Univerzitet Sarajevo  
Elektrotehnički fakultet

**Razrada projekta**Predmet: Ugradbeni sistemi

Tema: Double Snake igrica

*Članovi tima:*  
***Aldin Mašović  
Ćazim Rahić***

**U nastavku slijedi plan razrada zadatka u kombinaciji sa onim što je već urađeno do vježbi 2. U razradi će se navesti prototipovi ( onakvi kakvi trenutno mislimo da trebaju biti ) kao i još neke varijable koje smatramo bitnim za pomenuti u ovoj razradi.  
Ovdje je većina funkcija direktno povezana sa crtanjem putem displej-a, tako da ćemo u razradi pokušati hronološki dati funkcije koje će se otprilike tako i u main-u pozivati, umjesto da ih dijelimo po onima za displej, za igricu i sl.**

Prije svega, kreiran je ( dakle do vjezbi 2 već ) prebrojivi tip -> enum koji sadrži tipove kretanja ( pri čemu NemaKretnje ukazuje da dzojstik „miruje“):

*enum Kretanje{Gore=0,Dolje,Desno,Lijevo,NemaKretnje};*

Također, pošto su pozicije zmija i pozicija hrane veoma često korištena stavka u programu, pametno je kreirati klasu za Poziciju sa getterima i setterima i atributima x i y ( kao alternativa moze std::pair ).

*Class Pozicija -> atr: x,y; metode: getteri i setteri  
(Aldin Mašović)*

Potrebno je zatim nacrtati zmije. Zmije će predstavljati vezane pravougaonike koji ce s crtati pomocu fillrect metode displey-a.  
Zmija predstavlja **vektor Pozicija** koji sadrzi pozicije svih njenih pravouganika, tkao da metoda za crtanje zmija će biti:

*CrtajZmiju1(vector<Pozicija>& snake);  
CrtajZmiju2(vector<Pozicija>& snake);  
(Ćazim Rahić)*

Sada je potrebno im odrediti početne pozicije na početku igrice. Bitno je napomentui da moramo odrediti i kolika će na početku biti veličina zmije (npr 3), tako da ćemo svim tim elementiam vektora (prva 3 npr) morati sett-ovati pozicije. Zmije će najvjerovatnije na početku se kretati jedna prema drugoj pa će tako i biti postavljene pozicije.

*void StartnaPozicijaSnake1(vector<Pozicija>& snake);  
void StartnaPozicijaSnake2(vector<Pozicija>& snake);*

*(Aldin Mašović)*

Sljedeće je svakako **kretanje zmije.** To dakle ovisi i od dzojstika ali i bez dzojstika zmija treba nastaviti da se krece u smjeru u kojem se do tada kretala.

Da ne pravimo dvije metode, dovoljno je da u ovoj imamo neku logicku vrijednost koaj ukazuje da li se radi o prvoj ( ili o drugoj ). Unutar ove metode ćemo ovisno od provjere različitih uslova pozivati druge metode  
( radi preglednosti koda). Pa tako npr, jedan dio ove funkcije okvirno će izgledati ovako:

*Void kretanjeZmije(vector<Pozicija>& snake, bool prva){  
 ...  
 Kretanje trenutno;  
 If(prva) trenutno = kretanjeDzojstika1();  
 else trenutno = kretanjeDzojstika2();*

*...*

*pomjeriGlavu(snake,trenutno);*

*...*

*}*

*(Ćazim Rahić)*

Dalje se ovo kretanje može koristiti za još neke funkcionalnosti. Npr, zmija se ne moze iz skroz desnog kretanja nastaviti kretati lijevo na pomjeranje dzojstika, tako da mozemo jasno uz varijablu „trenutno“ to osigurati, i sl.  
Funkcija pomjeri glavu će uzeti vrijednost koju je vratio pomjeriDzojstik1 (Ili 2) i primjeniti to na zmiji, gdje će nakon svega toga se iscrtati i rep zmije.

