

Upgrade Dino Game



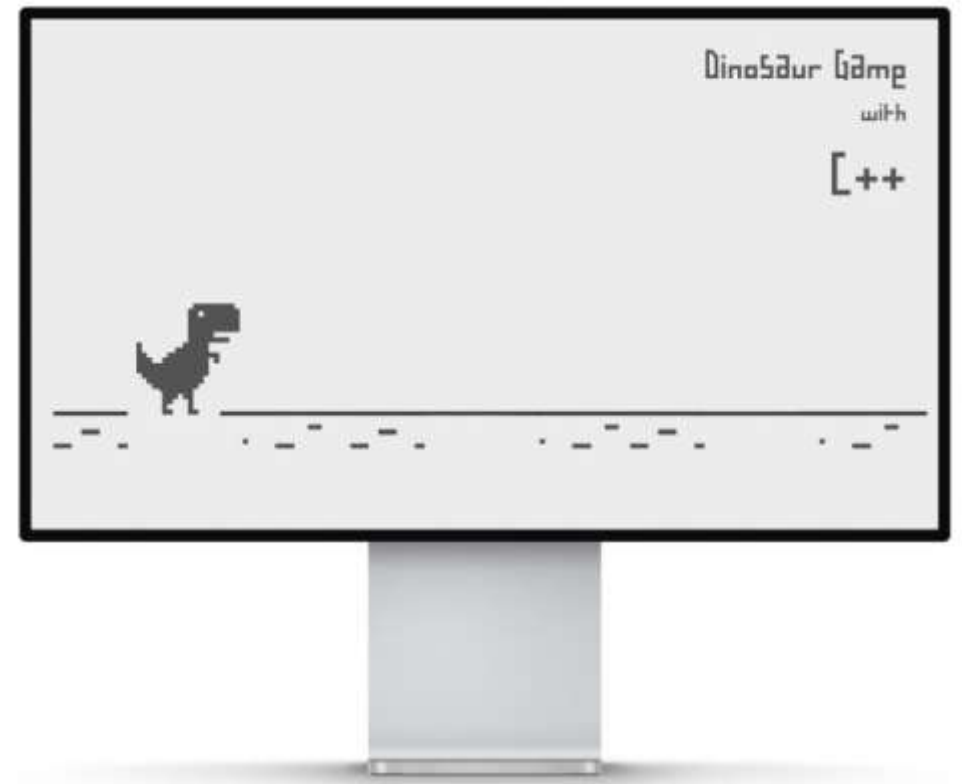
목차

1 Intro

2 Game

3 Epilogue

1. Intro



1. Intro

- 배경

직관적인
구현 가능

배운 지식의
활용

발전하는
과정

C++의
특징 고려



- 목표

시각적인 것 보다는 기능적인 게임

수정과 갱신이 용이한 게임

배운 내용을 활용하는 게임

1. Intro

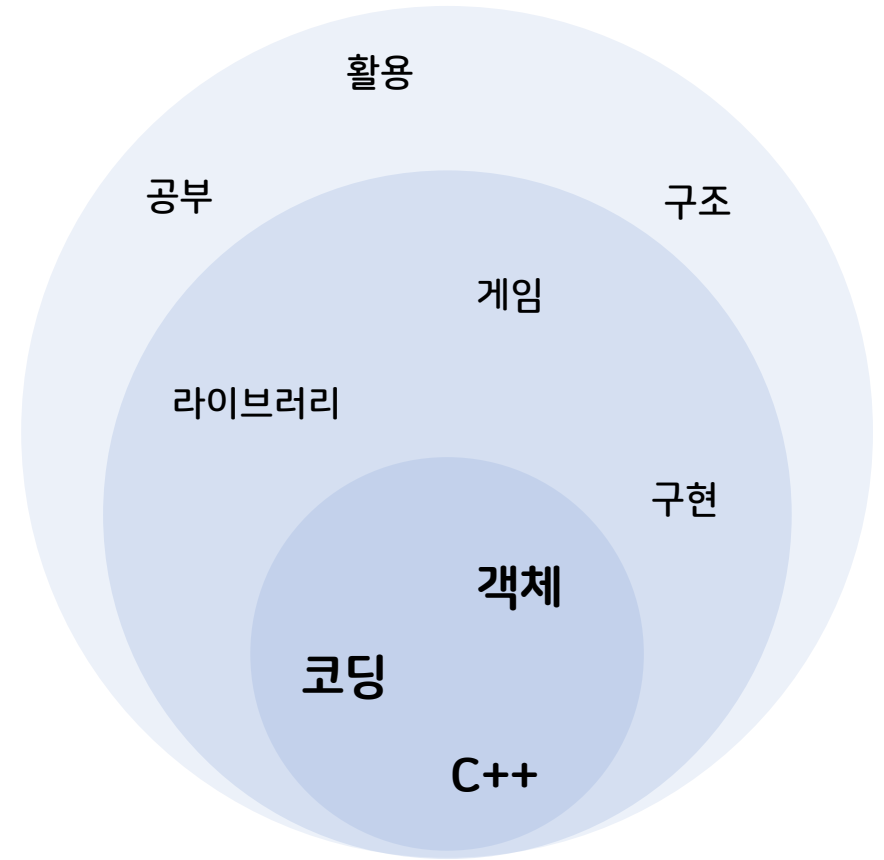
