

Gymnázium Antona Bernoláka
Námestovo, Mieru 307/23

Záverečná Práca z Informatiky

Hra „HAD“

2010
Zákamenné

Autor:
Matej Ridzoň
Trieda: VIII.OB

Čestné prehlásenie:

Čestne prehlasujem, že túto prácu som robil sám v súlade s etickými normami.

Podpis:

Obsah

0 Úvod	4
1 Hra HAD	5
1.1 Poblematika	5
1.2 Špecifikácia Prostredia premenných a súborov	5
1.3 Priebeh hry a zoznam procedúr a funkcií	6
1.4 Menu hry	6
1.5 sekundárne procedúry	7
1.6 Vykresľovanie hada a samotný pohyb hada	7
1.7 generovanie jedla a koniec hry	8
1.8 unit pohyb	9
2 Šírenie hry	10
2.1 Publikovanie hry	10
3 Zoznam použitej literatúry	11
4 Prílohy	12

Úvod

Prečo som chcel spraviť tento projekt? Hry ma už od dávna oslovovali, najmä logické a simulácie. Samozrejme nechýbal dôvtip pre strategické myslenie. A práve hra Had ma oslovila svojou jednoduchosťou, ale postupom hry už aj náročnosťou, kedy si treba všímať ako Had ide.

Mojím cieľom bolo teda vytvoriť hrateľnú verziu hry, avšak zdrojový kód som vyvíjal tiež s takou precíznosťou, aby sa dal čo najľahšie pochopiť a aby sa dalo v tomto projekte pokračovať. Samozrejme projekt sa dá nájsť aj na webe <http://snakegameinpure.sourceforge.net/> kde som projekt umiestnil pod licenciou GNU-GPL, aby sa dalo v ňom s prehľadom pokračovať.

Ďalšou podmienkou bola snaha vytvoriť kód ľahko použiteľný a upraviteľný do objektov a potom následné používanie objektov aj pre implementáciu ďalších možností. Ako napríklad umelá inteligencia (AI), online aj offline multiplayerová hra.

1 Hra HAD

1.1 Problematika

Hlavným problémom, alebo lepšie povedané výzvou bolo zosúladenie všetkých procedúr prvkov a rôznych drobností, ako aj kódu do jedného prehľadného celku. Našťastie mi v tom trochu pomohlo OOP. No v prvom prípade bolo treba vyriešiť posun hada a jeho zmenu smeru. Ďalšia otázka vychádzla z toho, že ako budú rôzne časti tela hada vyzerat', potom aj celkový vzhľad hry. Ako má vyzerat' menu. Pri postupnom tvorení vzišla aj otázka zadávania textu do grafu (jeho načítavanie) a efektné zobrazenie.

Nakoniec zohrala vo všetkom rola efektivity, najmä pri vykresľovaní do grafu. S ostatnými drobnosťami mi pomohol tiež neskôr spravený AI simulačný systém.

Celý princíp programu preto rozdelím na menšie časti (procedúry) zaradom tak ako sú v programe a nakoniec vysvetlím celý beh programu. Keďže vývojový diagram by bolo pre túto situáciu kresliť obťažné, rozhodol som sa pre slovný postup a takéto objasnenie problému.

1.2 Špecifikácia Prostredia premenných a súborov

V kóde programu sú použité základné knižnica balíka rtl prekladača fpc. Ich názvy sú „graph“ a jej doplnok „winCRT“ na podporu sekundárnych funkcií ako načítavanie z klávesnice.

Na riadkoch 1-39 v prílohe na DVD súbor „a.webarchive“ sú zobrazené farebne rozlíšene direktívy a rôzne typy a premenné definované v programe. Riadky 5-16 udávajú rôzne konštanty použité v hre, ako názov Hry, položky v menu, direktívy hada a pod. Všetky sú vyčlenené von, pre jednoduchú úpravu.

```
5. const P=1; H=2; L=3; D=4;           //definicia smerov (P - pravá, LP - Ľavá-  
    Pravá)  
6.    LD=5; LP=6; LH=7; DP=8; DH=9; PH=10;  
7.    DL=5; PL=6; HL=7; PD=8; HD=9; HP=10;  
8.    NazovHry='Hra HAD :)';  
9.    Menu1='ZACNI HRU!'; Menu2='NASTAVENIA'; Menu3='POMOC'; Menu4='NAJLEPSIE  
    VYSLEDKY'; Menu5='KONIEC';  
    ...
```

Na riadkoch 18-27 sú definované typy pre premenné, hlavne pre tie, ktoré sa používajú ako externé dáta. To znamená nastavenia hry a najlepšie skóre. Všetky sú uložené do typového súboru.

