WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA
z siedzibą w Rzeszowie

Dokumentacja projektu

Przedmiot: **Programowanie**

Tytuł projektu: Cube Checkers 3D

Prowadzący:
Dr Marek Jaszuk

Wykonawcy:
Oleh Korsunskyi (w58957),
Nazarii Korniienko (w58975),
Krzysztof Pudysz (w58995)

Grupa: SP02/11

Rzeszów, 2019

TREŚĆ PROJEKTU

1. Założenie projektowe (wymagania biznesowe)

Głównym założeniem naszego projektu jest tworzenie gry komputerowej Warcaby 3D. Program będzie przypominać zwykłe Warcaby z planszą, która składa się z 64 pól, ale z nowym, wymyślonym przez nas designem. Gra będzie przeznaczona dla dwóch użytkowników, którzy potrafią grać na jednym komputerze. Design figur będzie w stylu geometrycznym, a mówiąc dokładniej, poszczególne figury będą sześcianami.

Gra zostanie zbudowana na silniku Unity, który idealnie pasuje do tworzenia prostych gier 3D. Poszczególne modele 3D, które będą wykorzystywane w tym projekcie, zostaną zrobione w programie 3D MAX. Programem dla napisania kodu jest Visual Studio, który łatwo się łączy z Unity. Cały projekt będzie napisany w języku programowania C#. Harmonogram (diagram Gantta) zostanie zrealizowany w środowisku Hygger. Projekt będzie realizowany na podstawie systemu kontroli wersji – GitHub.

Głównym celem tego projektu jest zrozumienie poszczególnych kroków tworzenia aplikacji desktopowych w oparciu o silnik Unity, znalezienie i rozwiązanie problemów, które pojawią się w czasie tworzenia gry. Dodatkowym celem jest nauczyć się pracować w zespole i pomagać sobie nawzajem.

