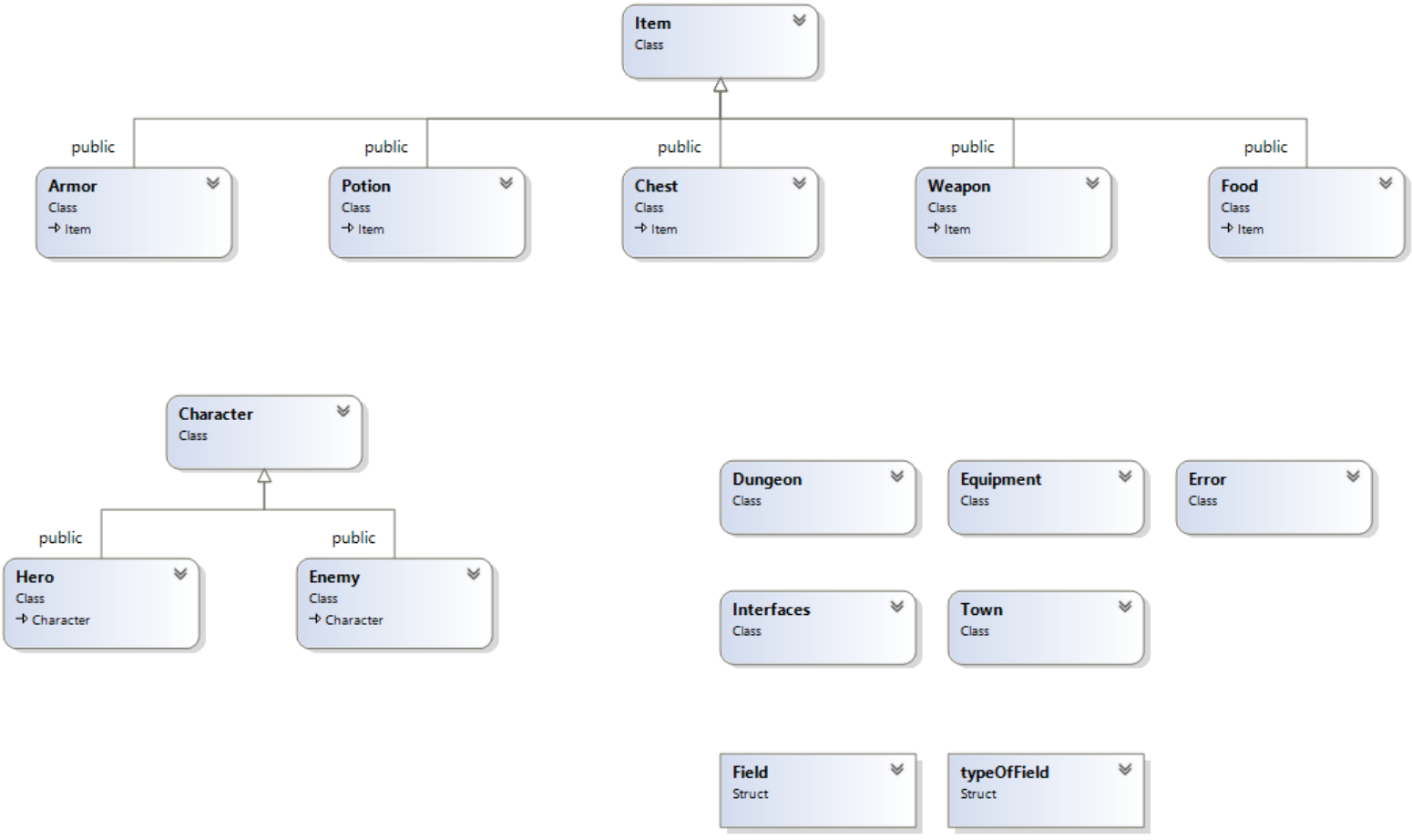
Imię nazwisko: Jakub Woźniak
Kierunek, grupa: Informatyka Stosowana, 09
Sprawozdanie z projektu Programowanie Obiektowe.

