Bu projede ilk olarak kullanılan sistemler, programlar ve robotlar önem addediyor. Bunlar Linux işletim sistemi, Rviz veya Gazebo gibi programlar kurulumu, çalışması projenin başlangıç kısmını oluşturacaktır.

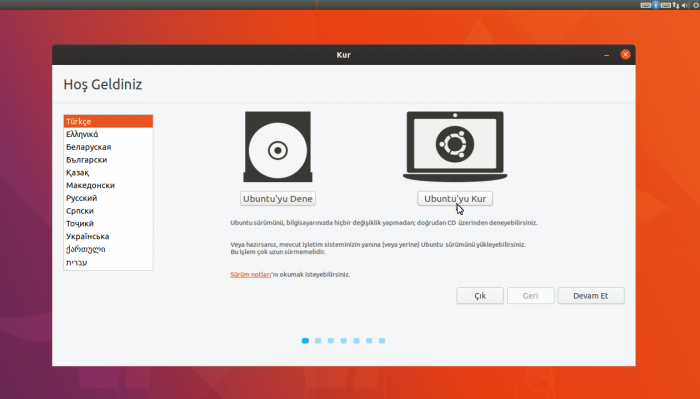
## 7.1 Proje Kurulumu

Projeye başlamak için ilk olarak Linux işletim sistemi kurulumu gerekiyor. Bu 2 şekilde yapılabilir. İlk olarak gündelik hayatta daha çok kullanılan Windows işletim sistemi içeresinde bir sanal işletim sistemi kurmak. Bu yöntem sayesinde bilgisayarın belleğinden, hafızasından sürekli yer kaplayacak bir alan ayırmaktan kurtulmuş oluyor. Kurulan bu sanal işletim sistemi Windows işletim sistemi çalışırken normal bir program açıyormuş gibi açılmak suretiyle kullanılıyor. Bu sistem başlatıldıktan sonra gerekli olan kurulumlar yapılıp asıl istenen projeler yapılmaya başlanır. Lakin bu sistem genellikle çok yavaş çalışır. Bu yüzdendir ki çoğu proje ya burada yapılamaz ya da çok fazla hata ile karşılaşılır. Bu proje bahsi geçen sebeplerden ötürü Windows işletim sistemine ilaveten Linux işletim sistemi normal kurulumlu (dual boot) şekilde bilgisayara kurulmuştur.

### 7.1.1 Linux Ubuntu 16.04 Kurulumu

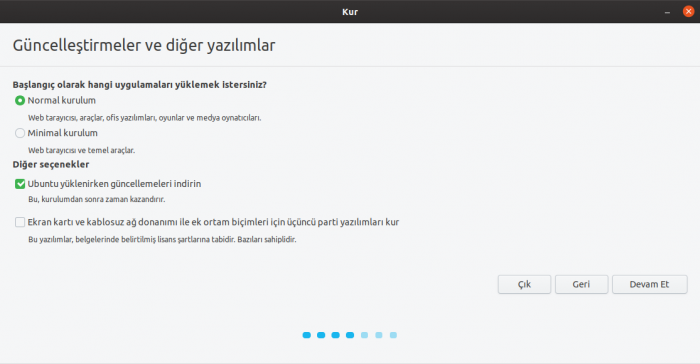
Linux, 1991 yılında Finlandiyalı bir üniversite öğrencisi tarafından daha eski bir sistem olan UNIX mimarisinden esinlenerek sıfırdan yazılmıştır. Yazılan bu sistem 1992 yılında daha çok kullanıcıya hitap edebilmek ve bu sayede sistem içerindeki hataların giderilmesini sağlamak amacıyla özgür yazılım olarak yayınlandı. Bu sayede Dünya çapında tanınırlığa ulaşan sistem pek çok gönüllü yazılım uzmanları katkısıyla geliştirilmiştir. [8] Ubuntu ise Linux çekirdeği temel alınarak geliştirilen açık kaynak kodlu bir işletim sistemidir. İlk sürümü 2004 yılında yayınlanmıştır. Masaüstü, bulut ve nesnelerin internetine yönelik geliştirilen türevleri bulunur. [9] Bu projeye başlayabilmek adına ilk olarak Ubuntu 16.04 kurulumunun gerçekleşmesi gerekir. Bunun için kurulum dosyasının internet yardımıyla indirilmesi gerekir. Bu yapılırken dikkat edilmesi gereken husus hangi sürümün indirilmek istendiğidir. İki çeşit sürüm vardır. Bunlardan birinin yanında “LTS” yazar. Bu kısaltma indirilmek istenen sürümün uzun ömürlü olduğunu gösterir. Diğeri ise daha kısa ömürlüdür ve bu süre sonunda güncelleme yapılması gerekir.

