POKER

Nasz projekt jest oparty na karcianej grze w pokera. W zamyśle chcieliśmy stworzyć program, który symuluje rozgrywkę w pokera i wyświetla ją graficznie na ekranie. Jedynym możliwym trybem gry miało być starcie z sześcioma graczami SI. Wzorowaliśmy się na popularnej grze World Series Of Poker. Cały program jest napisany w języku C++. Używamy również biblioteki graficznej SFML 2.4.

Jak działa nasz program?

W programie stworzyłem dwie struktury: Karta i Gracz. Każda karta ma swoją wysokość kolor oraz teksturę. Gracz natomiast ma 2 karty, pieniądze, pieniądze na stole oraz kilka zmiennych oraz funkcji pomocniczych. Poker składa się z dwóch głównych pętli(1 obsługująca rozgrywkę- druga kolejne gry) oraz wielu funkcji. Najważniejszymi z nich jest funkcja zarządzająca pojedynczą rundą oraz funkcja obsługująca licytację. Pierwsza z wymienionych na przemian wywołuje drugą oraz pokazuje kolejne karty w miarę postępu rozgrywki. Funkcja zajmująca się licytacją wywołuje w przypadku rundy gracza (mojej) obsługę zdarzeń oraz podświetlanie przycisków- ogólnie komunikację z użytkownikiem. Jeśli natomiast ruch należy do któregoś z graczy sterowanych przez komputer zostaje wywołana funkcja AI- „sztucznej inteligencji". Komputer potrafi obliczyć sobie szansę na wygraną przy danym ułożeniu kart, fazie gry oraz liczbie graczy. Aby nie być przewidywalnym AI potrafi zablefować (generowane losowo) oraz podbija na różny sposób przy tej samej szansy na wygraną, aby nie być zbyt przejrzystym co zdecydowanie ułatwiłoby użytkownikowi (mi) odgadnąć, czy komputer ma lepsze karty ode mnie i odpowiednio zareagować. Liczeniem szansy na wygraną zajmował się Andrzej i w dalszej części raportu opisze po krótce jak to robił. Dużym problemem stało się dla nas porównywanie ułożeń kart ze sobą, aby po pokazaniu piątej karty na stole można było wyłonić zwycięzcę. Sytuacja była o tyle trudniejsza, że dla każdego gracza należało najpierw wybrać 5 z 7 kart a następnie porównać najlepsze układy wszystkich graczy będących w grze. Tę przeszkodę rozwiązałem pisząc gro funkcji typu karta z poszczególnymi ustawieniami. Funkcja na wejściu otrzymywała wektor 5 kart, a na wyjściu zwracała pustą kartę jeśli nie było w danym zbiorze kart zadanego ustawienia lub najwyższą kartę w ustawieniu w przeciwnym wypadku. W ten sposób miałem już możliwość jakiegoś porównania układów kart ze sobą. Następnie przeciążyłem dwuargumentowy operator „<", który zwracał mi czy pierwszy układ kart jest lepszy od następnego. Mając możliwość dokładnego porównania ze sobą dwóch układów kart napisałem algorytm sortowania bąbelkowego, który sortował tablicę wektorów wszystkich możliwych kombinacji zbioru 5 elementowego zawartego w zbiorze 7-elementowym. Najlepszy z tych kombinacji był układem optymalnym dla danego zawodnika. Następnie, aby porównać ze sobą graczy napisałem funkcję, która przydzielała graczowi punkt za każdego innego gracza od którego jest lepsza. W ten sposób jednoznacznie określiłem kto wygrał (być może ex equo) oraz dalsze miejsca co jest kluczowe przy rozdawaniu pieniędzy za rundę. Gracz nie może dostać więcej pieniędzy niż iloczyn pieniędzy z którymi wszedł i graczy grających w danej rundzie. Rozdawaniem pieniędzy zajmuje się inna funkcja, która respektuje wszystkie zasady gry w pokera. W programie pojawia się wiele funkcji pomocniczych określających np. ile aktualnie jest grających graczy, ile weszło All-inem, ile

