

EngineerArc - Podręcznik gracza
Zasady gry z perspektywy gracza

Ziemowit Dworakowski

v0.2 (Dec 2025)

Spis treści

0.1	Witamy w <i>EngineerArc</i>	1
0.2	Twoja postać	1
0.3	Rzuty kością	3
0.4	Zasoby	5
0.5	Tryby gry w <i>EngineerArc</i>	5
0.6	Praca zespołowa, odgrywanie ról i metamechaniki	6
0.7	Pytania i odpowiedzi	6
0.8	Zapis przykładowej partii	8

EngineerArc - Player Handbook

0.1 Witamy w EngineerArc

EngineerArc to system do gier fabularnych (ang. *Tabletop Role-Playing Game*, (TRPG)) zaprojektowany jako pomoc dydaktyczna dla uczelni technicznych. Zamienia proces uczenia się, eksperymentowania i rozwiązywania problemów we wspólną opowieść, w której studenci grają rolę inżynierów mierzących się z autentycznymi wyzwaniami technicznymi i organizacyjnymi. System jest celowo prosty, wymaga jedynie kilku liczb i statystyk, jednej kostki do gry i odrobiny wyobraźni. W przeciwieństwie do typowych systemów RPG, *EngineerArc* nie polega na pokonywaniu potworów ani zbieraniu skarbów. Jego celem jest symulowanie procesu zdobywania doświadczenia inżynierskiego - poprzez rozwiązywanie problemów charakterystycznych dla obszaru Twojej specjalizacji z zastosowaniem logiki, kreatywności, planowania eksperymentów i pracy zespołowej. Rzuty kostką reprezentują próby rozwiązania problemów a każdy sukces lub porażka to krok w stronę zrozumienia zachowania złożonych systemów. Nie musisz znać innych systemów RPG by grać w *EngineerArc*. Jeśli potrafisz wyobrazić sobie hipotetyczną sytuację, omówić dostępne opcje i podjąć decyzję, masz już wszystko, co potrzebne. Zasady pozostają proste, aby uwaga była skupiona na rozumowaniu, współpracy i odkrywaniu. Z czasem gracze zauważają, że nawyki ćwiczone w grze: kwestionowanie założeń, refleksja nad błędami, planowanie alternatywnych rozwiązań i argumentowanie swoich decyzji — są tymi samymi nawykami, które charakteryzują dobrych inżynierów w świecie rzeczywistym. *EngineerArc* może pojawiać się na różnych zajęciach i warsztatach: jako krótka aktywność na ćwiczeniach albo dłuższa symulacja prowadząca projekt semestralny. System w niniejszym podręczniku opisany jest pod kątem specjalizacji mechatronicznej, ale zmiana wybieralnego zestawu umiejętności pozwala zaadaptować go do dowolnego kierunku studiów technicznych. Towarzyszący niniejszemu podręcznik dla Mistrzów Gry uwzględnia przykładowe adaptacje do różnych kierunków studiów. Co najważniejsze, gra tworzy bezpieczną przestrzeń do nauki. Tu błędy są interesujące, a nie kosztowne, a sukces jest wspólny, a nie konkurencyjny. Kości decydują o rezultatach, ale sens wynika z tego, jak razem je interpretujecie.

Idea przewodnia: nauka jako opowieść

Zanim pojawiły się podręczniki, równania i slajdy, ludzie uczyli się, opowiadając historie. Ktoś mierzył się z wyzwaniem, próbował różnych podejść i dzielił się tym, co zadziało, a co zawiodło. *EngineerArc* przywraca tę tradycję do dydaktyki akademickiej. Grupa studentów bierze udział w technicznej przygodzie, tworząc wspólnie wyobrażoną historię dotyczącą projektowania, budowania, zbierania informacji i rozumowania w zakresie

inżynierii mechatronicznej. Każde spotkanie kręci się wokół scenariusza — małego fragmentu zawodowej rzeczywistości.

Mistrz Gry (MG) będący doświadczonym w danym obszarze nauczycielem akademickim przedstawia problem do rozwiązania: może to być wadliwy przegub robota, konieczność optymalizacji procesu produkcyjnego czy nawet udział w dużym projekcie wdrożeniowym. Gracze omawiają, co robić, wybierają działania i rzucają kostką, aby zobaczyć, jak wyobrażony świat reaguje na ich działania. Wspólnie tworzą narrację historii, w której rezultat zależy zarówno od wiedzy, jak i od gotowości do głośnego myślenia.

MG pełni rolę narratora i facylitatora. Opisuje świat, ustala trudność zadań i interpretuje wyniki. Nie jest przeciwnikiem, lecz przewodnikiem, który dba o to, by niepewność prowadziła do budowania zrozumienia. Jego głównym zadaniem jest zadawanie dobrych pytań: "Skąd wiesz, że to zadziała?", "Co jeśli dane z czujników są błędne?", "Kto powinien wyjaśnić to klientowi?". Każde z nich zaprasza graczy do rozumowania, współpracy i refleksji.

Gracze z kolei wcielają się w tym wspólnym świecie w inżynierów o wybranych specjalizacjach. Łączą kompetencje techniczne z cechami osobistymi: wytrzymałością, ciekawością, komunikacją. Podejmują inicjatywę, proponują rozwiązania i uczą się na konsekwencjach błędów. Sesja często przebiega jak miniaturowy projekt badawczy — hipoteza, eksperyment, wynik, refleksja — ale z dozą nieprzewidywalności, która utrzymuje żywość gry.

