

# Projekt - Calico

## Etap 3

APRO2 2023L

Stanisław Ciszewicz, Stanisław Dolinski, Michał Kozłowski, Jan Sakowski

Politechnika Warszawska, Cyberbezpieczeństwo

16 czerwca 2023

### Spis treści

<b>1. Ocena poprzedniego etapu</b>	2
<b>2. Etap III</b>	2
2.1. Welcome stage	2
2.2. Lobby stage	3
2.3. Początek rozgrywki	4
2.4. Rozgrywka	4
2.5. Zakończenie rozgrywki	5
<b>3. Podsumowanie etapu</b>	5

## 1. Ocena poprzedniego etapu

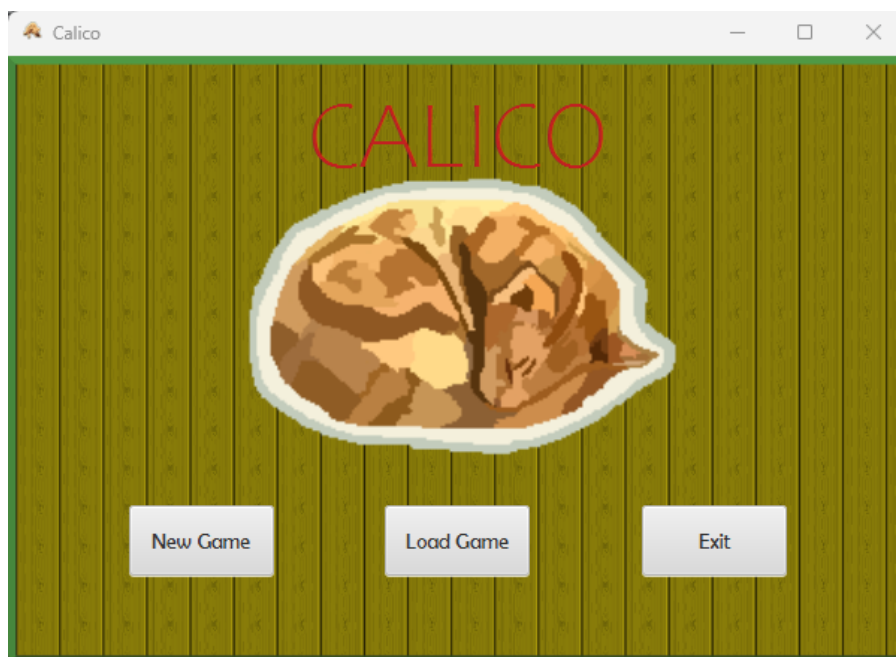
Przed przystąpieniem do pracy nad tym etapem dokonaliśmy analizy pracy wykoanej przez poprzednią grupę. Pierwszego przeglądu i wstępnej powierzchownej analizy dokonaliśmy 3 dni po przejęciu repozytorium od poprzedniego zespołu (21 maja 2023 roku). Pierwszą rzeczą, która rzuciła nam się w oczy była przejrzystość, jakość i stopień przemyślenia każdej z metod wykonanej w poprzednim etapie. Kod był zrozumiały, a w razie jakichkolwiek wątpliwości członkowie poprzedniego zespołu służyli nam pomocą. Dodatkowo, w kodzie znajdowały się testy jednostkowe głównych metod testujące różne warianty właśnie odbywającej się rozgrywki. Były one uwiarygodnieniem poprawności działania gry - pryncypalnie nie trafiliśmy podczas relizacji GUI na żadne poważniejsze bugi. Niestety, od samego początku zwróciliśmy uwagę na jeden dość istotny mankament - brak działającej klasy serwer i klient (jednego z głównych założeń poprzedniego etapu, bez której niemożliwa była praca nad wersją online). Zwróciliśmy się tym samym z prośbą do poprzedniego zespołu (wpier ustną, następnie za pomocą sekcji issues w repozytorium projektowym). Tym samym zgodnie z zaleceniem dr. Krystiana Ignasiaka, po ponownym zgłoszeniu problemu w issues, zrezygnowaliśmy z relizacji wersji sieciowej rozgrywki i skupiliśmy się jedynie na wersji offline. Warto jednak wspomnieć, że poprzedniemu zespołowi udało się ostatecznie zaimplementować klasy serwer i klient 10 czerwca 2023 roku - było to jednak zdecydowanie zbyt późno, aby móc w pełni zrealizować założenia III etapu (cała historia tego problemu znajduje się w naszym repozytorium w zakładce issues). Dodatkowo, musieliśmy na nowo zaimplementować metody zapisu i wczytywania stanu gry - poprzednia grupa zapisywała stan gry poprzez serializację obiektów, my zdecydowaliśmy się na zapis do pliku JSON, który jest bardziej efektywny i w prostszy sposób mogliśmy z niego sczytywać potrzebne dane. Podsumowując, praca poprzedniej grupy pozwoliła w łatwiejszy sposób przejść do relizacji interfejsu graficznego - mogliśmy w szybszy sposób zrozumieć specyfikę i logikę kodu, nie tracąc czasu na domysły. Mimo wszystko, projekt nie zawierał wszystkich elementów składających się na wymagania II etapu, co nie pozwoliło nam w pełni zrealizować naszej części zadania.

