

Report of Pokémon Game Development

(포켓몬 게임 설계 프로젝트 보고서)

© 2019 Sang Won Kang

과목: [CSE2035-01] C프로그래밍

담당교수: 서강대학교 컴퓨터공학과 김 지 환

개발자: 강상원 (20191559)

- **프로젝트 수행 환경**

Visual Studio 2017

- **프로젝트 개요**

- I. **설계 문제 및 목표**

본 프로젝트에서는 C언어를 활용하여 텍스트 형식의 ‘포켓몬’ 게임을 구현한다.

- II. **구현 사항**

1. **게임 시작:** 1을 입력하면 ‘새로 시작’, 2를 입력하면 프로그램이 종료된다.

- 사용자의 이름을 입력하고, 스타팅 포켓몬 3마리 중 1마리를 택한다.

- 스타팅 포켓몬 선택 후에 리포트 기능(2.5)을 수행한다.

2. **메뉴 선택**



- 2.1. **배틀:** 사용자가 지닌 포켓몬의 레벨에 따라 랜덤으로 야생 포켓몬이 등장한다.

- 등장한 포켓몬을 잡거나, 이기거나, 도망친다.

- 만약 패배한다면 다음 포켓몬을 내보내고, 다음 포켓몬이 없다면 도망친다.

- 2.2. **도감:** 현재 도감 상태를 출력한다. (발견되지 않음/발견됨/잡음)

- 2.3. **지닌 포켓몬:** 사용자가 지닌 포켓몬들을 보여준다.

- 지닌 포켓몬들의 상태 (Type, HP, 공격, 방어, 특수공격, 특수방어, 스피드)를 본다.

- 지닌 포켓몬들의 순서를 바꾼다.

- 2.4 **포켓몬 센터:** 사용자가 지닌 포켓몬의 HP를 모두 회복한다.

- 2.5 **리포트:** 트레이너 명과 지닌 포켓몬 (타입)들을 콘솔/txt 파일로 출력한다.

2.6 박스: 사용자가 지닌 포켓몬 중 선택한 포켓몬을 박스로 이동시킨다.

-박스에서 포켓몬을 꺼내거나, 박스에 포켓몬을 넣을 수 있다.

2.7 종료: 게임을 종료한다.

- **데이터 구조**

-포켓몬들의 진화 과정을 저장하기 위해 Linked List를 이용하였다.

-이중 포인터형 **head를 이용하여 전체 포켓몬의 진화 계통도를 참조할 수 있게 하였다.

-Structure(구조체)를 이용하여 포켓몬의 기본 정보값을 저장한다.

-Nested Structure를 이용해 각 포켓몬의 Stat값을 포켓몬 정보값 Structure(구조체)에 변수로 저장한다.

-사용자 정보를 담고 있는 Structure(구조체)는 동적 할당으로 최대 6마리의 포켓몬 정보를 저장할 수 있도록 메모리를 할당한다.

-Box(박스)는 동적 할당으로 포켓몬 저장 공간을 할당하였다.

POKE *dic를 이용해 포켓몬의 데이터를 진화 계통별로가 아닌 단순 나열식으로도 한번 더 저장하였다.

- **추가 구현 사항**

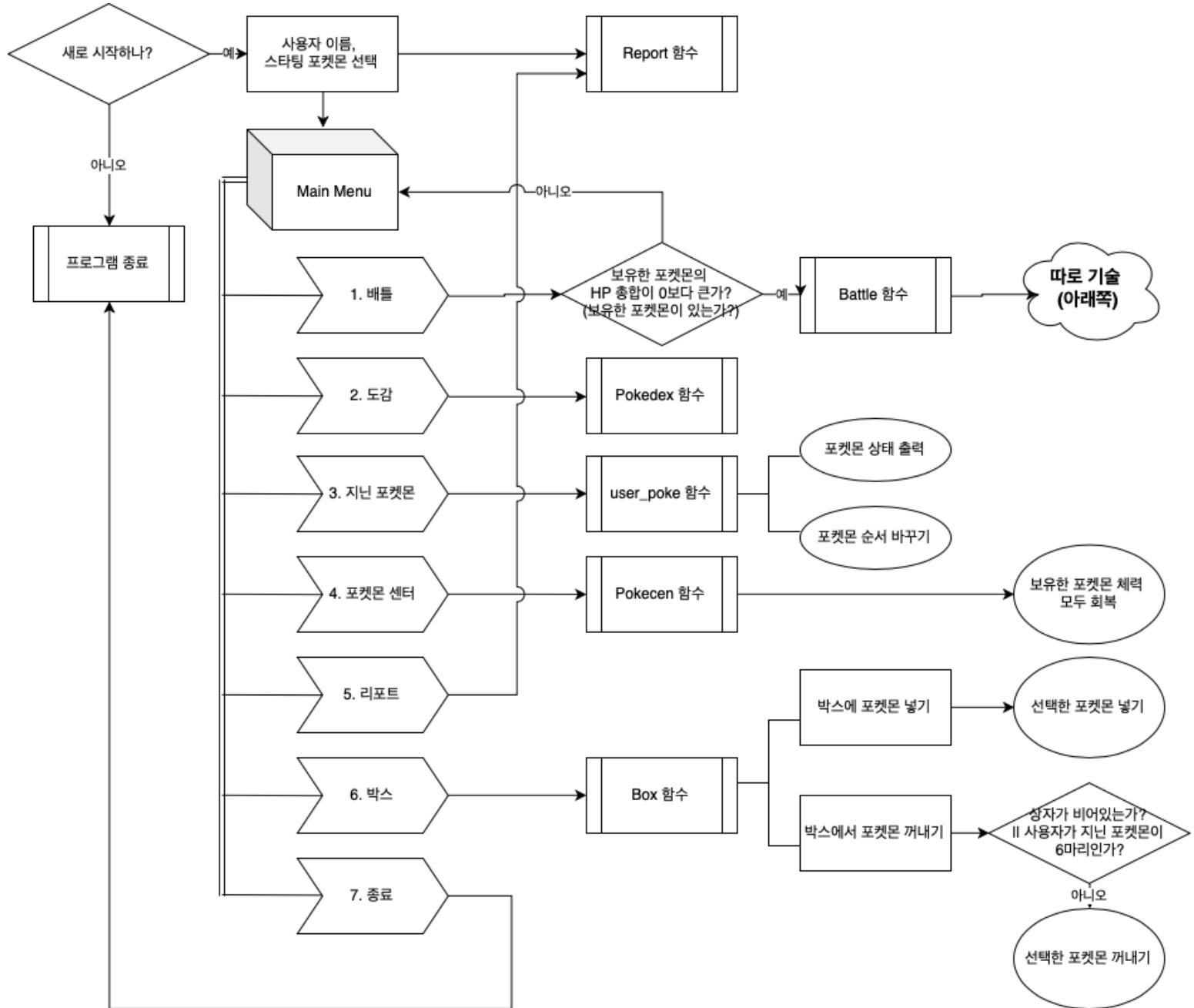
1. 포켓몬 센터

포켓몬 센터에서 사용자가 지닌 포켓몬의 HP를 모두 회복한다.