Fabula ma znaczenie, bo łączy wiedzę z kontekstem. Rozwiązanie problemu w *EngineerArc* to nie tylko uzyskanie poprawnej liczby; chodzi o zrozumienie, dlaczego ta liczba ma znaczenie, jak ją uzyskano i co może pójść źle następnym razem. Na tym polega istota *EngineerArc*: inżynieria jako wspólna opowieść o uczeniu się. Nie grasz, by uciec od rzeczywistości — grasz, by spojrzeć na nią z nowej perspektywy: bezpiecznie, twórczo i wspólnie. Główną zaletą zastosowania systemu w praktyce jest przyspieszenie procesu zdobywania doświadczeń. Zanim jeszcze przyszedł inżynier zdobędzie swoje pierwsze praktyczne doświadczenia projektowe w trakcie wielomiesięcznej czy wieloletniej praktyki, ma możliwość zanurzyć się w scenariuszach hipotetycznych, których w ciągu kilku godzin gry może przeżyć bardzo wiele. To pozwala na niemal-bezpośredni transfer doświadczenia eksperckiego od nauczyciela do studenta, bowiem projektowane scenariusze problemowe mogą odzwierciedlać charakterystyczne wyzwania w obszarze ekspertyzy danego nauczyciela.

0.2 Twoja postać

Każdy gracz w *EngineerArc* kontroluje postać: sfabularyzowaną wersję studenta lub inżyniera funkcjonującego we wspól-

nym świecie wyzwiań technicznych. Ta persona reprezentuje Cię w grze: mierzy się z zadaniami, rozwija kompetencje, współpracuje z zespołem i, czasem, w ciekawy sposób ponosi porażki. Nie musi być identyczna z Tobą, ale powinna być wiarygodna w realiach uczelni technicznej. Traktuj ją jak destylację swojej tożsamości zawodowej — osobowość zbudowaną z ciekawości, wytrwałości i rosnącego zestawu narzędzi umiejętności.

Kroki tworzenia persony

1. **Imię i krótki kontekst.** Nadaj postaci imię i jednozdaniowy opis: *"Ciekawski student z laboratorium robotyki, który woli sprzęt od teorii"*, albo *"Perfekcjonistyczny analityk zafascynowany informacją pozyskiwaną z sygnałów drganiowych"*. Ta lekka narracja pomaga innym wyobrazić sobie Twoje zachowanie w zespole. Zmiana imienia (wykorzystanie fikcyjnego) pozwala na emocjonalne oddzielenie się od problemów spotykających persone: *"To nie ja nie umiem rozwiązać tego problemu. To Mike, mój awatar w świecie gry ma z tym problem"*
2. **Rozdział punktów atrybutów.** Otrzymujesz 20 punktów do podziału pomiędzy cztery atrybuty: Wiedza, Rozumowanie, Fizyczność oraz Charyzma. Każdy atrybut musi mieścić się w zakresie 3–10. Wyższa wartość oznacza większą naturalną siłę w danym obszarze. GM może dopuścić wyjątki, jeśli koncepcja je uzasadnia, albo narzucić dodatkowe ograniczenia (np. *"żaden atrybut nie wyżej niż ..."*).
3. **Wybór umiejętności startowych.** Wybierz kilka umiejętności z dwunastu w siatce, które pasują do Twojego tła. Ilość punktów do podziału oraz dodatkowe reguły podziału punktów zależą od kontekstu danego scenariusza - np. scenariusze symulujące "sytuacje zawodowe" będą zakładały większą ilość dostępnych punktów niż scenariusze symulujące "naukę na studiach". W porozumieniu z MG możesz dodać własne lub dodatkowe umiejętności, jeśli wymaga tego specjalizacja.
4. **Zapis na karcie postaci.** Karta postaci zawiera informacje o postaci (w tym miejsce na wskazówki do odgrywania), atrybuty i umiejętności - wraz z krótką ściągawką "jak je interpretować". W trakcie gry będziesz ją mieć przed sobą.

Cztery atrybuty

Wiedza mierzy to, co już wiesz: fakty, równania, normy i narzędzia teoretyczne. Odpowiada za zadania oparte na pamięci lub stosowaniu formalnej wiedzy. Wysoka Wiedza oznacza solidne przygotowanie i dobre nawyki akademickie.

Rozumowanie reprezentuje to, jak myślisz, a nie tylko *co* wiesz: dostrzeganie wzorców, logicznych powiązań i samokorektę, gdy coś "nie gra". Często podsuwa wskazówki, alternatywne ścieżki lub wczesne wykrycie błędów. Wysokie Rozumowanie bywa "ostatnią deską ratunku", gdy wszystko inne zawodzi.

Fizyczność łączy koordynację, wytrzymałość, odporność psychiczną i świadomość bezpieczeństwa. Określa, jak długo możesz wykonywać prace praktyczne, czy potkniesz się o kable i jak pewną rękę masz przy lutowaniu lub pomiarach. Wyższa Fizyczność daje więcej rzutów w testach wydłużonych.

Charyzma obejmuje komunikację, perswazję i pracę zespołową. Decyduje, jak dobrze potrafisz wyjaśnić rozumowanie, negocjować zasoby czy dbać o dobrą dynamikę grupy. W sesjach zespołowych Charyzma często otwiera drzwi, których sama wiedza nie otworzy.

Umiejętności i kompetencje

Jeśli cztery atrybuty opisują, *jak* podchodzisz do problemów, to umiejętności opisują, *co* potrafisz zrobić. To dyscypliny, techniki i obszary praktyki, które czynią każdego inżyniera wyjątkowym. W *EngineerArc* umiejętności łączą teorię z działaniem: pokazują, gdzie inwestowałeś czas i jak doświadczenie przekształca się w biegłość.