1. Wstęp (opis projektu)

Opisywany projekt (Darker Dungeon) to gra typu dungeon crawler z elementami gry RPG. Jest ona po części wzorowana na grze Darkest Dungeon. Program posiada funkcję nowej gry i wczytania poprzedniej. Nowa gra zaczyna się od stworzenia bohatera, wyboru jego imienia i rasy. Następnie wybieramy nazwę miasta dla stworzonej rozgrywki. Ekran główny to panel miasta z wyborem akcji. Do odwiedzenia mamy różne obiekty które w zależności od podjętej decyzji mogą zmienić nasz ekwipunek (handlarze zbroi, broni, mikstur życia, jedzenia). Możliwość zapisu gry tylko w momencie przebywania w mieście. Kolejną z opcji jest eksploracja podziemi. Do wyboru jest trzy rodzaje z nich: łatwy, średni i trudny. Sukces dla zbadania każdego z nich polega na znalezieniu wszystkich skrzyni ze złotem odpowiednio: 10, 20, 50. Po zdobyciu wszystkich skrzyni zwiększa się licznik punktów gry o: 12,22 lub 32.Na mapie występują po dwa rodzaje przeciwników na każdy poziom: szkielet i zombie, duch i wampir, lisz i kościany smok. Gra kończy się gdy gracz uzyska wymaganą ilość punktów (600, wcześniej założeniem było pokonanie głównego przeciwnika, z walką dostępną od 500 punktów).

2. Diagram klas:



3. Opis klas, metod, funkcji globalnych, użytych bibliotek.

Zmienne globalne: GamePoints – zmienna z ilością punktów zwiększająca się po każdym udanym zbadaniu podziemi**, GameDays:** zmienna przechowuję ilość dni jakie spędził na rozgrywce bohater.

Plik nagłówkowy lib.h zawierający biblioteki iostream, string, time.h (losowanie), vector (dla klas Potion i Food), algorithm (przy odcz/zap pliku zamienia funkcja replace), Windows dla ustawienia okna początkowego i jego wymiarów.

Klasa Error: klasa z kontrolą błędów, funkcja errorWhat() zwracająca treść błędu.

Klasa Character: klasa abstrakcyjna przechowująca zmienne wspólne dla klas Hero i Enemy (name, race, health, stamina, attack, defence, dexterity) oraz funkcje związane z walką (quickAttack, powerAttack, defend, checkStamina, checkHealth, wait, dogde, suffer, isDefeated). Deklaracja przyjaźni z klasą Town.

Klasa Enemy: klasa odpowiedzialna za tworzenie przeciwników, dodatkowa zmienna przechowująca wartości doświadczenia i czy obiekt został pokonany. Zawiera funkcje odpowiedzialne za stworzenie konkretnego przeciwnika. Akcje: walka i przeładowanie operatora << w celu jego wyświetlenia.

Klasa Hero: przechowuję dodatkowo zmienne odnośnie poziomu postaci i jego procesu levelowania (experience_points, hpEveryLVL, stEveryLVL, level, maxHealth, maxStamina) oraz obiekty typu Equipment, Weapon i Armor. Funkcje: eat/health zwracające odpowiednią ilość punktów życia, goldAmount ilość złota, buy funkcja kupowania ale tez podnoszenia złota (wartość -), changeWeapon/Armor zmiana broni/zbroi. butPotion/Food kupowanie mikstur/jedzenia. Funkcje dla ekranu miasta: sleepInInn (życie wytrz. na maks. wart.) i sleepInCity (życie i wytrz. +50). Funkcja maxHPandST zwracająca prawdę jeśli wartości maksymalne dla health i stamina. Funkcja potions/foodCounter zwracająca wartość całkowitą odpowiednio misktur/jedzenia w ekwipunku. Operator przeładownia w celu wyświetlania <<.

Klasa Equipment: złożona ze zmiennych gold - ilość złota, wektory packOfFood i packOfPotion przechowujące ilość obiektów Food i Potion. Funkcje: addPotion i addFood zwiększa ilość o jeden. Funkcje: useFood usePotion zmniejszają ilość o jeden zwracają odpowiednią ilość punktów zdrowia, goldAmount zwraca ilość złota , sizeOfFood/PotionPack ilość obiektów, klasa buy zmienia wartość gold.

Klasa Item: klasa abstrakcyjna przechowująca zmienne łańcuchowe itemName (nazwa) i itemType (opis). Funkcja getName i getDescr zwracająca obiekty itemName i itemType.

Klasa Potion: zmienne całkowite price i healthRecover funkcja usePotion zwracajaca wart. healthRecover

Klasa Food: zmienne całkowite price i healthRecover funkcja eatFood zwracająca wart. healthRecover.

Klasa Weapon: klasa broni, zmienne całkowite price i baseAttack zmienna double weight. Funkcje presentWeapon wyświetla statystyki obiektu, getPrice, getAttack i getWeight zwraca wartość obiektów price, baseAttack i weight.

Klasa Armor: klasa broni, zmienne całkowite price, baseDefence, stamina, hp . Funkcje presentArmor wyświetla statystyki obiektu, getPrice, getDefence getStamina getHP zwraca wartość obiektów price, baseDefence,stamina,hp.