## 2. Etap III

W tym etapie zrealizowaliśmy implementację interfejsu graficznego do gry Calico. Na tym etapie gra posiadała zaimplementowaną logikę, którą należało połączyć z efektem pracy obecnego etapu. Do implementacji interfejsu graficznego użyliśmy biblioteki JavaFX - głównie dlatego, że członkowie naszego zespołu mieli z nią styczność w poprzednich projektach. Sam interfejs graficzny był testowany ręcznie (poprzez granie w grę i szukanie błędów), gdyż testy jednostkowe są niezwykle trudne do zaimplementowania w przypadku złożonej wieloscenowej aplikacji. Cała dokumentacja wszystkich klas, z których korzystamy z GUI, jest zawarta w javadoc. W tym rozdziale omówimy sceny i rozwiązania zaprojektowane przez nasz zespół.

### 2.1. Welcome stage

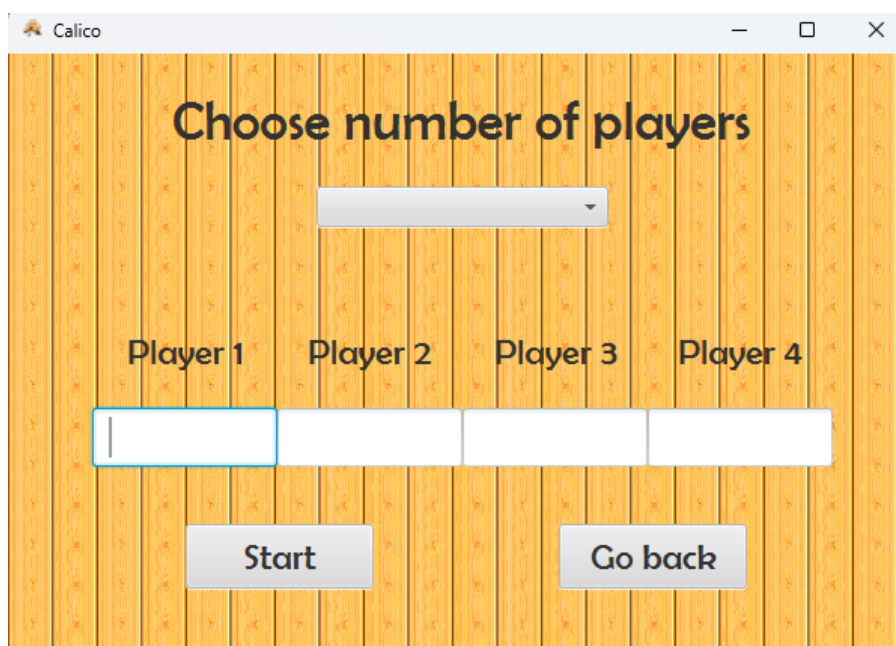
Pierwszą sceną ukazującą się graczom po uruchomieniu rozgrywki jest scena tytułowa. Znajdują się na niej trzy główne przyciski: New game, Load Game i Exit. Po naciśnięciu Load Game powinniśmy móc załadować planszę w formacie JSON (z powodu braku implementacji przez poprzednią grupę, musieliśmy w ostatniej chwili napisać cały kod nowego zapisu). Ostatecznie pokazuje się opcja wybrania pliku, jednak najprawdopodobniej niewłaściwie działa sam zapis do pliku. Po naciśnięciu przycisku EXIT gra zapyta czy na pewno chcemy zamknąć okno, a następnie gra zostanie zamknięta. Po naciśnięciu przycisku New Game zostaniemy przeniesieni do kolejnej sceny.



Rys. 1.

