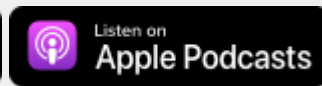


Pierwsze kroki w IT

PODCAST



Jak zrobić grę w 48h? Game Jam z producentem Wiedźmina

Gość: Maciej Miąsik

Wszystkie polecane materiały i linki znajdziesz na stronie odcinka:

<https://devmentor.pl/b/jak-zrobic-gre-w-48h-game-jam-z-producentem-wiedzmina>

Dziś moim gościem jest Maciej Miąsik, weteran branży GameDev i producent m.in. Wiedźmina. Maciej opowie nam o samej branży, jak również o evencie game jam, którego jest współorganizatorem. Maćku, dziękuję, że przyjąłeś moje zaproszenie na rozmowę.

Dziękuję, miło mi tu być.

Mnie jest jeszcze bardziej miło. Zaczniemy trochę inaczej –

oglądałem twoje wywiady w innych miejscach i zazwyczaj przedstawiał cię prowadzący. Ja wolałbym, żebyś zrobił to sam. To pewnie trochę nietypowa sytuacja dla ciebie, ale od razu powiem – chciałbym, żebyś nie był skromny. Więc zaczynaj.

Tak, zawsze trudno przedstawić się samemu. Jestem twórcą gier. Tak się określám, ponieważ w dziedzinie tworzenia gier robiłem bardzo dużo różnych rzeczy i nie mam takiej jednoznacznej roli w tym procesie. Byłem i jestem programistą. Jestem do pewnego stopnia projektantem gier. Bardzo długo zajmowałem się procesem produkcji, czyli zarządzaniem produkcją i podobnymi rzeczami. Łatwiej byłoby, gdybym mógł powiedzieć, że jestem znanym grafikiem odpowiedzialnym za świetne modele 3D, ale nie jestem grafikiem. Zajmowałem się kiedyś trochę grafiką, ale było to dekady temu i byłem w tym słaby. Na szczęście przestałem się tym zajmować i zostawiłem to ludziom, którzy to potrafią o wiele lepiej. Sam wciąż zajmuję się pozostałymi aspektami tworzenia gier. Z takich bardziej artystycznych dziedzin – udźwiękowiałem większość swoich gier. Trochę znam się na dźwięku. Nie jestem ekspertem w tej dziedzinie, ale mam o tym pewne pojęcie. Kiedy robię swoją małą grę, lubię sobie ją udźwiękować, bo wtedy mam odrobinę artystycznego zacięcia. Gry robię od ponad 35 lat. Pierwsza premiera mojej pierwszej gry komercyjnej – bo wcześniej robiłem też różne projekty do szuflady, bo trzeba było się gdzieś nauczyć – miała miejsce w 1992 roku, czyli jeszcze w ubiegłym wieku.

Rozumiem, że było to Electro Body?

Tak. Potem robiłem różne gry, między innymi przygodowe, które były sprzedawane na całym świecie. Otarłem się o Wiedźmina, Wiedźmina 1,

gdzie byłem szefem produkcji. Potem zajmowałem się różnymi innymi rzeczami. Później znowu zrobiłem swoją grę indie Beat Cop. Od jej premiery minęło chyba ponad 7 lat, ale to był taki powrót do korzeni – znowu tworzyłem grę w małym zespole, dużo własnymi rękami. Bardzo mi na tym zależało, bo to w grach lubię najbardziej – że mogę zrobić naprawdę dużo, pokazać: to zrobiłem ja. To była gra, nad którą pracowały cztery osoby, więc każdy mógł wskazać bardzo konkretne rzeczy, które stworzył. To daje mi największą satysfakcję – kiedy mogę powiedzieć, że coś wymyśliłem, zrealizowałem, widzę efekt i biorę za niego odpowiedzialność. Dużo się przez te lata działo.

Może tylko dodałbym, że Electro Body, o którym wspomnieliśmy, to była pierwsza taka komercyjna gra na PC – piszesz o tym również na swojej stronie, więc myślę, że warto to dodać.

Jakby się tak dokładnie przyjrzeć, to oczywiście nie była to pierwsza polska gra, bo branża gier miała swój epizod w latach 80., kiedy w Polsce istniała firma tworząca komercyjne gry na Zachód – jeszcze za czasów komuny. Zniknęła jednak wraz z upadkiem systemu. Branżę, którą mamy teraz, w pewnym sensie zakładaliśmy od nowa. Wtedy mówiliśmy, że to takie otwarcie. Nie wiedzieliśmy tego, ale Electro Body było pierwszą profesjonalnie wyprodukowaną, opublikowaną i sprzedawaną w pudełkach grą – taki był zamysł na ten produkt.

Jest też ten Firefight, który był dla Electronic Arts. Myślę, że warto o tym wspomnieć.

Tak, dwa-trzy lata później udało nam się zdobyć kontrakt dla Electronic Arts. Był to taki naprawdę spory sukces i wyprodukowaliśmy grę

Firefight, którą Electronic Arts wydało w 1996 roku. Tylko dwie polskie gry były opublikowane przez Electronic Arts: Firefight i Bullet Store dwadzieścia lat później.

A czy wtedy podpisywaliście umowę faksem?

Tak, to były czasy, kiedy umowy się podpisywało faxem, urządzeniem nieznanym już dziś młodzieży. Cały kontrakt to były dosłownie metry papieru, bo wychodził z rolki i gdzieś tam się wyciągało tę rolkę. Na końcu trzeba to było podpisać, a potem ten sam papier przepuścić z powrotem przez faks w drugą stronę.

Czy to było w języku angielskim?

Tak.

Mieliście tam kogoś, kto wam w ogóle wytłumaczył, co tam jest?

Nie, oczywiście, że nie mieliśmy o tym pojęcia — to była totalna partyzantka. Byliśmy młodzi, w firmie pracowali głównie dwudziestokilkulatkowie. Ja miałem wtedy dwadzieścia sześć, może dwadzieścia siedem lat, inni byli w wieku od dwudziestu dwóch do dwudziestu siedmiu. Robiliśmy coś, czego nikt wcześniej w Polsce nie robił — podpisywaliśmy kontrakt z dużym zachodnim wydawcą. Nie mieliśmy zielonego pojęcia, co dokładnie w nim było, bo cały dokument napisany był prawniczym angielskim, takim typowym „legalese”. Interesowało nas właściwie tylko jedno — czy kasa się zgadza, czyli czy te pieniądze, które nam obiecano, faktycznie dostaniemy. A co dokładnie tam jeszcze było zapisane...

Ale rozumiem, że najpierw podpisaliście umowę, a dopiero potem

dostaliście pieniądze?