Bu proje için uzun ömürlü sürüm uygundur. İndirilen bu dosyanın kurulum kısmında ise ilk önce bu dosyayı bir harici belleğe aktarıp bilgisayarı tekrar başlatmak gerekiyor. Bilgisayarın BIOS ayarlarından sabit bellek yerine bu harici bellek ile çalıştırılması gerekir. Çalıştırdıktan sonra artık kurulum aşamaları başlayacaktır. İlk olarak dil seçimi yapılması gerekir. Burada dil seçimi yapıldıktan sonra Ubuntu kur seçeneği ile yükleme başlayacaktır.



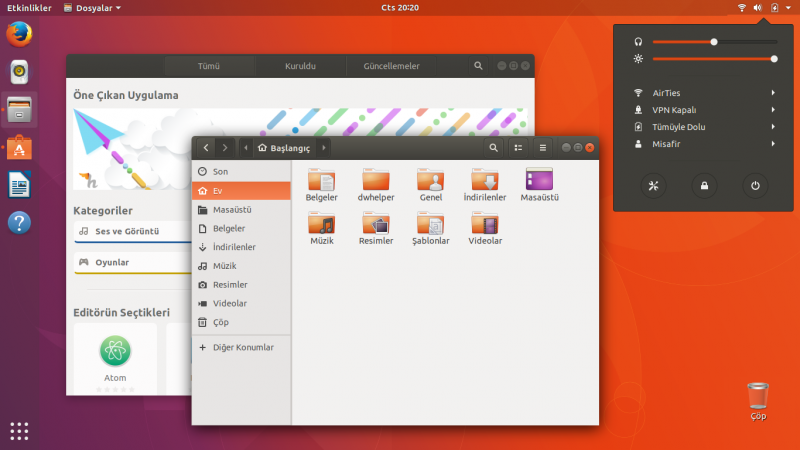
**Şekil 7.1:** Linux Ubuntu 16.04 Kurulumu Dil Seçimi.

Sonra klavye düzeni seçmemiz gerekir. Genellikle burada normal şartlarda klavye düzeni otomatik olarak belirleneceğinden bir seçim yapılmaz. Bu işlemden sonra devam edip internet bağlantısı ekranına gelinir. Kurulum anında bir internet tarayıcısına bağlanmak zorunlu değildir. İsteniliyorsa sonradan bu özellik yapılabilir. Bu ekrandan da sonra güncelleştirme ve diğer yazılımların seçimleri gelir. İki şekilde gerçekleşir. İlki normal kurulum diğeri ise minimal kurulumdur. Burada normal kurulum tercih edilir. Dikkat edilmesi gereken kısım ise bu yükleme gerçekleştirilirken güncelleştirmelerin de yüklenmesidir. O yüzden o seçenekte seçilmelidir.



**Şekil 7.2:** Linux Ubuntu 16.04 Kurulum Güncelleştirme Ekranı

Bu aşamadan sonra yükleme gerçekleşecektir. Bu yüklemeden sonra parola oluşturulması istenir. Oluşturulan parola sonrasında Ubuntu masaüstü açılır.