Każda umiejętność ma poziom 0–10. Wartość 0 oznacza jedynie ogólną świadomość tematu; 10 — biegłość ekspercką, możliwą na podstawie wieloletniej praktyki w obszarze. Orientacyjnie przyjętą można, że ocena „5” z odpowiednich kursów podczas studiów przekłada się na poziom „5” w danej umiejętności. Wyższe wartości pochodzą z aktywności dodatkowych, kół naukowych, doświadczenia zawodowego itd. W trakcie gry zwykle dodajesz wartość umiejętności do odpowiedniego atrybutu, aby uzyskać łączną zdolność do realizacji danego zadania.

Na przykład gracz z Wiedzą akademicką 7 i Dynamiką strukturalną 5 rzucający kością k20 podczas diagnozowania problemów drganiowych powinien rzucić *poniżej* 12 aby osiągnąć sukces.

Siatka umiejętności mechatronicznych

Poniżej przedstawiono standardową siatkę umiejętności wykorzystywaną na kierunkach mechatronicznych i interdyscyplinarnych. Razem odzwierciedlają one zestaw umiejętności wymaganych, by płynnie poruszać się między systemami fizycznymi, logiką sterowania, elektroniką, obliczeniami i komunikacją.

Nie wszystkie umiejętności pojawiają się bezpośrednio w każdym scenariuszu - ale od kreatywności graczy może zależeć możliwość ich wykorzystania do obejścia problemów które przed graczami stoją. Być może, pomimo braku umiejętności dedykowanych z teorii sterowania można wykorzystać wysokie wartości umiejętności pracy z profesjonalnymi źródłami oraz kodowania - by napisać kontroler bezpośrednio na podstawie poprawnie zidentyfikowanego przykładu z literatury?

Ponieważ inżynieria jest rozległa, w niektórych scenariuszach zasadne może być wprowadzenie dodatkowych umiejętności wspierających jak optomechanika, energetyka odnawialna, termodynamika czy robotyka autonomiczna. Czasem inicjatywa może wyjść ze strony gracza zainteresowanego poprowadzeniem konkretnie zdefiniowanej persony, czasem MG ze względu na konkretny scenariusz narzuci wybór nowych umiejętności wspierających wymaganych w danym problemie.

Rule

- **Programowanie:**
Programowanie do analizy, automatyzacji i sterowania (Python, MATLAB, C/C++).
- **Urządzenia mechatroniczne:**
Integracja komponentów elektronicznych, robotycznych i mocy.
- **Modelowanie CAD:**
Projektowanie części i zespołów do symulacji lub produkcji.
- **Sygnały i systemy:**
Modelowanie i analiza dynamiki układów w dziedzinie czasu i częstotliwości.
- **Sztuczna inteligencja i systemy decyzyjne:**
Metody oparte na danych, uczenie maszynowe i optymalizacja.
- **Pomiary i czujniki:**
Dobór, kalibracja i obsługa czujników z dbałością o dokładność.
- **Dynamika strukturalna:**
Drgania, rezonanse i odpowiedzi dynamiczne struktur.
- **Automatyka i sterowanie:**
Projektowanie i strojenie układów sprzężenia zwrotnego, programowanie PLC.
- **Systemy wbudowane i mikrokontrolery:**
Tworzenie firmware'u i obsługa interfejsów sprzętowych czasu rzeczywistego.
- **Materiały i wytwarzanie:**
Wiedza o materiałach i metodach wytwarzania elementów mechatronicznych.
- **Raportowanie i prezentacja:**
Jasna komunikacja wyników w dokumentacji i wizualizacjach.
- **Praca ze źródłami fachowymi:**
Czytanie i interpretacja artykułów naukowych, norm i dokumentacji.

Budowanie podejścia

Nie ma jednego "poprawnego" sposobu projektowania postaci w *EngineerArc*. Różne podejścia ujawniają odmienne mocne strony — tak jak w realnych zespołach.

Generalista. Rozdziel punkty równomiernie. Być może nigdy nie zdominujesz jednego obszaru, ale wniesiesz wkład do niemal każdego problemu i szybko się zaadaptujesz. Generalista często bywa facylitatorem zespołu, zmieniając rolę w miarę rozwoju projektu.

Specjalista. Zainwestuj mocno w jeden obszar — np. Wiedza akademicka + Dynamika strukturalna, albo Programowanie + SI. Błyszczysz, gdy pojawia się "Twoje" zagadnienie, ale poza strefą komfortu opierasz się na zespole.

Odbicie siebie. Możesz zaprojektować personę na podobieństwo własnego stylu uczenia się i działania. Niektóre scenariusze wręcz do tego zachęcają. Jeśli wiesz, że wolisz teorię od praktyki, albo komunikację od kodowania, przedstaw to w rozkładzie punktów. Gra staje się wtedy bezpieczną przestrzenią do eksperymentowania ze strategiami, ćwiczenia rzadziej używanych umiejętności oraz sprawdzania "jak moje preferencje przełożą się na sposób działania w warunkach przyszłej kariery".

0.3 Rzuty kością

W *EngineerArc* kości istnieją z jednego powodu: reprezentują niepewność realnej pracy inżynierskiej. Nawet najbardziej kompetentny inżynier doświadcza czynników losowych — szumu pomiarowego, zmęczenia, bałaganu w danych i czasem szczęścia. Kość to zwinny sposób pokazania, że rzeczywistość nie biegnie po szynach. Wszystkie działania w grze używają jednej dwudziestościennej kości — k20. Gdy próbujesz czegoś, co może się powieść lub nie, rzucasz raz i porównujesz wynik ze swoimi zdolnościami. W przeciwieństwie do wielu systemów RPG *EngineerArc* stosuje zasadę „niżej-lepiej” (roll-under): im niższa Twoja ostateczna liczba, tym lepiej. Sukces odnosisz, gdy suma rzutu i modyfikatora trudności jest *mniejsza lub równa* Twojej wartości „Atrybut + Umiejętność” właściwej dla zadania.