2. 박스

포켓몬을 최대 6마리까지밖에 못 지니고 다니는 사용자를 위해 포켓몬을 저장, 교환할 수 있는 박스를 만든다.

● 순서도



다음 쪽에 Battle 함수 상세 흐름도 계속….

<Battle 함수의 순서도>

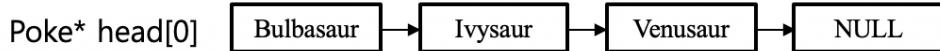


- 기능 및 함수 설명

- Insert 함수

sogang_poke.txt 파일에서 포켓몬 정보를 읽어 와서 head[i]의 끝에 노드를 추가한다.

가령 for(int i=0; i<3; i++) Insert(&head[0]); 이라 한다면 텍스트 파일의 처음 세 줄을 head[0]의 노드로 저장한다.

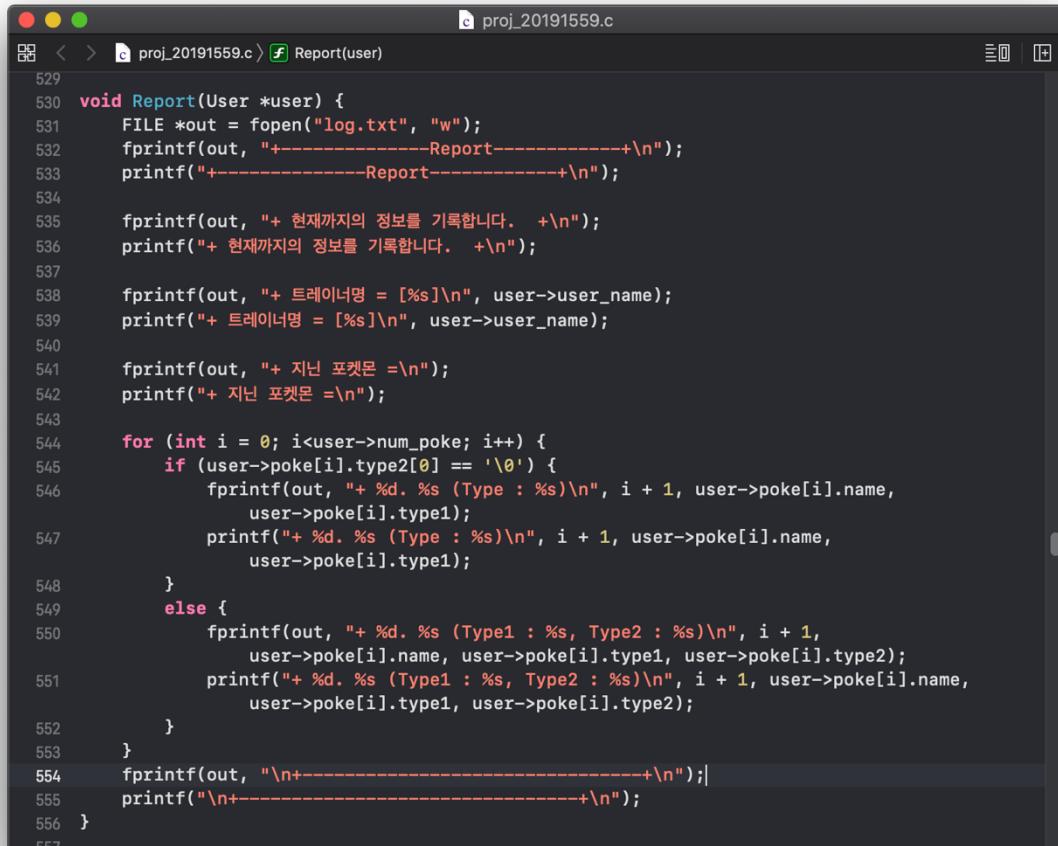


Type0이 하나만 있는 포켓몬과 두 개 있는 포켓몬을 구별하기 위해 ',' 의 index를 센 후 type2가 있는 경우 strcpy함수를 이용해 집어넣어준다.

```
745
746 void Insert(POKE** head) {
747     int len = 0;
748     POKEMON* tmp = (POKE*)malloc(sizeof(POKE));
749
750     fscanf(in2, "%d%s%d%d%d%d\n", &tmp->dex_num, tmp->name, tmp->type1,
751             &tmp->stat.HP, &tmp->stat.Attack, &tmp->stat.Defense, &tmp->stat.Contact,
752             &tmp->stat.Speed);
753     tmp->link = NULL;
754     if (*head == NULL) *head = tmp;
755     else {
756         POKEMON* idx = (POKE*)malloc(sizeof(POKE));
757         idx = head[0];
758
759         while (1) {
760             if (idx->link == NULL) break;
761             idx = idx->link;
762         }
763         idx->link = tmp;
764     }
765
766
767
768     for (int j = 0; j<30; j++) {
769         len++;
770         if (tmp->type1[j] == ',')
771             break;
772     }
773     tmp->type1[len - 1] = '\0';
774     tmp->type2[0] = '\0';
775     if (len>0)
776         strcpy(&tmp->type2[0], &tmp->type1[len]);
777 }
```

