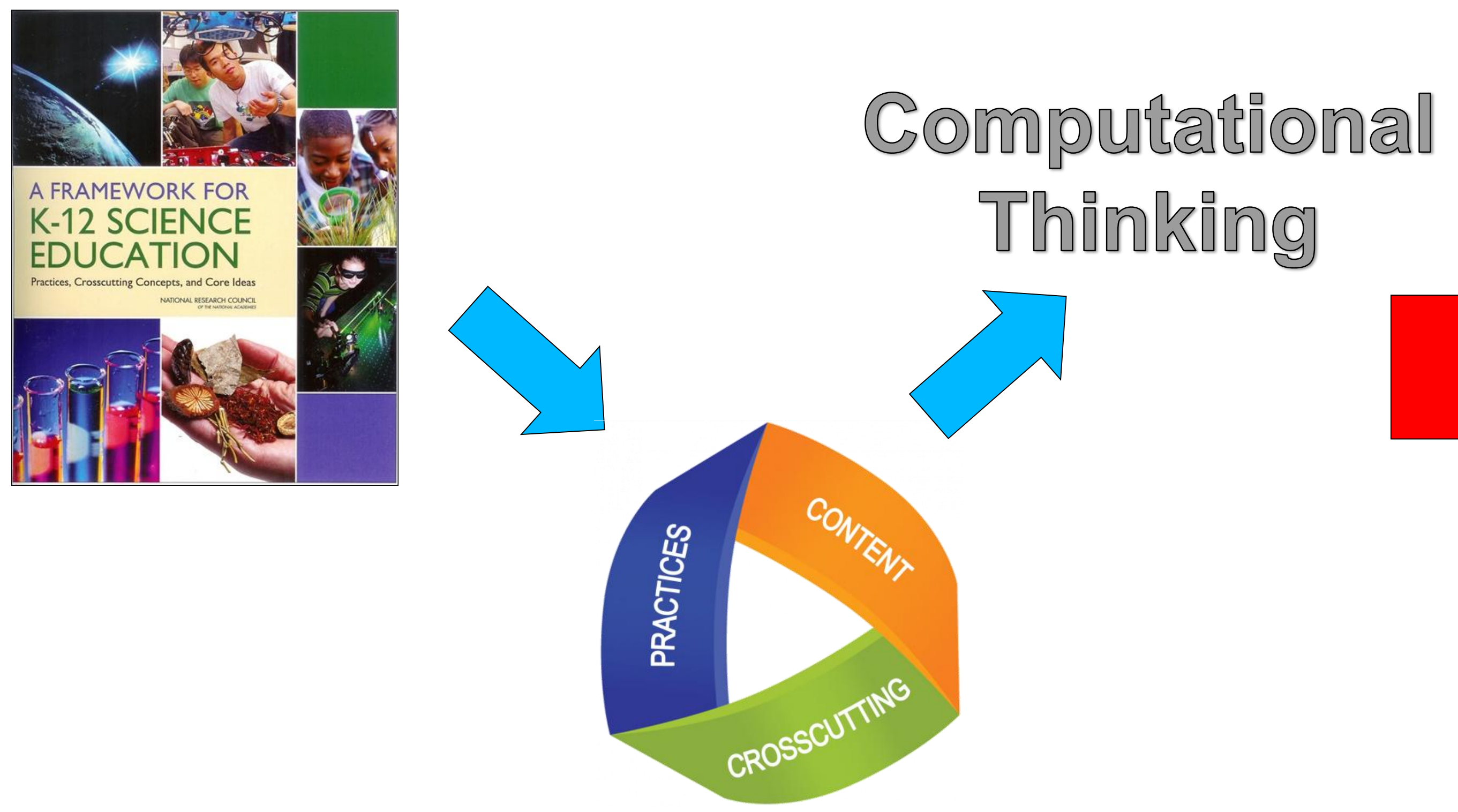
Summary and Conclusion

- 9th grade students were taught to use computational modeling and computational thinking during their forces instruction in a Modeling Instruction physics classroom.
- About one third of the students were completely successful in completing a computation assignment.
- Student success on the proctored assignment was closely tied to how students synthesize knowledge of physics (force and motion) and computation (iterative processes).
- Students who described iterative processes but had not yet connected the concepts of force and motion were unable to create precise computational models.

