Plagiarism Scan Report

Summary	
Report Genrated Date	28 Mar, 2018
Plagiarism Status	**100% Unique**
Total Words	396
Total Characters	2752
Any Ignore Url Used	

Content Checked For Plagiarism:

```
#pragma once #include  #include  #include  #include "FCheckCross.h" using FText =
std::string; using int32 = int; void Introduction(); void PlayGame(); FText
ValidGuessAccept(); bool RetryGame(); void PrintGameLog(); void wait(int seconds);
FCheckCross CCGame; int main() { bool bPlayAgain = false; do { Introduction();
PlayGame(); bPlayAgain = RetryGame(); } while (bPlayAgain); return 0; } void
Introduction() { std::cout << "Welcome to Checks And Crosses, a fun word game.\n";
std::cout << std::endl; std::cout << " } { ___ " << std::endl; std::cout << " (o o) (o o) " <<
std::endl; std::cout << " /-------\\ / \\ /-------\\ " << std::endl; std::cout << " / | BULL |O O|
COW | \\ " << std::endl; std::cout << " * |-,--- | |------| * " << std::endl; std::cout << " ^ ^ ^
^ " << std::endl; std::cout << "Can you guess the " << CCGame.GameTry+3; std::cout <<
" letter isogram I'm thinking of?\n"; std::cout << std::endl; return; } void PlayGame() {
CCGame.Reset(); int32 MaxTries = CCGame.GetMaxTries(); while (!CCGame.IsGameWon()
&& CCGame.GetCurrentTry() <= MaxTries) { FText Guess = ValidGuessAccept();
FCheckCrossCount CheckCrossCount = CCGame.ValidGuessSubmit(Guess); std::cout <<
"Checks = " << CheckCrossCount.Bulls; std::cout << ". Crosses = " <<
CheckCrossCount.Cows << "\n\n"; } PrintGameLog(); return; } FText ValidGuessAccept() {
FText Guess = ""; EGuessStatus Status = EGuessStatus::Wrong_Status; do { int32
CurrentTry = CCGame.GetCurrentTry(); std::cout << "Try " << CurrentTry << " of " <<
CCGame.GetMaxTries(); std::cout << ". Enter your guess: "; std::getline(std::cin, Guess);
Status = CCGame.CheckGuessValidity(Guess); switch (Status) { case
EGuessStatus::Wrong_Length: std::cout << "Please enter a " <<
CCGame.GetHiddenWordLength() << " letter word.\n\n"; break; case
EGuessStatus::Not_Isogram: std::cout << "Please enter a word witout repeating
letters.\n\n"; break; case EGuessStatus::Not_Lowercase: std::cout << "Please enter all
lowercase letters.\n\n"; break; default: break; } } while (Status != EGuessStatus::Cool);
return Guess; } bool RetryGame() { if (CCGame.GameTry == 4) { std::cout << "You've
reached the end of this game"<
```