Upravljanje modelom tenka na visinskoj mapi u 3D prostoru pomoću Wii kontrolera

***Predmet:***Sistemi Virtualne Realnosti

***Mentor:***prof. dr Dragan Ivetić

***Autor:***Ivan Perić

***Datum:***15.01.2010.

***URL projekta:***http://code.google.com/p/gimii/source/browse/#svn/trunk/TankOnAHeightmap

# Uvod

### Zadatak

Projekat upravljanja modelom tenka u 3D prostoru pomocu Wii kontrolera ima za cilj demonstraciju detektovanja i moguće upotrebe pokreta kontrolera u 3 pravca pomoću akcelerometra koji je sastavni deo kontrolera i pomoću IR bar-a koji omogućava precizniju detekciju pokreta kontrolera u nekim pravcima.

### Alati i modeli

Za razvoj interaktivne animacije su korišteni sledeći alati i modeli:

Razvojna okruženja:

* Microsoft XNA Studio 3.0 (<http://en.wikipedia.org/wiki/Microsoft_XNA>)
* Microsoft Visual Sudio 2008 (<http://en.wikipedia.org/wiki/Microsoft_Visual_Studio>)

API i biblioteka:

* Gimii WiiApi (<http://code.google.com/p/gimii/source/browse/#svn/trunk/WiiApi>)
* WiimoteLib (<http://www.codeplex.com/WiimoteLib>)

Modeli:

* FBX model tenka i 3D heightmap okruženje (<http://creators.xna.com/>)

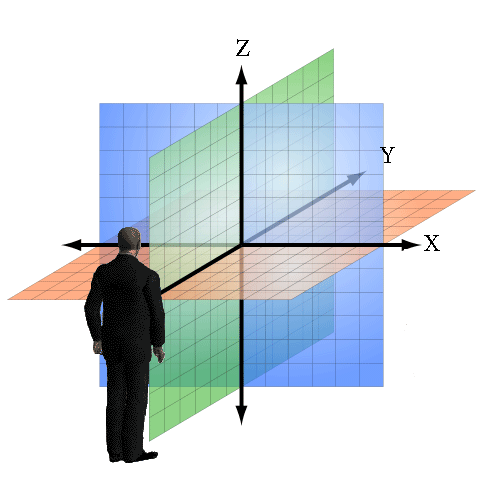
Hardware:

* Wii kontroler (<http://en.wikipedia.org/wiki/Wii_Remote>)
* IR bar – ručno pravljen sa dve IR LED diode koje se napajaju preko USB-a

### Akcelerometar i IR bar

Korišteni akcelerometar je [ADXL330](http://www.analog.com/en/prod/0,2877,ADXL330,00.html) koji je ugrađen u Wii kontroler. On je korišten da bi se detektovali pokreti u pravcu vertikalne ose (Z-ose) i jedne od horizontalnih osa (X-ose) koja su bila dovolja da bi omogućila skretanje tenka u levo i desno (X-osa) i ubrzavanje tenka unapred i kretanje unazad (Z-osa). Ručno pravljeni IR bar je korišten da bi se detektovali pokreti u pravcu druge horizontalne ose (Y-ose) koji u ovoj interaktivnoj animaciji omogućuju menjanje pogleda u odnosu na tenk. Wii kontroler ima ugrađen [PixArt](http://www.pixart.com.tw/) optički sensor koji je upotrebljen da bi se izračunavanjem promene rastojanja izmedju dve IR diode realizovala detekcija kretnje u Y-osi.

Prethodno spomenute ose, kao i kretanje Wii kotrolera po njima su prikazani na slikama 1, 2 i 3.

Slika 1 Slika 2

**Slika 3**



# Korisničko uputstvo

### Upravljanje

Pre pokretanja aplikacije potrebno je Wii kontroler postaviti u horizontalan položaj (±30˚) jer je pri tome tenk u mirovanju. Kada se kontroler nagne ka dole, tenk počinje da ubrzava do ograničene maksimalne brzine. Ukoliko se nagne ka gore, tenk se kreće ubrzano unazad do određene brzine.

Skretanje tenka jednom od dva smera (levo-desno) se obavlja naginjanjem kontrolera na stranu na koju želimo da skrenemo. Slično važi i za kretanje u nazad osim sto je logika skretanja obrnuta kao i kod vožnje automobila.