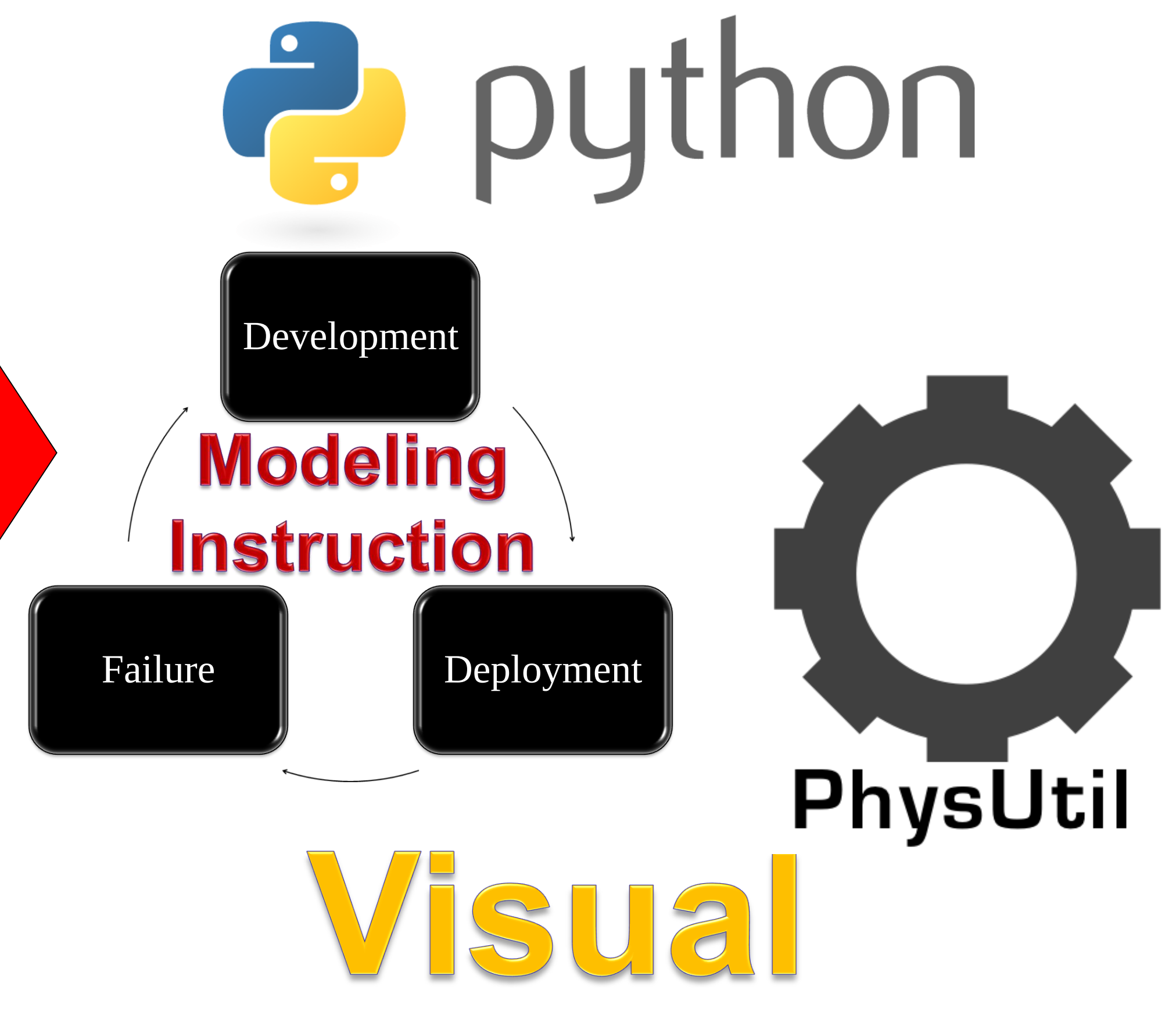
References

[1] H. Quinn, et al., A Framework for K-12 Science Education. National Academies Press, 2012.
[2] VPython. http://vpython.org
[3] M.D. Caballero, et. al., arXiv preprint:1207.0844v1.
[4] PhysUtil. Version 1.22. Georgia Tech PER, 2012.
[5] J.M. Wing, "Computational Thinking", CACM, 49, 3, 2006.
[6] M.D. Caballero, et al., Phys. Rev. ST PER. (2012).

