**Technická univerzita v Košiciach**

Fakulta elektrotechniky a informatiky

*Katedra kybernetiky a umelej inteligencie*

*Zadanie z Humanoidných technológii*

Ovládanie robota Nao pomocou gest

Meno a priezvisko: Ján Cabadaj

Damián Sedlák

Dominik Dujčák

Odbor: Inteligentne systémy

Ročník: 4.

Akademicky rok: 2017/2018

**Obsah**

[1. Zadanie 3](#_Toc499204220)

[2. Softvérová príručka 3](#_Toc499204221)

[2.1. Kinect 3](#_Toc499204222)

[2.1.1. Hlavné okno 3](#_Toc499204223)

[2.1.2. Upravené triedy knižnice 3](#_Toc499204224)

[2.2. Cloud 4](#_Toc499204225)

[2.2.1. Funkcia webovej aplikácie 4](#_Toc499204226)

[2.3. Nao 4](#_Toc499204227)

[2.3.1. Funkcie pre riadenie klbov 4](#_Toc499204228)

[2.3.2. Funkcie pre riadenie robota 5](#_Toc499204229)

1. Zadanie

Vytvorte program, ktorý pomocou gest človeka, bude ovládať robota.

1. Softvérová príručka

Na zrealizovanie nášho zadania sme použili:

* Kinect + počítač,
* Cloudové prostredie Azure,
* Robot Nao.
  1. Kinect

Pomocou Kinectu rozpoznávame gestá človeka. Na rozpoznanie sme použili knižnicu Vitruvius. Knižnica má zadefinované niektoré základné gestá a obsahuje funkcie na ich rozpoznanie. V zadaní sme použili gestá knižnice, ktoré sme si upravili podľa svojich potrieb.

* + 1. Hlavné okno
* **public MainWindow()** – inicializácia, vytvorí sa spojenie s Kinectom a zaregistrujú sa funkcie, ktoré sa budú volať pri príchode snímku z Kinectu a tiež pri rozpoznaní gesta knižnicou
* **void Reader\_MultiSourceFrameArrived(...)** – funkcia volaná pri príchode snímku, vykreslí snímok v používateľskom rozhraní a tiež vykreslí skelet pri jeho rozpoznaní
* **void GestureController\_GestureRecognized(...)** – funkcia volaná pri rozpoznaní gesta, aktualizuje posledné rozpoznané gesto v rozhraní a spustí proces ktorý ho odošle na cloud
* **static async Task SendData(string data)** – zabezpečuje odoslanie stringu na cloud pomocou metódy HTTP POST
  + 1. Upravené triedy knižnice

V knižnici Vitruvius boli upravené premenné WINDOW\_SIZE, MAX\_PAUSE\_COUNT v súbore Gestures\Gesture.cs tak, aby rozpoznávanie gest fungovalo čo najpresnejšie. Hodnoty sme volili na základe experimentov.

Ďalej boli upravené niektoré z podmienok pre jednotlivé segmenty gest pri preddefinovaných gestách v priečinku Gestures\Segments. Po týchto úpravách sme dosiahli požadované správanie a nebolo potrebné definovať ďalšie gestá.

* 1. Cloud

Keďže knižnice pre prácu s Kinectom sú určené hlavne pre jazyk C# a Nao sa programuje v Pythone, museli sme vymyslieť spôsob komunikácie medzi nimi. Vytvorili sme webovú aplikáciu, ktorej účel je pri POST požiadavke uložiť prichádzajúce dáta a pri GET ich následne poskytnúť.

* + 1. Funkcia webovej aplikácie

Hlavným prvkom webovej aplikácie je súbor Data.aspx.cs. Obsahuje iba jednu funkciu, ktorá pri príchode požiadavky určí či ide o GET alebo POST a na základe toho si buď dáta do globálnej premennej uloží, alebo ich z nej vyčíta a odošle ako odpoveď.

* 1. Nao

Robot Nao vykoná príkazy človeka, rozpoznané Kinectom. Vo funkcii *main* je celé riadenie robota. Na začiatku sú definovane globálne premenne a následne je riešená komunikácia s cloudom. Ďalej je definovaná trieda *Nao*, kde sú definovane riadiace úkony a pohyby kĺbov robota.

**Globálne premenné:**

* *robot\_ip*  - IP adresa Naa,
* *data\_url*  - URL adresa cloudu,
* *ready\_string* - riadiaca hodnota z clodu,
  + 1. Funkcie pre riadenie klbov
* **StiffnessOn (self)** – zaseknutie kĺbov (aby sa mohol hýbať)
* **StiffnessOff(self)** - vypnutie motorov
* **Set(self,command)** – premenná *command* je úkon, ktorý robot vykoná. Premenná môžeme mat nasledovne hodnoty:
  + Stand – postavenie robota,
  + Sit - posadenie robota,
  + LyingBack - ľahnutie robota.
* **Move(self,kolko)** – chodza robota. V premennej *kolko* sa definuje v metroch o koľko sa ma robot Nao pohnúť.
* **MoveTo(self,kolko,y,uhol)** – chôdza robota aj do strán

**Wave(self, hand)** – kývanie rúk robota. Vo funkcii sú definovane presne pohyby kĺbov pre mávanie, keďže mávanie rúk nie je základná funkcia robota. Premena *hand* definuje, ktorá ruka zakýva. Môže nadobudnúť hodnoty:

* + left
  + Right
    1. Funkcie pre riadenie robota
* **DefineCommands(self)** – riadenie robota pomocou hodnôt z Cloudu.

def DefineCommands(self):

self.Commands = {b'JoinedHands' : self.Stop, # STOP

b'WaveRight' : self.WaveRight, # WAVE RIGHT

b'WaveLeft' : self.WaveLeft, # WAVE LEFT

b'Menu' : self.LyingBack, # LYING BACK

b'SwipeUp' : self.Stand, # STAND

b'SwipeDown' : self.Sit, # SIT

b'SwipeLeft' : self.TurnLeft, # MOVE/TURN LEFT

b'SwipeRight' : self.TurnRight, # MOVE/TURN RIGHT

b'ZoomIn' : self.MoveForward, # MOVE FORWARD

* **Stop(self)** - zastavenie chôdze,
* **Stand(self)** - postupnosť krokov pre postavenie robota,
* **Sit(self)** - postupnosť krokov pre posadenie robota,
* **LyingBack(self)** - postupnosť krokov pre ľahnutie robota,
* **WaveRight(self)** - postupnosť krokov pre mávanie pravou rukou robota,
* **WaveLeft(self)** - postupnosť krokov pre mávanie ľavou rukou robota,
* **MoveForward(self)** – rovná chôdza robota,
* **TurnLeft(self)** - postupnosť krokov pre odbočenie vľavo,
* **TurnRight(self)** - postupnosť krokov pre odbočenie vpravo,