Example

Gracz chce użyć "Wiedzy z Dynamiki strukturalnej". Z Wiedzą akademicką 7 i Dynamiką strukturalną 5 ma łączną zdolność 12. Jeśli GM uzna zadanie za normalne (modyfikator 0), każdy rzut k20 na poziomie 12 lub niżej to sukces.

Example

Gracz chce zainstalować czujniki w maszynie. Ma Fizyczność 4 i Pomiary i czujniki tylko na 3, więc łącznie 7. Normalnie potrzebowałby rzutu 7 lub niżej, ale MG stwierdza, że dostęp do maszyny jest łatwy i czasu jest dużo, więc deklaruje zadanie jako relatywnie proste (modyfikator -10). Teraz każdy rzut 17 lub niżej jest sukcesem.

Example

Umiejętności łączą się z atrybutami zależnie od kontekstu. Interpretacja uszkodzenia w maszynie może standardowo opierać się na połączeniu "Wiedzy" i "Dynamiki strukturalnej", a negocjacje dostępu do laboratorium — na "Charyzmie" i "Raportowaniu i prezentacji". Czasem związek jest mniej oczywisty i GM pozwoli zespołowi uzasadnić, która umiejętność i który atrybut najlepiej wspiera dane rozwiązanie — takie dyskusje są częścią nauki. Gracz może chcieć zaprojektować sterownik na podstawie "wiedzy zapamiętanej ze studiów" ("Wiedza" + "Automatyka i sterowanie") ale może również uargumentować, iż sterowana przez niego persona zauważyła dużą zmienność operacyjną i środowiskową co utrudni zbudowanie sterownika z zastosowaniem klasycznych metod - i w związku z tym musi wymyślić kreatywne, adaptacyjne rozwiązanie ("Rozumowanie" + Automatyka i sterowanie").

Modyfikatory trudności odzwierciedlają wymagania sytuacji. Proste, rutynowe zadanie może otrzymać -5, eksperckie wyzwanie +5, a problem na poziomie badawczym +10. Modyfikator dodaje się do wyniku rzutu, więc wyższe liczby zmniejszają szanse powodzenia. Czasem w większych scenariuszach modyfikator może być bardzo wysoki — służy wtedy jako bariera, którą należy przełamać za pomocą **Zasobów** (zob. sekcja 0.4).

Rule**Przykładowe modyfikatory trudności (dodawane do rzutu):**

- -10 — trywialne, "każdy absolwent powinien dać radę"
- -5 — praca rutynowa
- 0 — zadanie specjalistyczne / umiarkowanie trudne
- +5 — wyzwanie eksperckie
- +10 — trudność "wymagająca doktoratu"
- > +10 — poziom specjalny zarezerwowany dla finalizacji scenariuszy
- *ukryty* — modyfikator ustalony przez MG, niewyjawiony przed rzutem

Dwa skrajne wyniki zasługują na uwagę. Naturalne 20 to zawsze **porażka krytyczna** — coś idzie źle niezależnie od umiejętności. MG opíše realistyczny problem: zwarcie, uszkodzony plik, błędne założenie. Naturalna 1 liczy się jako **wzmocniony sukces**, tj. wystąpią dodatkowe pozytywne efekty (szybszy eksperyment, czystsze dane, wrażenie na przełożonym) o ile test jest zaliczony ("Szczęście nie zastąpi kompetencji"). Pamiętaj też, że porażka rzadko się marnuje — prowadzi do refleksji lub zmiany planu, a MG może pozwolić zespołowi omówić, co poszło nie tak. Kości czynią zdarzenia niepewnymi, nie arbitralnymi — uruchamiają rozmowę o rozumowaniu i poprawie. Niezaliczo-

ny test oznacza, że do kolejnej próby trzeba zmienić podejście, system nie pozwala na cykliczne próbowanie "aż wyjdzie".

Example

Gracz próbuje wykonać analizę modalną zbiornika ciśnieniowego. Niestety, jego persona nie przyswoiła w tym obszarze bogatej wiedzy na studiach (Wiedza 3), brakuje mu również kompetencji w obszarach dynamiki strukturalnej (2) i analizy sygnałów (2). Zadanie jest nietrywialne i wymaga specjalistycznego podejścia (poziom trudności +5). Gracz mimo wszystko próbuje rozwiązać zadanie. Decyduje się zastosować narzędzia analizy sygnałów by odkryć częstotliwości drgań własnych. Jego wynik na kości to "1". Gdyby test był zaliczony, oznaczałoby to szczególnie skuteczne działanie (np. ułatwienie kolejnego testu bądź analizy) - ale w naszym przypadku wynik kości powiększony o modyfikator poziomu trudności to "6" - a musielibyśmy wyrzucić poniżej "5" ("Wiedza" + "analiza sygnałów") by rzut był udany. Jeśli personie brakuje podstawowych kompetencji by rozwiązać zadanie, idealny rzut nie będzie w stanie jej uratować, MG deklaruje porażkę próby ("Zebrałeś sygnały za pomocą kilku akcelerometrów, ale brakuje Ci umiejętności by poprawnie wyznaczyć widmo częstotliwościowe")

Rule

- 20 — porażka krytyczna
- 1 — wzmocniony sukces (o ile test i tak byłby zaliczony)

Rodzaje testów

Nie każde wyzwanie inżynierskie jest takie samo. Jedne to rezultaty szybkich procedur bądź obserwacji, inne mogą prowadzić do zróżnicowanych efektów w zależności od umiejętności i szczęścia, a część to długie, wyczerpujące procedury. *EngineerArc* używa trzech prostych rodzajów testów, by pokryć te sytuacje.