Oczywiście, że tak. Jak zwykle, pieniądze dostawaliśmy w transzach — tak jak w każdym tego typu projekcie czy kontrakcie. To nas głównie interesowało. Nie byliśmy w stanie ocenić reszty zapisów — to było głupie, ale nie mieliśmy pojęcia, jak się takie rzeczy załatwia. Dziś wyglądałoby to zupełnie inaczej: prawnicy, negocjacje, każdy warunek sprawdzony pięć razy. A wtedy nawet nie było do kogo z tym pójść. Kogo miałem szukać? Prawnika, który zna amerykańskie prawo, mówi po angielsku i jeszcze rozumie branżę gier oraz kwestie licencji? Przecież wtedy nikt w Polsce nie miał o tym pojęcia. No więc podpisaliśmy.

A to jeszcze dopytam — czy w tym kontrakcie były określone jakieś wymagania dotyczące gry? Na przykład, na jakim sprzęcie miała działać albo jakie miała spełniać warunki?

Tak, głównym wymaganiem było to, żeby gra działała pod Windows 95. Podpisywaliśmy kontrakt w 1995 roku, chyba pod koniec, po premierze Windows 95, która odbyła się chyba we wrześniu. Oni powiedzieli wtedy, że chcą, by gra była przygotowana właśnie pod Windows 95, bo Microsoft bardzo promował wtedy Windows jako system do gier. Było wtedy hasło Games for Windows. Założenie było takie, że gry z CD-ROM-u miały działać podobnie jak na konsolach: wkładało się płytę CD-ROM, gra uruchamiała się automatycznie i można było od razu grać, bez instalacji. Takie były wytyczne. To oczywiście częściowo pokrywało się z wymaganiami samego systemu Windows.

To były czasy procesora Pentium, który się pojawił. Dzięki temu

kontraktowi nabyliśmy swoje pierwsze Pentium 66 MHz z szalonymi 16-ma megabajtami RAM-u. I to był wypasiony komputer. I mieliśmy ten kontrakt, zrobiliśmy tę grę. Ta gra nie okazała się takim sukcesem, jakiego się spodziewaliśmy. To była tania produkcja — my mieliśmy wobec niej większe oczekiwania niż Electronic Arts. Dla nich to była po prostu niedroga gra do katalogu, a dla nas coś, co miało otworzyć drzwi do światowej branży gier. No, nie stało się tak. Ale miło to wspominać, bo zrobiliśmy naprawdę dobrą grę, porządną, z multiplayerem. Można było grać po modemie albo w lokalnej sieci. W pamięci wielu osób ta gra gdzieś została. No ale niestety — Porsche sobie po niej nie kupiliśmy.

To ja tylko dodam, jeszcze zanim przejdziemy do kolejnego tematu, że jesteś również wykładowcą na Polsko-Japońskiej Akademii Technik Komputerowych oraz w Warszawskiej Szkole Filmowej. To drugie może być istotne, bo jeszcze o tym nie wspominałeś. Kiedy pytałem cię wcześniej o twoje związki z branżą IT, od razu zaznaczyłeś, że GameDev to nie jest branża IT — choć oczywiście jest z nią powiązany. Może rozwiniesz ten wątek?

Tak – przez lata branżę gier GameDev próbowano przypiąć do IT, bo komputery. Od pewnego czasu zaczęliśmy się bronić, bo GameDev to nie jest branża IT. Łączy się z IT, bo po pierwsze, piszemy programy komputerowe, bo gry to są jednak programy komputerowe. Używamy komputerów wyłącznie do pracy. Korzystamy z narzędzi, infrastruktury informatycznej itd. To są rzeczy wspólne dla mojej branży i IT. Mamy mnóstwo programistów. Natomiast GameDev to jest branża rozrywkowa — część szeroko rozumianej kultury. Bardziej zależy nam, żeby być klasyfikowani właśnie w ten sposób, a nie wrzucani do IT, bo tak

naprawdę do IT nie bardzo pasujemy. My nie rozwiązujemy problemów ludzkości, nie sprawiamy, że ludziom żyje się lepiej, nie ratujemy życia — to zupełnie inny obszar. Dlatego wolimy być kojarzeni z rozrywką, z czymś lżejszym.

Oczywiście to nie znaczy, że wszystkie gry są lekkie czy głupie — absolutnie nie. Po prostu mierzymy się z innymi wyzwaniami i tematami niż branża IT. Ale jeśli chodzi o IP Boxa, to jak najbardziej chętnie z niego korzystamy — w końcu programistów mamy całkiem sporo.

Muszę przyznać, że mimo wszystko – nawet jeśli mówisz, że GameDev to branża rozrywkowa i część kultury – w mojej głowie ciągle to jest IT. Może dlatego, że kiedy ktoś mówi o tworzeniu gier, ja od razu widzę programistę piszącego kod. Wiem, że ktoś musi stworzyć scenariusz, zaprojektować te wszystkie poziomy, musi być grafik, który to robi. W niektórych produkcjach ktoś musi też wziąć kamerę i nagrać ludzi, a potem wrzucić to do gry. Rozumiem, że wszystko się na to składa. Ale zastanawiam się, czy wiele osób nie myśli podobnie jak ja, że GameDev to jednak IT.

Ja wiem. Tak jak mówię, to było kiedyś powszechne. *Co robisz? Zajmujesz się grami? To pewnie jesteś programistą* – takie było skojarzenie. Ale programiści to jest generalnie mniejszość w zespołach growych. Są oni bardzo istotną mniejszością. Najczęściej zarabiają najwięcej ze wszystkich.

Tak jak mówiłeś, są najdrożsi.

Tak, są najdrożsi, najwięcej zarabiają. Często, jeśli potrzebujemy takich programistów – szczególnie gdy robimy coś w zaawansowanej

technologii, w silnikach itd. – to są to ludzie niesłychanie wyspecjalizowani, bardzo, powiedzmy, „łebscy”, z ogromną wiedzą i umiejętnościami, drodzy, i tak dalej. Ale to jest tylko garstka. Cała reszta w projektach to są ci wszyscy pozostali: to są artyści, designerzy, ludzie od produkcji, testerzy. To są gigantyczne tłumy. Oczywiście, jak zawsze opowiadam o tym, kto jest potrzebny do stworzenia gry, to de facto programista zazwyczaj jest niezbędny. Chyba że robimy grę na gotowym systemie, określoną grę – są RPG makery, gdzie możesz sobie wyklikać swojego RPG-a, nie mając programisty. Takie narzędzie załatwia większość, właściwie całą funkcjonalność gry. Ty tylko wypełniasz to grafiką, treścią, questami i możesz nie mieć zielonego pojęcia o programowaniu. Ale tak naprawdę trochę musisz mieć, bo choćby logika gry musi być w jakiś sposób zakodowana, więc trochę tego zawsze będzie. Ten programista jest w każdym projekcie. Tyle mamy wspólnego z IT.

Zatrzymajmy się przy tych programistach. Zastanawiam się, jeśli ktoś nas słucha i chciałby właśnie wejść do branży GameDev w roli programisty, to od czego powinien zacząć?

Od nauki programowania, oczywiście. To jest tak: w programowaniu gier obowiązują te same reguły co przy programowaniu czegokolwiek. Czyli trzeba znać podstawy: tworzenie algorytmów, struktury danych, programowanie w różnych językach, ponieważ różnie to bywa w różnych systemach, grach czy silnikach.