➤ Report 함수

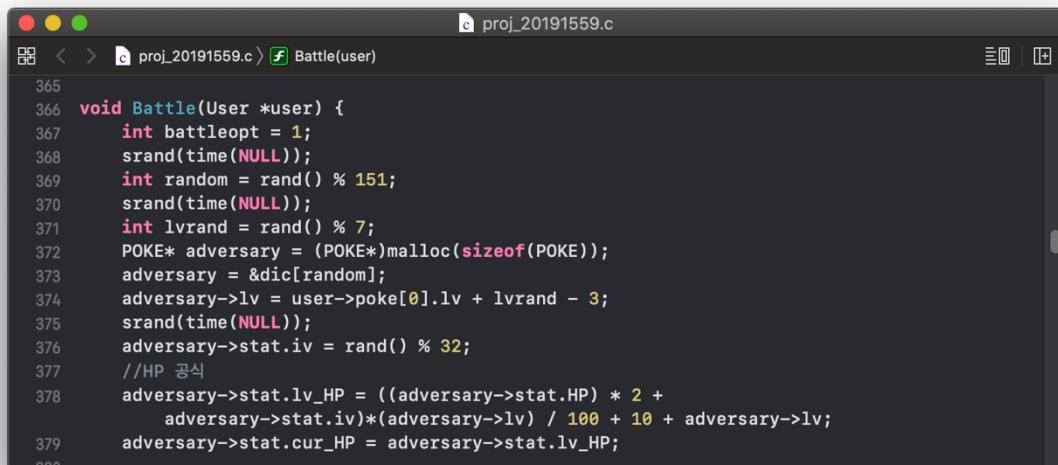
트레이너명과 보유한 포켓몬 (타입)을 콘솔과 log.txt에 출력한다.



```
proj_20191559.c Report(user)
529
530 void Report(User *user) {
531     FILE *out = fopen("log.txt", "w");
532     fprintf(out, "+-----Report-----+\n");
533     printf("+-----Report-----+\n");
534
535     fprintf(out, "+ 현재까지의 정보를 기록합니다. +\n");
536     printf("+ 현재까지의 정보를 기록합니다. +\n");
537
538     fprintf(out, "+ 트레이너명 = [%s]\n", user->user_name);
539     printf("+ 트레이너명 = [%s]\n", user->user_name);
540
541     fprintf(out, "+ 지난 포켓몬 =\n");
542     printf("+ 지난 포켓몬 =\n");
543
544     for (int i = 0; i<user->num_poke; i++) {
545         if (user->poke[i].type2[0] == '\0') {
546             fprintf(out, "+ %d. %s (Type : %s)\n", i + 1, user->poke[i].name,
547                     user->poke[i].type1);
548             printf("+ %d. %s (Type : %s)\n", i + 1, user->poke[i].name,
549                     user->poke[i].type1);
550         } else {
551             fprintf(out, "+ %d. %s (Type1 : %s, Type2 : %s)\n", i + 1,
552                     user->poke[i].name, user->poke[i].type1, user->poke[i].type2);
553             printf("+ %d. %s (Type1 : %s, Type2 : %s)\n", i + 1, user->poke[i].name,
554                     user->poke[i].type1, user->poke[i].type2);
555         }
556     }
557     fprintf(out, "\n+-----+\n");
558     printf("\n+-----+\n");
559 }
```

➤ Battle 함수

사용자가 선택한 배틀 옵션에 따라 수행한다. 자세한 구동 방식은 앞선 Battle 함수의 순서도에 나와 있다.



```
proj_20191559.c Battle(user)
365
366 void Battle(User *user) {
367     int battleopt = 1;
368     srand(time(NULL));
369     int random = rand() % 151;
370     srand(time(NULL));
371     int lvrand = rand() % 7;
372     POKEx adversary = (POKEx)malloc(sizeof(POKE));
373     adversary = &dic[random];
374     adversary->lv = user->poke[0].lv + lvrand - 3;
375     srand(time(NULL));
376     adversary->stat.iv = rand() % 32;
377     //HP 공식
378     adversary->stat.lv_HP = ((adversary->stat.HP) * 2 +
379                             adversary->stat.iv)*(adversary->lv) / 100 + 10 + adversary->lv;
380     adversary->stat.cur_HP = adversary->stat.lv_HP;
```

```
proj_20191559.c
```

```
381     printf("\n+-----Battle-----+\n");
382     printf("+ 앗! 야생의 %s(Lv.%d)가 나타났다!\n", adversary->name, adversary->lv);
383     dic[adversary->dex_num - 1].dex_state = 1;
384
385     while (1) {
386         printf("+ 무엇을 할까? > 1. 싸운다.\n");
387         printf("+ > 2. 포켓몬.\n");
388         printf("+ > 3. 잡는다.\n");
389         printf("+ > 4. 도망친다.\n");
390         printf("+ >> ");
391         scanf("%d", &battleopt);
392
393         if (battleopt == 1) {
394             //int pokenum=0;
395             while (1) {
396                 int fightresult;
397                 fightresult = Fight(adversary, user, &user->poke[pokenum]);
398
399                 if (fightresult == 3)
400                     break;
401
402                 else if (fightresult == 0 && pokenum < user->num_poke - 1) { // 나중에 진위 여부 확인할 것!
403                     pokenum++;
404                     break;
405                 }
406                 else if (fightresult == 0 && pokenum >= user->num_poke - 1) {
407                     printf("+ 더 이상 사용할 수 있는 포켓몬이 없다...\n");
408                     printf("+ %s는 %s로부터 간신히 도망쳤다...\n", user->user_name,
409                           adversary->name);
410                     pokenum++;
411                     return;
412                     //break;
413                 }
414                 else if (fightresult == -1) {
415                     pokenum++;
416                     return;
417                 }
418                 else if (fightresult == 1)
419                     return;
420             }
421
422             /*
423             else if (battleopt == 2) {
424                 user_Pokemon(NULL, user);
425             }
426             else if (battleopt == 3) {
427                 srand(time(NULL));
428                 int catchopt = rand() % 2;
429                 if (catchopt == 1) {
430                     Catch(adversary, user, NULL);
431                     return;
432                 }
433                 else printf("+ 아쉽다! 조금만 더 하면 잡을 수 있었는데!\n");
434             }
435             else {
436                 int runres = Run();
437                 if (runres == 1) break;
438             }
439         }
440     }
441 }
442 }
```