## 2.2. Lobby stage

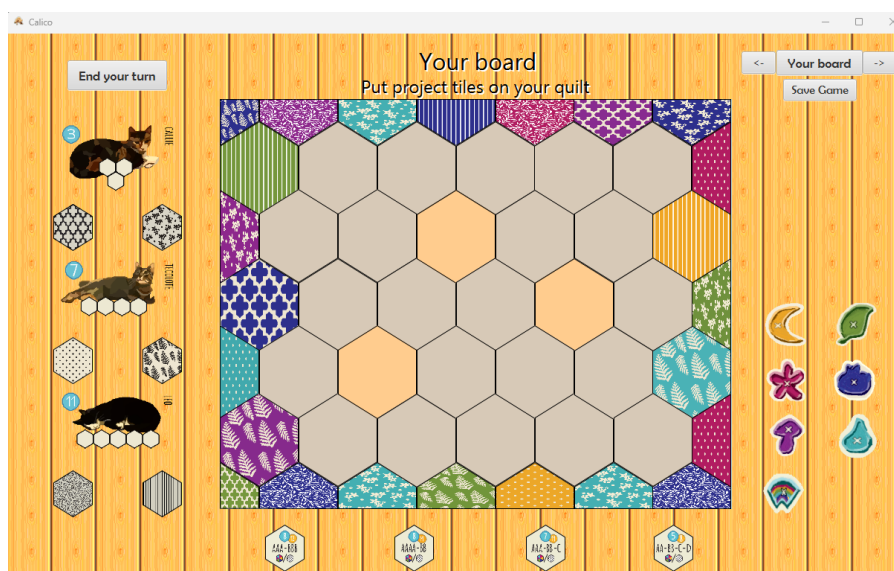
Po wybraniu przycisku New Game zostajemy przeniesieni na stronę, na której możemy wybrać liczbę graczy oraz nadać im imiona. Po wpisaniu danych naciskamy przycisk i rozpoczynamy rozgrywkę. Możemy również wrócić do poprzedniej strony - po naciśnięciu przycisku Go back.



Rys. 2.

### 2.3. Początek rozgrywki

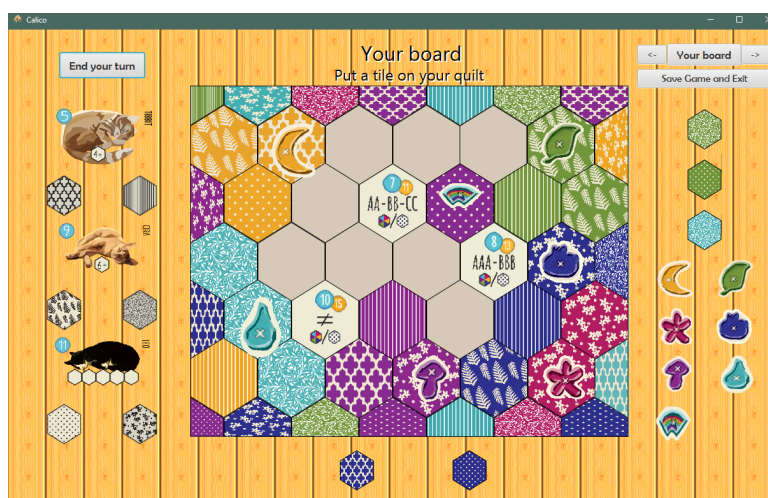
Po rozpoczęciu rozgrywki uruchamia się gra. Każdy z graczy ma swoją indywidualną planszę, jednak dla urealistycznienia rozgrywki dodaliśmy opcję poruszania się po planszach innych graczy (prawy górny róg) oraz powrotu do swojej planszy (gracza, który w tym momencie wykonuje ruch). W pierwszym kroku po wylosowaniu planszy gracz musi ułożyć na swojej planszy 3 z 4 wylosowanych na dole ekranu Project Tiles - po wykonaniu swojego ruchu gracz musi nacisnąć przycisk End your turn. Po wybraniu project Tiles gracz rozpoczyna rozgrywkę według zasad określonych w instrukcji gry. W prawym górnym rogu ekranu widnieje również przycisk Save game, jednak z powodu braku implementacji w poprzednim etapie nie jest ona w pełni funkcjonalna.



Rys. 3.

