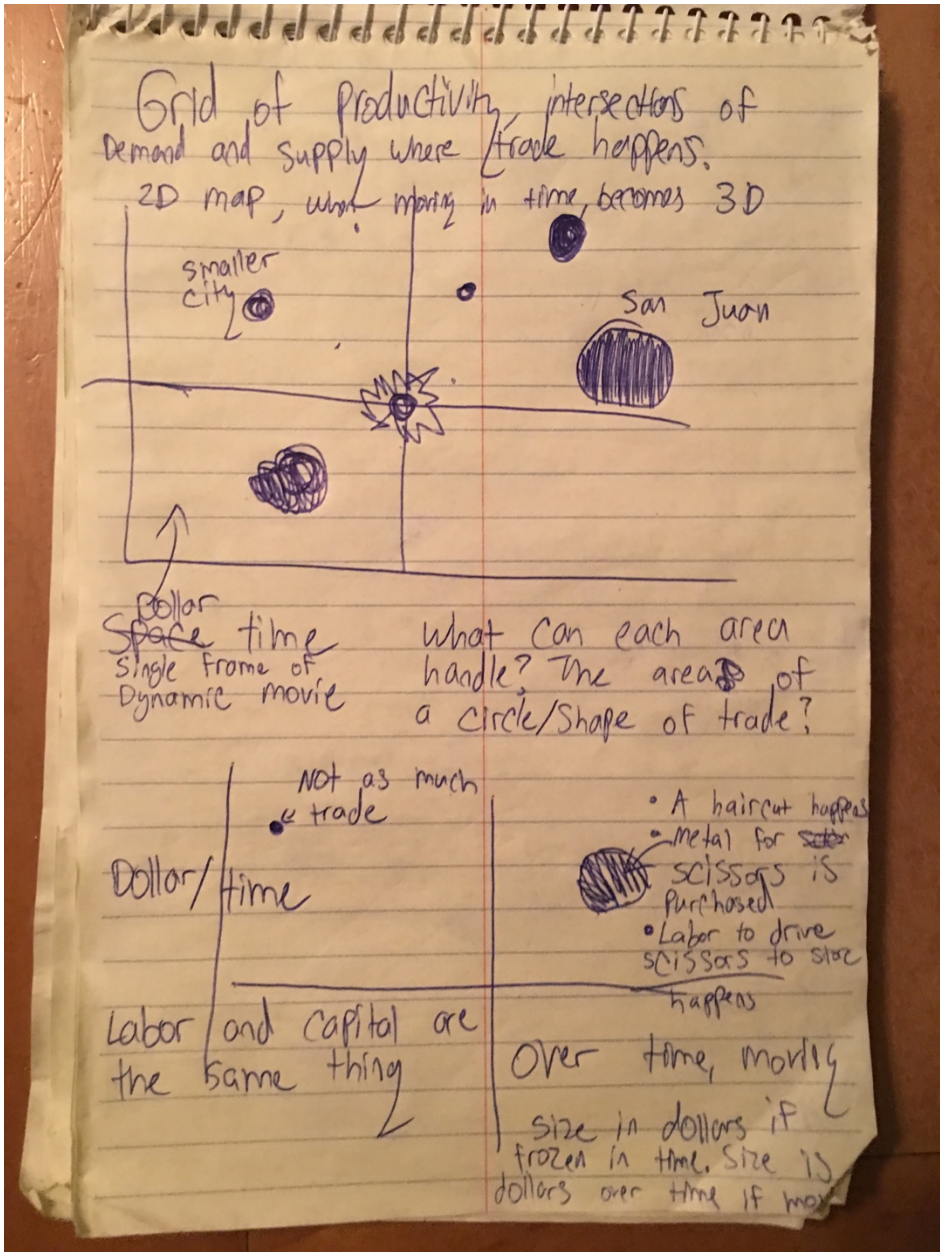