c++ 공부와 활용을 동시에 이룰 수 있는,

객체를 이용한 기능 추가와 수정에서 자유로운,

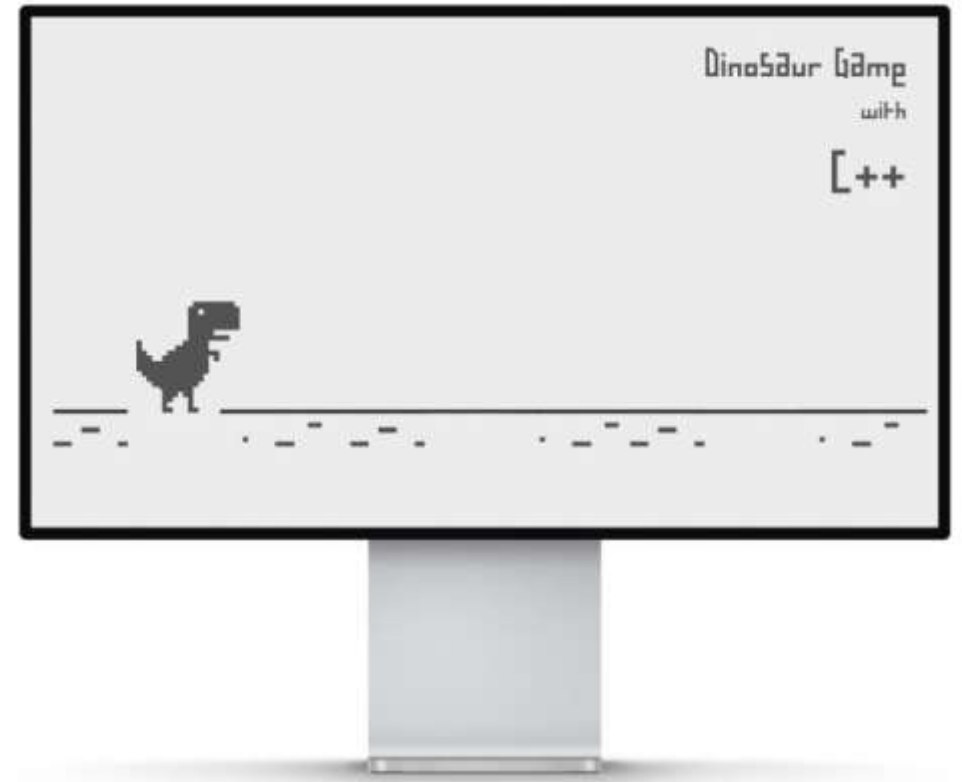
발생하는 이슈의 문제해결이 용이한,

원활한 협업과 분담을 할 수 있는,

Upgrade Dino 게임을 개발



2. Game



2. Game

• 키워드

콘솔창에서 실행되는 게임

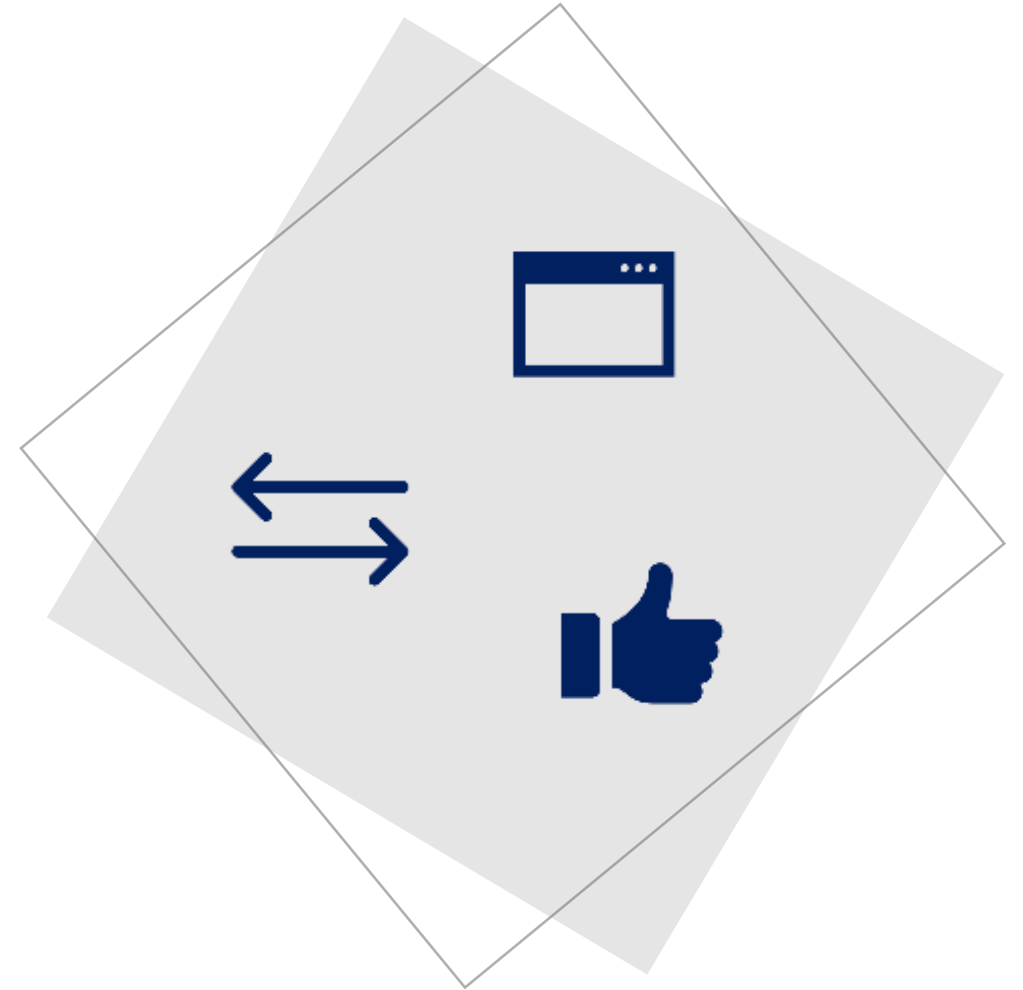
- 시각적 효과(SFML라이브러리 사용)보다는 c++ 공부와 활용에 중점