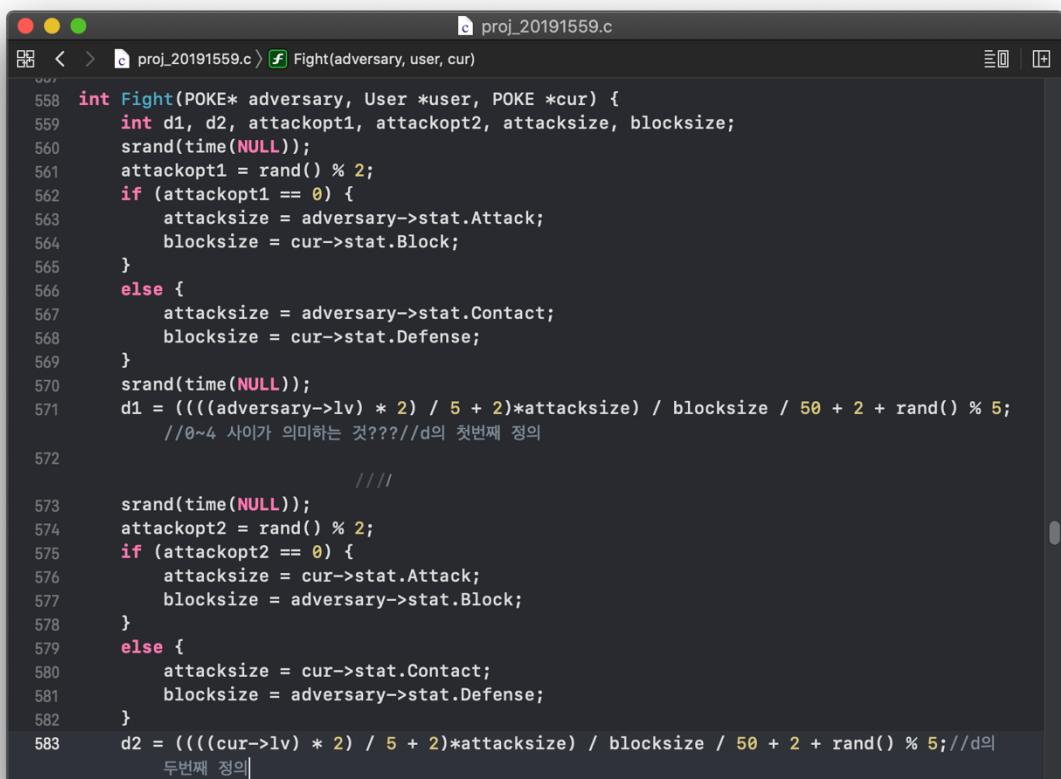
➤ Fight 함수

특수공격, 특수방어의 확률을 rand 함수를 이용해서 정해준다.

$d = (((\text{본인의 레벨} * 2) / 5 + 2) * \text{공격 또는 특수공격}) / \text{상대의 방어 또는 특수방어} / 50 + 2 + (0\sim4)$ 사이의 랜덤 값

의 방식으로 피해량을 계산하여 적용한다.

마찬가지로 Fight 함수의 자세한 구동 방법은 순서도에 명시되어 있다.



The screenshot shows a code editor window with the file 'proj_20191559.c' open. The cursor is positioned at the end of line 583, which contains the calculation for d2. A circled 'C' is overlaid on the left side of the screen.

```
558 int Fight(POKE* adversary, User *user, POKE *cur) {
559     int d1, d2, attackopt1, attackopt2, attacksize, blocksize;
560     srand(time(NULL));
561     attackopt1 = rand() % 2;
562     if (attackopt1 == 0) {
563         attacksize = adversary->stat.Attack;
564         blocksize = cur->stat.Block;
565     }
566     else {
567         attacksize = adversary->stat.Contact;
568         blocksize = cur->stat.Defense;
569     }
570     srand(time(NULL));
571     d1 = (((adversary->lv) * 2) / 5 + 2)*attacksize) / blocksize / 50 + 2 + rand() % 5;
572     //0~4 사이가 의미하는 것????//d의 첫번째 정의
573     /////
574     srand(time(NULL));
575     attackopt2 = rand() % 2;
576     if (attackopt2 == 0) {
577         attacksize = cur->stat.Attack;
578         blocksize = adversary->stat.Block;
579     }
580     else {
581         attacksize = cur->stat.Contact;
582         blocksize = adversary->stat.Defense;
583     }
584     d2 = (((cur->lv) * 2) / 5 + 2)*attacksize) / blocksize / 50 + 2 + rand() % 5;//d의
      두번째 정의|
```

```

587     if (adversary->stat.Speed > cur->stat.Speed) {
588         //선공-----
589         printf("%s는 %s에게 %d의 피해를 입혔다!\n", adversary->name, cur->name, d1);
590         cur->stat.cur_HP -= d1;
591         if (cur->stat.cur_HP <= 0) cur->stat.cur_HP = 0;
592         printf("+ %s(%d)          %s(%d)\n", cur->name, cur->lv,
593                                         adversary->name, adversary->lv);
594         //if(cur->stat.HP <=0) cur->stat.HP=0;
595         printf("+ HP = (%d/%d)      (%d/%d)\n\n", cur->stat.cur_HP, cur->stat.lv_HP,
596                                         adversary->stat.cur_HP, adversary->stat.lv_HP);
597
598     if (cur->stat.cur_HP == 0) {
599         int faintopt;
600         printf("+ %s는 쓰러졌다!\n", cur->name);
601         while (1) {
602             printf(" 1. 다음 포켓몬을 사용한다.\n");
603             printf(" 2. 도망친다.\n");
604             scanf("%d", &faintopt);
605             if (faintopt == 1) return 0;
606
607             else if (faintopt == 2) {
608                 int runres = Run();
609                 if (runres == 1) {
610                     //break;
611                     return -1;
612                 }
613             }
614         }
615         //후공-----
616         printf("%s는 %s에게 %d의 피해를 입혔다!\n", cur->name, adversary->name, d2);
617         adversary->stat.cur_HP -= d2;
618         if (adversary->stat.cur_HP <= 0) adversary->stat.cur_HP = 0;
619         printf("+ %s(%d)          %s(%d)\n", cur->name, cur->lv,
620                                         adversary->name, adversary->lv);
621         printf("+ HP = (%d/%d)      (%d/%d)\n\n", cur->stat.cur_HP, cur->stat.lv_HP,
622                                         adversary->stat.cur_HP, adversary->stat.lv_HP);
623
624     if (adversary->stat.cur_HP == 0) {
625         printf("+ %s를 쓰러트렸다!\n", adversary->name);
626         int upexp = ((adversary->lv) * (adversary->stat.HP)) / 5;
627
628         printf("+ %s는 %d만큼의 경험치를 얻었다!\n", cur->name, upexp);
629         cur->exp += upexp;
630         if (cur->exp >= ((cur->lv)*(cur->lv)*(cur->lv))) {
631             cur->lv++;
632             printf("+ %s는 lv.%d에서 lv.%d로 레벨업했다!\n", cur->name, cur->lv - 1,
633                                         cur->lv);
634             cur->stat.lv_HP = ((cur->stat.HP) * 2 + cur->stat.iv)*(cur->lv) / 100 +
635             10 + cur->lv;
636             cur->exp = 0;
637
638             int f = Find(head2, cur->dex_num);
639             if (f == 1) {
640                 Evolution(cur, cur->dex_num);
641             }
642             return 1;
643         }
644     }
645
646     //사용자의 포켓몬이 먼저 공격하는 경우
647     else {
648         //선공-----
649         printf("%s는 %s에게 %d의 피해를 입혔다!\n", cur->name, adversary->name, d2);

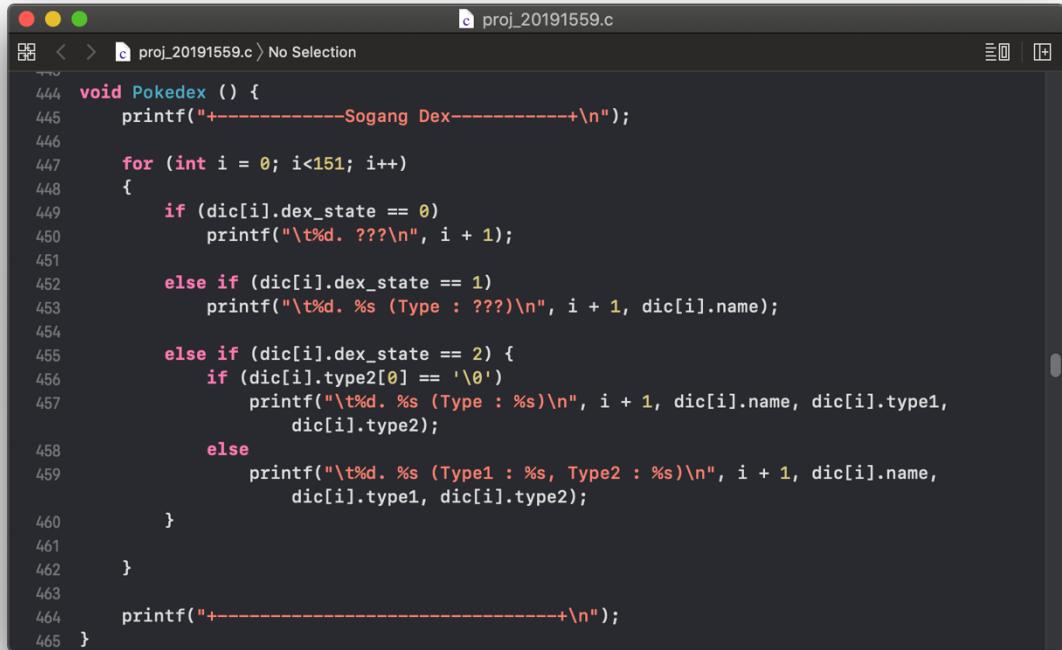
```