Nakoniec riadky 29-39 udávajú rôzne premenné potrebné pre hru.

```
var gd, gm: smallint;                //graf  
29. StartX, StartY, KoniecX, KoniecY: longint;           //rozmery pola  
30. CurX, CurY, LastX, LastY, CurSmer, LastSmer: longint; //smer hada 1  
31. Atyp, skore, priznak, rychlost, Arychlost, vyber, Hdlzka: longint;  
32. MenuX1, MenuY1, MenuX2, MenuY2: longint;           //pozicia menu  
33. c, c2: char;                                       //nacistany klaves
```

1.3 Priebeh hry a zoznam procedúr a funkcií

Samotný priebeh hry HAD a jeho inicializácia začína na riadkoch 980-1190 v zdrojovom kóde. Opíšem tu niektoré zaujímavé udalosti.

```
980.begin
981.Randomize;
982.detectgraph(gd, gm);
983.gm:=m800x600;           //nastavenie rozlisenia na 800x600
984.initgraph(gd, gm, '');
985.inicializuj;
```

Riadok 983 je dôležitý, hlavne pre grafické spracovanie hry, kde sa nastavuje veľkosť grafického okna v akom bude hra spustená. Teda ako vidno z kódu, hra ide pod rozlíšením 800x600 pixelov. Ďalej riadok 985 má v sebe procedúru inicializuj. Táto procedúra je v príloha a.webarchive, prípadne v zdrojovom kóde na riadkoch 347-368. Táto procedúra vlastne načíta všetky údaje zo vstupného súboru, v prípade žeby súbor neexistoval, vytvorí ho so stanovenými hodnotami.

1.4 Menu hry

```
986.3:
987.repeat
988.menu;
989.until vyber=-1;
```

Na pozícií 986 je návestie 3. Toto návestie je používané, keď skončí hra a teda aby sa znova zobrazilo celé menu. Funkcia menu (riadky 781-955) vykreslí menu a celé grafické rozhranie hry. Taktiež pripraví hru k štartu, ak sa vyberie v ponuke štart hry. Premenná vyber udáva výber možnosti v menu. Ak je vybraná prvá možnosť (štart hry) Tak sa cyklus skončí a pokračuje sa ďalej.



-obrázok menu hry. Toto uvidíte, keď spustíte hru.

```
785.ohranic;
786.vyciernimenu;
787.priznak:=0;
788.skore:=0;
789.Arychlost:=round(Srychlost*XnaY(Qrychlost, (subor.rychlost-1)));
```

Riadok 785 volá procedúru ohranic (v kóde na riadkoch 370-378), ktorá vlastne vytvorí jednoduchý šedý rám. Procedúra vyciernimenu vykreslí do šedého rámu čierny obdĺžnik, čím

vymaže všetky nežiadúce prvky. Riadok 789 nastavuje aktuálnu rýchlosť hada podľa vybranej úrovne. Keďže v nastaveniach môže byť rýchlosť 1-20, kde 20 je najrýchlejšia, musel som použiť Geometrickú postupnosť na úpravu premennej, ktorá sa bude používať v procedúre Delay.

Na riadkoch 790-809 je vložená implementácia výplne AI poľa. O tom bude napísané ďalej.

Na ďalších riadkoch pokračuje vykresľovanie položiek menu a potom ich výber. Podľa výberu sa zavolá ďalšia procedúra. Ak je výber 1, teda Štart Hry, ukončí sa cyklus v procedúre, kde sa čaká na vstup a pokračuje sa koncom procedúry, kde je inicializované vykreslenie hracieho poľa a vykreslenie Hada.

V prípade vybratia možnosti 2 sa zavolá procedúra nastavenia. 3 je pomoc, 4 highskóre a 5 ukončí vykonávanie programu príkazom Halt.

1.5 sekundárne procedúry

Ako sekundárne procedúry môžem označiť tie, ktoré sú v používanie v menu. To je: nastavenia, uprav, pomoc, highscores, zapisiskore. Všetky tieto procedúry majú podobnú funkciu ako menu. Líšia sa len v drobnostiach.

Procedúra nastavenia – v nej sa upravujú parametre hry ako šírka a výška poľa, rýchlosť a úroveň AI. Pre úpravu používa funkciu uprav, ktorá načítava len určité znaky a podľa toho vyhodnocuje, či znak môže byť zapísaný, alebo nie. Pri každom novom znaku upraví text a vykreslí ho do grafu. Tak sa dosiahne efekt postupného písania. Procedúra je v prílohe na riadkoch 508-557.

Procedúra pomoc – táto procedúra len zobrazí všetky možnosti hry. V prílohe na riadkoch 481-506.

Procedúra highscores – táto procedúra má podobnú funkciu ako pomoc, no vypíše prvých 16 najlepších dosiahnutých skóre. Zdrojový kód je v prílohe v riadkoch 451-479.

Procedúra zapisiskore – táto procedúra zistí, či aktuálne skóre hráča môže byť zapísané v tabuľke. Ak áno počká na zadanie mena a vsunie ho do tabuľky s najvyšším skóre. Tabuľku tiež utriedi. Na triedenie používa algoritmus Maxsort. V prílohe na riadkoch 559-625.