Test prosty — gdy wynik to jasne "tak" lub "nie". Rzucasz raz, stosujesz modyfikator poziomu trudności i sprawdzasz, czy suma nie przekracza wartości Atrybut + Umiejętność. Nadaje się do przypomnienia wzoru, podłączenia typowego układu, dotrzymania terminu raportu.

Test zaawansowany — gdy liczy się także *jakość* wyniku. Po zaliczeniu testu liczysz "poziom sukcesu" odejmując "Rzut + Modyfikator" od "Atrybut + Umiejętność". Im większa różnica, tym głębsze zrozumienie lub lepszy rezultat. 0-4: wynik pozytywny; 5-9: duży sukces; 10+: wynik spektakularny.

Test wydłużony — dla długich lub obciążających wysiłków: wieloetapowej akwizycji danych, złożonego montażu. GM określa, ile *udanych kroków* jest wymaganych (np. trzy) i ile masz prób. Liczba możliwych rzutów zazwyczaj będzie bazować na Twojej Fizyczności. Każdy rzut to jednostka czasu pracy. Jeśli osiągniesz wymaganą liczbę sukcesów zanim skończą się rzuty — zadanie ukończony; w przeciwnym razie zmęczenie, rozproszenie lub limity zasobów kończą sesję.

Rule**Test prosty:**

Zaliczenie, gdy: $Atrybut + Umiejętność \geq Rzut + Modyfikator$

Test zaawansowany:

Poziom sukcesu = $Atrybut + Umiejętność - Modyfikator - Rzut$

0–4: Sukces; 5–9: Duży sukces; 10+: Spektakularny wynik

Test wydłużony:

Co najmniej X udanych testów prostych w Y (Fizyczność) rzutach

Example

Mike projektuje obudowę do sterownika dla robota mobilnego. Jest świetnym ekspertem w tym obszarze i przygotował sobie wcześniej pełną analizę możliwych sytuacji, które mogą się zdarzyć ("Rozumowanie" (6) + "Modelowanie CAD" (8) = 14. Zadanie jest typowe (poziom trudności 0). Wynik na kości to zaledwie 2, co oznacza, że poziom sukcesu testu to $14 - 2 = 12$. MG deklaruje spektakularny sukces: Mike nie tylko przewidział w modelu odpowiednie zabezpieczenia przed rozbryzgami wody i pyłem, wykonał swój projekt w taki sposób, by środek ciężkości sterownika po wyposażeniu w elektronikę wypadł idealnie w środku obudowy a cały model był sparametryzowany i umożliwił w przyszłości szybkie analizy w przypadku zmian gabarytów elementów elektronicznych.

0.4 Zasoby

W realnej inżynierii sukces rzadko rozstrzygany jest za pomocą jednego pomiaru czy jednej analizy. Projekty są przyjmowane albo odrzucane na podstawie ścieżki dowodów: działających prototypów, raportów z testów, zadowolonych klientów, rzetelnej dokumentacji i wiarygodnych badań. Świat *EngineerArc* oddaje tę rzeczywistość poprzez **Zasoby**. Zasoby reprezentują namacalne lub weryfikowalne osiągnięcia, które wzmacniają Twoją pozycję podczas kluczowych ewaluacji. Mogą być przedmiotami, rzetelnymi dowodami, zaufaniem publicznym albo po prostu dobrze udokumentowanymi wynikami - wszystkim, co czyni Twoje finalne twierdzenie bardziej przekonującym. Mechanicznie: obniżają trudność działań wysokiej stawki, takich jak obrona projektu przed komisją, przekonanie inwestora czy prezentacja wyników sceptycznemu profesorowi. Każdy Zasób ma **poziom 2–6** odzwierciedlający siłę lub wiarygodność.

Gdy stajesz przed decydującym wyzwaniem, odpowiednie Zasoby *zmniejszają efektywną trudność* testu finałowego. Zamiast walczyć z pełnym "poziomem doktorskim" (+10), odejmujesz od modyfikatora *sumę poziomów* właściwych Zasobów. Dla graczy Zasoby tworzą motywację długoterminową: każdy sukces może dołożyć cegiełkę do przyszłego celu, a nawet skromne osiągnięcia stają się wartościowe jako część wiarygodnego łańcucha. Dla MG Zasoby porządkują długie symulacje i projekty semestralne bez nadmiaru buchalterii. W *EngineerArc*

świat "pamięta" Twoją dobrą inżynierię tworząc przesłanki na które możesz się powołać, gdy przyjdzie pora udowodnić, że Twój system rzeczywiście działa.

Tworzenie i używanie Zasobów

Ileokroć kończysz istotne zadanie, które daje rezultat rozpoznawalny przez świat, MG może ogłosić powstanie Zasobu i przypisać mu poziom w przybliżeniu zależny od wyniku testu. Notuj każdy Zasób na *Karcie Zasobów*: czym jest, jaki ma poziom, jak go osiągnięto i czemu może posłużyć. To projektowe portfolio wewnątrz gry.

Podczas trudnej oceny — zwykle na końcu scenariusza — możesz *wykorzystać lub zaprezentować* Zasoby istotne dla sprawy. MG zapyta, jak ich używasz: pokazujesz prototyp, odwołujesz się do analizy, cytujesz literaturę. Będziesz musiał naracyjnie uzasadnić ich związek z oceną. Poziomy właściwych Zasobów bezpośrednio *odejmują się* od trudności zadania.

