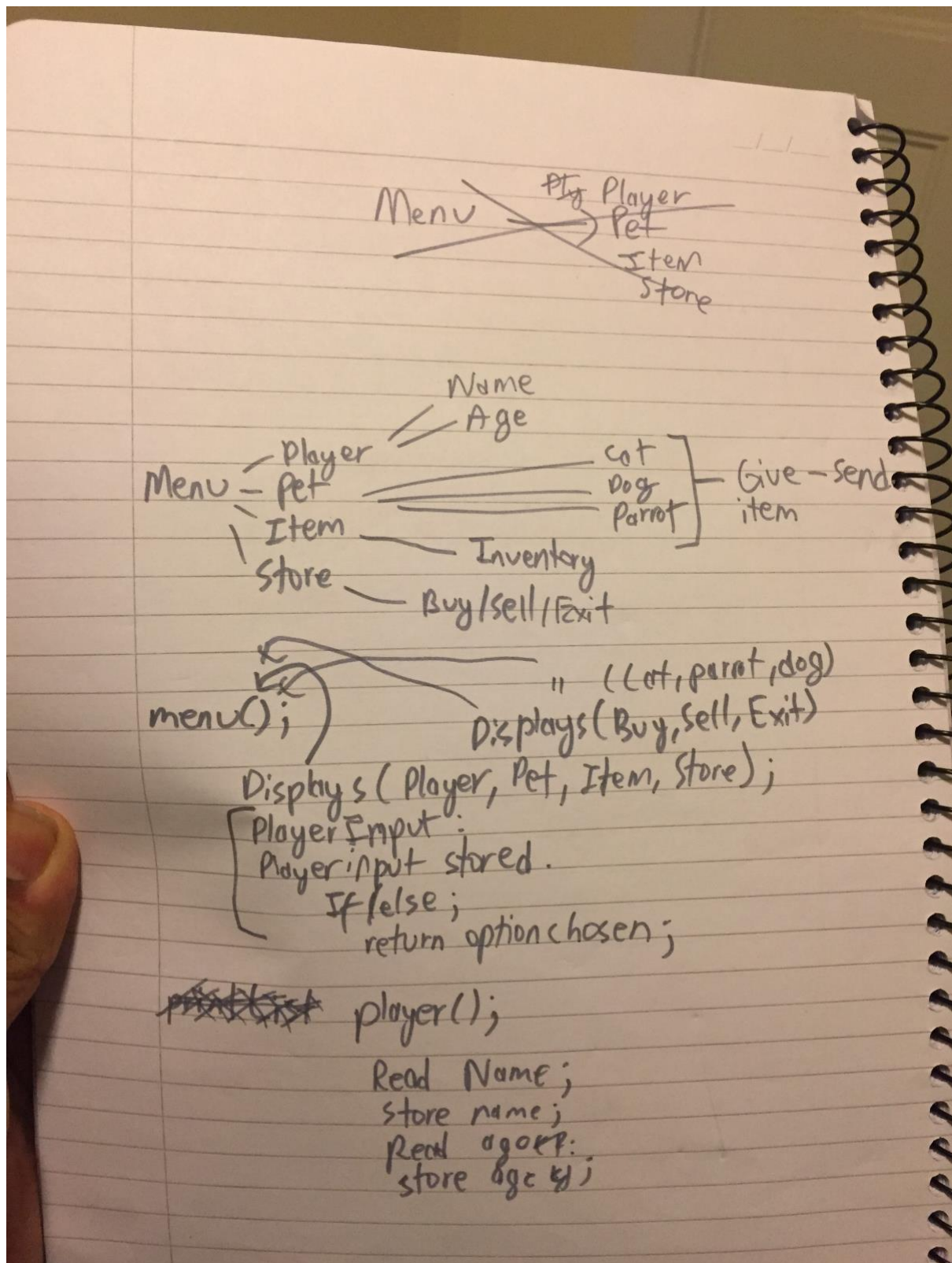


Documentation

Initial sketch of my idea:



UML Diagram

```
Program
Title = "Ishan's Awesome Virtual Pet 2.0";
GameLogic gl = new GameLogic();
Main();
```

```
Pet
public PetHUD pethud;
public string PetName;
public string PetColor;
public string PetAge;

public void attributes()
public string getName()
```

```
Dog : Pet

public static List
public Dog()
```

```
Cat : Pet

public static List
public Cat()
```

```
Parrot : Pet

public static List
public Parrot()
```

```
GameLogic
string path = "intro.txt";
Tools tool = new Tools();
Dog dog = new Dog();
Cat cat = new Cat();
```

```

Parrot parrot = new Parrot();
Store store = new Store();
Player player;
List<INameGet> listOfPet = new List<INameGet>();
List<string> petnames = new List<string>();
List<INameGet> listOfItems = new List<INameGet>();
List<string> itemnames = new List<string>();

public GameLogic()
public List<string>

```

```

Store

public List
public delegate int GetPrice(Item item);

public Store()
public static int BasePrice(Item item)
public static int FluctPrice(Item item)

```

```

Item : INameGet

public ItemType typeOfItem;

public string getName()
public Item(ItemType _item)

```

```

interface INameGet

string getName();

```

```

Player

public string PlayerName;
public string PlayerAge;

public Player()
public void inputPlayer()
public void outputPlayer()

```

```

public enum MainHUD

Player,
Pet,
Item,
Store,
Exit

```

```

public enum MainHUD

```

```
Player,  
Pet,  
Item,  
Store,  
Exit
```

```
public enum PetHUD
```

```
Cat,  
Dog,  
Parrot
```

```
public enum StoreHUD
```

```
Buy
```

```
public enum ItemType
```

```
Walking_Stick,  
Voice_Changer,  
Slide_Whistle,  
Ball,  
Milk_Bowl,  
Water_Bowl,  
Bird_Feed,
```

```
public enum SkillEnum
```

```
Walk,  
Talk,  
Slide,  
RollOver,  
Meow,  
Bark,  
Chirp,
```

```
Tools
```

```
public string playerInput;
```

```
public int menu
```

```
public string Spacer(string space)
```

Tools
public string playerInput;
public int menu
public string Spacer(string space)

This was a really interesting project because I completely changed the way I think about code now. Instead of doing top-down programming, I'm starting to explore bottom-up programming. I had an amazing time dealing with this project because of the conditions we were given. Even though I didn't feel the requirement for delegate in my project initially, as I started think more methodical, it made perfect sense. Thanks to Jethros help with how he approached bottom-up programming, I was able to make a reusable program.