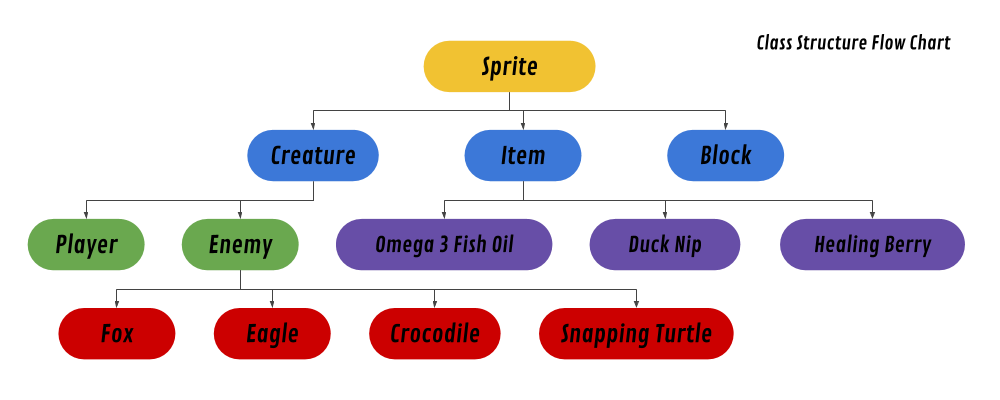
**Let’s Get This Bread**

A duck based platformer game.

The story is…

A duck is on a quest to find his golden feather. He was born with exactly one - metallic and shiny as the sun - but alas, it was stolen by a big bad goose. After minutes of training, he ventures on a great quest to get it back! And stops to eat some bread along the way…

**Class Structure**



**Sprite:** a wrapper for all objects in the game space

* **Variables**
  + X Pos and Y Pos (int)
  + Width and Height (int)
  + Image
  + Visible (boolean)
  + Solid (boolean) - whether or not a sprite can be passed through by another solid sprite
* **Instance Methods**
  + Load Image - loads an image from a file and adjusts the width and height of the sprite accordingly
  + Set Visible - hides/shows the image and toggles the value of visible
  + Get Bounds - returns the rectangle space of the image for collisions
* **Class Methods**
  + Collides - returns true if two sprites are overlapping

**Creature:** all moving, living creatures extend this class

* **Variables**
  + Standing, Walking, and Fighting Animations (Images)
  + Walk Speed (int)
  + Health (int)
  + Max Health (int)
  + Strength (int)
  + Accuracy (int)
* **Instance Methods**
  + Walk: speed, direction, time - play a walking animation and move in the desired direction
  + Take Damage - subtracts health based on a damage amount, if health drops to/below zero, run the kill method
  + Heal - increases a player’s health if the max has not been reached
  + Kill - removes the creature from the game space
  + Attack: victim creature - uses a damage and accuracy formula to attack another creature, which will then call take damage for said victim

**Enemy:** can move around attack attack player

* **Variables**
  + Loot - item array for items to drop on death
* **Instance Methods**
  + Move - will call the walk method in super based on algorithms

**Player:** Mr. Ducky himself

* **Variables**
  + Items (item array)
  + Bread (int)
* **Instance Methods**
  + Move - will call the walk method in super based on keyboard input

**Item:** an item to be used by the player

* No special function, exists for the purpose of a non-specific item array

**Omega 3 Fish Oil:** temporarily increases the player’s walk speed

* **Static Variables**
  + Boost (int)
* **Instance Methods**
  + Use - will use the item to increase the player’s walk speed

**Healing Berry:** regains the player’s lost health

* **Static Variables**
  + Boost (int)
* **Instance Methods**
  + Use - will use the item to set the player’s current health back to max

**Duck Nip:** permanentlyincreases the player’s strength and max health (like a level up!)

* **Static Variables**
  + Strength Boost
  + Health Boost
* **Instance Methods**
  + Use - will use the item to boost the player’s strength and max health

**Block:** any walls/floor tiles that the player may interact with

* **Variables**
  + Hits Left (int) - the number of remaining hits needed to break this block, will default to -1 if the block is indestructible
* **Instance Methods**
  + Hit - will subtract one from the number of hits remaining, if hits left is zero, then destroy the block (if hits left starts at -1, then the block will never reach zero, and it will never be destroyed)

**Enemies:**

* **TBD!** The enemies will be the most challenging component to code, and I am not sure if I will have the time to make them work properly, so I will leave them for a later time. Essentially, they will all need special melee and/or ranged attack methods and I haven’t figured out how I would like to do that just yet.