

ROBOT PROJESİ

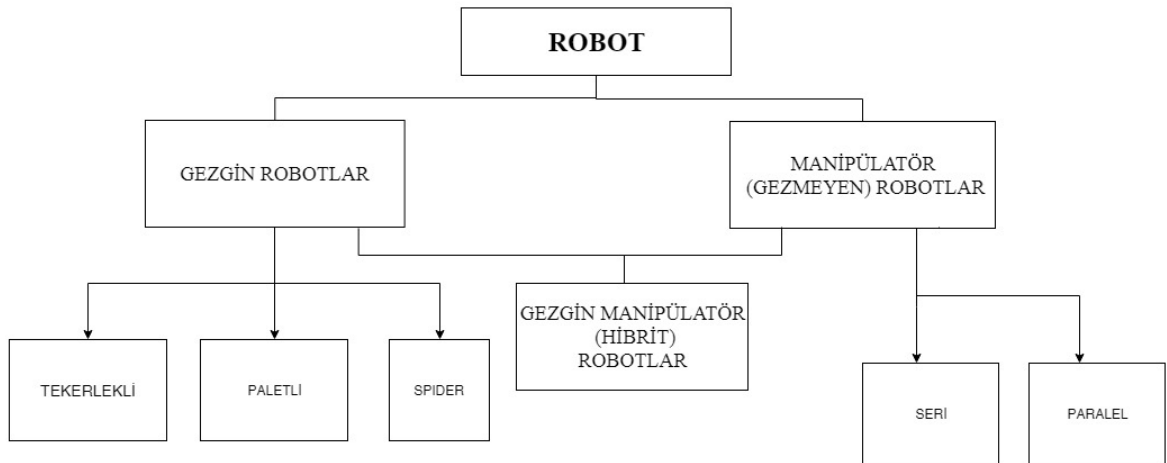
Projede farklı türdeki robotların hareket etme, yük taşıma özellikleri kullanılarak verilecek problemlerin çözümü istenmektedir.

Amaç: Proje gerçekleştirimi ile beraber öğrencilerin nesneye yönelik programlama yapısını anlaması ve çözüm sağlayabilmesi amaçlamaktadır.

Programlama Dili: Proje C++ veya Java dili kullanılarak gerçekleştirilecektir.

1. Problem Tanımı

Projede sizden istenen, her bir robot türü için ayrı ayrı sınıf oluşturmak yerine sistem yönetilebilirliğini sadeleştiren bir uygulama yazmanız. Şekil 1 projede ele alınan robot sınıflarını göstermektedir.



Şekil 1. Oluşturulması İstenen Robot Sınıfları

1.1 Önemli Özellikler

Sınıf özellikleri aşağıdaki kısıtlara göre belirlenecektir.

- Tüm robotların motor sayısı, yük miktarı (kaldıracağı (gezgin robotlar) veya taşıyacağı (Manipülatör robotlar)), robot tipi özellikleri ortaktır.
- Her gezgin robotun gezinme hızı vardır. Gezinme hızı artan sıraya göre; Spider, Paletli ve Tekerlekli gezgin robotlar olarak sıralanır. Yani Tekerlekli gezgin robotlar kendi sınıfı içinde en hızlı robotlardır. Tekerli robotların tekerlek sayısı, paletli robotların palet sayısı, spider robotların bacak sayısı özellikleri vardır. Gezin robotların kaldıracağı yük miktarlarında sınırlama yoktur bu nedenle çok ağır yükleri de kaldırabilirler.

Gezin robotlar özelliklerine göre gidecekleri yolun üzerindeki engelden geçebilirler veya geçemezler. Proje kapsamında Spider robotların engelden geçemeyeceği varsayılacaktır. Tekerlekli robotların engelden geçme süresi, Paletli robotlara göre daha kısadır. Tekerlekli ve Paletli robotlar için `EngelGecmeSuresiBul ()` metodu yazmanız gerekmektedir. Her iki sınıfta bu metodu kullanacaktır ancak hesaplamaları farklıdır; Tekerlekli robotlar için engelden geçme süresi hesaplama formülü ($motorsayisi \times 0,5$) sn iken, Paletli robotlar için ($motorsayisi \times 3$) sn olacak şekilde hesaplanacaktır.