Rule**Tworzenie Zasobów (orientacyjnie):**

- Sukces: Zasób poziomu 2
- Duży sukces lub podwójne sukcesy: Zasób poziomu 4
- Spektakularny wynik lub długa seria drobnych sukcesów: Zasób poziomu 6

Example

CEO ocenia finał Waszego projektu. Trudność bazowa wynosi +15. Twoja Charyzma 6 + Raportowanie i prezentacja 6 = 12. Normalnie nie byłbyś w stanie przekonać go do sukcesu (Nawet idealny rzut (1) będzie się liczył jak 16 a musisz wyrzucić "poniżej 12"), ale masz Zasoby warte 9 punktów: prototyp (poziom 6), analiza danych (poziom 4), referencje zadowolonego klienta (poziom 4), raport (poziom 2) i kwerenda patentowa (poziom 4) potwierdzająca nowość. Zmodyfikowana trudność to teraz -5 (15–20). Teraz każdy rzut o wartości 16 lub niżej to sukces — wiarygodne, dobrze udokumentowane zwycięstwo.

0.5 Tryby gry w *EngineerArc*

Każda sesja *EngineerArc* może wyglądać i brzmieć nieco inaczej. Czasem to krótkie, zwarte wyzwanie; czasem dłuższy projekt; czasem refleksyjna ścieżka powtórek. Zasady pozostają te same, ale ton i cel gry zmieniają się wraz z rozwojem historii i wyborem stylu prowadzenia przez nauczyciela (Twojego MG). Poniżej przedstawiłem trzy główne perspektywy różnicowania rozgrywki.

Rodzaj scenariusza — "w jakiej historii jesteś?"

Scenariusze zdarzeniowe koncentrują się na pojedynczym, jasno zdefiniowanym problemie. Analizujesz, planujesz i wy-

konujesz kilka kluczowych rzutów, by sprawdzić, czy pomysł działa. Krótkie i strategiczne — idealne na jedne zajęcia.

Scenariusze projektowe Mogą rozgrywać się przez kilka spotkań. Zespół gromadzi Zasoby przez udane działania: prototypy, raporty, testy, analizy. Na końcu te Zasoby ułatwiają ocenę finałową - tak jak dobre przygotowanie ułatwia realne projekty.

Scenariusze kariery dotyczą powtórzeń i wzrostu. Wyzwania wracają w podobnej formie, uczysz się rozpoznawać wzorce i poprawiać nawyki. Może nie być punktacji finałowej, za to jest refleksja nad tym, jak ewoluowało myślenie zespołu.

Etapy nauki — gdzie możesz spotkać *EngineerArc* ?

Etap 1 — Budowa postaci. Użyjesz idei karty postaci, by opisać siebie jako inżyniera: co Cię motywuje, gdzie widzisz swoje mocne strony i które kursy budują Twoje przyszłe kompetencje.

Etap 2 — Test terenowy. Faktycznie zagrasz w *EngineerArc* w scenariuszach rozwiązywania problemów, używając kości i dyskusji, by uczyć się kreatywności, pracy zespołowej i wyciągania wniosków z porażek.

Etap 3 — Optymalizacja. Wrócisz do osoby na poziomie magisterskim, by sprawdzić, jak Twoje zdolności pasują do celów zespołowego projektu, gdzie zespół ma słabe punkty i jak można je uzupełnić lub obejść.

Style prowadzenia — jak reaguje świat?

Tryb podawczy. GM aktywnie uczy poprzez grę, udziela jasnej informacji zwrotnej i wyjaśnień po rzutach, czasem przejmując kontrolę nad osobami prezentując właściwe podejścia do problemów. Ten tryb przypomina interaktywny film, w którym gracze czasem podejmują decyzję a czasem nie mają kontroli nad decyzjami swoich bohaterów.

Tryb eksploracyjny. Najbliższy klasycznemu prowadzeniu gier RPG. Wspólnie z GM współtworzysz historię. Świat reaguje na Twoje pomysły; dostajesz wskazówki, ale samodzielnie prowadzisz rozumowanie i samodzielnie podejmujesz wszystkie decyzje. To najbardziej otwarta i kreatywna forma gry.

Tryb realistyczny. Świat po prostu trzyma się swoich reguł. Złe pomysły mogą nie wymagać rzutu, dobre — zadziałają, jeśli są sensowne. Bardziej symulacyjny, wymagający, ale nagradzający pewnych siebie graczy.

0.6 Praca zespołowa, odgrywanie ról i metamechaniki

Gdyby *EngineerArc* opierał się wyłącznie na rzutach, byłby grą logiczną. Tymczasem o roli i wartości dydaktycznej decyduje to, jak ludzie współdziałają przy rozwiązywaniu zagadek. W realnych projektach rzadko wygrywa pojedynczy ekspert. Blyskotliwy algorytm bez dobrych danych zawodzi, doskonały projekt upada, gdy brakuje komunikacji, a nawet najsprawniejszy technik popełnia błędy, gdy jest zmęczony lub pracuje samotnie. Dlatego współpraca leży w sercu *EngineerArc*.

Każdy gracz wnosi inny zestaw atrybutów i umiejętności — i ta różnorodność jest zamierzona. Jeden student może błyszczyć Wiedzą akademicką i analizą danych, drugi umiejętnościami

manualnymi i starannym montażem, trzeci Charyzmą i negocjacjami. W trakcie sesji różnice przekładają się na komplementarne role. Analityk rozpoznaje, co trzeba zrobić i *dlaczego* coś zachowuje się tak, jak się zachowuje. Budowniczy przekuwa te wglądy w działanie: lutuje, koduje, testuje. Komunikator koordynuje, negocjuje i dokumentuje, dbając, by praca zespołu była widoczna, a zasoby dostępne.

Większość problemów w *EngineerArc* jest zbyt złożona na jeden rzut. Zespół może najpierw przywołać tło teoretyczne (Wiedza), potem zinterpretować nietypowe dane (Rozumowanie), wykonać pomiar (Fizyczność), a na końcu przekonać opiekuna projektu o wydłużeniu czasu testów (Charyzma). Każdy wychodzi na pierwszy plan, gdy jego mocne strony pasują do sytuacji. MG powinien dbać o balans - by każdy miał istotne decyzje do podjęcia.