2. Lista wymagań z podziałem na wymagania

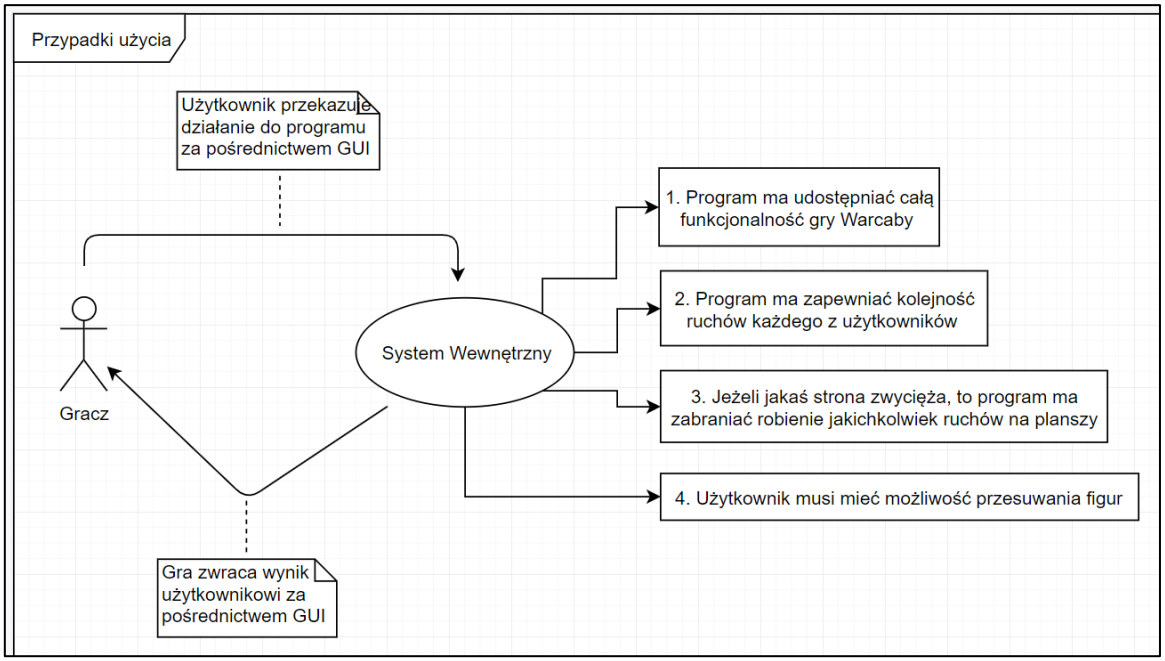
2.1 Wymagania funkcjonalne

1. Program ma udostępniać całą funkcjonalność gry Warcaby (tzn. przestrzegać wszystkich zasad tej gry).
2. Program ma zapewniać kolejność ruchów każdego z użytkowników.
3. Jeżeli jakaś strona zwycięża, to program ma zabraniać robienie jakichkolwiek ruchów na planszy.
4. Użytkownik musi mieć możliwość przesuwania figur za pomocą myszy komputerowej na dostępne pola, ale jednocześnie program ma zabraniać użytkownikowi robienie nieprawidłowych ruchów zgodnie z zasadami gry.