A postawiłbyś na coś konkretnego na samym początku? Wiadomo,

że mniejsza ilość technologii sprawi, że szybciej możemy wejść do branży.

Jeżeli wybierzesz sobie popularny silnik, niech to będzie Unity – bo moim zdaniem jest on przystępniejszy dla kogoś, kto dopiero zaczyna – to w Unity wszystko programuje się w C#. Musisz więc znać C# na jakimś podstawowym poziomie. Ja nie jestem biegłym programistą C#. Jakbyś mnie przepytował z reguł języka, jakichś konstrukcji, to większości z nich nie znam, to znaczy, rozumiem, jak działają, kiedy je zobaczę, ale nie znam ich „na wrywki”. Natomiast to mi nie przeszkadza. Potrafię zakodować całą grę w C# w Unity i będzie działała dość sprawnie. Jestem przeciętnym programistą C#. Muszę jednak umieć konstruować algorytmy, którymi zakoduję mechanikę gry. Muszę mieć tę umiejętność, a ona jest niezależna od języka, prawda? Co radzę? Radzę by szybko, jak najszybciej, zainteresować się zejściem na niższy poziom, czyli na poziom C++. Nie mówię już o samym C – fajnie by było go liznąć, chociażby z czystego C, żeby poczuć smak, jak się pracuje na tym poziomie, o jeden krok wyżej od Assemblera.

Albo niesmak.

No tak, żeby trochę pomyśleć, żeby po zmagać się z problemami zarządzania pamięcią, a nie polegać na Garbage Collectorze. Żeby wiedzieć, że kiedy się alokuje pamięć, trzeba ją w odpowiednim miejscu zwolnić. Żeby spróbować takich rzeczy, nauczyć się minimalizować alokacje w pętlach, znać te tematy. Uważam, że jeśli myśli się poważnie o programowaniu gier, to w C++ trzeba wejść jak najszybciej. Wejście w C++ otwiera nam też drogę do poważniejszego programowania w Unrealu, a to już jest dużo fajniejsza zabawa. Chociaż ja akurat nie

jestem fanem. Czekam, kiedy do Unreala wróci język skryptowy – podobno ma wrócić – bo nie chce mi się wracać do C++. Ja zacząłem programować w C++. Chociaż moja pierwsza gra, Electro Body, była w C. W C++ zaczęliśmy FireFight już był, prawda? Powiedzmy, dwa lata później zaczęliśmy – ale to było takie C z klasami. Jeśli chodzi o szablony, o template'y, to w C nie użyłem ich chyba nigdy w życiu, ale gdzie indziej mi się to zdarzało. Trzeba jak najszybciej zacząć się tego uczyć.

Oczywiście, moim zdaniem, trzeba znać Pythona, bo to uniwersalny język do pisania szybkich skryptów, z gigantyczną biblioteką do wszystkiego. Jeżeli potrzebujemy czegoś szybko – a potrzebujemy tego bardzo często przy tworzeniu gier – napisać skrypty, które automatyzują pewne procesy, przetwarzają jakieś zewnętrzne dane, to Python jest idealny. Jestem wielkim fanem Pythona i uważam, że jest to dla mnie estetycznie najfajniejszy język, z jakim mam do czynienia.

Warto też znać Lua, bo jest to często spotykany język skryptowy napędzający rozmaite, bardziej egzotyczne silniki gier, albo jest osadzany w grach do realizacji różnych zadań. A jeśli chcemy być specjalistami od największej platformy growej, czyli Robloxa, to cały Roblox jest kodowany w ich wersji Lua, ponieważ musieli ją odpowiednio zabezpieczyć na potrzeby platformy. Jeżeli chcemy tworzyć tak zwane doświadczenia robloxowe, musimy znać Lua, co nie jest specjalnie trudne, bo podręcznik Lua jest cieniutki, więc każdy jest w stanie go opanować.

Trzeba natychmiast otwierać się na inne języki programowania. Nie chodzi o to, by wchodzić w jakąś egzotykę. Ale jeśli już o tym mowa,

myślę, że odrobina wiedzy frontendowej czy też JavaScriptu może się przydać. Być może będziemy chcieli stworzyć jakąś aplikację i szybciej będzie to zrobić właśnie tymi narzędziami, na przykład wykorzystując Node.js i pisząc wszystko w JavaScriptcie. Ja akurat nie mam tej wiedzy i żałuję. Gdzieś mnie to ominęło. Nigdy tych rzeczy dobrze nie poznałem ani nie zrozumiałem, a to może być naprawdę bardzo przydatne – mam na myśli te frontendowe rzeczy, czyli JavaScript, CSS i tak dalej.

Ja może wejdę w słowo. Jakiś czas temu, razem z Piotrkiem, który jest współorganizatorem game jamu – o którym za chwilę będziemy mówić – nagraliśmy odcinek podcastu, w którym porównywaliśmy Pythona i JavaScript w GameDevie. Rozmawialiśmy o tym, co można zrobić w Pythonie, a co w JavaScriptcie. Ja sam byłem w szoku, że istnieją biblioteki, które pozwalają tworzyć takie fajne rzeczy, w sensie gry w przeglądarce, i to takie, które naprawdę świetnie wyglądały.

No właśnie. Kiedy myśli się o grach od strony programistycznej, możliwości są różne. Oprócz programowania samego gameplayu – czyli tworzenia mechanik gry – jeśli robimy coś w Unity, to najczęściej używamy czystego C# i nic więcej nie potrzebujemy. Natomiast jeśli wchodzimy w Unreal, mamy oczywiście Blueprinty, to jest ten rodzaj programowania wizualnego. Dość szybko jednak poczujemy potrzebę wyjścia do C++, ponieważ to faktycznie otwiera nowe możliwości i po prostu trzeba go znać. Z kolei, jeśli chcemy iść w kierunku programowania grafiki czy programowania engine'owego, wtedy dochodzą jeszcze języki shaderów.

Dochodzą tu kwestie związane z programowaniem równoległym,

ponieważ te wszystkie maszyny to właśnie programowanie i przetwarzanie danych równoległe. To są rzeczy, którymi trzeba się naprawdę mocno zainteresować. Z drugiej strony mamy gry webowe czy mobilne. Tam pojawiają się właśnie te techniki frontendowe, JavaScript. Możemy robić grę w Unity, a potem działa ona w WebGL-u w przeglądarce. Oczywiście, silnik taki jak Unity sprawia, że każda platforma to w zasadzie tylko wybranie platformy i zbuildowanie nowego builda. My możemy nie mieć o tym zielonego pojęcia. Zresztą, ja tak robiłem build na MacOS, choć nigdy nie miałem Maca. Po prostu w Unity wybiera się platformę, klika build, wyskakuje plik DMG, który wysyłam komuś i mówię: *powiedz mi, czy działa*. Tak to wyglądało.