pieniędzy może dany gracz maksymalnie wygrać. Sporą część objętości zajmują funkcje wczytujące tekstury z plików, oraz ustawiające sprajty na odpowiednich miejscach na ekranie. Cały kod składa się z dwóch części: pisanej przez Andrzeja Hermana, oraz pisanej przeze mnie(Michała Kucharczyka). Pisaliśmy na nasze nieszczęście całkowicie osobno przez co niektóre problemy rozwiązywaliśmy niezależnie, a więc dwukrotnie. Następną przeszkodą było dopasowanie do siebie części układanki. Andrzeja funkcja, serce pomyślunku AI przyjmowała argumenty w kompletnie innym formacie niż ja to miałem przewidziane. Nie były to kwestie czysto kosmetyczne jak zamiana tablicy na wektor, lecz poważniejsze typu: zamiast podania 7 kart funkcja przyjmuje 5, ale konkretnie wybranych. Na szczęście przy użyciu kilku funkcji „konwertujących" udało nam się scalić nasze kody. Jak to zwykle bywa w świecie informatyków nie było wyzwaniem napisanie tego programu. Wyzwaniem było szukanie, znajdowanie i niwelowanie błędów. Wyzwaniu mam nadzieję podołaliśmy, bo gra zdaje się chodzić bez zarzutu. Pisząc kod staraliśmy się to robić przejrzyście, aby nie pogubić się w oznaczeniach oraz dla ułatwienia czytaniu go przez osobę z zewnątrz.

Czego się nauczyliśmy?

A Dudek:

Moją „działką" w projekcie była grafika. Robiłem stół, postacie, napisy i wszystko związane z wyglądem wizualnym naszej gry. Całość obrabiałem w programie paint.net. Jest on łatwy w obsłudze, ma prosty interfejs, ale można robić w nim naprawdę przydatne i dobrze wyglądające grafiki. Zaskakująco przydatną opcją, z którą wcześniej nie miałem do czynienia jest funkcja warstw. Pozwala ona zachować ład i porządek podczas edytowania poszczególnych elementów w grze. Za sprawą tego, że często zmieniałem stworzone grafiki była to opcja, dzięki której zaoszczędziłem naprawdę mnóstwo czasu.

Moją pracę rozpocząłem od stworzenia tła. Jako, iż nie chciałem żeby było ono jednolite i nudne, zdecydowałem, że będzie ono zawierało fraktal oraz będzie ciemniało wraz z oddalaniem się od środka ekranu. Myślę, że dało to dobry efekt. Następnie zająłem się tworzeniem stołu oraz przycisków, co było nieco żmudną pracą. Jako, że każdy z nas miał nieco inną wizje projektu, stworzenie postaci było najbardziej problematycznym fragmentem mojej pracy. Początkowo postacie były przedstawione jako ludziki na fotelach, co jednak nie przypadło do gustu moim kolegom i zamieniłem je na powszechnie znane postacie okręgach zmieniające barwę w zależności od gry zawodnika (jest w grze/jest jego tura/zrzucił karty). Grafiki zazwyczaj pobierałem z internetu, lecz w niektórych przypadkach musiałem sam narysować potrzebne nam elementy jak stół czy żetony.

Najbardziej problematyczną częścią grafiki jest monotonia pracy i irytacja sporą ilością niezrozumiałych błędów. Nauczyłem się jednak wielu przydatnych rzeczy m.in. :

- obsługi programu paint.net z którym wcześniej nie miałem styczności

- tego, że nie można pobierać dużego obrazu i zmniejszać go, gdyż wyjdzie niewyraźny (np. z 1200x800 do 120x80 pikseli)

A Herman:

Andrzej Herman

Podczas tworzenia projektu moim głównym zadaniem było tworzenie funkcji mówiących o ustawieniu kart oraz późniejsze wykorzystywanie ich w grze (szacowanie prawdopodobienstwa wygranej, wsakzywanie zwycięzcy itp.). Tworzyłem struktury, tablice i wektory struktur, które następnie używałem w funkcjach. Aby program działał tak jak oczekiwałem, musiałem zdobywać różne informacje o c++ w internecie lub od znajomych, na przykład o strukturach, klasach. Osobiście bardzo cieszę się, że napisałem bardzo dużo przeróżnych pętli, tablic i wektorów, gdyż będę tego używał na maturze. Dodatkowo musiałem dogłębnie poznać zasady gry i wszystkie kontrowersyjne przypadki. Komputer aby grał jak najlepiej potrzebował sprawnego systemu obliczającego prawdopodobieństwo zwycięstwa i systemu informującego czy opłaca mu się grać dalej, czy zrezygnować. Prawdopodobieństwo szacowałem za pomocą schematów takich jak np. cztery karty do koloru, para itp. Znaczną część pracy włożyłem w zaobserwowanie jak zmieniają się szanse w zależności od układu kart na stole, ręce gracza czy ilości graczy pozostałych w grze, na stronie internetowej:

https://pl.pokernews.com/poker-tools/poker-odds-calculator.htm , skąd również czerpałem wiele informacji na temat rozgrywki. Szanse na zwycięstwo zależą od wielu czynników: od rundy, liczby graczy, dwóch kart gracza oraz kart które się pojawiły na stole. Funkcja szacująca prawdopodobieństwo na początku uwzględnia rundę. Następnie po kolei sprawdza czy gracz posiada jakieś ustawienie inne niż na stole, tzn: jeżeli gracz posiada dwie pary, ale na stole są też dwie takie same pary, prawdopodobieństwo na wygraną nie zwiększa się. Jeżeli natomiast na stole jest jedna para lub nie ma żadnej, prawdopodobieństwo wzrasta w odpowiedni dla rundy sposób. Następnie szanse zmieniają się, bo uwzględniana jest liczba graczy pozostałych w grze, w momencie szacowania. Na koniec otrzymujemy oszacowane szanse na zwycięstewo w procentach, które są wystarczająco bliskie rzeczywistości, aby komputer odpowiednio zaprogramowany potrafił podjąć odpowiednią dla sytuacji decyzję. W Programie tworzyliśmy skomplikowane funkcje dla odpowiednich operatorów, które umożliwiają np. porónanie układów kart lub wskazanie najlepszego układu pięciu kart z wszystkich siedmiu na stole i na ręce gracza. Znaczna część mojej pracy polegała na szukaniu poszczególnych błędów w programie, zazwyczaj związanych z porównywaniem układów kart. Porównywałem dogłębnie wiele układów, dochodząc dzięki nim do czasem nawet kosmetycznych błędów, które jednak miały znaczenie na końcowy rezultat. Jasny i przejrzysty kod ułatwiał mi tę pracę (o ile w ogóle jej nie umożliwiał). Pisząc program używałem takich form jak:

-Komentarze w c++

-Przestrzeń nazw std

-Instrukcja warunkowa if ... else

-Pętle, wychodzenie z pętli - break

-Losowanie liczb

-Tworzenie struktur

-Przekazywanie tablic jednowymiarowych do funkcji

-Zmienne przechowujące tekst

-Programowanie obiektowe

- Przekazywanie wektorów do funkcji

-obsługa plików pdn

Z większością z nich zapoznawałem się od zera, a część musiałem bardzo znacznie sobie przypomnieć. Uważam, że projekt w bardzo mocnym stopniu rozwinął mnie, przygotował do matury z informatyki i zainteresował tematem.

M Kucharczyk:

Ten projekt dał mi okazję na spróbowanie informatyki z zupełnie innej strony. Zwykle rozwiązywałem jakieś problemy w zadaniach, na które z góry wyznaczony był sposób rozwiązywania. Tutaj problemy pojawiały się same, a sposobów na ich rozwiązanie również było dużo. Praca nad takim projektem cieszy, bo z każdym dniem widać postępy i efektem tego wysiłku jest program, w który będę z chęcią grał.

Pisząc tak długi kod na pewno nabrałem wprawy w pisaniu, oraz rozwiązywaniu pewnych problemów z marszu. Nabrałem szerszej perspektywy jeśli chodzi o patrzenie na program. Nauczyłem się myśleć bardziej dalekowzrocznie, tworzyć najpierw projekt-szkielet całego programu, a dopiero później pisać na komputerze kod. Zdobyłem wiele umiejętności z zakresu informatyki pozaszkolnej- poznałem wiele narzędzi programistycznych takich jak:
-struktury
-przeciążanie funkcji i operatorów
-listy(wykorzystywane w pierwotnej wersji programu)
-wektory, pary
-przekazywanie zmiennych do funkcji poprzez kopię oraz referencję
- biblioteka graficzna SFML, która daje ogromne możliwości w zakresie tworzenia gier
Cały program staraliśmy się pisać w myśl cytatu „Dziel i zwyciężaj” co dało zaskakujące efekty, ponieważ nigdy nie uważaliśmy się za zdolnych stworzenia czegoś użytecznego z zakresu informatyki. Mimo złego nastawienia udało nam się jednak stworzyć grę, w którą mamy nadzieję użytkownicy będą grali z przyjemnością.

W skład twórców projektu wchodzą:

Adam Dudek 3A

Andrzej Herman 3A

Michał Kucharczyk 3A