협업과 분담

- 공통개발영역과 개인개발영역을 분리

문제 해결과 발전

- 매주 개발 목표를 설정 후, 주마다 업데이트 해나감



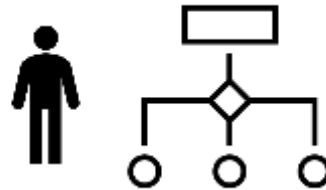
2. Game

- 개발과정



1, 2 Week

개발 목표 설정
활용 기능 협의
게임 구조 개발 및 조정



3 Week

분업 : 각자 캐릭터 개발



4 Week

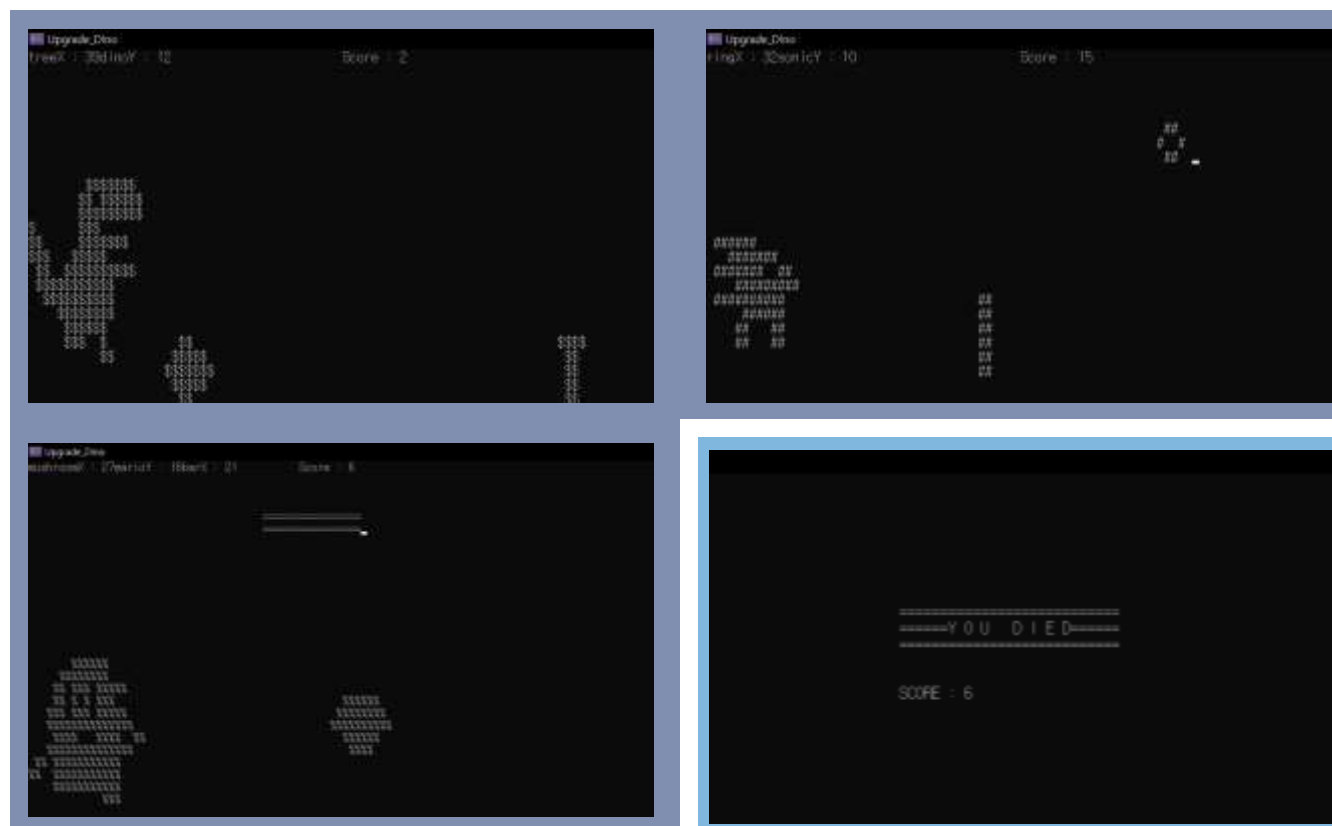
코드 분석 및 문제해결
코드 종합 -> 최종완성

2. Game

시작화면



게임화면



게임종료

2. Game

Update : Week 1, 2

개발 목표 설정

활용 기능 협의

게임 구조 개발 및 조정

[게임 구조 . 업데이트 내역]

클래스를 이용한 임의 캐릭터 생성 - 완료