To czasem załatwia sprawę, ale fajnie jest jednak mieć jakąś wiedzę o tym, jak te platformy działają. Chodzi o to, żeby nie rozkładać rąk, jeśli coś nie działa jak należy, albo nie działa optymalnie. Przyznam, że miałem takie epizody. Kiedy robiłem tę wersję na MacOS, Unity wszystko załatwiło. Wyprodukowałem, wysłałem i ktoś mówi *wszystko pięknie, tylko coś tam jest do góry nogami*. A ja na to: *co ja mam zrobić? Nie mam zielonego pojęcia, dlaczego na Macu jest coś do góry nogami, nie mam Maca, nie mogę tego sprawdzić, a nawet gdybym to sprawdził, to i tak nie wiem dlaczego*. Skończyło się na tym, że poprosiłem kolegę z IT, który akurat dobrze znał Maca. Opowiedziałem mu o problemie, on zajrzał na chwilę i mówi: *tak, już wiem. Trzeba tylko jeden parametr na Maca zmienić na odwrotny i będzie działać. Jak wstawisz to w define'y, będzie dobrze*. Zrobiłem to i zadziałało. Warto takie rzeczy wiedzieć. Trzeba się starać mieć najszerszą wiedzę na temat różnych możliwości w grach i jej szukać. Niestety, to wymaga własnej pracy. Tego raczej nie nauczysz się na studiach, bo nie ma takich kierunków. Trzeba rozwijać

się samodzielnie.

Co do tego rozwijania, to przeszliśmy przez te wszystkie tematy, ale chyba nie powiedziałaś o najważniejszym: po prostu trzeba zrobić taką grę. Pewnie nie jedną, żeby przekonać się, jakie są problemy i „z czym to się je”. Dzięki temu, idąc później do pracy, możemy po pierwsze pokazać, że już coś stworzyliśmy, a po drugie – pracodawca zobaczy, że nie będzie musiał do nas dokładać, tylko faktycznie nasza praca będzie miała wartość.

Wiesz, zrobienie gry jest obecnie bardzo proste. Mamy bardzo dobre narzędzia, mamy gotowe silniki, które zdejmują z nas sporo takich low-levelowych rzeczy, związanych z maszyną jako taką. Nie musimy się zastanawiać, jak wyświetlić trójkąt na ekranie, bo tam będzie odpowiednia translacja, która nam to załatwi. My tylko zdefiniujemy ten trójkąt. To jest duża pomoc. Możemy się więc skupić bardziej na rzeczach związanych z wymyśleniem, jak gra ma działać i co się w niej ma dziać. To też jest łatwe do zrobienia. Nawet jeśli nie mamy do końca pomysłu, jaka to ma być gra, ja zawsze mówię: *sklonujmy jakąś istniejącą grę*. Oczywiście prostą. Nie klonujmy Wiedźmina 3, tylko sklonujmy jakąś prostą grę. Może to być klasyczna gra, coś nieskomplikowanego. Robimy to po to, żebyśmy zgłębili problem, jak zrealizować to, co robi tamta gra. Pomijam już, że łatwo jest to przeanalizować, patrząc, jak ta gra działa, ale są też gotowe, nazwijmy to, designerskie analizy gier. Ktoś już dokładnie rozkminił, na przykład, jak porusza się każdy duszek w Pac-Manie, bo one poruszają się na różne sposoby. To jest opisane. W ten sposób spada z nas konieczność

zaprojektowania gry, co jest trudne. Mamy gotowy projekt. Teraz siadamy i myślimy: *Dobra, te ludki mają się zachowywać tak i tak, to jak mam napisać ten kod, żeby one się tak zachowywały?* Jak mają wyglądać interakcje? Jak wygląda interakcja ze światem? Jak świat jest zdefiniowany i zapisany, żeby go wyświetlić?

Kiedy się to zrobi, to się tego nauczymy. Nauczymy się zamiany pomysłu na jego konkretną realizację. To zresztą jest tak samo w każdym projekcie IT, ale w grach jest identycznie. Mamy to gotowe. A grafika i cała reszta – albo bierzemy coś gotowego, albo coś nabazgrolimy w dowolnym narzędziu, albo teraz to wygenerujemy w czymś, i robimy grę. Robimy ją od początku do końca.

I to jest ważne, co zawsze podkreślam: istotne jest doprowadzenie tego projektu do końca. Gra skończona to nie jest to, że *chodzi ludek i skacze po platformach*. Gra skończona to taka, gdzie chodzi ludek, skacze, są przeciwnicy, są dźwięki, jest muzyka, jest menu, settingsy, intro, outro, start, wygenerowany build – to jest kompletna gra. Chodzi o to, że każdy element zrobiliśmy i dopracowaliśmy, bo w każdym z tych elementów, które wydają nam się może nie tak istotne, bywają różne niespodzianki. Trzeba przez to przejść, żeby móc powiedzieć: *Dobra, to jest kompletny program*. On nie musi być super, nie musi być piękny, nie musi nadawać się do pokazania światu, ale jest kompletny.

Jak to przejdziemy raz, drugi, będziemy już mieli pojęcie, jak się te gry robi. W zasadzie, gdy zrobimy coś, co jest w jakiś sposób oryginalne – nasz pomysł, nasza grafika, czy coś innego – żeby to de facto wypuścić w świat, nie mówię za pieniądze, ale po prostu opublikować, powiedzieć: *Bierzcie, zrobiłem grę*, wysłać do znajomych, to już jesteśmy de facto

częścią GameDevu. Zrobienie gry to już jest ta przepustka. Nikt nam nie zabrania wejść, ale to jest to potwierdzenie, że jesteśmy częścią GameDevu. To, czy będziemy robić gry zawodowo, czy będziemy na tym zarabiać, to już inna historia.

To teraz przechodzę do kolejnego pytania. Czy taki game jam, o którym będziemy sobie mówić, może być dobrym wyznacznikiem wystarczających umiejętności do pracy w branży?

Game jam to jest takie poletko treningowe, jak ja to nazywam. Możemy tam bezpiecznie i bez konsekwencji popróbować, czy potrafimy to zrobić. Game jamy mają różny charakter i formę, ale bywa, że idziemy na taki game jam, spotykamy innych ludzi i decydujemy się popracować razem. I to jest gigantyczna szkoła współpracy, umiejętności współpracy, chęci współpracy. Pamiętajmy, że game jam jest zawsze pod presją czasu. Przeważnie są one ograniczone, na przykład do 24 czy 48 godzin – dłuższe to rzadkość. Mamy więc presję czasową. Musimy szybko nauczyć się dogadywać, współpracować, podzielić się zadaniami. Odpowiadamy za te zadania, a zazwyczaj nie ma tam nikogo, kto mówi: *teraz zrób to, teraz zrób tamto*. Każdy musi zrobić to, co uważa, że należy zrobić.

To są takie warunki, które nawet w firmach normalnie nie występują. Często mówią nam, co mamy robić, czasu jest więcej. A tutaj wszystko jest pod presją. Jeśli się w tym sprawdzimy, jeśli poprobujemy, to po pierwsze zasmakujemy tego i będziemy wiedzieć, czy to nas rajcuje. Może się okazać, że robienie gier wcale nie jest tym, co chcemy robić.