**Şekil 7.3:** Linux Ubuntu 16.04 Kurulum Masaüstü

### 7.1.2 Robot Operating System Kinetic Kurulumu

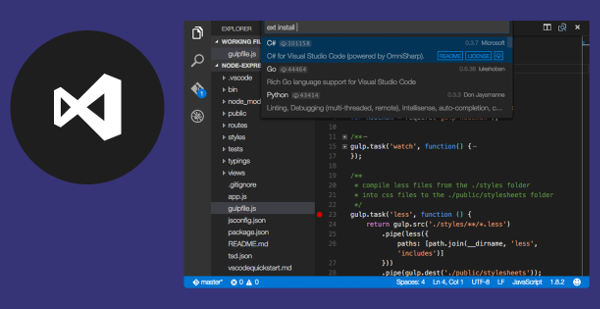
ROS robotların sensörler yardımıyla dış dünyadan aldığı verileri, bilgileri değerlendirip tekrar robota göndermeyi sağlayan açık kaynak kodlu bir yazılımdır. Bu yazılım sayesinde robotlar dış dünyadaki olaya cevap yani tepki verebilir. ROS konular (Topic) mantığıyla çalışır. Yani sensörler yardımıyla gelen bilgi bir konu yardımıyla yayınlanır (publish). Daha sonra gelen bu bilgiye cevap verebilmek için kodun ilgili kısmında o konuya abone (subscriber) olunur. Bu sayede bilgi akışı sağlanmış olur ve mesaj yerine ulaşmış olur. Bundan sonra verilecek tepki belirlenir. Tepki belirlendikten sonra ise tekrardan bu konuya verilecek tepki yayınlanır. Yayınlandıktan sonra tepkiyi verecek olan kısımda abone olur. Bundan sonra da tepki verilmiş olur. Bu sayede bilgi alışverişi tamamlanmış ve haberleşme sağlanmış olur. Bu haberleşme sistemi Python ve C++ gibi önemli yazılım dilleriyle beraber kullanılabilir olduğu için kullanımı ve uyarlanması kolaydır. Aynı zamanda açık kaynak kodlu olduğu için sürekli olarak gelişimini sürdürmektedir. Robot haberleşmeleri için önemli ve elzem olan ROS bu proje için de önemlidir. Proje yapımına başlayabilmek adına da ROS Kinetic sürümü bu proje uygundur. Kurulum kısmında ilk olarak 4 seçenek karşımıza çıkar. Bu seçenekler; Masaüstü-Tam Kurulum, Masaüstü Kurulum, ROS-Base ve Bireysel Paket.

Masaüstü-Tam Kurulum seçenekler arasında önerilen kurulum türüdür. Özellikle yeni kullanıcıların kurulum aşamasında ve sonrasında sıkıntı yaşamaması için tavsiye edilir. Bu kurulum türünde ROS Kinetic kurulumu yanında Gazebo, Rqt, Rviz, Robot-Jenerik Kütüphaneleri, 2D/3D simülatörleri, navigasyon ve 2D/3D algısı gibi bazı uygulamalar ve programların kurulumları gerçekleşir. Bu sayede özellikle yeni kullanıcılar bu programları ve ayarlamaları teker teker yapmaktan kurtulur.

Masaüstü Kurulum ise Masaüstü-Tam Kurulum seçeneğine göre daha yalın ve sadedir. Burada ROS Kinetic kurulumu yanında Gazebo, Rviz, Rqt ve Robot-Jenerik Kütüphaneleri gibi temel kısımlar kurulur. ROS-Base kısmında ROS paketi, derlemesi ve iletişim kütüphaneleri yüklenir. Herhangi bir kullanıcı arayüz aracı yoktur. Dördüncü ve son seçenek olan Bireysel Paket ise daha önce bu işlerle uğraşmış bilen kişiler için en iyi seçenektir. Çünkü burada yüklenilecek olan her şeyi kullanıcı kendi istediği şekilde seçebiliyor. Bu yüklemeyi yaparken istenilen uygulamalar, programlar paket isimleri ile birlikte sırayla yüklenir. Bu seçeneklerden birini seçtikten sonra ROS ortam kurulumu ve paket oluşturma bağımlılıkları vardır. Bu yüklemelerde yapıldığı zaman ROS Kinetic kurulumu tamamlanmış olur.[10]