- Her bir Manipülatör robotun yük taşıma kapasitesi, kol uzunluğu, taşıma hızı vardır. Manipülatör robotların konumu sabittir, üzerindeki kollar yardımıyla yükü bir yerden bir yere taşırlar. Manipülatör robotlar çok ağır yük kaldıramazlar bu nedenle robotların genel özelliği olan yük miktarı özelliğine ek olarak yük taşıma kapasitesi özelliğine sahiptir. Kol uzunluğu özelliği ise bir robotun bir yükü hangi mesafeye kadar taşıyacağını belirler (örneğin paralel robot sınıfı için kol uzunluğu 5 m belirlendiyse, bu sınıftan türetilecek nesne taşıyacağı yükü bulunduğu konumdan 6 m uzağa taşıyamaz). Taşıma hızı özelliği göz önünde bulundurulduğunda Seri robotlar çok yavaş, Paralel robotlar çok hızlıdır. Bu nedenle taşıma hızı özelliği için Seri robot nesnesinin Paralel robot nesnesinden hızlı değeri almaması için kontrol olmalıdır. Yük taşıma kapasitesi

özelliđi için ise; Seri robotlar ağır yük kaldıramaz, Paralel robotlar ise ağır yük kaldırabilir. Bu nedenle yük taşıma kapasitesi özelliđi için Seri robot nesnesinin Paralel robot nesnesinden daha fazla kg yük taşıma kapasitesi değeri almaması için kontrol olmalıdır. Ayrıca robota verilecek yük taşıma kapasitesini aşarsa o robot o yükü kaldıramaz (örn. seri robot sınıfı için taşıma kapasitesi 10 kg belirlendiğinde, bu sınıftan oluşturulacak yeni bir seri robot nesnesi 15 kg yükü kaldıramaz).

- **Gezgin Manipölatör (Hibrit) robotlar** ise hem gezgin hem Manipölatör özelliklerini taşırlar. Hem gezinme, hem de üzerine eklenen kol ile yükü bir yerden bir yere taşıma özelliđine sahiptir. Hareketli konumdan sabit konuma geçmek için sabitletme süresi (sn) özelliđine sahiptir. Yükü robotun üzerinden alıp, robot koluna geçirmek için ise geçiş süresi (sn) özelliđine sahiptir. Bu robotlar ayrıca engellerden geçebilirler.

Her sınıf için ortak olan özellikler:

- **Projede Encapsulation, Inheritance, Polymorphism, Abstraction yapılarından gerekli olanların kullanılması gerekmektedir.** Projede her yapıyı kullanmasanız bile, proje sunumu esnasında bu yapıların ne olduğunu bilip bilmediğinizi ölçecek sorular sorulacaktır.

Nesneye yönelik programlamada kalıtım konusu altında anlatılan özellikler kullanılarak sınıflar oluşturulmalıdır.

2. İsterler

Projede üç problemi çözmeniz beklenmektedir. Her üç problem için ilk yapılacak ortak işlem kullanıcı istediği robot sınıfından yeni bir nesne oluşturacak, daha sonra ilgili robota aşağıdaki ilgili problemi çözdürecektir.

PROBLEM 1

Bu problemde sizden gezgin robotların üzerine konulan bir yükü belirli bir konumdan kullanıcıdan alınacak direktifler doğrultusunda hareket ettirmeniz ve gezgin robotun konumunu bulmanız ve hareket için geçen toplam süreyi hesaplamanız istenmektedir.

Adım 1. 20x20 boyutlu ızgara oluşturmanız gerekmektedir, her bir ızgara 10 m x 10 m boyutlarında olduğu varsayılacaktır. (Izgaraları görselleştirmek için grafik kütüphanesi kullanmalısınız)

Adım 2. Bu ızgara üzerine engeller yerleştirilip yerleştirilmeyeceği bilgisi kullanıcıdan alınmalıdır. Eğer engel yerleştirilecekse, engel sayısı ve engellerin konum bilgileri (x,y) kullanıcıdan alınacaktır. Daha sonra engeller ızgaraya yerleştirilecektir. **Unutulmamalıdır ki Spider robotlar engelden geçemez.**

Adım 3. Gezgin robotun başlangıç konum bilgisi (x,y), daha sonra robotun başlangıç konumundan itibaren gideceği yön bilgileri kullanıcıdan alınacaktır. Yön bilgisi için örneğin kullanıcı 2 ızgara ileri, 5 ızgara geri, 1 ızgara sağa, 3 ızgara sola şeklinde olabilir. Yada örneğin 3 ızgara ileri, 5 ızgara sağa şeklinde de olabilir. **Burada dikkat edilmesi gereken durum robotun ızgaradan çıkıp çıkmayacağının belirlenmesidir.**

Adım 4. Tüm bu bilgiler doğrultusunda, robotun bulunacağı son konumu grafik kütüphanesi yardımıyla ızgara üzerinde gösterilecektir. Ayrıca başlangıç konumundan son konuma kadar geçen süre sn cinsinden hesaplatılmalıdır (**Süre hesaplatılırken eğer varsa engelden geçme süresi de eklenmelidir, hız m/s cinsinden olduğu varsayılacaktır**)

PROBLEM 2

Bu problemde sizden sabit konumda duran Manipülatör robotun kolları yardımıyla bir yükü belirli bir konumdan diğer konuma taşımanız istenmektedir. Kullanıcıdan alınacak direktifler doğrultusunda robot kolunu hareket ettirmeniz; taşınacak yükün konumunu ve yükü taşımak için geçen süreyi hesaplamanız istenmektedir.

