**Crystal Of Exchange**

****

**Projekt Kompetencyjny**

**Grupa:**

* **Dominik Rak**
* **Jakub Zawiasa**
* **Maciej Kopka**
* **Paweł Wolski**

**Założenia projektowe**

Celem projektu było stworzenie gry platformowej 2D z wykorzystaniem silnika gry UNITY oraz języka C#.

Gra nosi tytuł Crystal Of Exchange czyli Kryształ Zamiany. Wykorzystuje ona pomysł, w którym bohater potrafi przemieniać się w różne postacie z różnymi zestawami umiejętności.

W naszej grze bohater potrafi zamieniać się w maga, wojownika oraz łucznika.

Gra będzie zawierać 3 poziomy kolejno: las, piekło oraz krainę lodu. Dodatkowo po przejściu każdego poziomu będziemy musieli się zmierzyć z mocniejszym przeciwnikiem, bossem.

**Wykorzystane narzędzia i technologie**

**Silnik gry UNITY**  
Pozwala na stworzenie gier 2D oraz 3D. Dostarcza narzędzia do implementowania fizyki, animacji, skryptów, interfejsu, interakcji między obiektami, zdarzeniami oraz wielu innych komponentów.

**Visual Studio Code / Microsoft Visual Studio**Edytory kodu umożliwiające nam pracę w języku C# wraz z synchronizacją UNITY API.

**Doxygen**Generowanie dokumentacji do skryptów wykorzystując adnotacje w kodzie w formacie JAVA DOC.

**Adobe Illustrator, Adobe After Effect, Adobe Photoshop**Oprogramowanie graficzne umożliwiające nam stworzenie grafik oraz animacji.

**UNITY Cloud Collaborate**Implementacja GIT wbudowana w UNITY pozwalająca na bezpośrednie zarządzanie projektem z poziomu aplikacji UNITY oraz na udostępnienie projektu pomiędzy zespołem.

**Wykorzystane komponenty silnika gry UNITY**

**Transform** – przetrzymuje informację o pozycji, rotacji i skali obiektu oraz pozwala na zmienianie każdej z tych właściwości.

**RigidBody 2D –** komponent odpowiedzialny za fizykę obiektu, pozwala na zarządzanie masą, działaniem grawitacji, sposób detekcji kolizji (dyskretnie lub ciągle), blokowanie rotacji na każdej z osi oraz na wiele pomniejszych opcji.

**Collider 2D –** komponent, który wyznacza sferę podlegającą kolizji z innymi obiektami zawierającymi Collider. Kształt Collidera oraz jego rozmiar możemy ustalić z poziomu aplikacji UNITY.

**Sprite Renderer –** komponent wyświetlający grafikę przypisaną do obiektu oraz pozwalający na zmienianie sposobu wyświetlania

**Animator –** komponent zarządzający animacjami, w którym definiujemy stany oraz momenty przejścia pomiędzy nimi. Można w nim utworzyć parametry, które będą warunkami przejść oraz którymi możemy zarządzać z poziomu skryptów.

**Animation –** animacja, którą możemy utworzyć za pomocą serii grafik, nagrać ją z poziomu tworzenia sceny bądź zdefiniować klatka po klatce wykorzystując każdy z dostępnych komponentów podpiętych pod animowany obiekt. Na przykład animacja obiektu posiadająca komponent Transform, może zmieniać parametr Scale powodując zmniejszanie się lub zwiększanie się obiektów.

Animacja pozwala również na wywoływanie zdarzeń w konkretnych klatkach, które mogą uruchamiać funkcję ze skryptów.

**Audio Source –** źródło dźwięku, do którego podpinamy plik dźwięku. Umożliwia uruchomienie dźwięku w momencie gdy obiekt zostaje uruchomiony, w dowolnym momencie działania gry z poziomu gry, na zapętlenie dźwięku, głośności, efektów dźwiękowych, zasięgu dźwięku i tym podobnych.

**Tilemap –** komponent tworzony z grafik, za pomocą którego można „rysować” otoczenie wykorzystując narzędzie Tile Pallete.

**Canvas –** obiekt pozwalający na tworzenie interfejsu użytkownika z wykorzystaniem komponentów takich jak : **Button, Text , Image, Slider, Toggle, Panel.** Może być interaktywna lub nie.

**Canvas Group –** komponent podpinany pod Canvas pozwalający na ustalanie przezroczystości Canvas.

**Camera –** Obiekt, który jest naszym okiem na grę, możemy go umieścić statycznie w jednym miejscu bądź za pomocą skryptu podążać za innym obiektem.

**Audio Listener –** komponent podpinany pod kamerę aby ta mogła przechwytywać i odtwarzać dźwięki z obiektów z Audio Source.

**Sposób działania skryptów**

UNITY podczas działania gry uruchamia każdy skrypt podpięty pod aktywne obiekty na scenie.

Każdy skrypt, który działa w UNITY musi dziedziczyć z MonoBehaviour. Jest to obiekt posiadający w sobie funkcje wywoływane automatycznie w odpowiednich momentach.  
Wykorzystywane przez nas funkcję, które nadpisywaliśmy to:

* **Start()** – uruchamia się w momencie, gdy obiekt pojawia się na scenie
* **Awake() –** uruchamia się, gdy obiekt staje się aktywny
* **Update() –** wywoływana jest w każdej klatce podczas działania gry
* **FixedUpdate() –** podobnie do Update(), jednak nie wykonuje się z czasem zależnym od wygenerowania klatki, a ze stałym czasem 0.2s. Używana głównie do operacji związanych z fizyką gry.
* **LateUpdate() -**  identycznie jak Update() ale dopiero, gdy Update() zakończy działanie.

**Skrypty utworzone do zarządzania obiektami w grze**

Szczegółowe informacje o zmiennych oraz metodach w skryptach znajdują się w dokumentacji wygenerowanej za pomocą Doxygen.

**Skrypty przeciwników**

**Audio –** Skrypt, wykorzystywany przez animację do uruchomienia dźwięku podpiętego do skryptu w momencie uruchomienia event’u.

**BigFootController –** Skrypt zarządzający bossem dla Krainy Lodu, Wielką Stopą. Boss będzie wykonywał atak w skoku, rzucał kulą śnieżną, oraz chronił się przed strzałami po utracie połowy zdrowia.

**CollisionDetect –** Skrypt, który sprawdza czy kolizja, która nastąpiła z obiektem była związana z graczem, jeśli tak odpycha gracza o ile ten nie używa tarczy. Jeśli używa przeciwnik się od niej odbija.

**EnemyInteraction –** Skrypt odpowiadający za interakcję przeciwnika z graczem

**EnemyRangedAttack**

**EnemyRangedWeapon**

**EntMove**

**FollowPatrolAI**

**GemsAfterKill**

**InfernoBossMove**

**JumpingEnemy**

**PatrolAI**

**SimpleEnemyInfo**

**Skrypty przedmiotów i otoczenia**

**AidKit**

**Arrow**

**Coin**

**hidingTrap**

**MovingPlatform**

**MovingSaw**

**Saw**

**SpikeController**

**Skrypty Gracza**

**Archer**

**BowShooting**

**CameraFollow**

**CharacterController2D**

**PlayerInfo**

**PlayerMovement**

**RopeScript**

**ThrowHook**

**ThrowQubic**

**Warrior**

**WarriorAttack**

**Wizard**

**Skypty Interfejsu oraz zarządzania grą**

**backgroundMove**

**BackgroundScroller**

**EndLevel**

**Fade**

**GameOverScreen**

**LevelsManager**

**MainMenu**

**SceneLoader**

**Scrolling**

**UIController**