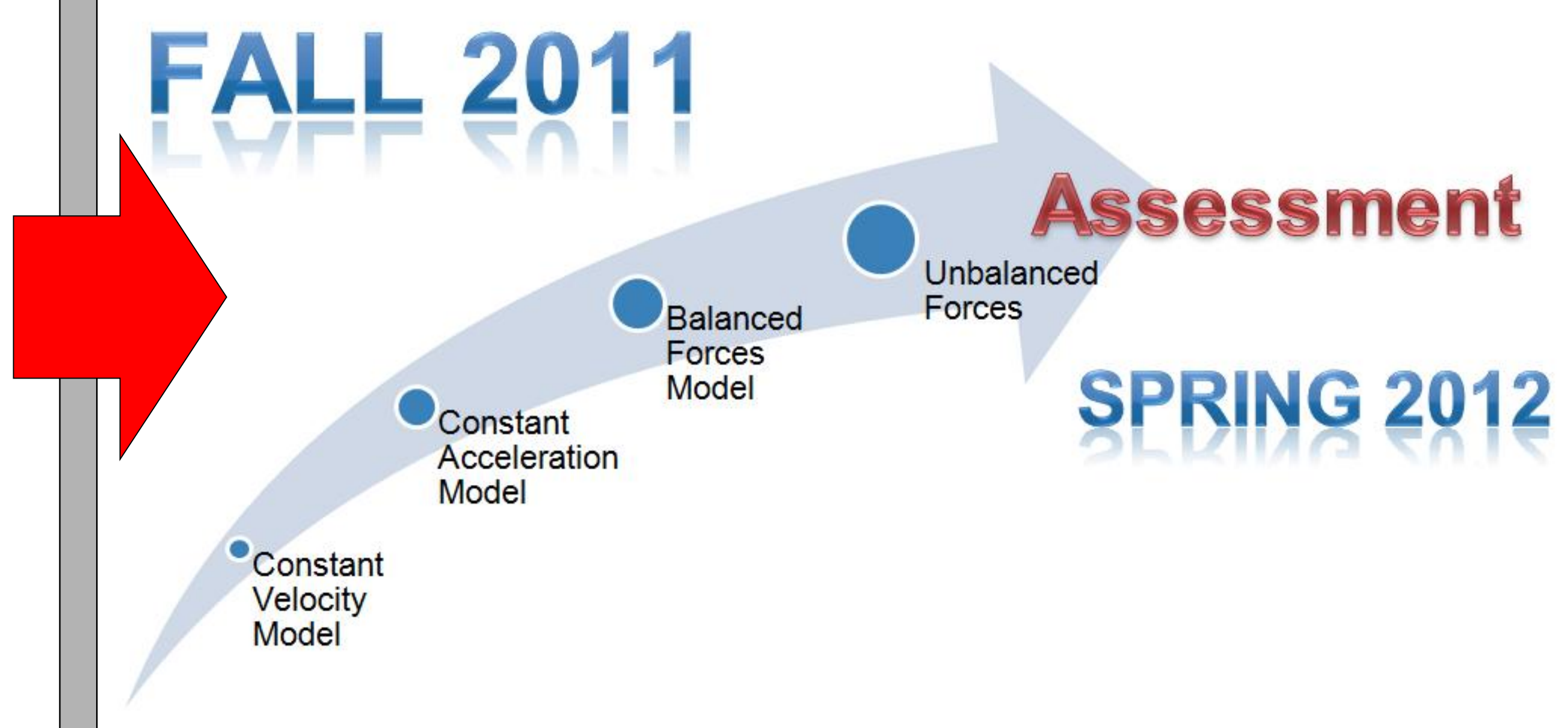
Motivation



How?