### 7.1.3 Visual Studio Code Kurulumu

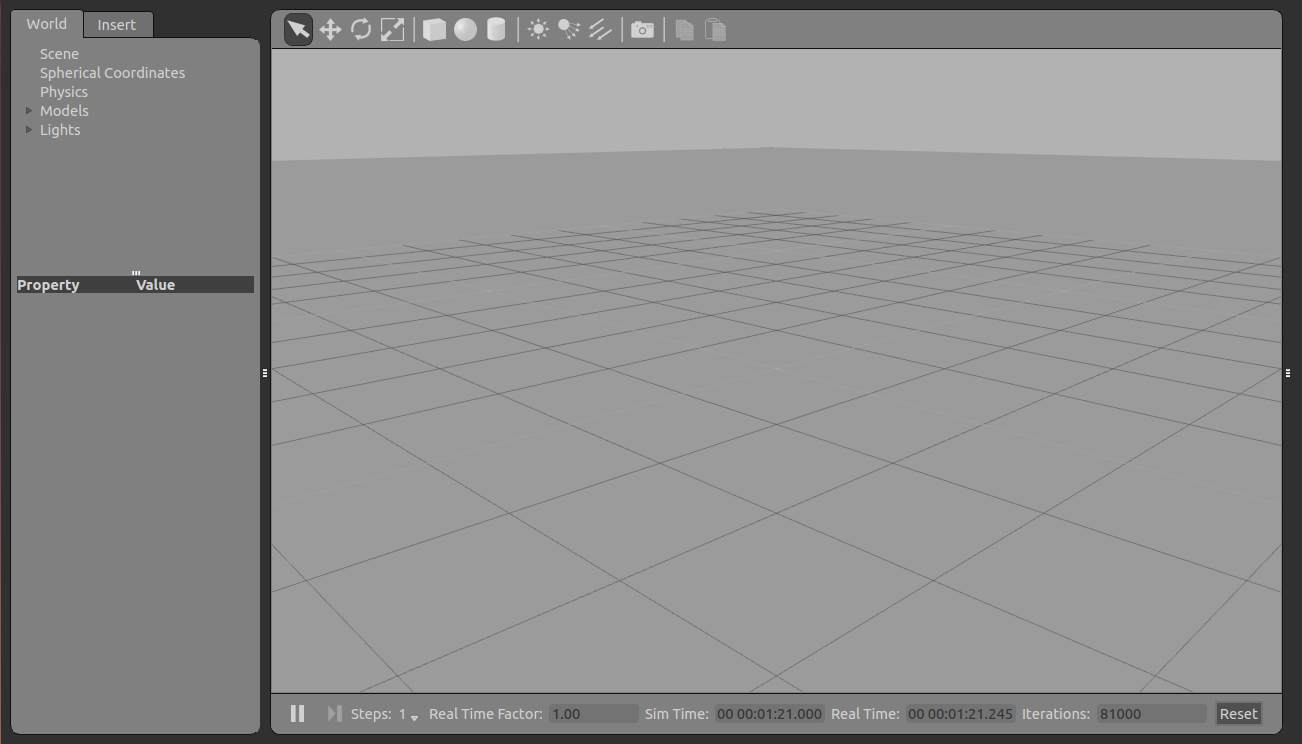
Visual Studio hem Windows işletim sisteminde hem de Linux işletim sisteminde çalışabilen açık kaynak editörüdür. C, C++, C#, Python, Node.js gibi yazılım dilleri için kullanımı oldukça kolaydır ve mümkündür. Linux’ta kurulum yapmanın 2 yolu vardır. Bunlardan ilki kurmak istenilen programın kullanılan sisteme uygun olan DEB veya RPM paketlerinin indirilip yüklenmesiyle olur. Diğer yöntem ise terminal kullanılarak yapılan yüklemelerdir. Terminal üzerinden kurulum yapabilmek için ilgili komutlar bilinmelidir. Yüklemenin doğru bir şekilde yapılabilmesi için yüklemenin yapılacağı yer ve kaplayacağı alan iyi bilinmelidir. Bu seçim yapıldıktan sonra kurulum başarılı bir şekilde gerçekleşir.