➤ Pokedex 함수

dic[i].dex_state 값에 따라 정보를 상이하게 출력한다.

-0이면 ???, 1이면 이름만, 2이면 이름과 Type 출력.



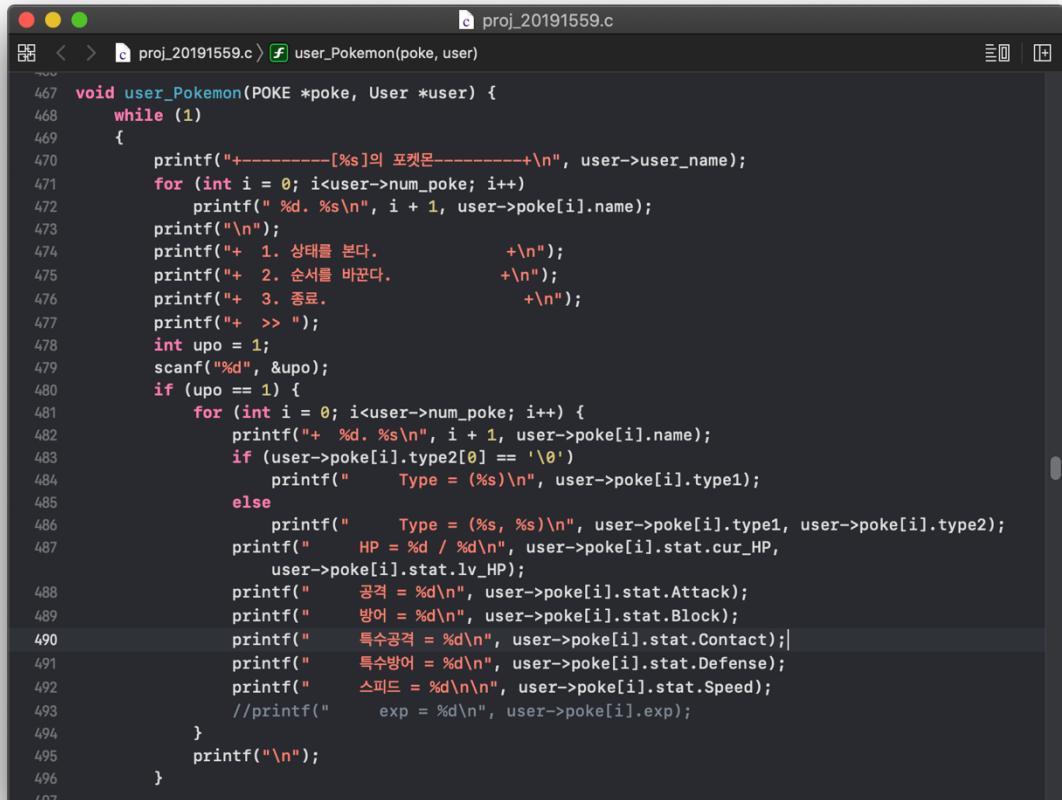
```
444 void Pokedex () {
445     printf("-----Sogang Dex-----+\n");
446
447     for (int i = 0; i<151; i++)
448     {
449         if (dic[i].dex_state == 0)
450             printf("\t%d. ???\n", i + 1);
451
452         else if (dic[i].dex_state == 1)
453             printf("\t%d. %s (Type : ???)\n", i + 1, dic[i].name);
454
455         else if (dic[i].dex_state == 2) {
456             if (dic[i].type2[0] == '\0')
457                 printf("\t%d. %s (Type : %s)\n", i + 1, dic[i].name, dic[i].type1,
458                         dic[i].type2);
459             else
460                 printf("\t%d. %s (Type1 : %s, Type2 : %s)\n", i + 1, dic[i].name,
461                         dic[i].type1, dic[i].type2);
462         }
463
464     printf("-----+\n");
465 }
```



2019 Sang Won Kang

➤ user_Pokemon 함수

사용자가 지닌 포켓몬의 상태를 보거나 순서를 바꾼다. 순서 바꾸기는 주솟값을 바꿔줌으로써 (Call by reference) 구현한다.

