****

*СТШ „Миленко Брзак Уча“ Рума*

*В.Караџића 70,Рума*

Предмет: **Програмирање**

Тема матурског практичног рада:

**ЈУНИ 2017.**

**С++ - Игрица „Змијица” реализована у IDE DevCpp**

*Ученик:*

***Максимовић Владислав***

*Професор:* ***Весна Илић Маравић***

Садржај

**Кратак осврт на IDE Dev CPP** ........................................ 3

**Инсталација отвореног кода**......................................... 5

**Резервисане речи и уграђене фукнције језика С++** ... 7

**Анализа задатака и утврђивање почетних захтева** .... 10

**Креирање костура апликације** ...................................... 11

**Тестирање и завршно уобличавање** ............................. 13

**Закључак** .......................................................................... 14

**Литература** ....................................................................... 14

**Кратак осврт на IDE DevCpp**

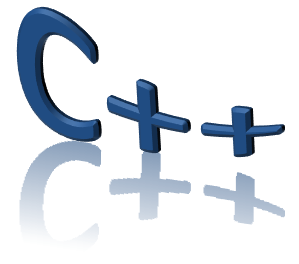
Рачунарски језици су доживели драматичан развој од првих електронских рачунара,грађених да помогну у телеметријским израчувањима током Другог светског рата.У почетку,програмери су радили са рачунарским инструкцијима ниског нивоа: машинским језиком. Те инструкције су биле представљене дугим низовима јединица и нула.Ускоро,креирани су asembleri да би заменили машинске инструкције са мнемоницима, као што су ADD и MOV, који су били лакши за рад.

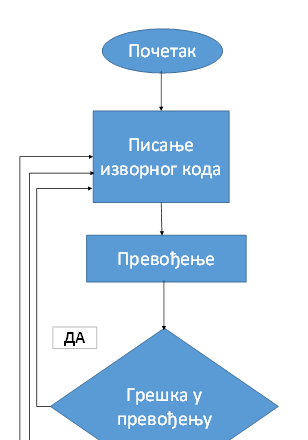
Временом појавили су се језици вишег нивоа,попут BASIC-a и COBOL-a. Ови језици су омогућили људима да раде са неким сличним речима и реченицама,попут LET I=100. Те инструкције су преведене у машински језик помоћу *интерпретатора и преводилаца. Интерпретатор* преводи програм док га чита, директно извршавајући наредбе програма. Преводилац преводи програмски код у прелазни облик. Овај корак се назива превођење и производи објектни фајл. Преводилац тада позива линкер, који од објектног фајла производи извршни програм.

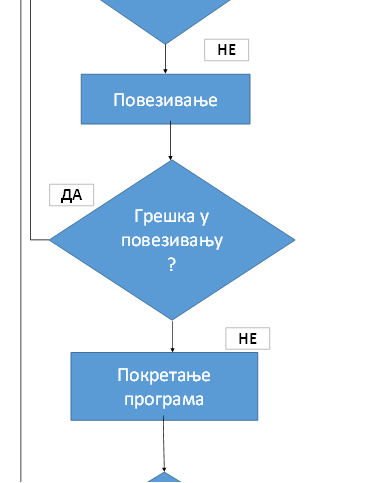
Пошто интерпетатори директно учитавају програмски код као што је написан и извршавају га на лицу места,лаки су прогамерима за рад. Преводиоци,са друге стране,уводе додатне кораке преводђенја и повезивање кода. Они производе програме, који су веома брзи сваки пут када се покрену. Међутим,сложен задатак превођења изворног кода у машински језик, који захтева пуно времена, већ је обављен.

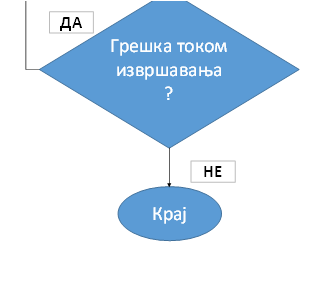
Друга предност језика који се преводе, попут C++, представља чињеница да можете дистрибуирати извршни програм и људима који немају преводиоце. Код језика који се интерпретирају морате да имате интерпретатор језика да бисте извршили програм.

Током многих година,основни циљ програмера је био да пишу кратке програме,који се брзо извршавају. Било је потребно да програм буде мали, пошто је меморија била скупа, а морао је да буде брз пошто је и рачунарска снага била скупа. Пошто рачунари постају све мањи,јефтинији и бржи и пошто цена меморије стално пада, приоритети су се променили. Данас је цена програмеског времена знатно виша од цене рачунара који се користе у пословању. Добро написан и лак за одржавање, код је данас главни циљ. Код лак за одржавање значи да, када посао то захтева, програм може бити проширен и унапређен без великих трошкова.