**Şekil 7.4:** Visual Studio Code Kurulum.

### 7.1.4 İHA Seçimi ve Gazebo Ortamı Kurulumu

Gazebo ortamını açabilmek için uygulama simgesine çift tıklamak veya terminalden Gazebo yazmak yeterlidir. Bu sayede boş bir Gazebo ortamı açılacaktır. Burada Gazebo ekranında bulunan seçenekler yardımıyla oluşturulacak olan projeye istenilen evler, arabalar, ambulanslar, itfaiye araçları, kafeler, tanklar, İHA’lar, İKA’lar vb. her şey eklenebilir. Gazebo sürümü yeterliyse güneş, rüzgar, ışıklandırma, karartma gibi faktörlerde istenmesi halinde bu ortamlara eklenebilir. Eklenen her cismin Gazebo ortamında bir matematiksel konumu, ağırlığı, kapladığı alanı, parlaklığı gibi fiziksel hayatta mevcut olan her türlü özelliğe sahiptir. Burada bu projede kullanılacak olan İHA ve İKA teknik özellikleri açısından özel ve karmaşık yapılara sahip olduğu için başlangıçta seçilip ortama eklenebilecek robotlar değil. Bu yüzden ilk olarak projede kullanılacak olan robotlar seçilip daha sonra bunlar için özel kurulum yapılır.



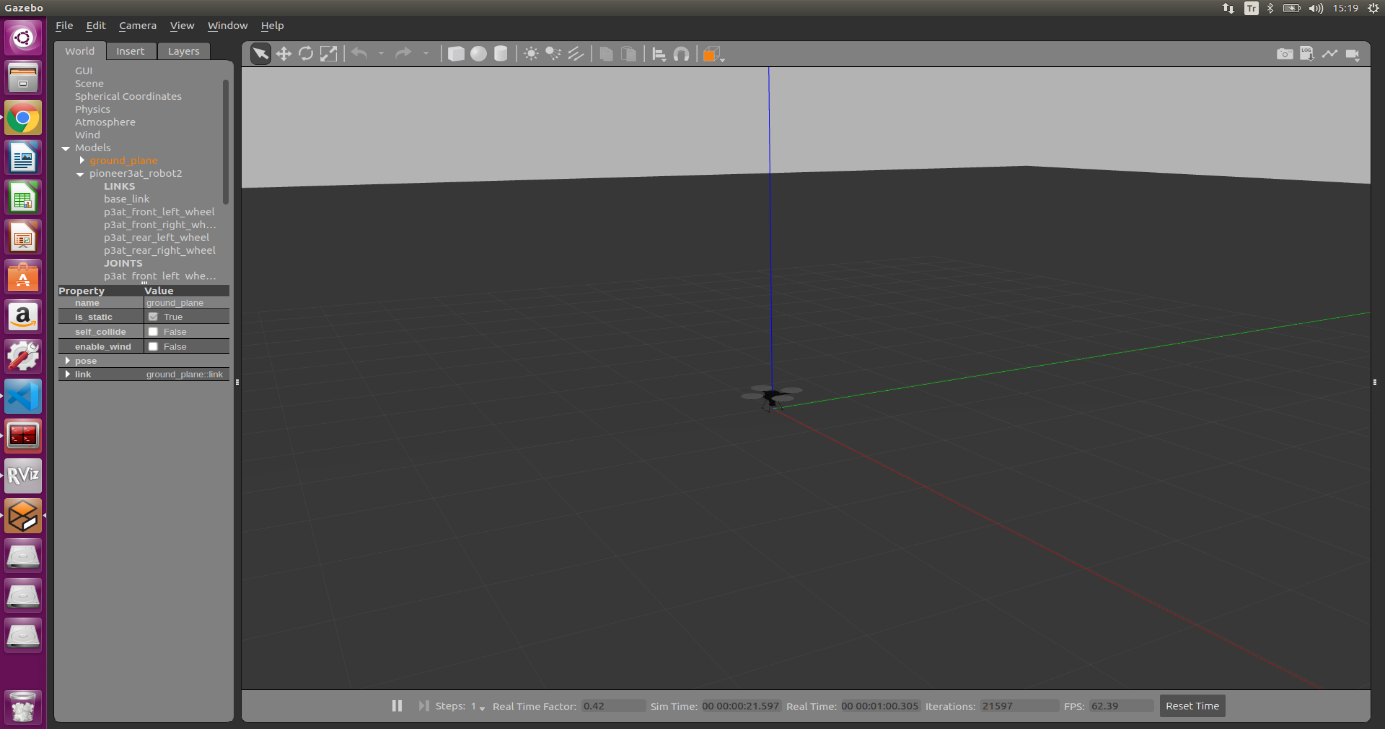
**Şekil 7.5:** Gazebo Boş Ortam Kurulumu.