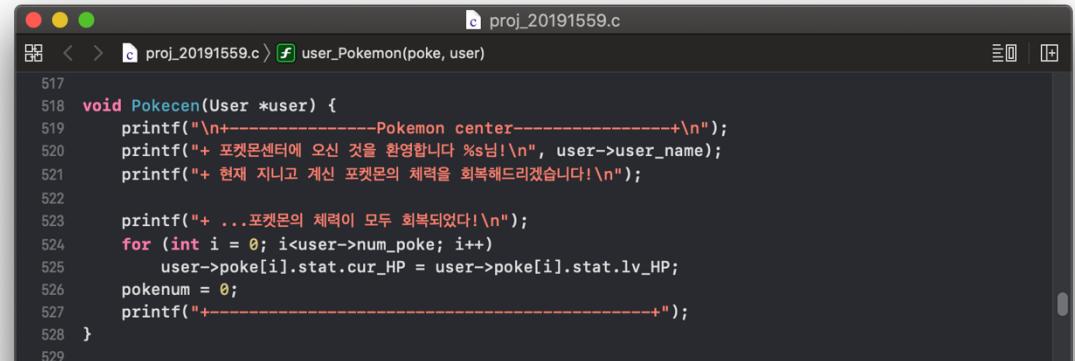


```
467 void user_Pokemon(POKE *poke, User *user) {
468     while (1)
469     {
470         printf("-----[%s]의 포켓몬-----+\n", user->user_name);
471         for (int i = 0; i<user->num_poke; i++)
472             printf(" %d. %s\n", i + 1, user->poke[i].name);
473         printf("\n");
474         printf("+ 1. 상태를 본다.          +\n");
475         printf("+ 2. 순서를 바꾼다.        +\n");
476         printf("+ 3. 종료.                  +\n");
477         printf("+ >> ");
478         int upo = 1;
479         scanf("%d", &upo);
480         if (upo == 1) {
481             for (int i = 0; i<user->num_poke; i++) {
482                 printf("+ %d. %s\n", i + 1, user->poke[i].name);
483                 if (user->poke[i].type2[0] == '\0')
484                     printf("      Type = (%s)\n", user->poke[i].type1);
485                 else
486                     printf("      Type = (%s, %s)\n", user->poke[i].type1, user->poke[i].type2);
487                 printf("      HP = %d / %d\n", user->poke[i].stat.cur_HP,
488                         user->poke[i].stat.lv_HP);
489                 printf("      공격 = %d\n", user->poke[i].stat.Attack);
490                 printf("      방어 = %d\n", user->poke[i].stat.Block);
491                 printf("      특수공격 = %d\n", user->poke[i].stat.Contact);
492                 printf("      특수방어 = %d\n", user->poke[i].stat.Defense);
493                 printf("      스피드 = %d\n", user->poke[i].stat.Speed);
494             //printf("      exp = %d\n", user->poke[i].exp);
495         }
496         printf("\n");
497     }
```

➤ Pokecen 함수

사용자가 지닌 포켓몬의 체력을 모두 회복한다.

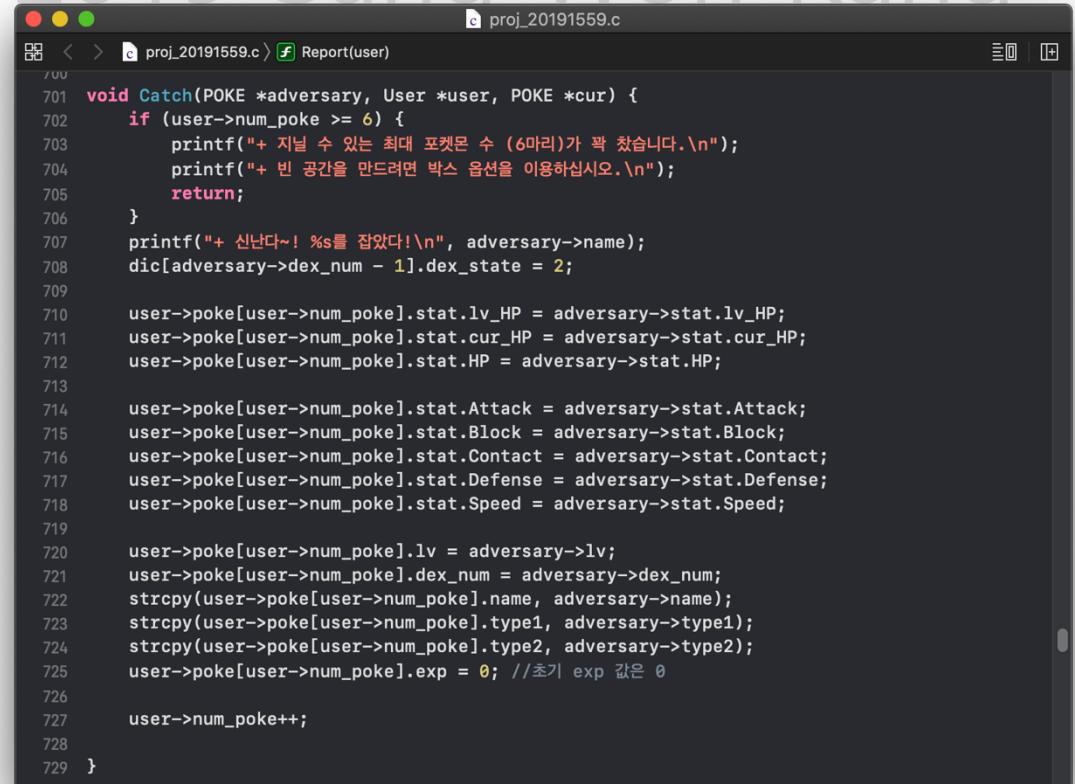
(cur_HP의 값을 lv_HP의 값과 같게 한다.)



```
proj_20191559.c
517 void Pokecen(User *user) {
518     printf("\n+-----Pokemon center-----+\n");
519     printf("+ 포켓몬센터에 오신 것을 환영합니다 %s님!\n", user->user_name);
520     printf("+ 현재 지니고 계신 포켓몬의 체력을 회복해드리겠습니다!\n");
521
522     printf("+ ...포켓몬의 체력이 모두 회복되었다!\n");
523     for (int i = 0; i<user->num_poke; i++)
524         user->poke[i].stat.cur_HP = user->poke[i].stat.lv_HP;
525     pokenum = 0;
526     printf("+-----+\n");
527 }
528 }
```