메인 함수에서의 캐릭터 임의 조정 - 완료

시작화면, 종료화면, 콘솔 창 생성 - 완료

[Check Point : 기본 셋팅]

콘솔 창의 크기와 제목을 지정하는 함수

```
void SetGameScreen()  
{  
    system("mode con: cols=100 lines=25");  
    system("title Upgrade_Dino");  
}
```

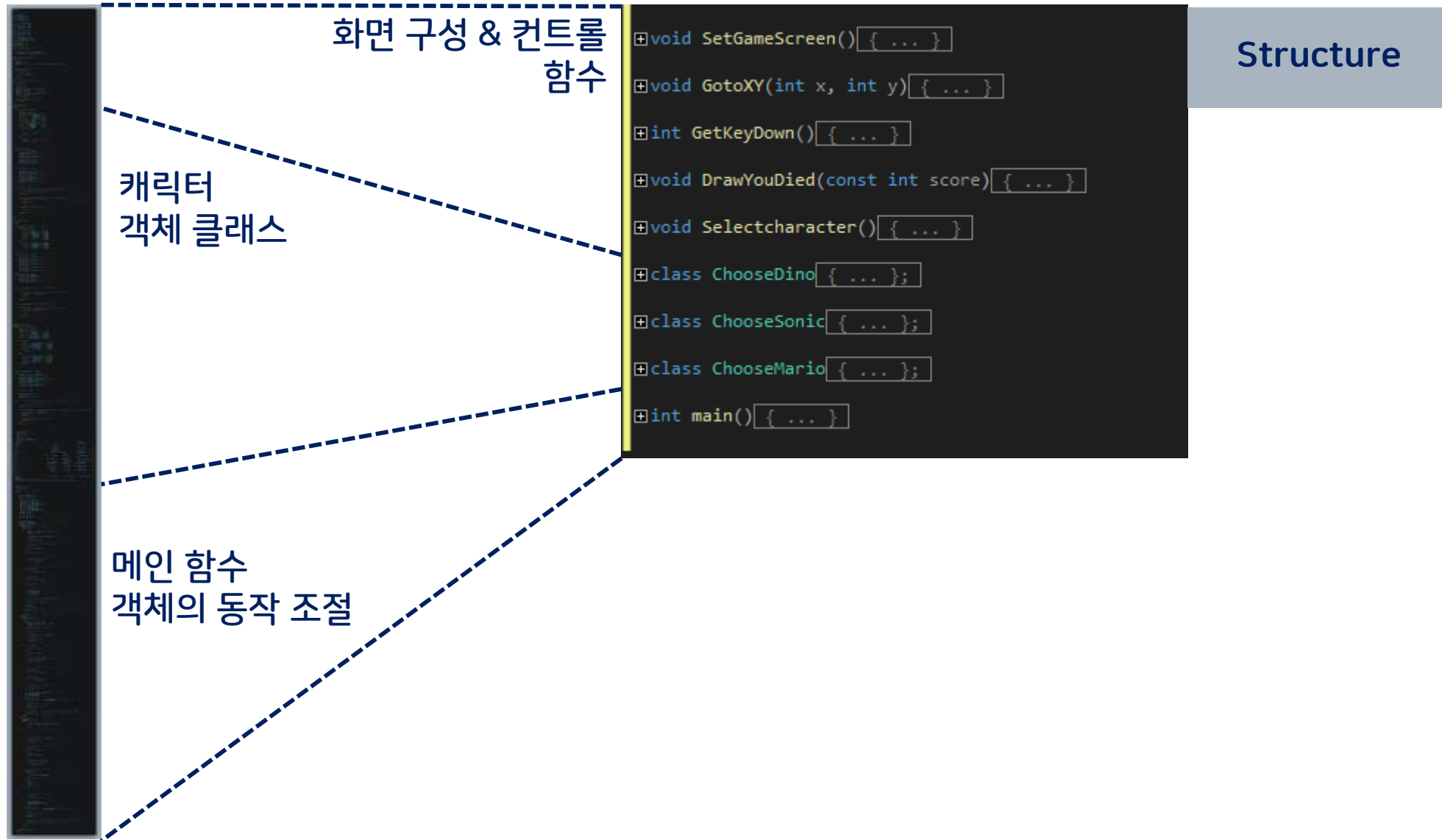
커서의 위치를 x, y로 이동하는 함수

```
void GotoXY( int x, int y)  
{  
    COORD Pos;  
    Pos.X = 2 + x;  
    Pos.Y = y;  
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), Pos);  
}
```