### 2.4. Rozgrywka

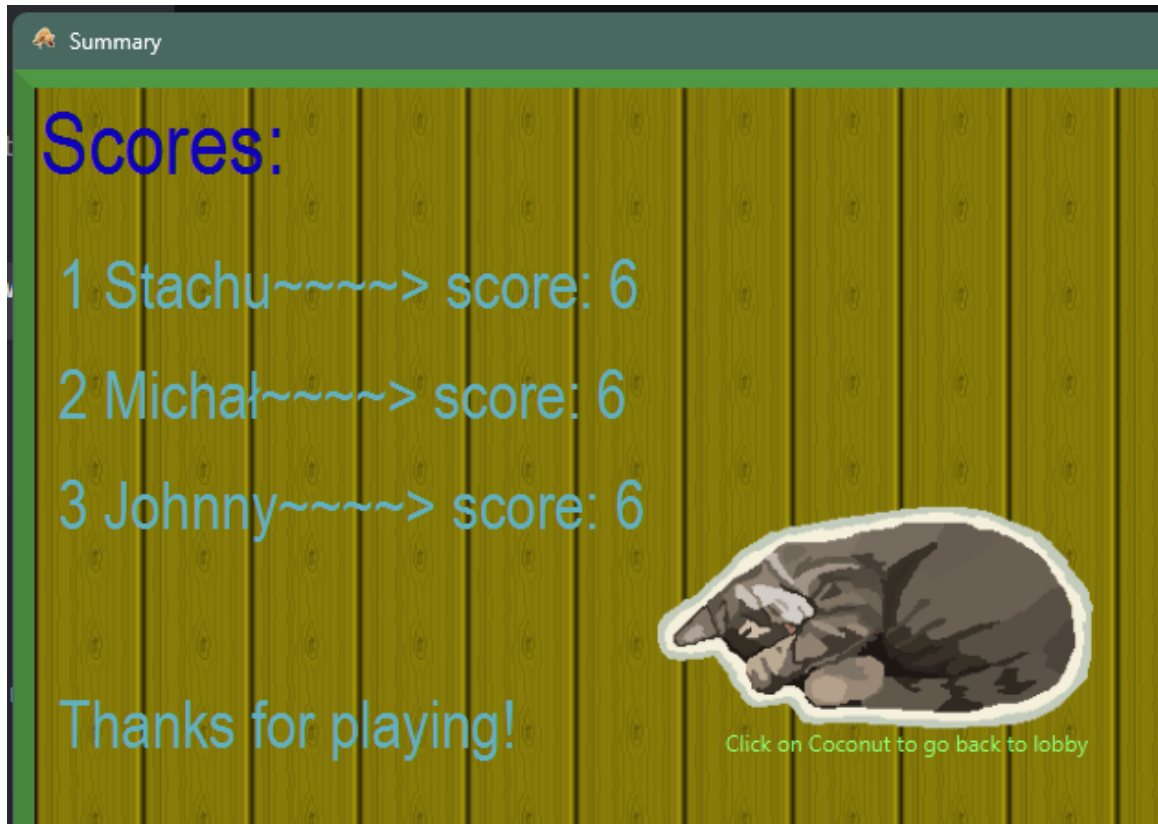
Gracze w tym etapie układają kafelki na swoich planszach tworząc kołderki dla kotków. Zgodnie z zasadami gracze mogą dodawać w międzyczasie znajdujące się po bokach planszy guziki oraz koty. Każdy kot ma swój preferowany układ kafelków oraz ich wzór, i tylko na takim można go położyć. Guziki kolorowe można położyć na zbiorze co najmniej trzech kafelków jednego koloru, po ułożeniu kafelka na tole i dobraniu nowego. Poniżej prezentujemy przykładowy przebieg rozgrywki.



Rys. 4.

## 2.5. Zakończenie rozgrywki

Rozgrywka, według instrukcji kończy się gdy wszyscy gracze ukończą swoje kołderki. Wtedy podliczane są punkty i wyłaniany jest zwycięzca.



Rys. 5. Wyniki zakończonej rozgrywki

## 3. Podsumowanie etapu

Wykonana przez nas ostatnia część dużego projektu z przedmiotu APRO2 sprawia, że całość wykonanej pracy w trzech etapach staje się połączona w funkcjonalną grę z pełną logiką i interfejsem graficznym. Niestety, podczas pracy nie udało nam się zrealizować wszystkich założeń tego etapu - głównie z powodu braku zaimplementowanych klas serwer i client oraz poprawnego sposobu zapisu do pliku. Wykoanie tego etapu okazało się bardzo dużym wyzwaniem, gdyż wymagało od nas zaimplementowania niestandardowych guzików, narysowania grafik oraz połączenia całego GUI ze skomplikowanymi zasadami gry.