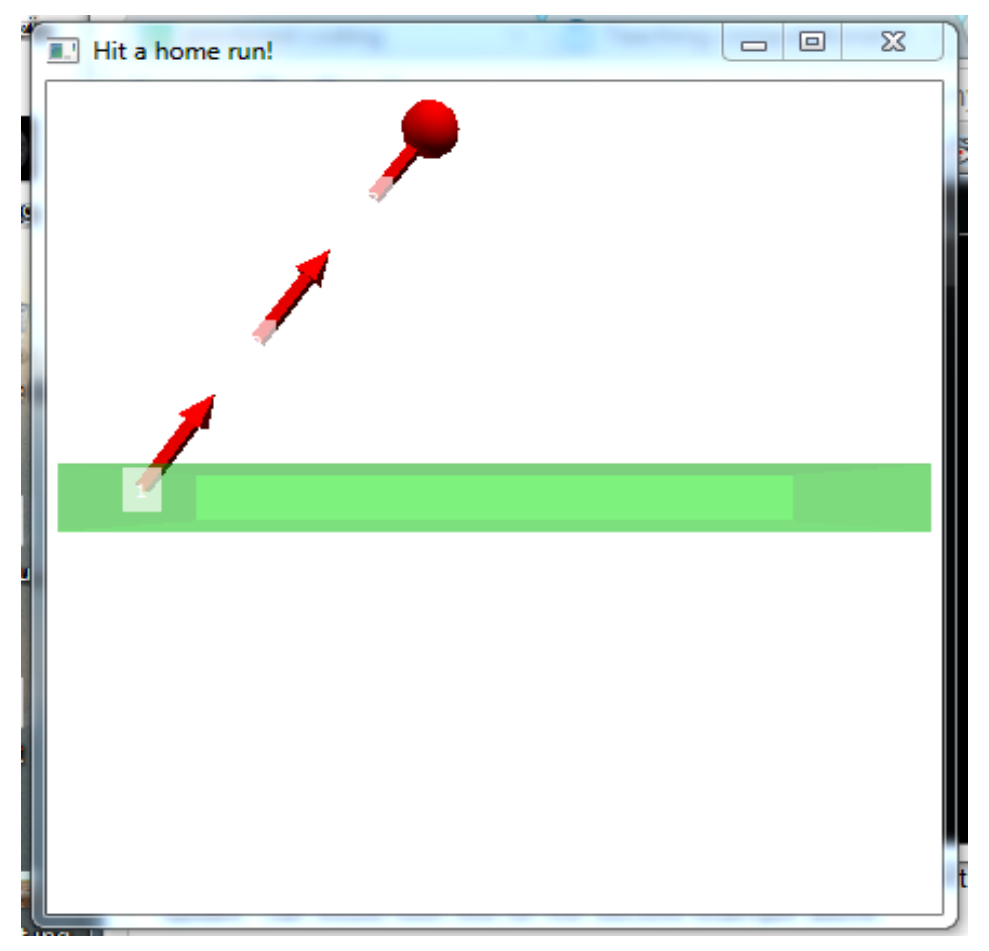


Timeline



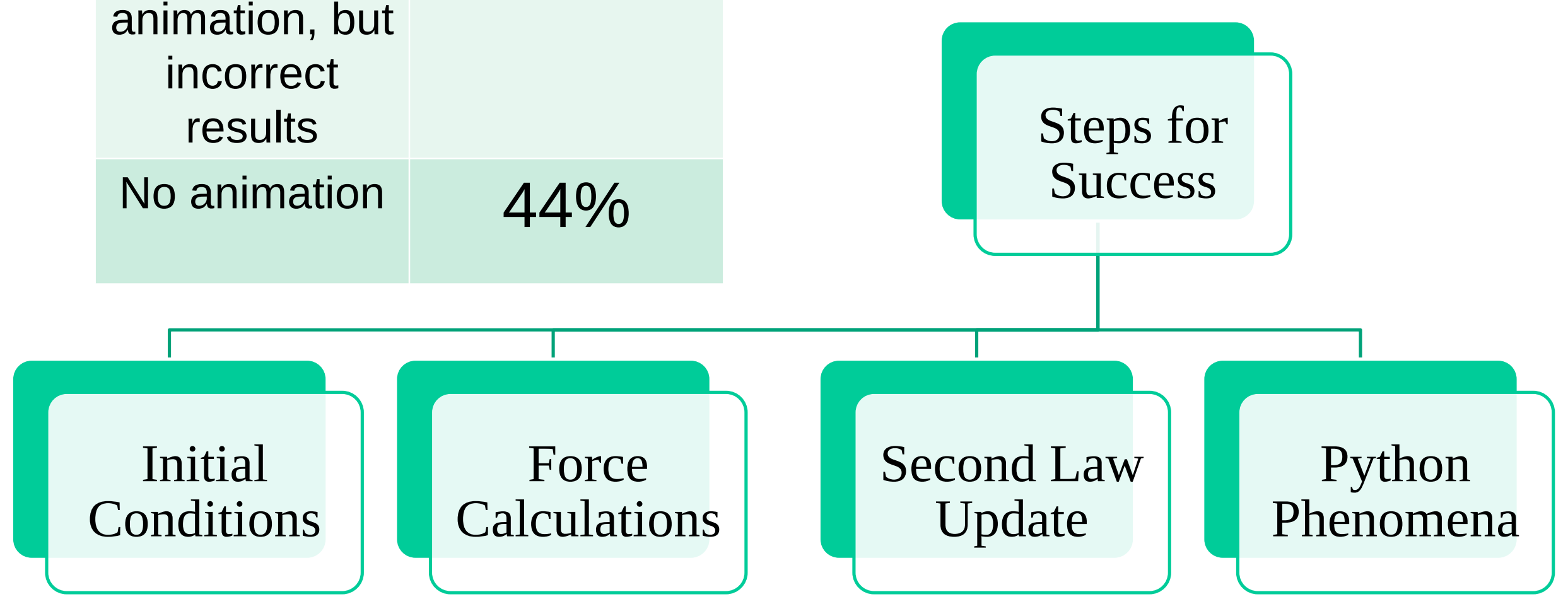
Assessments of Computational Thinking

Proctored Assignment



- Model the motion of a baseball immediately after it was hit.
- Students were given scaffolded code via an online homework tool that they were tasked to "fill in the physics"
- Students were given a **practice case** on Earth
- They were then given a **grading case** on the moon

Success rate	Percent
Correct results and animation	31%
Produced animation, but incorrect results	25%
No animation	44%



Almost one third of the students constructed a completely correct program. Another quarter avoided syntactic errors that would have produced no animation, but produced some number of errors either writing the integration algorithm alone (25%) or writing the integration algorithm and assigning initial conditions (75%). The remaining 44% either had small syntactic errors (36%) or had numerous physics and computational errors (64%).

Essay Question

- Investigated whether students success was predicated on reproducing an algorithm or did successful students make deeper connections between physics and the computational algorithm.
- Students broke down into three sometimes overlapping views. Force-causal, kinematic-observational, and iterative.
- Students who were force-causal were exclusively iterative.

> Student presenting a Force-causal view
>
> "The loop is constantly changing the velocity of the ball while the Fnet [net force] stays constant. It makes the ball fall faster with every loop that runs".

> Student presenting a Kinematic-observational view
>