키보드의 입력을 받고,
입력된 키의 값을 반환하는 함수

```
int GetKeyDown()  
{  
    if (_kbhit() != 0)  
    {  
        return _getch();  
    }  
    return 0;  
}
```

2. Game



The screenshot displays a Jupyter Notebook with the following sections:

- Import Libraries:** Imports `numpy`, `pandas`, `matplotlib.pyplot`, `seaborn`, `sklearn.metrics`, `sklearn.preprocessing`, `sklearn.linear_model`, `sklearn.model_selection`, and `sklearn.ensemble`.
- Data Loading:** Loads the `titanic.csv` file into a `pandas` DataFrame.
- Data Cleaning:**
 - Checks for missing values using `isnull().sum()`.
 - Imputes missing values for `age` and `embarked` using `fillna()`.
 - Converts `sex` to a binary variable (0 for male, 1 for female).
 - Creates a `family_size` feature based on `size` and `children`.
 - Creates a `single` feature based on `family_size`.
 - Creates a `married` feature based on `family_size` and `children`.
 - Creates a `status` feature based on `married` and `single`.
 - Creates a `status2` feature based on `status` and `family_size`.
 - Creates a `status3` feature based on `status2` and `family_size`.
 - Creates a `status4` feature based on `status3` and `family_size`.
 - Creates a `status5` feature based on `status4` and `family_size`.
 - Creates a `status6` feature based on `status5` and `family_size`.
 - Creates a `status7` feature based on `status6` and `family_size`.
 - Creates a `status8` feature based on `status7` and `family_size`.
 - Creates a `status9` feature based on `status8` and `family_size`.
 - Creates a `status10` feature based on `status9` and `family_size`.
 - Creates a `status11` feature based on `status10` and `family_size`.
 - Creates a `status12` feature based on `status11` and `family_size`.
 - Creates a `status13` feature based on `status12` and `family_size`.
 - Creates a `status14` feature based on `status13` and `family_size`.
 - Creates a `status15` feature based on `status14` and `family_size`.
 - Creates a `status16` feature based on `status15` and `family_size`.
 - Creates a `status17` feature based on `status16` and `family_size`.
 - Creates a `status18` feature based on `status17` and `family_size`.
 - Creates a `status19` feature based on `status18` and `family_size`.
 - Creates a `status20` feature based on `status19` and `family_size`.
 - Creates a `status21` feature based on `status20` and `family_size`.
 - Creates a `status22` feature based on `status21` and `family_size`.
 - Creates a `status23` feature based on `status22` and `family_size`.
 - Creates a `status24` feature based on `status23` and `family_size`.
 - Creates a `status25` feature based on `status24` and `family_size`.
 - Creates a `status26` feature based on `status25` and `family_size`.
 - Creates a `status27` feature based on `status26` and `family_size`.
 - Creates a `status28` feature based on `status27` and `family_size`.
 - Creates a `status29` feature based on `status28` and `family_size`.
 - Creates a `status30` feature based on `status29` and `family_size`.
 - Creates a `status31` feature based on `status30` and `family_size`.
 - Creates a `status32` feature based on `status31` and `family_size`.
 - Creates a `status33` feature based on `status32` and `family_size`.
 - Creates a `status34` feature based on `status33` and `family_size`.
 - Creates a `status35` feature based on `status34` and `family_size`.
 - Creates a `status36` feature based on `status35` and `family_size`.
 - Creates a `status37` feature based on `status36` and `family_size`.
 - Creates a `status38` feature based on `status37` and `family_size`.
 - Creates a `status39` feature based on `status38` and `family_size`.
 - Creates a `status40` feature based on `status39` and `family_size`.
 - Creates a `status41` feature based on `status40` and `family_size`.
 - Creates a `status42` feature based on `status41` and `family_size`.
 - Creates a `status43` feature based on `status42` and `family_size`.
 - Creates a `status44` feature based on `status43` and `family_size`.
 - Creates a `status45` feature based on `status44` and `family_size`.
 - Creates a `status46` feature based on `status45` and `family_size`.
 - Creates a `status47` feature based on `status46` and `family_size`.
 - Creates a `status48` feature based on `status47` and `family_size`.
 - Creates a `status49` feature based on `status48` and `family_size`.
 - Creates a `status50` feature based on `status49` and `family_size`.
 - Creates a `status51` feature based on `status50` and `family_size`.
 - Creates a `status52` feature based on `status51` and `family_size`.
 - Creates a `status53` feature based on `status52` and `family_size`.
 - Creates a `status54` feature based on `status53` and `family_size`.
 - Creates a `status55` feature based on `status54` and `family_size`.
 - Creates a `status56` feature based on `status55` and `family_size`.
 - Creates a `status57` feature based on `status56` and `family_size`.
 - Creates a `status58` feature based on `status57` and `family_size`.
 - Creates a `status59` feature based on `status58` and `family_size`.
 - Creates a `status60` feature based on `status59` and `family_size`.
 - Creates a `status61` feature based on `status60` and `family_size`.
 - Creates a `status62` feature based on `status61` and `family_size`.
 - Creates a `status63` feature based on `status62` and `family_size`.
 - Creates a `status64` feature based on `status63` and `family_size`.
 - Creates a `status65` feature based on `status64` and `family_size`.
 - Creates a `status66` feature based on `status65` and `family_size`.
 - Creates a `status67` feature based on `status66` and `family_size`.
 - Creates a `status68` feature based on `status67` and `family_size`.
 - Creates a `status69` feature based on `status68` and `family_size`.
 - Creates a `status70` feature based on `status69` and `family_size`.
 - Creates a `status71` feature based on `status70` and `family_size`.
 - Creates a `status72` feature based on `status71` and `family_size`.
 - Creates a `status73` feature based on `status72` and `family_size`.
 - Creates a `status74` feature based on `status73` and `family_size`.
 - Creates a `status75` feature based on `status74` and `family_size`.
 - Creates a `status76` feature based on `status75` and `family_size`.
 - Creates a `status77` feature based on `status76` and `family_size`.
 - Creates a `status78` feature based on `status77` and `family_size`.
 - Creates a `status79` feature based on `status78` and `family_size`.
 - Creates a `status80` feature based on `status79` and `family_size`.
 - Creates a `status81` feature based on `status80` and `family_size`.
 - Creates a `status82` feature based on `status81` and `family_size`.
 - Creates a `status83` feature based on `status82` and `family_size`.
 - Creates a `status84` feature based on `status83` and `family_size`.
 - Creates a `status85` feature based on `status84` and `family_size`.
 - Creates a `status86` feature based on `status85` and `family_size`.
 - Creates a `status87` feature based on `status86` and `family_size`.
 - Creates a `status88` feature based on `status87` and `family_size`.
 - Creates a `status89` feature based on `status88` and `family_size`.
 - Creates a `status90` feature based on `status89` and `family_size`.
 - Creates a `status91` feature based on `status90` and `family_size`.
 - Creates a `status92` feature based on `status91` and `family_size`.
 - Creates a `status93` feature based on `status92` and `family_size`.
 - Creates a `status94` feature based on `status93` and `family_size`.
 - Creates a `status95` feature based on `status94` and `family_size`.
 - Creates a `status96` feature based on `status95` and `family_size`.
 - Creates a `status97` feature based on `status96` and `family_size`.
 - Creates a `status98` feature based on `status97` and `family_size`.
 - Creates a `status99` feature based on `status98` and `family_size`.
 - Creates a `status100` feature based on `status99` and `family_size`.
 - Creates a `status101` feature based on `status100`