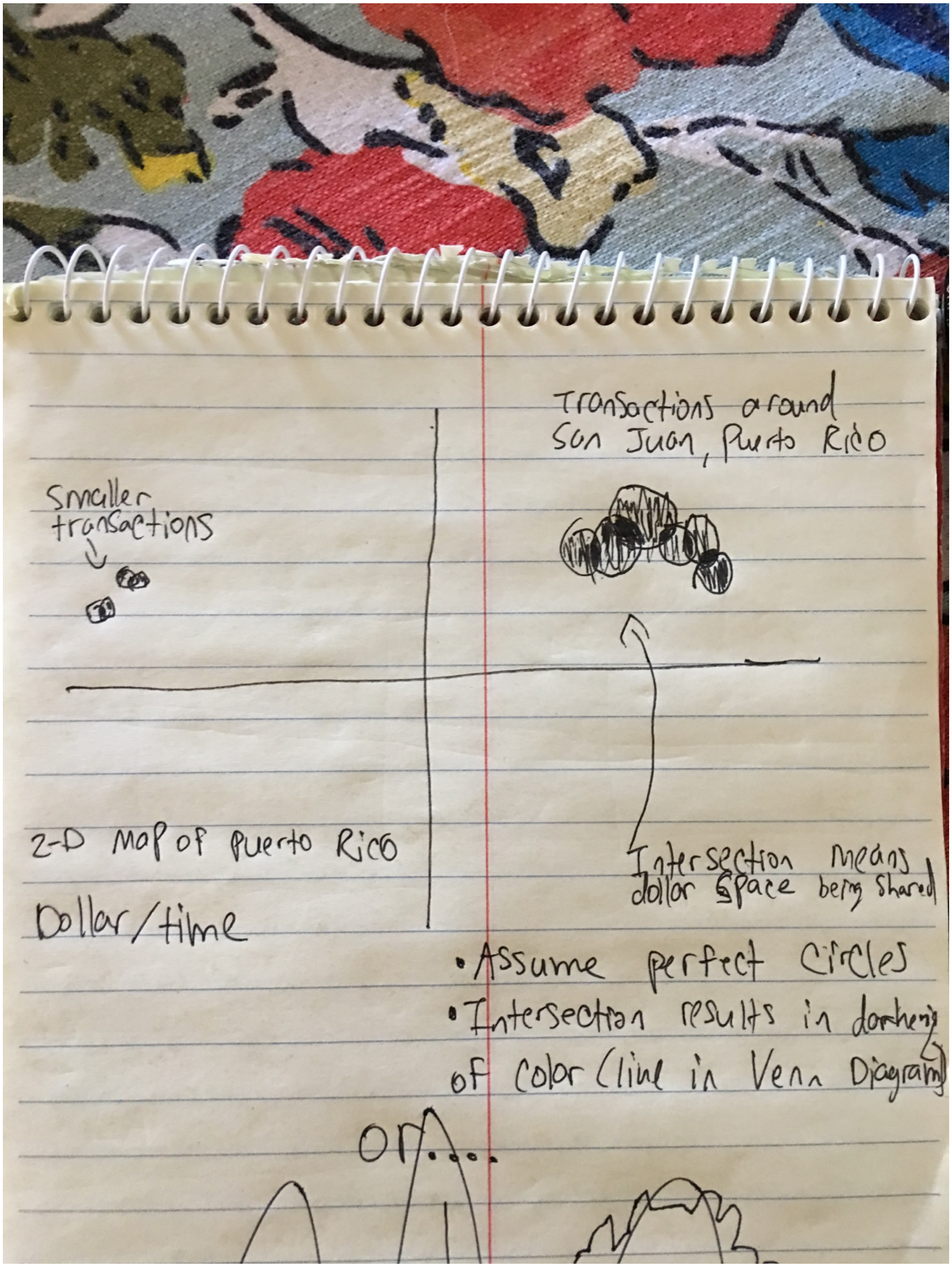
# Grid of Productivity