***Слика 1. Кораци у развоју С++ програма***

**Инсталација отвореног кода**

**(open sоurce software)**

**“Open source”**

Отворени софтвер се односи на софтвери чији је изворни код доступан унутар “open source” лиценце свим корисницима који га могу мењати,дограђивати и побољшавати његов садржај. То значи да уз „open source” програме долази и читав изворни код у неком програмском језику, што није случај са плаћеним софтвером.

Модел отверног кода и развојна сарадња конкуренције из више независних извора остварили су широк опсег дизајнерских перспектива, више него што би иједна развојна софтверска компанија самостално могла да оствари у дужем периоду.

**Инсталација**

Након упознавања са основним терминима прелазимо на инсталацију програма тј. платформе на којој сам написао игру уз помоћ литературе, професора и стекнутог знања за 3 године учења програмског језика С++.

Након преузимања инсталације програма (*заинтересовани ме могу контактирати путем мејла за додатне информације* [**daksrapsody@gmail.com**](mailto:daksrapsody@gmail.com) )

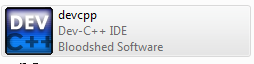
C:\Users\Vladislav\Desktop\Privremeno\Maturski\Materijal za maturski\Screenshot_4.png

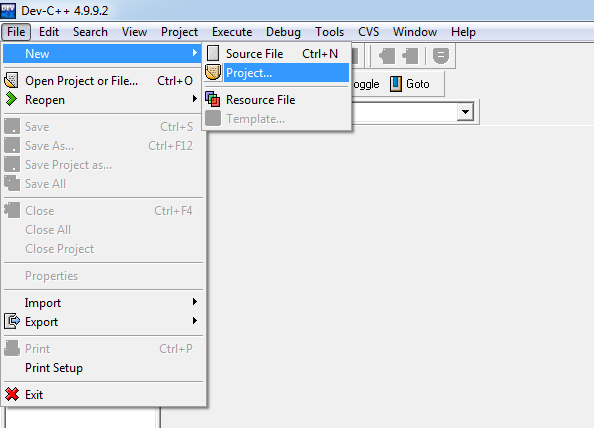
***Слика 2. Изглед инсталације***

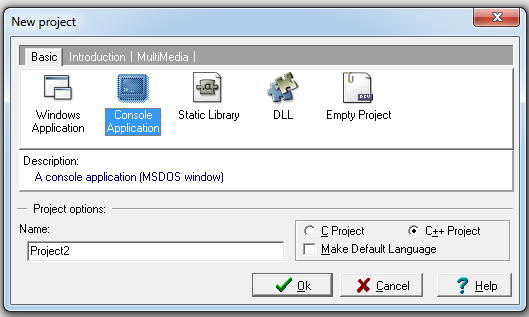
Покренемо инсталацију, изаберемо на којем језику желимо да буде наша инсталација. Када смо то изабрали пратимо кораке инсталације(*потврдимо да прихватамо услове инсталације, чекирамо да желимо цео програм да инсталирамо, и кажемо где ћемо програм сместити, сачекамо док се инсталира, и завршимо инсталацију*).

Када смо завршили са нашом инсталацијом, коначно улазимо у едитор развојног графичког окружења у којем пишем код за игру.

***Слика 3. Иконица програма***





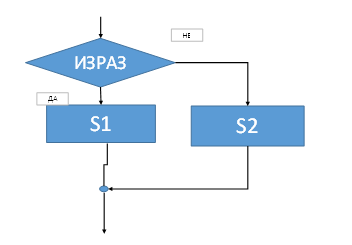


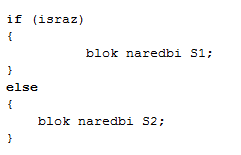
***Слика 4. Илустровано отварање новог пројекта у нашем програму***

**Резервисане речи и уграђене функције језика С++**

Након покретања програма креирамо нови пројекат. У наредном делу упознаћу Вас са неким озновама програмској језтика C++ које су коришћене у изради игре.

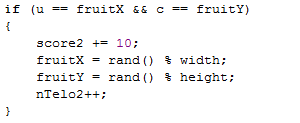
**Наредба “if-else”**





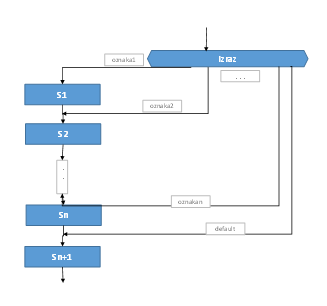
Алгоритам који илуструје управљачку структуру **if-else** могао би да се протумачи: „Уколико израз има тачну вредност ( стрелица означена са „да“), онда се извршава блок наредби S1; иначе уколико израз није тачан (стрелица означена са „не“), извршава се блок наредби S2 „

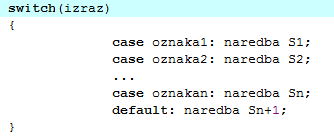
* **НАПОМЕНА**: коришћење вичитастих заграда “{ “ , “}” испред и иза блока наредби није нам неопходно уколико се он састоји из само једне наредбе !