Bu projede kullanılmak için uygun olan birkaç farklı quadrotor vardır. Ardrone, Hector Quadrotor gibi. Lakin en uygunu Hector Quadrotor’dür. Bu robot modeli Gazebo’nun kendi kütüphaneleri içinde yoktur. Onun için özel bir kurulum ister. Bunun için Hector Ortamı internetten indirilir. Daha sonra gerekli kurulum yapılır. Kurulumu yapılan bu ortamın etrafında dağlık engeller vardır. İlk olarak bunun giderilmesi lazım. Çünkü bu projede istediğimiz alan engelsiz ve düz olmalıdır. Bunu yapabilmek için Gazebo’yu çalıştırabilmek için kullanılan .launch uzantılı dosyaya giriyoruz. Burada dağlık engebeli alanın oluşmasına neden olan kodlar silindikten veyahut değiştirildikten sonra dosya kaydedilir ve Gazebo tekrardan başlatılır.



**Şekil 7.6:** Hector Quadrotor Gazebo Ortamı Kurulumu.

Şekil 7.6 ‘da Hector Quadrotor ortamı başlatıldı. Şekil 7.6’da da görüldüğü gibi dağlık ve engebeli arazinin ne kadar sık ve ne kadar yüksek olduğu görülür. Yukarıda da bahsedildiği gibi bu projede daha düz ve engebesiz alan gerekir. Bunun sebebi İKA üzerine yapılacak inişin daha düzgün ve kazasız olmalı. Aynı zamanda İHA’nın olası bir kaza yapması engellenmeli. Bu yüzden dağlık arazi silinir ve engelsiz düz bir ortam oluşturulmuş olur. Düzenlenen bu ortam sayesinde kullanılan İHA ve İKA başarılı bir şekilde verilen koordinatlara gidebilecektir.



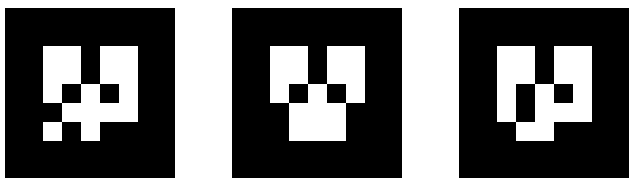
**Şekil 7.7:** Hector Quadrotor Ortam Kurulum-2.

Burada görüldüğü üzere henüz Pioneer-3AT yani İKA kurulumu yapılan ortamın içinde bulunmamaktadır. Bunun için ilk olarak Pioneer-3AT ortamının kurulması gerekir. Bu kurulumdan sonra Pioneer-3AT ortam dosyası yani .launch uzantılı dosya içerisinde bulunan Pioneer-3AT spawn kodlarının Hector Quadrotor ortam dosyasına atılması gerekir. Bu işlem gerçekleştikten sonra Hector Quadrotor ortamı her başlatıldığında hem Hector Quadrotor hem de Pioneer-3AT aynı ortamda başlatılır. Bu sayede birbirleri ile haberleşmeleri ve birbirlerine göre konumlanmaları mümkün hale gelir. Bu şu şekilde açıklanabilir; Hector Quadrotor başlangıç halinde bir farklılık olursa yani yükselmeye başlarsa, ileri veya geri giderse bu hareketleri ve konum değişiklikleri Pioneer-3AT’ye iletilebilir. Aynı şekilde Pioneer-3AT hızlanıp yer değiştirirse veya etrafında dönerse Hector Quadrotor’a iletilebilir.