Prilikom promene pogleda, potrebno je usmeriti kontroler ka diodama pritisnuti A dugme i potom približavati ili udaljavati kontroler u zavisnosti od toga da li želimo da se približimo tenku ili udaljimo. Zbog toga je preporučljivo da diode budu vidljive kontroleru kada je on u horizontalnom položaju, tj. kada tenk miruje, da bi se zumiranje moglo obaviti tada neometano.

### Alternativni mod

Pritiskanjem dugmadi 1 i 2 na kontroleru je omogućeno menjanje moda u kome se vrši upravljanje tenkom. Drugi mod je dodat da bi se podržao Nintendo Wiimote Wheel (Slika 4) koji je dodatak namenjen igrama vožnje i trkanja.

U drugom modu skretanje se vrši intuitivnim naginjanjem volana na levo ili desno kao i kod pravog automobila, dok se ubrzavanje i kretanje u nazad izvrše prlikom naginjanja volana unapred ili unazad, respektivno.

Opcija zumiranja nije preporucljiva u ovom modu jer Nintendov dodatak prekriva optički senzor koji registruje udaljenost kontrolera od IR bar-a. U oba moda je omogućena vibracija prilikom kretanja.



**Slika 4**

# Programersko uputstvo

### Inicijalizovanje

Da bi detektovanje pokreta Wii kontrolera bilo moguće, prvo je neophodno instancirati objekat tipa WiiFabrika koji kreira objekat tipa Kontroler koji reprezentuje kotroler u kodu. Taj kod izgleda ovako:

*WiiFabrika wiiFabrika = WiiFabrike.dobaviInstancu();*

*wiiFabrika.postaviTipKontrolera(WiiTip.WII\_KONTROLER);*

*Kontroler kontroler = wiiFabrika.kreirajInstancu;*

### Detektovanje pokreta

Ukoliko je objekat koji reprezentuje Wii kontroler instanciran, potrebno je nabaviti vrednosti akseleratora instanciranog objekta tipa Kontroler i uporediti ih sa graničnim vrednostima koje su zadate u kodu. Na primer:

*if (kotroler.Stanje.Akcelerometar.X > pozitivnaGranica) {*

*okretanjeTenka -= 1;*

*} else if (kontroler.Stanje.Akcelerometar.X < negativnaGranica) {*

*okretanjeTenka += 1;*

*}*

### Vibracija

Ukoliko je tenk u stanju kretanja, potrebno je aktivirati vibraciju, a ukoliko je stao, potrebno je da se ona dektivira. To se radi na sledeći način:

*if (tenkUPokretu == true) {*

*kontroler.postaviVibrator(true);*

*} else {*

*kontroler.postaviVibrator(false);*

*}*

### Zumiranje

Ukoliko je objekat koji reprezentuje Wii kontroler instanciran, potrebno je nabaviti pozicije dveju detektovanih IR dioda i potom izračunati njihovu razdaljinu i uporediti je sa postojećom. Ukoliko trenutna razdaljina nije podešena ili je jednaka 0, to znači da je zumiranje upravo uključeno i da će novoizmerena razdaljina postati početna. Zumiranje je diskretizovano, tj. vrši se u odredjenim koracima tek kada apsolutna razlika izmedju nove razdaljine tačaka bude preko odredjene granice.

*if (kontoler.Stanje.Dugmici.A == true) {*

*if (kontroler.Stanje.Senzori[0].Nadjen && kontroler.Stanje.Senzori[1].Nadjen) {*

*if (distance < 0) {*

*distance = Math.Sqrt(Math.Pow((kontroler.Stanje.Senzori[1].x – kontroler.Stanje.Senzori[0].x), 2) + Math.Pow((kontroler.Stanje.Senzori[1].y – kontroler.Stanje.Senzori[0].y), 2));*

*} else {*

*double newDistance = Math.Sqrt(Math.Pow((kontroler.Stanje.Senzori[1].x – kontroler.Stanje.Senzori[0].x), 2) + Math.Pow((kontroler.Stanje.Senzori[1].y – kontroler.Stanje.Senzori[0].y), 2));*

*if (Math.Abs(distance - newDistance) > granicaRazmaka) {*

*rezultat = cameraDistance + (distance – newDistance)\*mnoziteljRazmaka;*

*cameraDistance = rezultat;*

*distance = newDistance;*

*}*

*}*

*}*

*}*