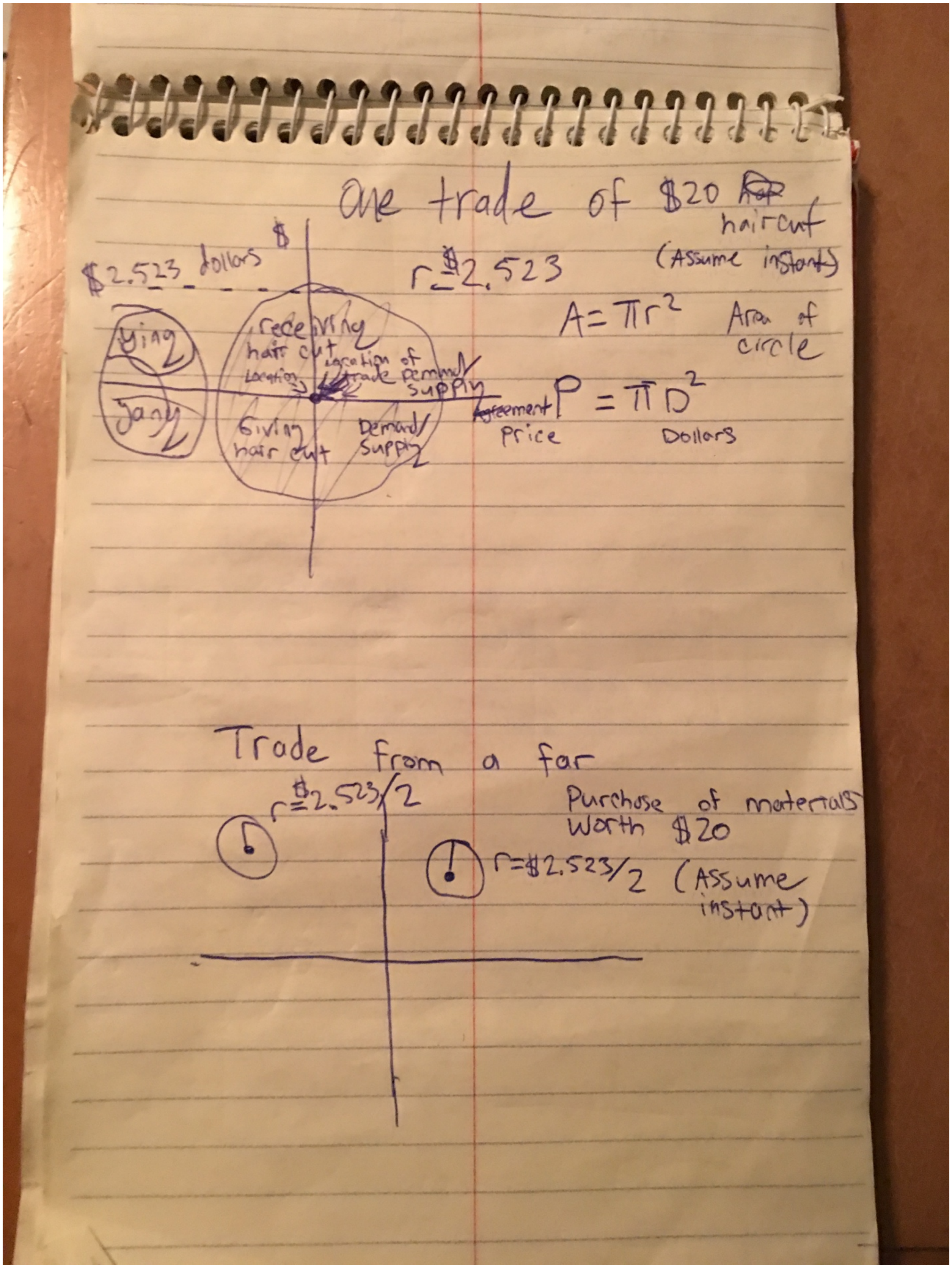



One way someone can visualize the market. If San Juan, Puerto Rico had $25,000 dollars worth of transactions at this one instance (includes all trade, including interest payments/receivables), then the area of San Juan would be 25000 in the grid. It is a perfect circle because I arbitrarily picked a center to be the center of San Juan and it's frozen in time.

A more accurate reading of dollar space is to plot where every individual transaction happens. Intersections result in shared dollar space (Think Venn Diagram)


Transactions around San Juan, Puerto Rico
Smaller transactions
2-D Map of Puerto Rico
Dollar/time
Intersection means dollar space being shared
•Assume perfect circles
•Intersection results in darkening of color (like in Venn Diagram)
or...