### 7.1.5 AR-Tag Kurulumu ve Ar-Track-Alvar Kütüphanesi Kurulumu

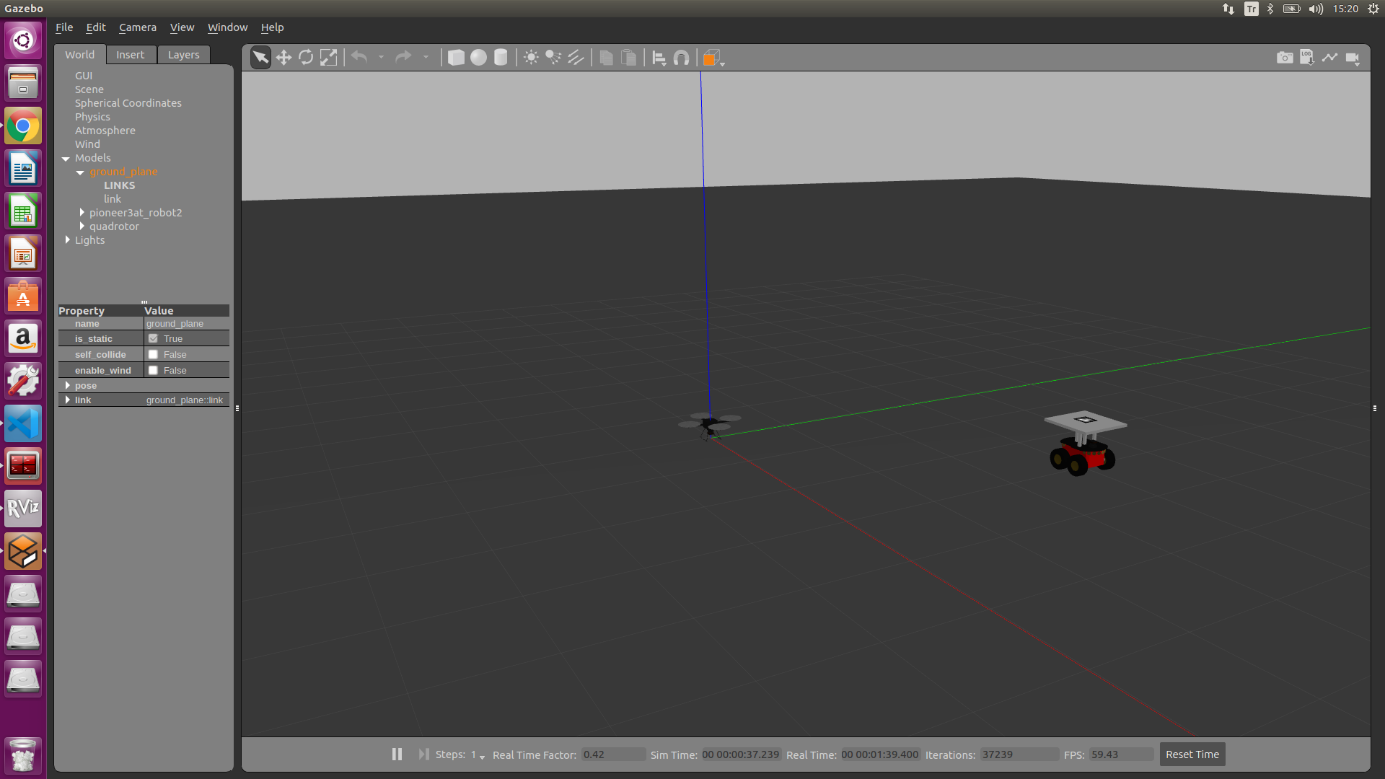
Ar-Track-Alvar kütüphanesi açık kaynak kodlu bir AR etiketi izleme kütüphanesidir. Bu kütüphane sayesinde platformlar veyahut zeminlere eklenen AR etiketleri algılanabiliyor. Bu algılama sayesinde eğer etiket hareketliyse yani konumu değişiyorsa takip edilir. Bu da AR etiketi bir hareketli platform üzerindeyken yani yer değiştirirken örneğin onun üzerine iniş yapılabilmesini sağlar. Ar-Track-Alvar kütüphanesiyle değişken boyutlu ve çözünürlüklü veri kodlamasına sahip bir AR etiketi oluşturulur. Bu kütüphane AR etiketinin bir adet oluşturulması yanında birden fazla AR etiketi mevcut olduğu durumlarda oluşturulan etiketlerin pozunu belirleme ve izleme, daha iyi poz tahminleri için isteğe bağlı olarak kinect derinliği verilerini (kinect kamera mevcut olduğunda) entegre edebilir.