Kiedy zetkniemy się z tymi problemami, z koniecznością rozwiązywania ich, z dogadywaniem się z ludźmi, z samodzielnością – nie wszyscy to od razu umieją – to jest fajny test. Nikt na nikogo tam nie nakrzyczy, jeśli na game jamie tej gry nie zrobi, albo zrobi ją słabą. Wszyscy wiedzą, że to jest wyzwanie, i to trudne.

Ja na przykład bym nie dał rady, to znaczy, ja się do game jamów nie nadaję. Po pierwsze, jestem za stary. Fajnie podejmuje się takie wyzwanie jak się jest młodym, pełnym sił. Jak sam byłem młody to zabrałem się za robienie gier zawodowo i już jakby przepadło. Wtedy nie było game jamów, a teraz już mi się po prostu najzwyczajniej w świecie nie chce, to jest dla mnie zbyt duże wyzwanie – ale bardzo podziwiam tych, którzy podejmują takie wyzwanie, próbują coś zrobić i coś tam skleją. To będzie jakaś gierka, ale oni dobrze się zabawią i mnóstwo nauczą o sobie, nauczą się o tym, jak pracować. To jest super pole treningowe. Jeśli ktoś się waha, zastanawia, nie wie, czy się nadaje, to niech spróbuje, najzwyczajniej w świecie. Niech idzie na game jam albo niech podejmie wyzwanie game jamowe. Bo są game jamy, na których nie trzeba fizycznie być, można sobie poświęcić weekend w domu, narzucić te wymogi sobie, z kolegą czy koleżanką: *teraz siadamy i robimy grę*.

I przetestować. Jeśli nie wyjdzie, nikt nam nie będzie czynił wyrzutów, nie wyrzuci nas z pracy, bo jej nie ma, nikt nas nie skrzyczy, ani nie dostaniemy dwójki, czy jaką tam się teraz daje ocenę. Nic nam nie grozi. Będzie nam co najwyżej głupio, że się nie udało. Ale to też jest lekcja, że być może następnym razem lepiej się przyłożymy, następnym razem lepiej przemyśle pewne rzeczy, nie rzucę się na robienie czegoś

wielkiego, gdy mam na to 48 godzin, tylko zrobię coś małego. Tak dużo pozytywnych lekcji można z game jamu wynieść. Tylko trzeba się przełamać, to jest wyzwanie. Ja bym nie poszedł, bo bym nie dał rady, ale wiem, że jest sporo młodych ludzi, którzy jednak powiedzą: *A ja pójdę, bo ja się nie boję.*

A ja to tak sobie wyobrażam, że warto też spróbować. Będzie też możliwość wygrania, w sensie pierwszego miejsca. Zawsze można się pochwalić takiemu pracodawcy: *wiem, że jeszcze nie mam doświadczenia komercyjnego, ale zobacz, tutaj wygrałem pierwszą nagrodę na game jamie, gdzie w jury występował Maciej, więc to też może być super opcją.*

Ale rzeczywiście, mamy przypadki gier game jamowych, które zostały potem przetworzone w komercyjne produkty. Nie będę wspominał Superhota, który był grą game jamową, ale potem, dwa i pół roku później, i dużo pieniędzy później, stał się gigantycznym sukcesem komercyjnym. Jest też wiele małych gier, które gdzieś tam zaczynały swoją ścieżkę, i to zarówno w Polsce, jak i za granicą – możemy takich przykładów pewnie znaleźć sporo – które zaczynały właśnie jako gra game jamowa, i potem zamieniały się w prawdziwą grę. To, co ty powiedziałaś, to jest bardzo istotna rzecz: teraz jest dużo łatwiej zatrudnić programistę, który jednak ma doświadczenie w grach. Kiedyś było bardzo trudno znaleźć takiego programistę, bo było ich po prostu mało.

Programiści owszem przychodzili na rozmowy o pracę i mówili, *tutaj takie systemy robiłem, tu takie*, ale co mi z tej wiedzy? Te systemy nijak nie przystają do tego, co robimy w grach. W pewnym momencie

stwierdziłem, że ja chcę zobaczyć grę, którą zrobił kandydat. Była taka historia, że przyszedł młody chłopak, który nie miał dużego doświadczenia właśnie w komercyjnych projektach IT, ale powiedział *ja tu zrobiłem taką grę o świętym Mikołaju. Bardzo prosta gra – jest jakaś grafika, ale ja już całą zrobiłem*. Został zatrudniony i pracuje u nas. Jest w branży już od kilkudziesięciu lat.

Ja wolę, gdy przychodzi człowiek, szczególnie człowiek bez doświadczenia, taki potencjalny junior i mówi: *ja bardzo chciałbym pracować przy grach, być programistą, nie mam doświadczenia, ale byłem na game jamie, zrobiłem taką grę, tutaj wydłubałem z kolegą taką grę*. I wtedy ja – *pokaż mi kod*. Popatrzę: *no dobra, widzę, ten kod tutaj trochę jest nieuporządkowany, tutaj trochę jest nie tak, ale widzę, że tu już są jakieś zaczątki porządku, widzę, że zaczynasz się rozwijać i ten kod jest coraz porządniejszy*. Mówimy tu już o konkretach, o grze. Nie muszę się zastanawiać – opowiadasz mi swoją karierę przy produkowaniu jakichś systemów, których nie znam i nie możesz pokazać mi kodu, bo jest to kod korporacyjny i właściwie rozmawiamy nie wiadomo o czym. Podsumowując: budowanie takiego portfolio, nawet małych projektów growych, jest o wiele lepsze niż nie robienie tego i powiedzenie, *ja bardzo bym chciał programować grę*, bo to jest zupełnie inna rozmowa i to jest mile widziane, a już sukcesy game jamowe, bardzo dobrze działają.

Dlatego wszystkich zainteresowanych zapraszamy na game jam, który odbędzie się 7–9 listopada w tym roku. Maciej będzie na miejscu, będzie możliwość sprawdzenia swoich sił, można też dołączyć online, więc wszystkich zapraszamy, a w szczególności

dlatego, że w tym roku ten game jam ma pomóc Łukaszowi, który niestety walczy z chorobą i potrzebuje Waszej pomocy. Wszelkie datki, które prześlecie, trafią właśnie do niego. Piotr organizuje to wydarzenie i postanowił, że wszystkie pieniądze, które Wy przekażecie, przeznaczy na pomoc dla Łukasza. Maciej też zna Łukasza, więc pewnie dlatego bierze udział w organizacji i pomaga w całej akcji, prawda Maciej?

Tak, tak. Obaj – i ja, i Piotr – znamy Łukasza. Kiedy Łukasz skontaktował się ze mną i poprosił o nagłośnienie swojej sprawy, o zbiórce pieniędzy na leczenie, ja to rozpowszechniłem w swoich mediach społecznościowych. Wtedy odezwał się Piotr. Powiedział: *słuchaj, może byśmy zrobili ten game jam?* On nie organizuje go po raz pierwszy. Dodajmy, że to jest game jam w Sierpcu, więc jest to okazja do odwiedzenia Sierpca i skansenu, który się tam znajduje. Całość odbywa się w bardzo fajnym miejscu.