Choć Wiedza napędza większość testów technicznych, pozostałe trzy atrybuty modyfikują lub rozszerzają to, co sama wiedza może osiągnąć. Poza udziałem testach dedykowanych działają dodatkowo jako *metamechaniki* — narzędzia wpływu na kontekst wokół zadania. **Rozumowanie** daje podpowiedzi: gdy zespół utknie, dobry rzut na Wgląd może ujawnić brakującą zmienną, mylące założenie lub sprytne obejście. **Fizyczność** zapewnia wytrzymałość: pozwala powtarzać wymagające testy dłużej, prowadzić przedłużone pomiary i zachować sprawność podczas nocnych sesji. **Charyzma** dostarcza zasobów: obejmuje komunikację, perswazję i morale. Wysoka Charyzma pozwala wynegocjować lepsze narzędzia, pożyczyć instrument albo zmotywowować zmęczonych kolegów.

Rozsądne korzystanie z metamechanik zamienia grupę jednostek w działający zespół. W praktyce oznacza to ustalanie, kto i w jakiej kolejności powinien działać. Kolega o wysokiej Charyzmie najpierw wywalczy zgodę na eksperyment, który następnie wykona gracz z wysoką Fizycznością, prowadzony przez analityka z wysoką Wiedzą akademicką, który zinterpretuje wyniki. Kości rozstrzygają niepewność, ale rozmowa ujawnia strategię i współpracę.

Praca zespołowa w *EngineerArc* nie jest wyścigiem po punkty. Prawdziwy postęp dzieje się, gdy gracze słuchają się nawzajem, kwestionują założenia i łączą mocne strony. Po każdym scenariuszu warto refleksyjnie spojrzeć nie tylko na to, *co* zadziałało, ale *dlaczego*: czy problem rozwiązano dzięki znajomości wzoru, zauważeniu anomalii, czy dzięki planowaniu i komunikacji? Te refleksje łączą fikcyjne zdarzenia z realną pracą akademicką, podkreślając kompetencje miękkie, o które trudno w typowych ćwiczeniach. Dobre zespoły uczą się równoważyć role: czasem analityk powinien odpuścić i oddać prowadzenie "budowniczemu"; czasem komunikator musi zakończyć przeciągającą się dyskusję. Gra nagradza świadomość tych przejść bardziej niż pojedynczy rzut.

0.7 Pytania i odpowiedzi

Co jeśli nigdy nie grałem w RPG i nie mam pojęcia, jak to działa? Nie ma problemu — *EngineerArc* został zaprojektowany z myślą o osobach, które nigdy wcześniej nie grały w gry fabularne. Nie musisz zapamiętywać długich podręczników ani znać żargonu graczy. Twój Mistrz Gry (zazwyczaj nauczyciel) poprowadzi Cię krok po kroku w trakcie sesji.

U podstaw RPG leży po prostu uporządkowana rozmowa: ktoś opisuje sytuację, Ty decydujesz, co Twoja postać zrobi,

a gdy wynik jest niepewny — wykonuje się rzut kością. I to wszystko, czego potrzeba, by zacząć.

Mechanika *EngineerArc* jest celowo prosta, przejrzysta i łatwa w użyciu, byś mógł skupić się na myśleniu, rozwiązywaniu problemów i uczeniu się — a nie na zapamiętywaniu zasad.

Czy muszę nauczyć się zasad przed grą? Nie. *EngineerArc* wprowadza się tak jak inne gry fabularne — możesz po prostu dołączyć, a Mistrz Gry wyjaśni potrzebne rzuty i mechaniki w trakcie. Warto jednak przejrzeć najważniejsze zasady (wyróżnione ramki "Zasada" w tym podręczniku), by wiedzieć, mniej więcej, czego się spodziewać.

Czy potrzebuję jakiegoś sprzętu, by grać? Nie — wystarczy kartka papieru i coś do pisania. *EngineerArc* używa jednej dwudziestościennej kości (k20), ale zapewni ją Mistrz Gry więc nie musisz przynosić niczego specjalnego.

Czy to jest oceniane? Co jeśli całkowicie zawalimy scenariusz? Nie. *EngineerArc* to środowisko do nauki, a nie narzędzie oceniania. Kości i mechaniki istnieją po to, by wprowadzić napięcie, ciekawość i refleksję — nie by przydzielać oceny. Nie można "oblać" scenariusza w sensie akademickim.

Nawet jeśli projekt zespołu się rozsypie, system zawiedzie lub plan doprowadzi do katastrofy, to w porządku — oznacza to po prostu, że dotarliście do jednego z najlepszych momentów do nauki. Gdy coś pójdzie źle, Mistrz Gry poprowadzi krótką rozmowę: co się stało, które założenia zawiodły i co można poprawić. Czasem odtworzycie kluczową scenę z nowym planem, czasem pójdziecie dalej i zmierzycie się ze skutkami. Jedyną prawdziwą porażką jest brak zaangażowania. Błędy, nieoczekiwane wyniki i spektakularne awarie są częścią opowieści — i często tym, co zapamiętacie najlepiej.

Co jeśli nie rozumiem tematu technicznego w danym scenariuszu? To zupełnie w porządku — *EngineerArc* ma pomagać w nauce, a nie sprawdzać to, co już wiesz. W zależności od trybu gry, czasem po prostu "udajesz, że wiesz, co robisz", a Mistrz Gry naprowadza Cię na poprawne rozwiązanie, tłumacząc po drodze pojęcia. Innym razem opierasz się na realnej wiedzy, a świat reaguje naturalnie — także na błędy. W obu przypadkach cel jest ten sam: poznawać idee techniczne w bezpieczny sposób i uczyć się na podstawie tego, jak Twoje decyzje zadziały.