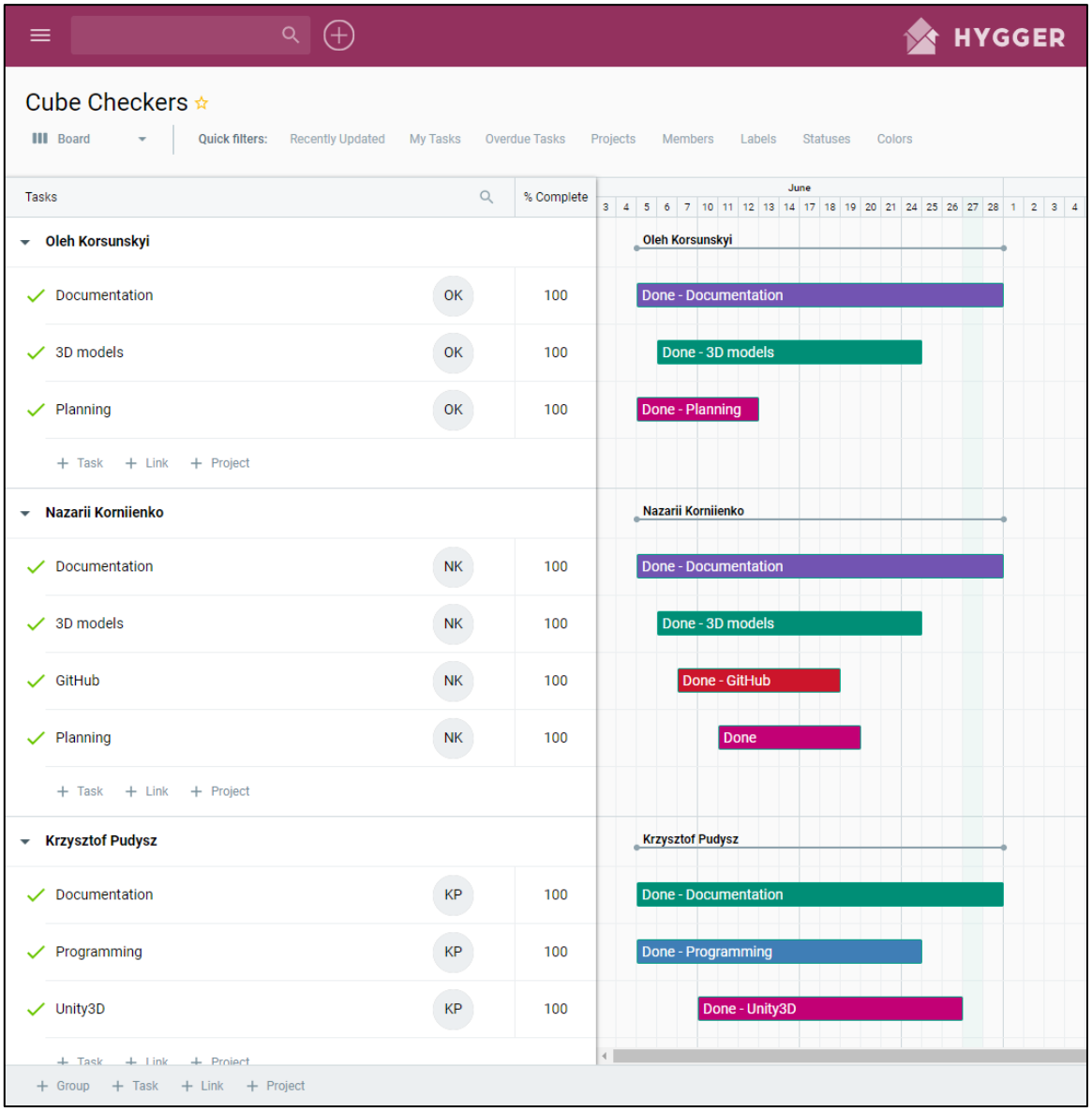
2.2 Wymagania niefunkcjonalne

1. Program ma być wykonany w stylu figur geometrycznych.
2. Plansza i figury mają być przedstawione w dwóch kolorach.
3. Program ma powstać na silniku Unity3D.
4. Interfejsem programu ma być GUI.
5. Program jest przeznaczony dla dwóch użytkownik.
6. Program ma działać na systemie Windows.
7. Dodatkowe oprogramowanie - .Net Framework 4.6.1 lub nowszy.

3. Diagram przypadków użycia z wyróżnieniem aktorów



4. Harmonogram realizacji projektu na diagramie Gantta



5. Opis techniczny projektu

Projekt składa się z dwóch klas, CheckersBoard oraz Piece. Klasa CheckersBoard składa się z 16 metod, natomiast klasa Piece z 2 metod.

5.1 CheckersBoard

Pierwsza metoda to Start. Znajdują się tu wszystkie informacje odnośnie tego co ma się stać po włączeniu programu.

Druga metoda to Update. Znajdują się tu informacje na temat przesuwania figur na planszy oraz możliwość zaznaczania figury i przesunięcia jej w inne miejsce. Ta funkcja również zmienia turę graczy.

Trzecia metoda to UpdateMouseOver. Ta metoda śledzi położenie kursora myszy za pomocą RaycastHit. Z punktu, w którym znajduje się kursor myszy jest wysyłana wiązka laserowa i po kolizji z planszą, na którą jest naniesiony BoxColider odczytywana jest wartość, gdzie znajduje się kursor myszy.

Czwarta metoda to UpdatePieceDrag. Ta metoda również używa RaycastHit, tym razem do przenoszenia figur. Bez tej metody przemieszczanie figur za kursorem myszy byłoby niemożliwe.

Piąta metoda to SelectPiece. Przyjmuje argumenty x, y i jest to położenie figur na planszy. Plansza składa się z pól X od 0 do 8 leżących w poziomie oraz z pól Y od 0 do 8 leżących w pionie. Sprawdzane tu jest czy gracz nie próbuje wyciągnąć jakiejś figury za dozwolone pola planszy oraz czy gracz musi ruszyć jakąś figurą, która ma zbić inną. Sprawia to, że nie może wykonać ruchu innymi figurami.

Szósta metoda to TryMove. Znajdują się w niej głównie informacje na temat ruchu figur. Jeżeli figura nie ruszyła się, to ma wrócić na swoje dawne miejsce. Sprawdza też czy dany ruch jest dozwolony oraz czy jakaś figura nie została zbita podczas ruchu.

Siódma metoda to EndTurn. Sprawdzane jest w niej czy jakaś figura została zbita i czy może wykonać kolejny ruch, jeżeli nie to kończy turę gracza. Również tutaj odbywają się promocje po dojściu figury na koniec planszy – figury mogą poruszać się wtedy również w tył.