Sada moramo detektovati **kretanje dzojstika1 i kretanje dzojstika2.** Pošto postoji 5 varijanti iz enum Kretanje koje smo ranije spomenuli, moramo razdvojiti da ova funkcija tacno to i vraca, odnosno to će joj biti povratni tip:

*Kretanje kretanjeDzojstika1(){*

*...*

*If(x < odrVrijednost ) return Lijevo;*

*else If(x > odrVrijednost) return Desno;*

*...*

*else return NemaKretnje; //Dzojstik miruje*

*}*

*(Ćazim Rahić)*

Jasno je da cemo na isti način uraditi i za drugi dzojstik samo što će x i y se odnositi na njega. OdrVrijednost će se odrediti nakon testiranja zadatka detaljnije na vjezbama. U slučaju da se vrati da nema kretanja, zmija **nastavlja da se kreće u smjeru u kojem je do tada išla.**

Nadalje, pomenuli smo funkciju pomjeri glavu, koja nije nešto slozena, pa cemo samo napisati njen prototip:

*Void pomjeriGlavu(vector<Pozicija>& snake,Kretanje trenutno);*

*(Aldin Mašović)*

Dalje, kao i u svakoj snake igrici, jako su bitne kolizije zmija sa samim sobom, zmija sa okvirom i u ovom slucaju, medjusobno zmija.   
Pošto se samo radi o provjeri pozicija zmija, napisat ćemo opet samo prototipe funkcije, dok je implementacija na osnovu dosadasnje razrade skoro pa intuitivna:

*Bool udarilaUOkvir(vector<Pozicija>& snake);  
bool udarilaSamuSebe(vector<Pozicija>& snake);  
(Aldin Mašović)*

*bool sudarileSeGlavama(vector<Pozicija>& snake1,vector<Pozicija>& snake2);  
bool udarilaZmijaDrugu(vector<Pozicija>& snake1, vector<Pozicija>& snake2);  
(Ćazim Rahić)*

Sada, kada smo naveli funkcije koje određuju ishod, onda moramo se pobrinuti i o ishodu. Njega možemo specificirati kao enum opet:

*enum Ishod{PrvaIzgubila,DrugaIzgubila,Nerijeseno,Nista};*

Sada možemo provjeriti da li je neka zmija izgubila, pa okvirno ta metoda moze izgledati ovako:

*bool izgubila(vector<Pozicija> &snake1,vector<Pozicija> & snake2){*

*if(udarilaUOkvir(snake1) || udarilaSamaSebe(snake1) ||*

*udarilaZmijaDrugu(snake1,snake2))return true;*

*return false;*

*}*

*(Aldin Mašović)*

Na kraju možemo sve gore navedeno funkcije u kombinaciji sa ishodom ukomponovati u jednu metodu koja validira kretnju čiji dio može izgledati ovako:

*Ishod ValidirajKretnju(vector<Pozicija>& snake1, vector<Pozicija>& snake2){*

*...*

*If(izgubila(snake1,snake2) and izgubila(snake2,snake1) or sudarileSeGlavama(snake1,snake2) ) return Nerijeseno;  
else if(udarilaZmijaDrugu(snake1,snake2)) return PrvIzgubila  
...*

*return Nista;*

*}*

*(Aldin Mašović)*

Još nam ostaju funckije vezane za crtanje, brisanje hrane, odredjivanje pozicije iste i za koliziju zmije sa hranom i automatskim rastom zmije.

Hrana će predstavljati jedan pravougaonik na displeju koji će se kreirati uz pomoć funkcije rand() ( naravno koja ce se skalirati na displej ). Pa prototipovi funkcija su:

*void crtajHranu(Pozicija &h);  
void brisiHranu(Pozicija &h);  
void novaHrana(vector<Pozicija>& snake1, vector<Pozicija>& snake2);  
bool pojedenaHranu(Pozicija &prva, Pozicija &druga,Pozicija& h);  
(Ćazim Rahić)*

U planu je još i početni meni i metode za isti, koje će okvirno izgledati ovako ( moguće su promjene još ):

*Enum Meni{Start,HighScore}  
void CrtajMeni(Meni& meni);  
void PromjenaMenija(Meni& meni);  
void selektujOpcijuMenija(Meni& meni);*

*(Ćazim Rahić)*

*Score ce se vrlo lagano dobijati sljedećom metodom:*

*int score(vector<Pozicija>& snake){*

*return velicinaZmije(snake)-3; //jer tri su pocetna*

*}*

*(Aldin Mašović)*

Ovo je dakle razrada projekta i potrebnih metoda. U main-u kada se napravi će biti otprilike ovako pozivane, dok će većina metoda biti u while(true) petlji, osim onih koje iscrtavaju početne stvari.  
Moguće su promjene u nekim metodama ili čak izbacivanje nekih ( Ili dodavanje drugih metoda). Ovo je samo okvirna razrada zadatka.

*Main će skupa raditi oba člana tima.*