Czy możemy dyskutować z Mistrzem Gry, jeśli uważamy, że mamy rację? Tak — i nawet należy to robić, oczywiście z zachowaniem wzajemnego szacunku i mając na uwadze co jest celem gry. W *EngineerArc* dyskusja to część procesu uczenia się. Jeśli Twoja koncepcja ma sens, wyjaśnij swoje rozumowanie — tak jak na spotkaniu projektowym lub w laboratorium. Mistrz Gry będzie miał ostatnie słowo, ale dobre argumenty poparte logiką lub danymi często wpływają na bieg historii. Nieporozumienie nie jest problemem — to właśnie ono czyni inżynierię ciekawą. Liczy się sposób obrony swojego stanowiska, a nie to, czy kości "dały wygraną". Pamiętaj jednak, że *EngineerArc* w pierwszej kolejności ma pozwolić na transfer eksperckiego doświadczenia nauczyciela oraz symulowanie scenariuszy rzeczywistych - kości są tu drugorzędne. Jeśli jakiegokolwiek podejście jest z punktu widzenia rzeczywistego scenariusza

absurdalne, MG może nie pozwolić na rzut albo zignorować "udany wynik" jeśli nie widzi możliwości pozytywnego zinterpretowania takiego rzutu.

Co jeśli kości sprawią, że nie uda mi się coś, co w rzeczywistości potrafię? To część gry — i bardziej realistyczna, niż się wydaje. Kości nie mierzą, jak jesteś zdolny, lecz reprezentują czynniki spoza Twojej kontroli: szum pomiarowy, awarie urządzeń, zmęczenie, presję czasu, brak narzędzi lub zwykłego pecha.

Czasem nawet idealny plan zawodzi — i właśnie to symuluje rzuty. Niepowodzenie nie znaczy, że czegoś nie rozumiałeś; oznacza, że w tej historii warunki nie były sprzyjające. Często porażka prowadzi do ciekawego odkrycia lub drugiej próby z nowym pomysłem — dokładnie jak w prawdziwej pracy inżynierskiej.

Czy muszę odgrywać rolę, zmieniać swój głos lub zachowywać się teatralnie? W większości sesji *EngineerArc* po prostu grasz sobą. Z wyjątkiem kilku scenariuszy specjalnych, pomagających eksplorować przyszłe ścieżki kariery, Twoja persona jest w zasadzie Tobą — może jedynie nieco bardziej pewną siebie lub doświadczoną. Nie ma potrzeby "grać", zmieniać głosu ani udawać, chyba że masz na to ochotę. Wystarczy wyobrazić sobie, jak zachowałbyś się w danej sytuacji, i opisać to własnymi słowami.

W wielu grach fabularnych używa się czasu teraźniejszego ("Sprawdzam czujnik", "Otwieram panel", czy w wersji bardziej klasycznej "atakuję smoka"), by utrzymać wrażenie immersji w świecie gry, ale możesz mówić też bardziej opisowo ("W tej sytuacji sprawdziłbym czujnik"), jeśli tak Ci wygodniej. Obie formy są poprawne — wybierz tę, która pozwala Ci skupić się na problemie, nie na występie.

Co jeśli nie lubię gier lub to podejście wydaje mi się dziwne? To całkowicie w porządku — *EngineerArc* nie wymaga bycia "graczem". Nie musisz lubić gier ani opowieści fantastycznych, by skorzystać z tego systemu. To po prostu uporządkowany sposób myślenia, który wykorzystuje wyobraźnię i element losowości, by spojrzeć na prawdziwe problemy inżynierskie z różnych perspektyw.

Gdy spróbujesz, zobaczysz, że przypomina to raczej wspólne opracowywanie planu rozwiązania realistycznego problemu z odrobiną swobody i pytaniem "co by było, gdyby?".

Czy mogę grać w *EngineerArc* ze znajomymi? Raczej nie — przynajmniej nie w taki sposób, w jaki system został zaprojektowany. Możesz oczywiście stworzyć własną historię i użyć zasad *EngineerArc*, ale świat gry ma odzwierciedlać naszą rzeczywistość, zwłaszcza w zachowaniu technologii. "Lore" gry przekłada się bezpośrednio na prawdziwą wiedzę mechatroniczną. Dlatego system najlepiej sprawdza się w środowisku dydaktycznym, prowadzonym przez osobę z doświadczeniem inżynierskim lub naukowym, która potrafi ocenić, czy działania graczy mają sens w kontekście rzeczywistego świata.

Czy istnieją dodatkowe materiały o grze? Tak. Istnieje *Podręcznik Mistrza Gry EngineerArc*, czyli uzupełniający tom przeznaczony dla nauczycieli i prowadzących, którzy chcą

używać EngineerArc w dydaktyce. Zawiera wskazówki dotyczące tworzenia scenariuszy opartych na własnej wiedzy, porady narracyjne dla początkujących Mistrzów Gry, metody adaptacji *EngineerArc* do różnych dziedzin nauki i techniki oraz gotowe przykłady scenariuszy. Jeśli jesteś nauczycielem i chcesz wykorzystać ten system w swojej praktyce, skontaktuj się ze mną w celu uzyskania informacji jak można otrzymać do niego dostęp.

0.8 Zapis przykładowej partii

W niniejszym rozdziale znajduje się fragment jednego ze scenariuszy testowych rozgrywanych w systemie *EngineerArc*. Został tutaj zawarty by ułatwić osobom niezwiązanym ze światem TTRPG zrozumienie "jak to właściwie wygląda". By zachować spójność i kompaktowość przykładu obszerne fragmenty narracyjne zostały w większości pominięte - pozostawiając jedynie fragmenty dialogów i podejścia do problemów.