➤ Catch 함수

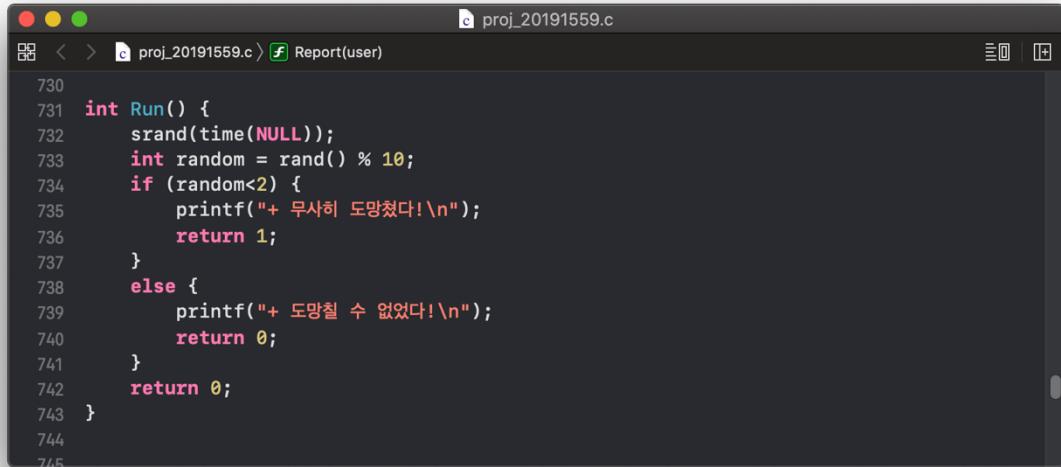
포켓몬을 잡는데 성공할 시 User Linked List에 adversary pokemon의 정보를 저장한다.



```
proj_20191559.c
700
701 void Catch(POKE *adversary, User *user, POKE *cur) {
702     if (user->num_poke >= 6) {
703         printf("+ 지닐 수 있는 최대 포켓몬 수 (6마리)가 꽉 찼습니다.\n");
704         printf("+ 빈 공간을 만드려면 박스 옵션을 이용하십시오.\n");
705         return;
706     }
707     printf("+ 신난다~! %s를 잡았다!\n", adversary->name);
708     dic[adversary->dex_num - 1].dex_state = 2;
709
710     user->poke[user->num_poke].stat.lv_HP = adversary->stat.lv_HP;
711     user->poke[user->num_poke].stat.cur_HP = adversary->stat.cur_HP;
712     user->poke[user->num_poke].stat.HP = adversary->stat.HP;
713
714     user->poke[user->num_poke].stat.Attack = adversary->stat.Attack;
715     user->poke[user->num_poke].stat.Block = adversary->stat.Block;
716     user->poke[user->num_poke].stat.Contact = adversary->stat.Contact;
717     user->poke[user->num_poke].stat.Defense = adversary->stat.Defense;
718     user->poke[user->num_poke].stat.Speed = adversary->stat.Speed;
719
720     user->poke[user->num_poke].lv = adversary->lv;
721     user->poke[user->num_poke].dex_num = adversary->dex_num;
722     strcpy(user->poke[user->num_poke].name, adversary->name);
723     strcpy(user->poke[user->num_poke].type1, adversary->type1);
724     strcpy(user->poke[user->num_poke].type2, adversary->type2);
725     user->poke[user->num_poke].exp = 0; //초기 exp 값은 0
726
727     user->num_poke++;
728 }
729 }
```

➤ Run 함수

rand()함수를 이용해 확률 계산 후 도망치는 동작을 수행한다.

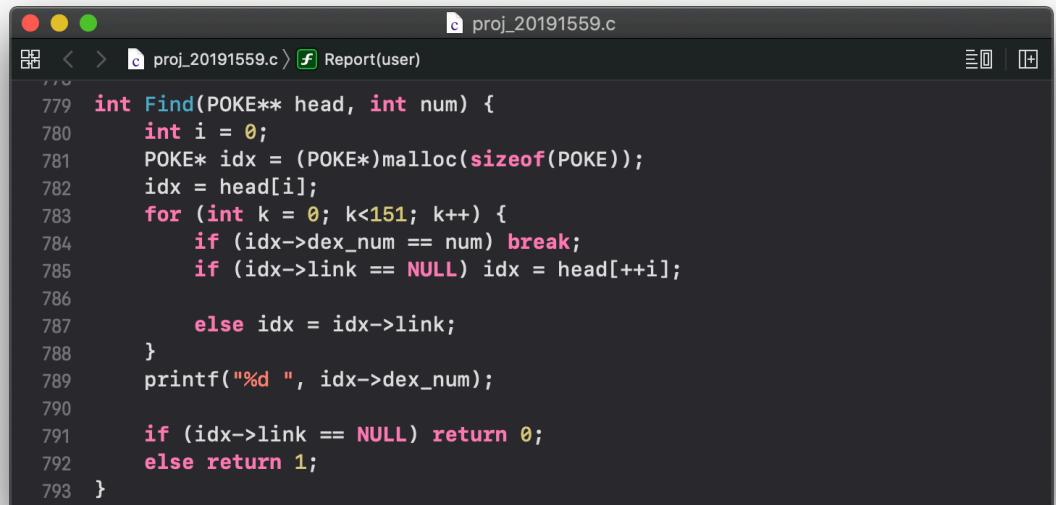


```
730
731 int Run() {
732     srand(time(NULL));
733     int random = rand() % 10;
734     if (random<2) {
735         printf("+ 무사히 도망쳤다!\n");
736         return 1;
737     }
738     else {
739         printf("+ 도망칠 수 없었다!\n");
740         return 0;
741     }
742     return 0;
743 }
744
745
```

➤ Find 함수

**head (전체 포켓몬의 진화 계통도 참조)와 포켓몬 번호를 입력받아 그 포켓몬이 진화가 가능한 포켓몬인지 판별한다. 진화 가능하다면 1을 return, 최종 진화 단계여서 더 이상 진화할 수 없는 포켓몬이면 0을 return한다.

(구분 방식: 그 다음 노드에 NULL이 들어가는지 여부 검사)