> "The loop's purpose is to use the acceleration of the ball to affect the ball's velocity and position. The loop is run every .01 seconds (deltat). It re-updates the velocity and position of the ball at that interval."

	Force-causal & Iterative-local view	Kinematic observational & Iterative-local view	Primarily iterative-local view	No classification
Correct results and animation	20.6	3.4	3.4	6.9
Produced an animation, but incorrect results	10.3	3.4	3.4	6.9
No animation	6.9	6.9	6.9	13.8

Think-aloud Interview

- 5 students were given the original scaffolded code from proctored assignment on paper and asked to fill in the missing code in a think-aloud environment.
- Students were asked questions about how they define a force, and how forces, motion, and the integration loop were related.
- 3 students presented a force-causal and iterative-local views on the essay question.
- 1 student had previously presented a kinematic-observational view but expressed a force-causal and iterative view during the interview.
- 1 student presented a primarily iterative-local view on the essay and during the interview.
- Students who presented force-causal and iterative views were able to explain their programs more effectively both programmatically and physically.

> Force-causal, Iterative-local view
>
> "To predict the velocity you would have to do baseball.v = initial velocity of the baseball plus gravity times time. *That would give me the new velocity after* [the execution of] *every single loop. And then you need to update the position based* on the loop."

> Incorrect computational model
>
> "*Force generally [is] acquired through motion.* There's always force acting on an object."