Piotr zaproponował *może zrobimy game jam i tym razem połączymy go z pomocą dla Łukasza? Postaram się szybko porozmawiać z lokalnymi władzami, z lokalnymi przedsiębiorcami, z hotelem, w którym to się odbywa.* Piotrek jest w Sierpcu dobrze osadzony. I to jest właśnie bardzo fajne – to, że coś takiego dzieje się lokalnie, w mniejszej miejscowości, i że może to zorganizować ktoś, kto generalnie, można powiedzieć, wszystkich tam zna. Bardzo mi się to podoba. Podoba mi się przede wszystkim to, że Piotrkowi się chce. Wiecie, to jest coś.

Trzeba też powiedzieć, że on to robi non-profit i jeszcze poświęca swój czas, niektóre rzeczy musi sfinansować sam, a to nie jest małe przedsięwzięcie.

Tak, bardzo to szanuję. Strasznie mi się podoba, gdy ludzie wychodzą poza ramy działalności biznesowej. Oczywiście ktoś może powiedzieć, że on na pewno jakoś na tym zyskuje, bo rośnie jego rozpoznawalność czy sława – i bardzo dobrze. Jeśli przyniesie mu to w przyszłości jakieś korzyści biznesowe, to tylko lepiej. Natomiast to jednak jest wysiłek. Jednak trzeba sobie czegoś odmówić, zaangażować się. Nie ma tu żadnej bezpośredniej gratyfikacji – nikt nie przychodzi z kopertą pieniędzy. Po prostu się pracuje, rozmawia, organizuje, załatwia sprawy i trzeba nad tym wszystkim trzymać pieczę. To naprawdę spory wysiłek, więc bardzo doceniam, że w ogóle chce mu się to robić. Byłem w jury poprzedniej edycji game jamu i bardzo mi się to podobało. To, że on znowu chce to robić, a przy okazji jeszcze w szczytnym, charytatywnym celu – to świetne.

A teraz wróćmy do tej nagrody, o której rozmawialiśmy. Załóżmy, że mamy osobę albo zespół, który zgarnął główną nagrodę i ma teraz świetną grę w portfolio. Myślą o karierze w GameDevie i zastanawiają się, jak pokierować swoją drogą, żeby trafić do produkcji typu Wiedźmin, czyli gier AAA – tych największych. Co powinni zrobić, żeby rozkręcić karierę i dojść tam, gdzie planują?

Nie da się zrobić nic na skróty, nie będę ukrywał. To jest proces raczej żmudny, który trwa. Trzeba po prostu być coraz lepszym w tym, co się robi. Jeśli jestem programistą, to rozwijam się programistycznie. W tym wypadku – jeżeli naszym celem jest praca w AAA np. w CD Projekcie, który przechodzi teraz na Unreala, to robię wszystko, żeby stać się specjalistą od Unreala. Muszę znać C++ i to w tej wersji, która jest

wykorzystywana w Unrealu. Muszę poznać architekturę silnika. Oczywiście – mogę być bardziej ambitny i próbować pójść w kierunku czysto silnikowym, czyli zobaczyć, jak się programuje GPU i pójść w tym kierunku. Mogę zająć się programowaniem rozgrywki – wtedy interesuje mnie jak działa silnik na tym poziomie. Uczę się interfejsów pomiędzy blueprintami, które są taką podstawową metodą programowania C++. To pierwszy krok – zdobywanie wiedzy.

Drugi krok to jest zdobywanie doświadczenia. Z tym zawsze jest trudniej, bo co innego jest przeczytać książkę albo przerobić jakiś kurs, ale prawdziwe doświadczenie wynika z tego, że robimy jakieś projekty. Najlepszą drogą jest chyba jednak przejście przez mniejsze firmy robiące projekty Unrealowe w mniejszej skali i zdobywanie doświadczenia. To niestety potrwa parę lat. To nie jest tak, że ja dzisiaj postanowię być w CD Projekcie i tam pójde – to się nie wydarzy. Owszem to się zdarza, ale rzadko. Znam przypadki, kiedy stażyści CD Projekt zostawali w firmie. Jeżeli jest program stażowy i jeżeli możemy się na niego załapać w CD Projekcie, to oczywiście idziemy i korzystamy.

Warto jeszcze wymienić pewien aspekt, który jest poza merytoryką (jeśli można to tak ująć), ale jest bardzo istotny: jeśli naszym celem jest dostanie się do dużej firmy w Polsce (mówimy o Polsce, bo to bardzo często ta kariera tak wygląda, że duża firma w Polsce, a potem duża firma za granicą), to bardzo istotnym aspektem jest też danie się poznać ludziom z tej firmy – jest to aspekt networkingowy i socjalny.

Musimy pojawiać się na eventach, na których są rekruterzy z danej firmy – trzeba się niestety pojawiać na jakichś konferencjach. To kosztuje i jest

dużo większym wyzwaniem. Trzeba się gdzieś dać poznać, porozmawiać z ludźmi, popytać ich, wykazać się. Nie ukrywam, to jest trudne, bo wymaga otwartości i pewnej dozy ekstrawertyzmu, z czym w IT generalnie jest różnie. Ale chodzi o to, by dać się poznać, poznać tych ludzi, przestać być anonimem, który gdzieś tam tylko aplikuje. Trzeba być kimś, kto zadaje pytania na wykładach, dyskutuje o problemach. Trzeba pokazać, że nie jesteśmy człowiekiem znikąd, tylko kimś, kto się tematem interesuje i ma coś do powiedzenia. To jest jedna kwestia, a druga – to oczywiście ciągle budowanie tej wiedzy i doświadczenia. Trzeba być też aktywnym, być może właśnie w taki sposób, że się coś organizuje, uczestniczy w kołach studenckich czy w różnych wydarzeniach.

Chodzi o to, że im bardziej jesteśmy widoczni, im bardziej jesteśmy aktywni, im bardziej ktoś nas zapamiętał, tym lepiej. To nie jest tak, że nie aplikujemy. Oczywiście, gdy jest nabór na jakąś pozycję, do której byśmy pasowali, z jednej strony aplikujemy, ale z drugiej – jeśli już kogoś znamy – możemy podejść i powiedzieć *śłuchaj, złożyłem CV, możesz jakoś pomóc? Możesz komuś powiedzieć, żeby spojrzeli przychylnym okiem, żeby przenieśli moją aplikację z dołu na szczyt kupki?* Te wszystkie rzeczy działają.

Oczywiście, możemy też po prostu liczyć na łut szczęścia i to się zdarza. Ale z tego, co widzę, ludzie aktywni, tacy, którzy dają się zauważyć, którzy coś próbują robić, oni częściej docierają do lepszych miejsc, bo po prostu się ich zna. Jeżeli ktoś jest aktywny, tworzy jakieś swoje projekty, organizuje choćby game jamy, robi takie rzeczy, to łatwiej jest go wyłować i to jest dobry krok. Takie rady mogę dać.