1.6 Vykresľovanie hada a samotný pohyb hada

Na riadkoch 990-1129 v prílohe a.webarchive, prípadne v zdrojovom kóde hry Je umiestnený hlavný cyklus hry – pohyb hada. Tento cyklus používa procedúry ako upravhada, jedlo, zobazskore a pauza.

Na začiatku cyklu je vždy načítanie smeru (šípky). Podľa toho, aká bola posledná stlačená sa nastaví smer hada, teda podľa toho sa vyhodnotí príkaz case.

```
1002. 72:begin                                     //hore
1003.   case GetPixel(CurX, CurY-7) of
1004.     black:atyp:=0;
1005.     green:zjedol;
1006.     lightgray:atyp:=2;
1007.     white:Ozjedol(H);
1008.   end;
1009.   CurSmer:=H;
1010.   if not (ord(c2)=80) then
1011.     begin
1012.       upravhada(atyp, LastX, LastY, LastSmer, CurX, CurY, CurSmer);
1013.       c2:=c;
1014.       CurY:=CurY-7;
1015.     end
```

```

1016. else
1017. begin
1018. c:=c2;
1019. goto 2;
1020. end;
1021. end;

```

Na riadku 1003 je zistenie čo sa na danej pozícii nachádza. Ak je tam čierno – had sa môže pohnúť. Ak zeleno – znamená to, že tam bolo jedlo ak sivá – had narazil a ak biela, zisti sa, či had nenarazil do seba. Podľa týchto údajov sa zavolá procedúra upravhada, ktorá upraví buď aj dĺžku hada, alebo ho vyplní dočervena ak narazil, alebo ho prípadne len posunie. Procedúra sa nachádza v prílohe a.webarchive, alebo v zdrojovom kóde na riadkoch 222-345. Procedúra volá procedúry zistismer a vymazblok.

Procedúra zistismer ma za úlohu zistiť aký bol posledný smer hada. Rozpísaná je v prílohe a.webarchive, alebo zdrojovom kóde na riadkoch 58-88.

Procedúra vymazblok vymaže blok hada (časť tela) tak, že ho prekryje čiernym štvorcem veľkosti 7x7.

Nakoniec podľa zistení smerov a typu posunu hada sa upraví had. Procedúra volá teda volá procedúru telohada, kde sú



-Takto vyzerá priebeh hry, ihrisko, had v ňom, jedlo a skóre

zapísané postupy vykresľovania špecifických blokov hada. Procedúra je na riadkoch v prílohe a.webarchive, alebo v zdrojovom kóde na riadkoch 115-220.

1.7 generovanie jedla a koniec hry

V každej úprave, ak bolo zjedené jedlo, zavolá sa procedúra jedlo v prílohe a.webarchive, alebo v zdrojovom kóde na riadkoch 90-113, ktorá buď vygeneruje jedlo, alebo ak had zaplnil celú plochu, nastaví príznak ukončenia na 4.

Na riadkoch v 1130-1190 v prílohe a.webarchive, alebo zdrojovom kóde, sú zobrazené všetky možnosti narazenia.

```

1130. case priznak of
1131. 1:begin //zrusenie Esc-om
1132. setcolor(yellow);
1133. setfillstyle(solidfill,yellow);
1134. SetTextStyle(DefaultFont,HorizDir,1);
1135. SetTextJustify(CenterText,TopText);
1136. if s='' then s:='0';
1137. OuttextXY(((subor.KoniecX-StartX) div 2)+StartX,StartY+10,Zrus1);
1138. OuttextXY(((subor.KoniecX-StartX) div 2)+StartX,StartY+24,Vypisskore+s);

```



```

1139. if not zapissskore(((subor.KoniecX-
      StartX) div 2)+StartX,StartY+36,skore) then Delay(1000);
1140. setcolor(black);
1141. OuttextXY(((subor.KoniecX-StartX) div 2)+StartX,StartY+10,Zrus1);
1142. OuttextXY(((subor.KoniecX-StartX) div 2)+StartX,StartY+24,Vypisskore+s);
1143. goto 3;
1144. end;

```

Tieto časti programu vypíšu typ ukončenia (dôvod konca hry) a zistia, či je možné zapísať highskóre. Potom sa presunú do návestia 3.

1.8 unit pohyb

Tento unit v sebe implementuje jednoduchý pohyb hada. Používa na svoju simuláciu pole, v ktorom sú určené smery pohybu hada. Nie je zatiaľ úplne dokončený, ale funguje pre párnú dĺžku poľa (20,22,24,...). Táto unit je vzorom jednoduchého AI. Je v prílohe b.webarchive, alebo v zdrojovom kóde pohyb.

2 Šírenie hry

2.1 Publikovanie hry

Hra, ako aj zdrojový kód je umiestnená na webe <http://snakegameinpure.sourceforge.net> .
Vzťahuje sa na ňu licencia GNU-GPL, teda je voľne šíriteľná.

3 Zoznam použitej literatúry

<http://wiki.freepascal.org>

4 Prílohy