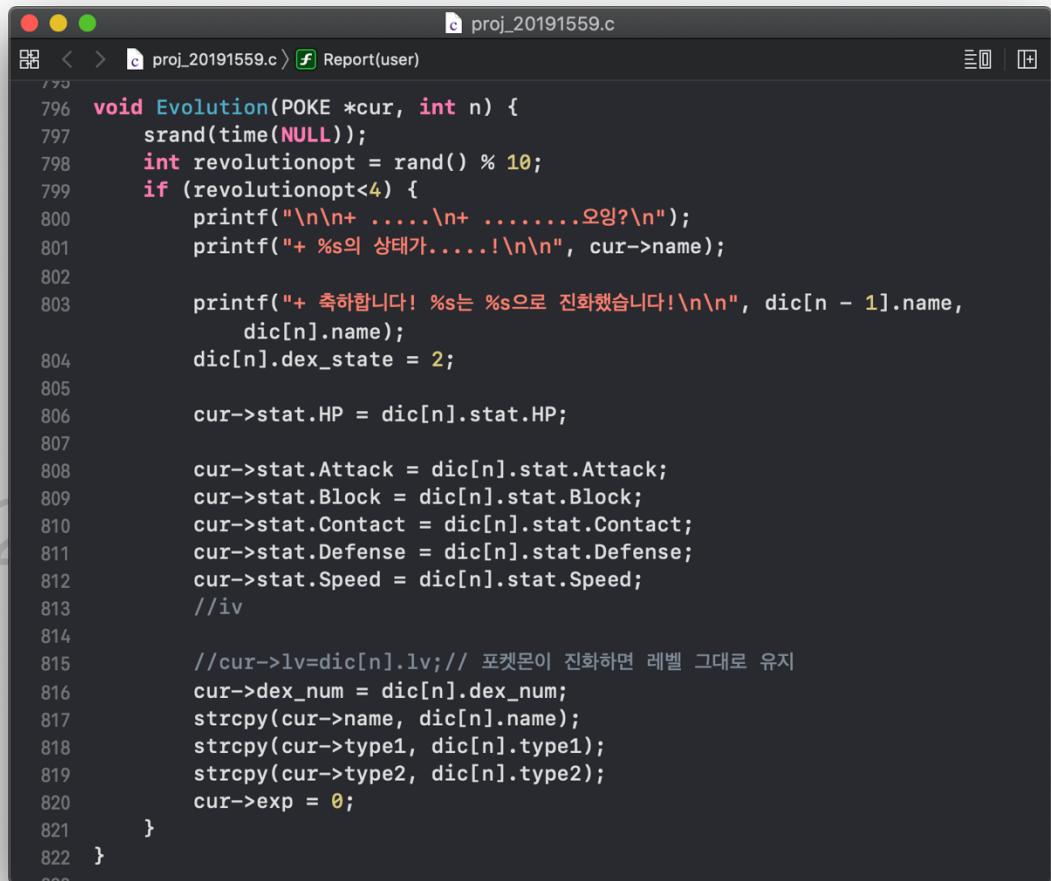


```
779 int Find(POKE** head, int num) {
780     int i = 0;
781     POKE* idx = (POKE*)malloc(sizeof(POKE));
782     idx = head[i];
783     for (int k = 0; k<151; k++) {
784         if (idx->dex_num == num) break;
785         if (idx->link == NULL) idx = head[++i];
786
787         else idx = idx->link;
788     }
789     printf("%d ", idx->dex_num);
790
791     if (idx->link == NULL) return 0;
792     else return 1;
793 }
```

➤ Evolution 함수

Evolution 함수는 Find 함수를 통해 진화 가능 여부가 검증된 포켓몬을 입력으로 받는다.

rand()함수를 이용하여 40%의 확률로 진화 작업을 수행하며, 진화한 포켓몬의 도감을 ‘소유 상태’인 2로 바꾼다.



```
796 void Evolution(POKE *cur, int n) {
797     srand(time(NULL));
798     int revolutionopt = rand() % 10;
799     if (revolutionopt<4) {
800         printf("\n\n+ .....오잉?\n");
801         printf("+ %s의 상태가.....!\n\n", cur->name);
802
803         printf("+ 축하합니다! %s는 %s으로 진화했습니다!\n\n", dic[n - 1].name,
804             dic[n].name);
805         dic[n].dex_state = 2;
806
807         cur->stat.HP = dic[n].stat.HP;
808
809         cur->stat.Attack = dic[n].stat.Attack;
810         cur->stat.Block = dic[n].stat.Block;
811         cur->stat.Contact = dic[n].stat.Contact;
812         cur->stat.Defense = dic[n].stat.Defense;
813         cur->stat.Speed = dic[n].stat.Speed;
814         //iv
815
816         //cur->lv=dic[n].lv;// 포켓몬이 진화하면 레벨 그대로 유지
817         cur->dex_num = dic[n].dex_num;
818         strcpy(cur->name, dic[n].name);
819         strcpy(cur->type1, dic[n].type1);
820         strcpy(cur->type2, dic[n].type2);
821         cur->exp = 0;
822     }
823 }
```

➤ Box 함수

두 가지 옵션을 수행한다.

- 1. 박스에 사용자가 지닌 포켓몬 넣기
- 2. 박스에서 포켓몬 꺼내기

만약 박스가 비어있는데 2번 옵션을 선택한다면 “상자가 비어있습니다”를 출력하고 함수를 종료한다.

```
proj_20191559.c Report(user)

825 void Box(User *user) {
826     int boxopt;
827     printf("-----Box-----+\n");
828     printf("+ 무엇을 하시겠습니까?\n");
829     printf(" 1. 박스에 포켓몬 넣기\n 2. 박스에서 포켓몬 꺼내기\n");
830     printf("+ > ");
831     scanf("%d", &boxopt);
832
833     if (boxopt == 1) {
834         for (int i = 0; i<user->num_poke; i++) {
835             if (user->poke[i].type2[0] == '\0')
836                 printf("+ %d. %s (Type : %s)\n", i + 1, user->poke[i].name,
837                     user->poke[i].type1);
838             else
839                 printf("+ %d. %s (Type1 : %s, Type2 : %s)\n", i + 1,
840                     user->poke[i].name, user->poke[i].type1,
841                     user->poke[i].type2);
842         }
843         int inopt;
844         printf("+ 넣을 포켓몬 :\n");
845         printf("+ > ");
846         scanf("%d", &inopt);
847
848         BOX.poke[boxnum] = user->poke[inopt - 1];
849
850         printf("inserted name : %s\n", BOX.poke[boxnum].name);
851         boxnum++;
852         for (int i = inopt - 1; i<user->num_poke - 1; i++)
853             user->poke[i] = user->poke[i + 1];
854         user->num_poke--;
855     }
856     if (boxopt == 2) {
857         if (boxnum == 0)
858             printf("+ 상자가 비어있습니다.\n");
859             return;
860         if (user->num_poke >= 6) {
861             printf("+ 사용자의 포켓몬 보관함이 이미 꽉 찻습니다.\n");
862             return;
863         }
864         for (int i = 0; i<boxnum; i++) {
865             if (BOX.poke[i].type2[0] == '\0')
866                 printf("+ %d. %s (Type : %s)\n", i + 1, BOX.poke[i].name,
867                     BOX.poke[i].type1);
868             else
869                 printf("+ %d. %s (Type1 : %s, Type2 : %s)\n", i + 1,
870                     BOX.poke[i].name, BOX.poke[i].type1, BOX.poke[i].type2);
871         }
872         int outopt;
873         printf("+ 꺼낼 포켓몬 :\n");
874         printf("+ > ");
875         scanf("%d", &outopt);
876         user->poke[user->num_poke] = BOX.poke[outopt - 1];
877         user->num_poke++;
878
879         for (int i = outopt - 1; i<boxnum - 1; i++)
880             BOX.poke[i] = BOX.poke[i + 1];
881         boxnum--;
882     }
883 }
```