Update : Week 3

각자의 캐릭터 개발

[캐릭터 개발 . 업데이트 내역]

- 공룡 캐릭터 생성 - 완료
- 소닉 캐릭터 생성 - 완료
- 마리오 캐릭터 생성 - 완료

공통 캐릭터 생성 - 완료

소닉 캐릭터 생성 - 완료

마리오 캐릭터 생성 - 완료

[illegible]

캐릭터를 그리는 메서드

[illegible]

게임 내 물체를 그리는 메서드

```
void DrawTree(int treeX)
{
    GotoXY(treeX, TREE_ROOT_Y);
    cout << "$$$";
    GotoXY(treeX, TREE_ROOT_Y + 1);
    cout << " $$ ";
    GotoXY(treeX, TREE_ROOT_Y + 2);
    cout << " $$ ";
    GotoXY(treeX, TREE_ROOT_Y + 3);
    cout << " $$ ";
    GotoXY(treeX, TREE_ROOT_Y + 4);
    cout << " $$ ";
}
```

캐릭터와 물체의 충돌처리 메서드

```

bool isMatchInOrder(const int tree, const int root, const int dir) {
    // 왼쪽의 노드 중량이 현재 노드 <= 현재값
    // 중량의 총합이 현재 노드 <= 값이면 중량은 0이
    GetSum(0, 0);

    cout << "tree : " << tree << ", dir : " << dir << endl;
    // 최종 중량의 근거를 표시해줌
    if (tree == 0 && root == 4 &&
        dir == 0)
        return true;

    if (root == 7 && root == 4 &&
        dir == 1)
        return true;

    return false;
}

```

2. Game

Update : Week 4

난이도 패치, 코드 종합

[난이도 패치 . 업데이트 내역]