***Пример петље у мојој игри.***

**Наредба “switch”**



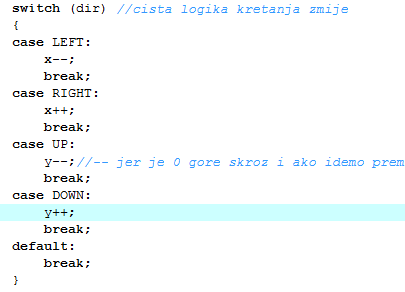
****

Зависно од вредности израза izraz , који мора бити целебројног типа(или знаковног, јер се у језику С знаци представљају као целобројни подаци), овом наредбом врши се преношење управљања на једну од више наредби S1, S2…, Sn, Sn+1. Ове групе наредби у наредби switch не стављају се између заграда { и }. Поједине од ових група наредби могу бити и празне,ако се за више различитих вредности израза izraz треба скочити на исту наредбу

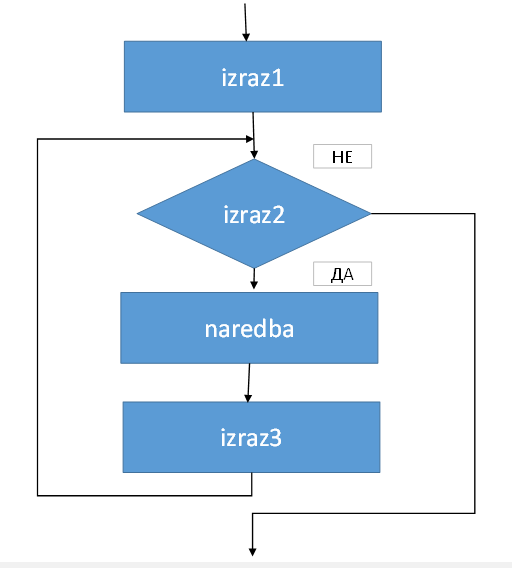
Ознаке(лабеле) oznaka1,oznaka2,…oznakan које се стављају после службене речи case и завршавају се знакм „:“ морају бити константе или константни изрази целебројног типа.

Садржај овог приказа можемо тумачити корак по корак на следећи начин:

* Испитујемо вредност израза izraz;
* Уколико је вредност тог израза једнака вредности oznaka1, онда се програм извршава од наредбе naredba1;уколико није,испитује се да ли је израз једнак вредности oznaka2;
* Уколико је вредност izraz једнака вредности oznaka2, извршава се naredba2; уколико није,испитује се да ли је izraz једнак вредности oznaka3 и процес се понавља онолико пута колико пута се понавља кључна реч case;
* Уколико ни једна од вредности oznaka1, oznaka2, ..није једнака вредности izraz, онда програм прелази на ред са ознаком default.
* **НАПОМЕНА:** Ознака(лабела) default је опциона(дакле, није обавезна) и уколико она не постоји, ток програма наставља извршавањем прве наредбе иза наредбе switch

****

***Пример друге петље.***

**C:\Users\Vladislav\Desktop\Privremeno\Maturski\Materijal za maturski\Screenshot_12.pngНаредба “for”**

Први део наредбе (izraz1) представља припрему за улазак у петљу. Овде се постављају почетне вредности. Уколико је то потребно,иницијализација може да садржи и више операција за постављање почетних вредности које се међусобно одвајају зарезом

Други део (izraz2) је исптивање (тестирање) вредности бројача(најчешће). Уколико је достигнута одређена вредност, излази се из петље. Значи, тело петље (naredba) извршаваће се онолико пута колико је потребно да бројач (или неки други парамтери) достигне тачно одређену вредност.

Трећи део (izraz3) је операција измене вредности бројача и она се врши на крају сваког проласка кроз петљу. Практилно, у овом делу се врше припреме за извршавање наредног проласка кроз петљу

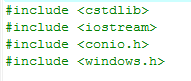
C:\Users\Vladislav\Desktop\Privremeno\Maturski\Materijal za maturski\Screenshot_22.png

**Анализа задатака и утврђивање почетних захтева**

Сад када сте упознати са неким основама и највећим делом програма тј. игре коју пишем, упознајем Вас са анализом задатка , тј. како сам утврђивао почетне захтеве, и шта и како сам требао да урадим из почетног погледа и размишљања.