To ja sobie tak wyobrażam: mamy tu kilkanaście, kilkadziesiąt różnych projektów. Wygrywa jedna gra i Maciej mówi: *ale to jest super programista, super gość. Chyba będę miał dla niego jakiś projekt, w którym mógłby uczestniczyć.* Oczywiście, nie chodzi mi o to, żebyś składał deklaracje, ale czy takie sytuacje też mogłyby się zdarzyć? Nie mówię, że to akurat o tobie.

Oczywiście, że się zdarzają. Może niekoniecznie w game jamach, bo game jam ma bardzo często charakter bardziej rozrywkowy. To jest jednak zabawa. Nie ma takiego ciśnienia, że musimy to zrobić. Są też inne rozmaite konkursy, w których zespoły przedstawiają swój projekt. Jest taki konkurs – nazywa się ZTGKA i odbywa się co roku w Łodzi. Jest to rozwinięcie konkursu studenckiego, który stał się konkursem otwartym. Przeznaczony jest dla studentów i uczniów. Możliwe, że już go poszerzono. Można znaleźć stronę tego konkursu. Bierze w nim udział wiele zespołów, a na finały przyjeżdżają przedstawiciele dużych firm, także tych z sektora AAA. Oni tam nie przyjeżdżają tylko oglądać gry, ale szukać talentów – ludzi, których warto zaprosić do współpracy. I to jest bardzo dobre miejsce, bo to są ludzie, którzy wykonali wysiłek, czyli zrobili tę grę konkursową, zgłosili się, czyli już wychodzą poza średnią. I lepiej się wyciąga, nazwijmy to, potencjalnych pracowników z takiego grona, niż z grona anonimowych ludzi, którzy aplikują i wysyłają CV.

To jest jasne. Ale to się gdzieś łączy z tą rozpoznawalnością. Im bardziej jesteśmy aktywni, na różnych polach, tym jesteśmy bardziej zapamiętywalni. Ja wolę rozmawiać z kimś, kogo znam, niż z kimś kompletnie mi nieznanym. On może być nawet lepszy, ale nigdy nie miałem z nim styczności. Dla mnie to jest bardzo duże ryzyko. Muszę go

poznać, a to dopiero wyjdzie w praniu. Natomiast tutaj mam kogoś, z kim się zetknąłem, widzę, że jest aktywny, widzę, co robi, mieliśmy okazję wielokrotnie porozmawiać. To jest jasne, że najpierw pójdę w tym kierunku. Nie mówię, że on się nada, nie mówię, że on będzie bardzo dobry, ale z mojego punktu widzenia jest to osoba mniej nieznana.

Więc dajmy się poznawać na wszelkie sposoby – czy to właśnie uczestnicząc w game jamach, w konkursach, czy też w ogóle jakąkolwiek aktywnością. Ten aspekt znajomości – może to źle brzmi – ale że znamy ludzi i oni nas znają, jest bardzo, bardzo istotny. To był zresztą istotny aspekt zajęć, które kiedyś prowadziłem na kursie Game Dev School. Nie tylko oferowaliśmy tam rzetelną wiedzę od prawdziwych praktyków, ale każdy kursant miał możliwość tych praktyków poznać i dać się poznać. I już nie był anonimowy. Już był człowiekiem, którego widzieliśmy, z którym rozmawialiśmy, któremu zadawałem pytania. To jest zupełnie inne podejście. To jest moja rada i wiem, że jest trudna, ponieważ sam nie jestem specjalnie ekstrawertyczny i dobry w nawiązywaniu kontaktów. Wiem, że to jest trudne. To jest w ogóle jeden taki „skill”, który polecałbym rozwijać każdemu w IT – czyli umiejętność nawiązywania kontaktów, umiejętność rozmawiania z ludźmi. To jest bardzo trudne, wiem to po sobie.

To tym bardziej warto pojawić się na game jamie, żeby móc potrenować. Aby bardziej was zachęcić do uczestnictwa – powiedz proszę, bo wiem, że starasz się o jakieś dodatkowe nagrody. Czy już coś wiemy na ten temat, czy możemy o tym powiedzieć, czy to jeszcze nie jest ten moment?

Jeszcze nie wiem. Na pewno uda nam się zdobyć trochę kluczy do gier

od różnych polskich firm. Myślę, że nie będzie z tym problemu, jeśli poproszę moich znajomych, żeby trochę tych kluczy mi dali, żebyśmy mogli je rozdać zwycięzcom. To na sto procent się da załatwić, bo już rozmawiałem i wiem. Myślimy i próbujemy zdobyć jakieś bardziej namacalne nagrody. Prowadzimy rozmowy. Będzie GIC, czyli Game Industry Conference w Poznaniu. Myślę, że będzie okazja zaczepić tam jeszcze rozmaitych ludzi. Przyznam, że robimy to trochę na wariackich papierach. To znaczy, decyzja Piotrka, żeby to zrobić zapadła jakieś trzy tygodnie temu. Trzeba było wybrać rozsądny termin, żeby to nie było w zimie i to nam trochę ograniczyło czas. Zazwyczaj tego typu rzeczy przygotowuje się dość długo. Wtedy jest czas na rozmowy.

To dobrze, to zrobmy tak, że na stronie skansengamejam.pl będzie informacja o nagrodach. Piotrek organizuje też live 29 października. Co prawda to będzie już po publikacji tego odcinka, ale pewnie będzie nagranie, więc też je podlinkujemy czy udostępnimy. Będzie tam więcej konkretnych informacji – szczegóły na pewno na stronie, którą podałem.

Myślimy o nagrodach, żeby nie było. To jest tak, że jasne, ten game jam ma dwa cele. Jeden to jest zebrać jak najwięcej pieniędzy dla Łukasza, ale drugi cel to jest to, żeby ludzie się zabawili, a ci, którzy naprawdę się wykażą, dostali nagrody. Na pewno będą jakieś nagrody lokalne. Nikt ze zwycięzców z pustymi rękami nie wyjdzie. Ale my też postaramy się załatwić jeszcze jakieś fajne nagrody dla uczestników, także tych uczestników zdalnych. Bo nie wszyscy oczywiście muszą być w Sierpcu, chociaż zachęcam, bo to jest naprawdę fajna okazja, żeby tam pojechać. Można dołączyć do game jamu zdalnie. Staramy się jakoś tę

pulę nagród skonstruować. Mam nadzieję, że się uda.

Jasne. To może jeszcze wróciłibyśmy właśnie do samych gier, do tematu, o którym dzisiaj rozmawiamy. Mamy gry AAA, o których trochę wspomnieliśmy. Mamy też gry indie, czyli takie mniejsze. Chyba te drugie obecnie bardziej wolisz, przynajmniej z tego, co słyszałem. Więc jak to wygląda z perspektywy pracownika? Chodzi mi o różnice między nimi.