공통 캐릭터 난이도 설정 - 완료

소닉 캐릭터 난이도 설정 - 완료

마리오 캐릭터 난이도 설정- 완료

[Check Point : 메인 함수]

제어문 : 캐릭터 선택 및 시작/실패

```
int input_key;
SelectCharacter();
input_key = _getch(); // _getch 함수는 cin과 달리 입력과 동시에 각 줄마다
switch (input_key) {
case 48: // ABCD 코드를 48은 1은 한씩한다.
    chosen(the_A);
```

```
case 50: // 30이 코드 50은 2번 씩  
do {  
    while (true) //한 번씩 10만 번
```

```
base 51 // ADD: 코드로 받은 2를 의미
Choose Mario C:
while (true) //반 반복 대안 부분
```

제어문 내 객체 생성 및 난이도 조절

```

chooseline()
while(1) { //무한 반복
    //출력: 현재 노드와, 현재 노드의 왼쪽과 오른쪽 자식
    if (A->left != 0 || A->right != 0) {
        cout<<endl;
    }

    //현재 노드 출력, 비어있지 않은데 종료
    if (getch() == 'q' || A == root)
    {
        cout<<endl;
        isRoot = false;
    }

    //왼쪽과 오른쪽 자식 출력, 비어있지 않은데 종료
    if (isRoot)
    {
        dir = 'L';
        dirV = prevV;
    }
    else
    {
        dir = 'R';
        dirV = prevV;
    }
}

```

[illegible]

2. Game

Update : Issue

[이슈체크. 업데이트 내역]

충돌 함수 이슈 발생 - 해결

장애물 출현 빈도수 이슈 발생 - 해결

[Issue Check]

충돌 범위 시각화

충돌 메서드에서 캐릭터와 물체가 충돌하는 범위가 너무 크거나 작아서 예상치 못한 상황에 게임이 종료되어버리는 문제 발생

-> 좌표값을 실시간으로 출력시켜 충돌 범위 시각화 및 조정 가능

```
cout << "mushroomX : " << mushroomX << "marioY : " << marioY << "barX : " << barX << endl;
//적절한 충돌 좌표 범위를 찾는다
if (mushroomX <= 6 && mushroomX >= 4 &&
    marioY > 7)
{
    return true;
}
if (barX <= 6 && barX >= 1 && marioY < 5) {
    return true;
}
return false;
```

장애물 출현 빈도수 조정

장애물들이 캐릭터에 다가오는 순간이 겹쳐 어떠한 경우의 수에도 회피할 수 없는 상황이 발생

-> 두 장애물이 일정거리 이상 가까워지면 하나를 빨라지게 하여 해당 문제 해결

```
mushroomX -= 3;
if (mushroomX <= 0)
{
    mushroomX = MUSHROOM_ROOT_X;
}

if (barX < 20) // bar가 일정 거리이상 가까워지면 속도가 빨라지게 함.
    barX -= 4;
else
    barX -= 2;
if (barX <= 0)
{
    barX = BAR_X;
}
```

2. Game

1. Dino : Easy_mode



[공룡게임 - 난이도 : 쉬움]

시작화면에서 키보드 숫자 1 키 를 선택하면 공룡게임 시작

돌 장애물과 나무 장애물이 각각의 다른 속도와 빈도수로 공룡을 향해 접근한다.

Z키를 누르면 점프를 하게 되는데, 장애물을 피할시 스코어가 1점씩 오르지만, 부딪히면 게임이 종료된다

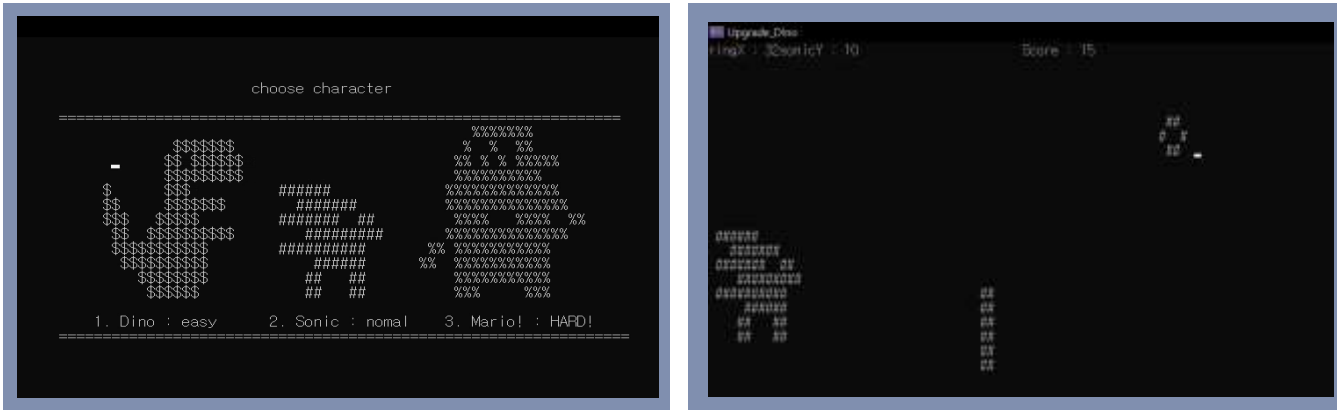
충돌범위를 여유롭게 설정하여 회피율이 높으며, 높은 수준의 컨트롤 능력이 요구되지 않는다.