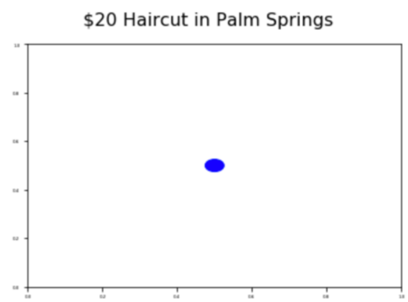
**One Transaction of a $20 haircut**



## Translating into Python

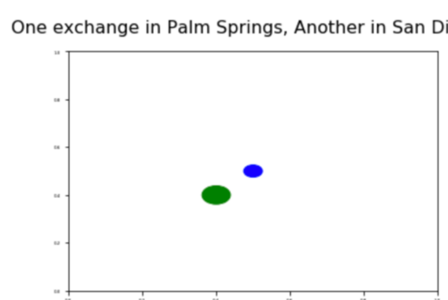
In [126]:
```
#Representing a $20 haircut (assume instant) in Palm Springs, CA
#Let (.5, .5) be 2D cooridinates of Palm Springs, CA on planet Earth, wherever we decide to put center
#Let 0.02523 be $2.523, making the area of the circle $20
#Area of Circle = pi*r**2
#$Amount from trade = pi*dollars**2
#$20 = pi*$2.523**2
circle2 = plt.Circle((0.5, 0.5), 0.02523, color='blue')

fig, ax = plt.subplots() # note we must use plt.subplots, not plt.subplot
# (or if you have an existing figure)
# fig = plt.gcf()
# ax = fig.gca()
fig.suptitle('$20 Haircut in Palm Springs', fontsize=16)
#ax.add_artist(circle1)
ax.add_artist(circle2)
#ax.add_artist(circle3)
```

Out[126]: <matplotlib.patches.Circle at 0x146d0dfd0>



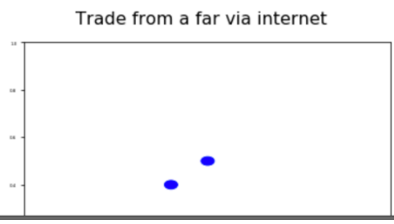

In [127]:
```
#Representing a $20 haircut in Palm Springs and $45 purchase of metal scissors in San Diego,CA
#Let blue be Palm Springs, CA and green be San Diego (Model doesn't have true scale atm)
#Haircut in Palm Springs
circle2 = plt.Circle((0.5, 0.5), 0.02523, color='blue')

#Metal for barbors in San Diego
#pi*$3.845**2 = $45
circle3 = plt.Circle((0.4, 0.4), 0.03845, color='green')

fig, ax = plt.subplots() # note we must use plt.subplots, not plt.subplot
# (or if you have an existing figure)
# fig = plt.gcf()
# ax = fig.gca()
fig.suptitle('One exchange in Palm Springs, Another in San Diego', fontsize=16)

#ax.add_artist(circle1)
ax.add_artist(circle2)
ax.add_artist(circle3)
```

Out[127]: <matplotlib.patches.Circle at 0x146cbf390>



In [128]:
```python
#Representing trade from a far
#Purchase of $20 of metal in san diego, the buyer is in Palm Springs

#Palm Springs is the center
#This part requires real data
circle2 = plt.Circle((0.5, 0.5), 0.01784, color='blue')
#GDP of San Diego = culmination of all trade (dollar value, includes both purchase of labor amd purchase of goods/servi
#math.pi*$3784.5**2 = $45,000,000
circle3 = plt.Circle((0.4, 0.4), 0.01784, color='blue')

fig, ax = plt.subplots() # note we must use plt.subplots, not plt.subplot
# (or if you have an existing figure)
# fig = plt.gcf()
# ax = fig.gca()

fig.suptitle('Trade from a far via internet', fontsize=16)

#ax.add_artist(circle1)
ax.add_artist(circle2)
ax.add_artist(circle3)

#XY are coordinates on map
#Area is in dollars, but can be used for any unit (e.g. number of xyz proteins in dna)
#Sum of circles is $20
```

Out[128]:
```
<matplotlib.patches.Circle at 0x14563f048>
```




The above graphs are 2D maps and the area/sum of the circles is the total transaction amounts. One can color code certain transactions (e.g. all buyers are green, all sellers are yellow or all interest payments are red and all interest receivables are green OR all debt is red and all surplus is green) anyway one desires. I believe a more accurate portrayal of trades would include Venn Diagrams.