İşaretçiler (Marker) genellikle 20 farklı etiket içerir. Bu etiketlerin boyutu minimum 5x5 iken maksimum 9x9 santimetre olmalıdır. Bu boyutlardan büyük veya küçük olursa AR etiketi algılanamayabilir. Bu da işaretli bir zemine iniş yaptırılmak istenin bir İHA’nın iniş yapamamasına sebep olur. Bu yüzden işaretleyicilerin boyutuna dikkat edilir. Her işaretleyicinin görüntüde hızlı algılanmasını sağlayan geniş bir siyah kenarlığı ve iç kısmında kimliğini gösteren ikili matrisi vardır. [11]



**Şekil 7.8:** AR-Tag Görünümü.

Oluşturulan bu AR-Tag esasında bir fotoğraf karesi gibidir. Bunu bir cismin üzerine entegre etmek gerekir. Bunun içinde Blender isimli programdan AR-Tag kurallarına uygun bir küp tasarımı yapılıp bu fotoğraf karesi tasarımı yapılan küpün bütün yüzeylerine entegre edilir. Bu işlemden sonra uygulama içinde bir takım işlemlerden sonra .dae uzantılı bir dosya oluşturulur. Bu dosya yardımıyla AR-Tag’i Pioneer-3AT’ye entegre edilebilir. Bu entegre işlemini Pioneer’e ait bir gövde tasarımı dosyası olan .xacro uzantılı dosya içerisinde, oluşturulan .dae dosyasını kullanarak bu AR-Tag Pioneer-3AT üzerine eklenir. AR-Tag’in İKA üzerindeki konumunu düzeltmek ve bu İKA üzerine bir platform eklemek için tekrardan .xacro uzantılı dosyaya belirli bir yükseklikleri olan 4 adet çubuk şeklinde nesneler eklenmiştir. Bu nesneler platformu sabit bir yükseklikte tutacaktır. Bu sayede AR-Tag üzerine güvenli bir iniş yapılabilecektir.



**Şekil 7.9:** Hector, Pioneer ve AR-Tag Eklenmiş Ortam.

## 7.2 Proje Ortamı Çalıştırılması ve Komutlar

İlk olarak Gazebo ortamının başlatılması gerekir. Bunu yapabilmek için hector\_quadrotor\_demo/launch klasörü içine outdoor.launch dosyası oluşturulmalı. Daha sonra terminalden

**“cd catkin\_ws/devel”** yazılıp o konuma gidilir. Burada **“source setup.bash”** komutu yazılır sonra Gazebo ortamı başlatılır. Bu başlatma için **“ roslaunch hector\_quadrotor\_demo outdoor.launch”** komutu çalıştırılır ve Gazebo ortamı başlar.

Daha sonra terminalde başka bir pencerede Hector Quadrotor’un motorlarının çalıştırılması gerekir. Bunun için;

**rosservice call /enable\_motors “enable: true”** komutu yazılır. **Enable : true** komutu geri dönerse Hector uçuşa hazır hale gelir.

Artık yazdığımız kodların çalıştırılması gerekir. Bunun için terminalden sırayla pioneer3at\_konum.py, markerla\_inis\_dene.py ve arayuz.py çalıştırılır. Bu sayede proje başarılı bir şekilde başlatılmış olur.

## 7.3 Oluşturulan veya Değişiklik Yapılan Dosyalar

İlk olarak AR-Tag oluşturuldu. Bunun için terminale **“rosrun ar\_track\_alvar createMarker”**

komutu yazıldı. Daha sonra istenilen bilgiye **“1234”** yazıldı. Sonra **2 kez Enter’a** basıldı. Tekrar bilgi istediğinde **“-1”** yazıldı ve **MarkerData\_1234.png** oluştu.

Bu oluşumdan sonra **Blender** adlı uygula yardımıyla oluşan fotoğraf 3 boyutlu simülasyon için bir küpe basıldı. Daha sonra **ar\_2020.dae** ve **ar\_2020.urdf** adlı dosyalar oluşturuldu. Burada **MarkerData\_1234.png** ve **ar\_2020.dae** dosyalarının konumu hector\_quadrotor\_description/meshes klasörü içerisinde olmalıdır. Oluşturulan diğer dosya yani **ar\_2020.urdf** ise hector\_quadrotor\_description/urdf klasörü içerisinde bulunmalıdır.

Daha sonra oluşturulan bu AR-Tag, Pioneer-3AT üzerine eklemek için **pioneer3at\_body.xacro** dosyasına eklendi. Gene aynı dosyaya AR-Tag’i üzerine oturtabilmek için platform ve bunun yanında GPS bilgisi alabilmek için GPS eklendi.