Klasa Chest klasa przechowująca wartość typu int goldInBox i bool opened. Funkcja isOpened zwracająca wartość opened, gold zwracająca wartość goldInBox, openTheChest zwracajaca wartość goldInBox

Klasa Interfaces klasa odpowiedzialna głownie za wyświetlanie komunikatów, najczęściej panelów rozgrywki z wyświetlonym menu na ekranie funkcje: printStart, printNewGame, printgameOver, printEndGame, printLoadGame, printSaveFile, printMerchant wyświetla trzy dostępne losowo wygenerowane zbroje które bohater może kupić, printBlacksmith podobnie jak Merchant tylko dla broni, printMedic, printInnkeeper podobnie, printSleepInSlums zmienia wartości health i stamina dla obiektu Hero, printDungeon menu rozgrywki w wyborze stopnia trudności podziemi, printFight wyświetla menu walki.

Klasa Town – zmienna łancuchowa nameOfCity i zmienna Hero mainHero. Funkcje Game odpowiedzialna za tworzenie i trwanie rozgrywki, visitMerchant/ Blacksmith/ Medic/ Innkeeper/ Dungeons i sleepInSlums odpowiedzialne za wyświetlenie odpowiedniego wyboru i przejście do niego. Funkcja save zapisująca aktualny stan gry (ekwipunek bohatera, broń, zbroję, statystyki bohatera, nazwe miasta i wartości zmiennych globalnych). Funkcja load odczytująca te wartości z pliku i tworząca po kolei kolejne obiekty.

typeOfField – struktura przechowująca wartości typu bool visited, hero, border i znak char what

Field – struktura przechowująca zmienna typu typeOfField type, Enemy *en, Chest *chest

Klasa Dungeon – zmienna Field **map, Enemy *group1, *group2, Chest *chests. Konstruktor klasy tworzy tablice 2d typu Field. Ustawia pozycje bohatera w lewym dolnym rogu mapy symbol @, jeśli bohater zbadał już dane pole jest ono oznaczane symbolem V. Mapa jest otoczona polami dla których wartość typeOfField.border jest prwadziwa- funkcja setBorders. Na mapie losowo umieszczane są dwie grupy przeciwników i skrzyni przy czym dla każdego z nich sprawdzana jest wartość dla pól en i ch aby nie było dwóch obiektów na jednym polu funkcja setChestsAndEnemies. Obiekty te na mapie zaznaczone sa znacznikiem ?. Klasa ta posiada następujące funkcję: printDungeon wyświetla zmienna map[x][y].type.what i menu dla poruszania się i użycia mikstury jedzenia. Funkcja moveHero zwraca wartość bool czy możliwy jest ruch bohatera w danym kierunku, isEvent również zwraca wartość bool w zależności czy pole zawiera skrzynię bądź przeciwnika. Kolejne funkcje to ExploreDungeon dla której odbywa się sterowanie postaci w zależności od podjętej akcji dochodzi do braku wywołania w funkcji, wywołania funkcji związanej z otwarciem skrzyni lub funkcja walki obiektu Hero i obiektu Enemy. Funkcja zwraca wartość typu całkowitego (1 pokonanie przeciwnika, 0 ucieczka z walki i tym samym z podziemi, 2 porażka, śmierć bohatera ekran porażki) przyjmuje obiekt Hero i Enemy. Funkcja Fight trwa do momentu gdy hp dla którejś z postaci nie spadnie albo będzie równe 0.

4. Podsumowanie projektu:

a) Co zostało zrealizowane

Tworzenie bohatera z kontrolą błędów. Proces levelowania postaci ze zwiększaniem atrybutów zależnych od rodzaju wybranej rasy. Losowe generowanie przedmiotów broni i zbroi u kupców. Leczenie przy pomocy użycia mikstur i jedzenia a także spania w karczmie czy na ulicy miasta. Losowe generowanie statystyk przeciwników i zawartości złota w skrzyni. Losowe generowanie każdego z poziomów podziemi.

b) Wnioski z realizacji projektu

Większość założonych celów została zrealizowana. Program został napisany w sposób obiektowy. Gra jest do pewnego stopnia grywalna. Jedna z opcji która nie została ukończona jest finałowa walka. Moment zakończenia gry został zmieniony tak by po odpowiedniej ilości punktów gra kończyła się sukcesem. Przeciwnicy, przedmioty są niezbalansowane. Występuje bug przy wczytywaniu gry włącza się tworzenie postaci, które później nie jest uwzględniane.