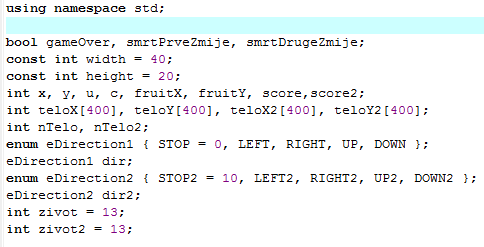
1. Направити мени игре
2. Направити оквир игре
3. Направити главу једне змије и њено тело
4. Направити главу друге змије и њено тело
5. Направити да се змије крећу
6. Направити ако се координате змије и воћкице потрефе да се повећа скор и добије један део тела
7. Направити више нивоа
8. Један ниво за две змије које се такмиче по скору само
9. Један ниво за две змије које се такмиче по скору и по животима
10. Један ниво за две змије које се такмиче по скору и по животима(и да губе живот додиривајући оквир)
11. Направити да у случају гублљења живота дође до краја игре,да се игра паузира,и да се види скор

**Креирање костура апликације**

Библиотеке које сам искористио су следеће: где су прве две по самом улажењу у програм већ дефинисане како би могли да корисимо неке команде за испис, упис и слично.

Трећа библиотека нам је омогућила да преко тастатуре крећемо се по пољима тј. да наш програм региструје притиснуте тастере на тастатури.

Четврта библиотека нам је омогућила да манипулишемо брзином кретања наших змијица.



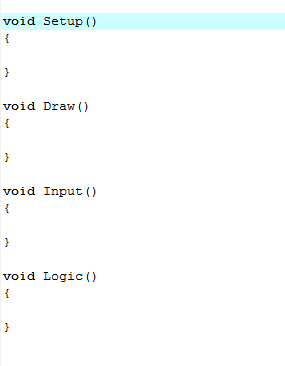
Иизвршио сам почетну иницијализацију (слика је мењана, променама у коду), користио сам неке резервисане речи како би дефинисао неке променљиве што се може видети са слике.

На пример Boolean сам користио како бих дефинисао трајање игре,трајање живота змија, const да би поставио константу на дужину и висину оквира игре, int да би дефинисао целобројне вредности главе једне и друге змије као и координате воћкице и живота и једне и друге змије.

Такође сам ограничио тело змије преко низа,онда сам дефинисао кретање тј. одлучио да неким променљивима дефинишем кретања лево, десно, горе, доле и мировање.

И на крају сам дефинисао да и једна и друга змија имају по 13 живота.

После тога сам направио 4 велике функције које ћу(сам) попуњавати и допуњавати временом и оне ће(су) чинити стуб моје апликације.



Ово су 4 функције моје апликације које представљају стуб апликације, које сам попуњавао с разним фукнцијама и петљама како би моју апликацију употпунио и завршио.

Касније сам позивао те функције који су типа void односно не враћају ништа али се у њима кад се међусобно споје извршава игрица.

**Тестирање и завршно уобличавање**

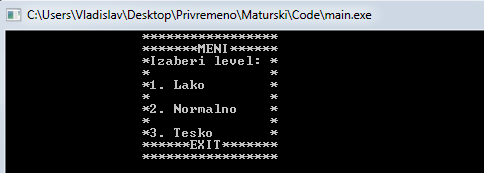
**Уобличавање**

Ни један програм не може да се заврши без грешака, пропуста и неких грешака, ни једна прва верзија не може да буде крајња, тако и моја апликација није.

Прошло је много тестирања и исправљања грешака, увођења нових детаља, мењање чак и окружења у ком сам радио.

Све моје пропусте и грешке желим да задржим за себе, зато што сваки успех кад се презентује презентује се само најбоље а зној и труд остаје за појединца.

**Тестирање**





**Закључак**

Пре свега желео бих да се захвалим професорима и школи на пруженој прилици да радимо са колико толико савременом опремом у свету рачунараства тј. IT sektoru.Велике заслуге за рад и мотивацију желео бих да посветим професорки Весни Илић, која би за ђаке који су желели да напредују и да уче излазила у сусрет како професионално тако и другарски.

У вези мог рада,волео бих да овај рад буде само основа за будуће генерације који ће похађати нашу школу и желети да уче програмирање, како у школи тако и индивидуално. Волео бих да већ можда и следеће године уз помоћ професорке Весне Илић мој рад буде реализован у неким визуалним техникама, тј. да се игра реализује и графички, а да овај рад буде само основа за то. Због тога сам остављао коментаре током целог кода, како би неко ко је тек почетник могао да релативно схвати шта се ту дешава.

Све у свему, надам се да ће овај рад некоме помоћи. За сва питања у вези кода или рада, можете да ме контактирате на мејл [**daksrapsody@gmail.com**](mailto:daksrapsody@gmail.com)

**Литература**

Литературу коју сам користио за овај рад:

1. “Naučite za 21 dan C++“ Jesse Liberty –prevod sedmog izdanja
2. Разни YouTube клипови (делови записани у свеску а линкови изгубљени)
3. Википедијa