게임종료

2. Game

2. Sonic : Normal_mode



[소닉게임 - 난이도 : 보통]

시작화면에서 키보드 숫자 2 키 를 선택하면 소닉게임 시작

화면 상단에는 점수를 얻을 수 있는 링, 하단에는 장애물이 각기 다른 속도와 빈도수로 접근한다

Z키를 누르면 점프를 하게 되는데, 장애물을 피할 시 스코어가 1점 씩 오르고, 링과 접촉하게 되면 10점이 오른다.

충돌범위를 딱 맞게 설정하여 회피하는데 어느정도 수준의 컨트롤 능력이 요구된다.

링 요소의 추가로 스코어 기록을 늘릴 수 있지만 반대로 링과 접촉하려다가 장애물과 충돌하는 것을 주의해야한다.



게임종료

2. Game

3. Mario : Hard_mode



[마리오게임 - 난이도 : 어려움]

시작화면에서 키보드 숫자 3 키 를 선택하면 마리오게임 시작

화면 상단에는 바형태의 장애물, 하단에는 독버섯이 각기 다른 속도와 빈도수로 접근한다

Z키를 누르면 점프를 하게 되는데, 장애물을 피할 시 스코어가 1점 씩 오른다.

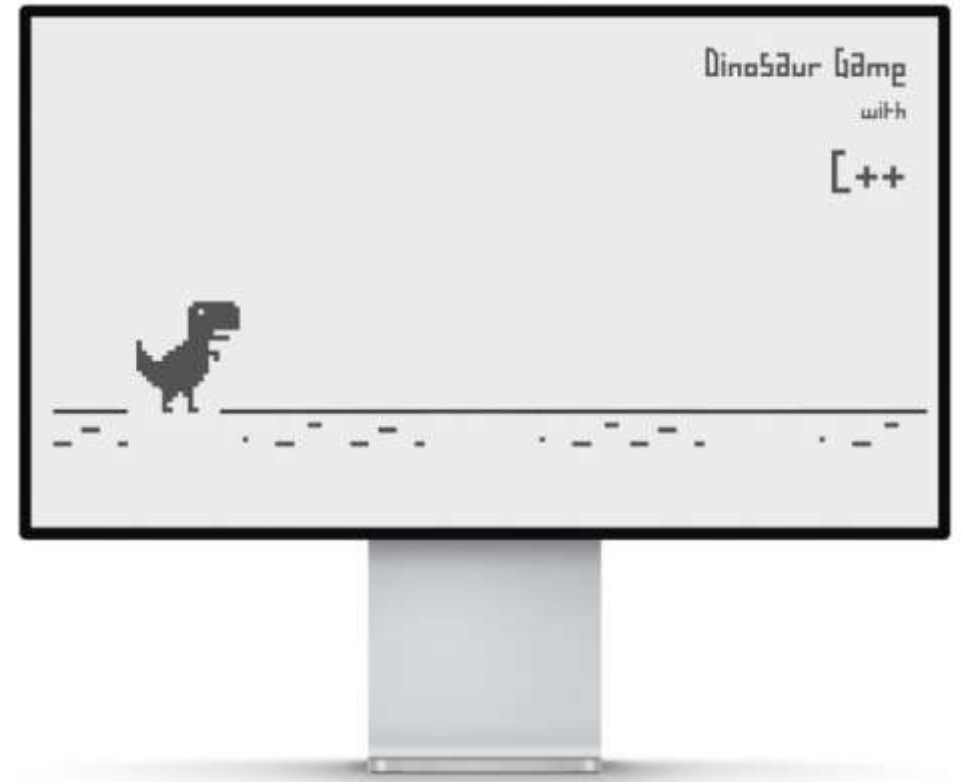
앞선 두 게임들과 달리 두개의 장애물이 상/하단 에서 출몰하기 때문에 높은 수준의 컨트롤 능력이 요구된다.

하지만 두장애물이 동시에 겹쳐 출몰하는 상황은 방지하여서 사용자의 실력에 따라 계속 게임을 이어나갈 수 있다.



게임종료

3. Epilogue



3. Epilogue

기획

주변의 가깝고 친근한 프로그램,
즐거운 경험과 이미지,
C++의 특성인 객체를 이용한 기능
추가와 수정에서 자유로운
미니게임(공룡게임)을 기획



과정

지속적인 업데이트와 업그레이드로 게임을
개발하는 과정속에서 계속해서 발전.
공룡 -> 소닉 -> 마리오-> 난이도



결과

초기 공룡게임의 컨셉과 업그레이드된 캐릭터들
프로젝트를 진행하면서 향상된 실력
Upgrade Dino

감사합니다