Ósma metoda to CheckVictory. Sprawdza czy któryś gracz wygrał.

Dziewiąta metoda to Victory. Jest pomocnicza do poprzedniej metody, wypisuje na ekranie który gracz wygrał rozgrywkę.

Dziesiąta i jedenasta metoda to ScanForPossibleMove. Jedna z metod jest przeciążona. To tutaj dodawane są do listy figury, które muszą wykonać ruch.

Dwunasta metoda GenerateBoard tworzy figury na planszy. Najpierw dla białej drużyny a następnie dla czarnej.

Trzynasta metoda to GeneratePiece robi to samo co dwunasta metoda z tą różnicą, że tworzy modele figur w miejscu, gdzie mają się znajdować.

Czternasta metoda to MovePiece. Na podstawie informacji co to za figura, ustawia ją w odpowiednim położeniu tak, aby na początku gry znalazła się na swoim polu.

Piętnasta metoda to Highlight. Podświetla figury, które muszą wykonać ruch, co sprawia, że gracz łatwiej może zobaczyć figury którymi ma się poruszyć

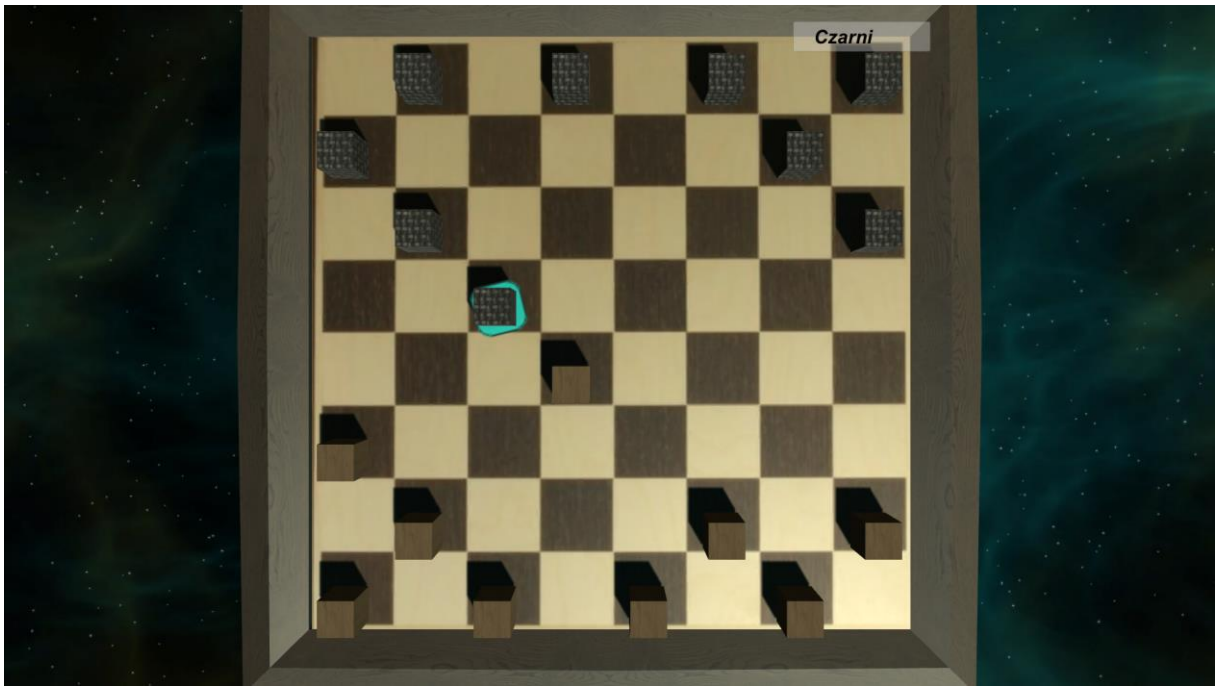
Szesnasta metoda to Alert. Powiadamia gracza o tym czyja jest tura.