Adım 1. 20x20 boyutlu ızgara oluşturmanız gerekmektedir, her bir ızgara 10 m x 10 m boyutlarında olduğu varsayılacaktır. (Izgaraları görselleştirmek için grafik kütüphanesi kullanmalısınız)

Adım 2. Robot konum bilgileri (x,y) kullanıcıdan alınacak ve robot ızgaraya sabitlenecektir. Robotun kolunun taşıyacağı yük miktarı kullanıcıdan alınacaktır. Yük miktarının robotun

taşıma kapasitesinin aşıp aşmadığı kontrol edilmelidir. Eğer yük miktarı aştıysa yeni yük miktarı istenmelidir.

Adım 3. Manipülatör kolunun yükü taşıyacağı yön bilgileri kullanıcıdan alınacaktır. Yön bilgisi için örneğin kullanıcı 2 ızgara ileri, 5 ızgara geri, 1 ızgara sağa, 3 ızgara sola şeklinde olabilir. Yada örneğin 3 ızgara ileri, 5 ızgara sağa şeklinde de olabilir. **Burada dikkat edilmesi gereken durum robot kolu uzunluğunun, yükün taşınacağı mesafeyi aşıp aşmadığının kontrolüdür.**

Adım 4. Tüm bu bilgiler doğrultusunda, robot kolunun taşıdığı yükün son konumu grafik kütüphanesi yardımıyla ızgara üzerinde gösterilecektir. Ayrıca başlangıç konumundan son konuma kadar geçen süre sn cinsinden hesaplatılmalıdır (**Hız m/s cinsinden olduğu varsayılacaktır**)

PROBLEM 3

Bu problemde gezgin Manipülatör robotlar için,

- Üzerine yükün konulduğu robot için başlangıç konumu kullanıcı tarafından belirlenecek, daha sonra robot yine kullanıcının yön direktifleriyle sabitleneceği noktaya kadar hareket edecektir. İlgili konumda robot hareketli durumdan sabit duruma geçecek, daha sonra yük robot koluna yüklenecektir. Robot kolu için yön bilgileri alındıktan sonra yük ilgili konuma taşınacaktır. Bu problemde robotun sabitlendiği konum ve toplam hareket için geçen toplam süreyi hesaplamamız istenmektedir.

Adım 1. 20x20 boyutlu ızgara oluşturmanız gerekmektedir, her bir ızgara 10 m x 10 m boyutlarında olduğu varsayılacaktır. (Izgaraları görselleştirmek için grafik kütüphanesi kullanmalısınız)

Adım 2. Bu ızgara üzerine engeller yerleştirilip yerleştirilmeyeceği bilgisi kullanıcıdan alınmalıdır. Eğer engel yerleştirilecekse, engel sayısı ve engellerin konum bilgileri (x,y) kullanıcıdan alınacaktır. Daha sonra engeller ızgaraya yerleştirilecektir.

Adım 3. Gezgin robotun başlangıç konum bilgisi (x,y), daha sonra robotun başlangıç konumundan itibaren sabitleneceği konuma kadar olan yön bilgileri kullanıcıdan alınacaktır. Yön bilgisi için örneğin kullanıcı 2 ızgara ileri, 5 ızgara geri, 1 ızgara sağa, 3 ızgara sola şeklinde olabilir. Yada örneğin 3 ızgara ileri, 5 ızgara sağa şeklinde de olabilir. **Burada dikkat edilmesi gereken durum robotun ızgaradan çıkıp çıkmayacağının belirlenmesidir.** Daha sonra robot sabitlenecek, yük robot koluna yüklenecektir. Robot kolunun yükü taşıyacağı yön bilgileri kullanıcıdan alınacaktır. Yön bilgisi için örneğin kullanıcı 2 ızgara ileri, 5 ızgara geri, 1 ızgara sağa, 3 ızgara sola şeklinde olabilir. Yada örneğin 3 ızgara ileri, 5 ızgara sağa şeklinde de olabilir. **Burada dikkat edilmesi gereken durum robot kolu uzunluğunun, yükün taşınacağı mesafeyi aşıp aşmadığının kontrolüdür.**

Adım 4. Tüm bu bilgiler doğrultusunda, robotun bulunacağı son konumu grafik kütüphanesi yardımıyla ızgara üzerinde gösterilecektir. Ayrıca tüm adımlar için geçen süre sn cinsinden hesaplatılmalıdır (**Süre hesaplatılırken eğer varsa engelden geçme süresi de eklenmelidir, hız m/s cinsinden olduğu varsayılacaktır**)