Physics World – Part III

An experimental 2-dimensional, physics-heavy, sandbox-game-thing.

# Submission Guidelines:

In GitHub, create a branch for your project named “Part3\_Completed”.

Due at the start of our next class meeting

# Base Requirements (83)

* In GitHub, create a branch for your project named “Part3\_Completed”.
* Place an annotated version of this rubric in the root directory of your Unity project. Save it in Word .docx format. When I download your Part3\_Completed branch, I should find the rubric.
* Build a fully working SquishyBox, and place it in your game scene:
  + It should have four physics-based game objects as the corners, and a SpriteShape with a SpriteShapeVertexMatcher script.
  + Explain: In your scene, where is your SquishyBox?

The squishy box I made is to make it like a jelly, So in my game, it is on a base which on plane surface in space, it will have Meteorite from the universe which will attack the base. So the Squishy box is like a giant jelly which can catch the Meteorite smoothly. I call it a jelly self defense system.

* Build simple thing that makes use of physics, springs, and FrontBackFriction. If you like, you could make a blade of grass. If you would like to build something else, that’s fine too.
  + Explain: What did you build? What does it do?

I try to build an alien octopus. Because octopus is soft, so when player pass by or jump on them, the springs joint will works like the reflection of octopus’s skin to player which create a soft feeling.

# Stretch Goals:

* (+3) Prepare for, and participate in next Wednesday’s Show ‘n Tell.
  + Before 5PM PST next Wednesday, send me an email with one image from your project. You can send still image of a game scene, an image of a piece of code, or an animated gif showing moving things. Your choice. You pick what you want to share.
  + Be prepared to say 2-3 sentences about your image in our next class meeting.
  + Say 2-3 sentences about your image in our next class meeting.
* (+1 to +5) Add detailed comments to the SpriteShapeVertexMatcher script, explaining how it works:
  + Use medium detail. If I were to hold on to your code and show it to Game Dev 1 students next spring, those people should be able to read your code and understand the ideas behind the code.
  + If you chose to do this, write ‘yes’ here:
* (+1 to +5) Add detailed comments to the FrontBackFriction script, explaining how it works:
  + Use medium detail. If I were to hold on to your code and show it to Game Dev 1 students next spring, those people should be able to read your code and understand the ideas behind the code.
  + If you chose to do this, write ‘yes’ here:
* (+8 to + 20) Build another solid, spring-based object, using the same ideas as in the Squishy Box. You could make a tree, a bridge, a fancy box, or something else. Whatever you like. Whatever you think would fit well with the premise of your project.
  + The object should do physics-based stretch and squash.
  + The object should be rendered with a SpriteShape and a SpriteShapeVertexMatcher.
  + Explain: What did you make?
  + Explain: When and how does it stretch and squash?
  + Explain: Where is this thing in your scene?
* (+8 to +20) Build another physics-based object that uses FrontBackFriction. Something at least a little different from whatever you built for the Base Requirements.
  + The object should do physics-based stretch and squash ... when things move past / through it.
  + Explain: What did you make?
  + Explain: When and how does it stretch and squash?
  + Explain: Where is this thing in your scene?
* (+8 to +20) Create a new PatternMaker component that takes a SpriteShape Component, and procedurally sets the vertices of that SpriteShape to create some sort of pattern.
  + Explain: What pattern does this script make?
  + Explain: Where is this thing in your scene?
* (+1 to +20) Other. Something related to this week’s topics: springs, spring networks, sprite shapes, controlling sprite shapes via code, front/back friction, etc.
  + Explain: What is your nifty thing?
  + Explain: Where have you used this thing in your project?