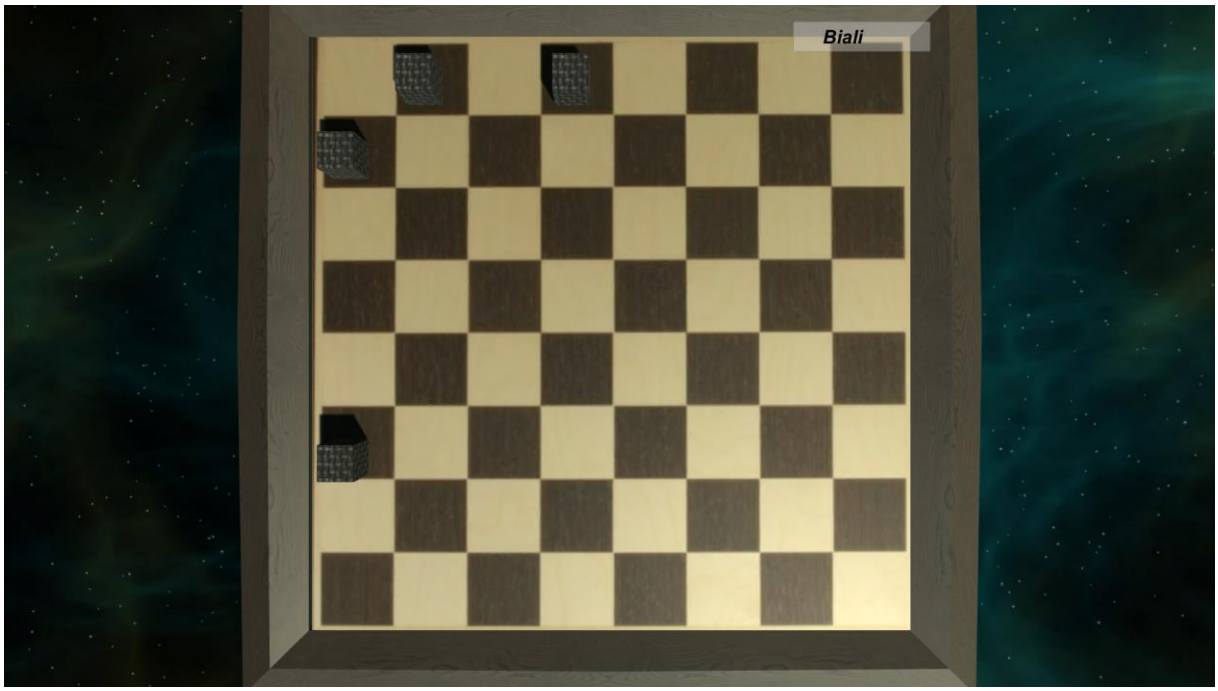
5.2 Piece

Pierwsza metoda to IsForceToMove. Dzieli planszę na cztery części w których sprawdzane są warunki czy figura obok nas nie jest tego samego koloru oraz czy miejsce, w które chcemy się przenieść po zbiciu figury jest wolne.

Druga metoda to ValidMove. Znajduje się w niej upewnienie, że gracz nie może przenieść figury na inną figurę oraz warunki czy ruch, który został wykonany jest dozwolony czy też nie.

6. Prezentacja warstwy użytkowej projektu





Użytkownik może przestawiać figury za pomocą myszy komputerowej, ma nacisnąć na figurę i przeciągnąć w dostępne pole.

7. System kontroli wersji

https://github.com/uitmers/checkers

8. Materiały źródłowe

1. https://www.youtube.com/playlist?list=PLLH3mUGkfFCVXrGLRxfhst7pffE9o2SQO (tutorial na YouTube) (28.06.2019).

9. Dokumentacja w Html Doxygen

https://github.com/uitmers/checkers/tree/master/WarcabyDesktop/Doxygen%20Warcaby