To są zupełnie inne rodzaje projektów i mają swoje wady i zalety. Ja preferuję gry indie głównie z tego powodu, że pozwalają mi się bardziej realizować – nazwijmy to – kreatywnie, artystycznie. Jeśli jestem jedynym programistą w grze indie, to jasne jest, że całość kodu tej gry spoczywa na mnie. To daje gigantyczny wpływ na cały jej kształt. Jest z tego bardzo duża satysfakcja – z tego, nazwijmy to, naklepanego kodu. I to, jak działa, co tam zrobiłem, jakie rozwiązania wymyśliłem – to daje mi dużo satysfakcji jako programiście. Ale oczywiście ja patrzę trochę szerzej na tworzenie gry indie niż tylko na programowanie. Chodzi o to, że każdy, kto pracuje przy tej grze, ma bardzo duży wpływ na nią jako całość. Niezależnie od działki, którą się wykonuje, ten wpływ jest zazwyczaj dużo większy. Bo jeżeli tworzymy grę w zespole kilkuosobowym, to dyskusja o wszystkich aspektach odbywa się w ramach tego zespołu. Nie ma tak, że to jest departament czegoś, a to jest departament czegoś innego, i oni tam robią swoje, a tylko jacyś dyrektorzy na górze łączą te rzeczy i omawiają tematy. Nie, tutaj każdy dyskutuje każdy temat. Mamy więc duży wpływ na to, co robimy. Mamy dużo satysfakcji, bo bardzo widać efekty naszej pracy.

Z kolei, przy tytule AAA, najczęściej będziemy specjalistą w pewnej wąskiej dziedzinie. Może być tak, że jest gigantyczna gra, a my będziemy specjalistą od pływania łódkami. I cały kod pływania łódkami i wszystkiego, co z tym związane, będzie, owszem, naszym dziełem, ale tego pływania łódkami w tych grach może nie za dużo jest i w zasadzie nie mamy się czym pochwalić. Powiemy, że *jak tam znajdziesz, to możesz wsiąść na łódkę i przepłynąć, i ta łódka to moje dzieło*. No super, ale to już, powiedzmy, nie robi takiego wrażenia, nie?

Musimy się z tym liczyć, że w grach AAA wykonujemy dość wąskie zakresy. Bardzo często nie mamy na to żadnego wpływu. To znaczy, to zadanie jest przez kogoś zaprojektowane i my je tylko realizujemy. Oczywiście robimy to jak najlepiej, chcemy, żeby było optymalne, żeby działało zgodnie z zasadami, z tym, co zostało zaprojektowane, no ale wpływ na to mamy niewielki. Możemy co prawda podyskutować o pewnych aspektach designu z tym głównym projektantem, który to robi, coś tam pozmieniać, ale nie możemy nagle przyjść i powiedzieć *nie podoba mi się, jak wygląda walka na miecze, ja miałbym na to inny pomysł*. Wtedy najczęściej powiedzą: *idź stąd, tam są drzwi*. I słusznie, bo nie mamy o tym zielonego pojęcia. Musimy się z tym pogodzić.

Ale z drugiej strony, duże projekty to są duże pieniądze. To jest, nie ukrywam, większa stabilność. Zazwyczaj te firmy, jeśli porywają się na dany projekt, to mają gigantyczne zaplecze finansowe. Taki celowy projekt ma mnóstwo gotówki, np. półtora miliarda złotych. To wystarczy na chwilę, nawet biorąc pod uwagę rozmiar tych firm. Mamy więc tę stabilność, że będą nam regularnie płacić, wiemy, jak wygląda nasza ścieżka kariery, gdzie możemy awansować, co się z tym łączy. Te

awanse często są automatyczne z upływem czasu pracy. To są rzeczy, o których w projektach indie się nie myśli. Wiesz, kto tam myśli o jakichś awansach?

Ale z drugiej strony zastanawiam się, czy to faktycznie jest dużo bezpieczniejsze? No bo jednak w takim dużym projekcie typu Wiedźmin to chyba łatwiej zwolnić jedną, dwie, pięć osób i w zasadzie chyba nikt się z tym nie patyczkuje, bo po prostu jesteś wierszem w Excelu. Natomiast w projekcie indie to jest zespół. Dobra, zaciśnijmy pasa, może każdy obniży swoją pensję, nie wiem, o 500 złotych, jakoś razem dotrwamy. I wtedy jest chyba trochę lepiej?

Tak, to jest ważny aspekt, bo wiesz, to jest tak, że w tych dużych projektach generalnie się siedzi i robi. Tyle, że jeżeli taki duży projekt wpada w problemy, to te problemy są z drugiej strony bardzo duże. To znaczy: bum, zwalniamy trzysta osób. Już nie mówię, że zwalniamy tysiąc, nie? Bo to są rzeczywiście pozycje w Excelu i jak się tam to wyzeruje, to nagle: puf, wynik wyskakuje i nie ma sentymentów, wszyscy wylatują.

A w grze indie to jest kombinowanie: może coś zrobimy, może zaciśniemy pasa, może przetrwamy, bo chcemy to robić, bo to jest fajne, bo widzimy gdzieś tam cel i rezultat. Jeżeli to jest taki naprawdę projekt indie, to wszyscy są zaangażowani. Ale też wszyscy mają szansę partycypować w sukcesie. A ten sukces czasem się zdarza i czasem to jest spektakularny sukces. Wtedy nie jesteśmy na łasce kogoś, kto być może da nam bonus, a może go nie da, bo uzna, że musimy zainwestować w kolejny projekt czy coś tam. Czasem tutaj przychodzą

pieniądze i nagle mówię: no dobra, to teraz mamy tort i dzielimy się nim. Trzeba sobie rozważyć, co dla nas jest większym ryzykiem.

Projekty indie są jednak bardziej ryzykowne z punktu widzenia finansowego, bo zwyczajnie to finansowanie mają wyliczone co do złotówki. Różne fluktuacje powodują, że te problemy finansowe bardzo często dość szybko się objawiają. Bywa też, że łatwiej jest je załatać, bo są relatywnie niewielkie, więc czasem jakaś modyfikacja i nagle już jesteśmy na zero. A w dużej firmie to jest nagle tragedia, bo nie ma skąd wziąć kolejnych stu milionów złotych. To jest taki wybór, który każdy musi jakoś dokonać sam.

Jest jeden bardzo istotny aspekt dużych projektów i nie da się go nijak zrekompensować w małym – w dużych projektach można robić rzeczy, których nigdy nie da się zrobić w małych. I są ludzie, którzy świadomie wybierają właśnie te duże projekty, chociaż na przykład strasznie narzekają, że to korpo, że okropne, że coś tam, że biurokracja, że nie mogą robić tego, co chcą. Czasem pytam takiego kolegę, który regularnie narzeka i pracuje w bardzo dużej zachodniej korporacji growej: *dłaczego? Przecież mógłbyś pracować w dziesiątkach innych firm. Znasz ludzi, którzy odchodzą z tych korpo i zakładają właśnie jakieś takie zespoły indie, tych bardzo ludzi lubisz, mógłbyś z nimi pracować.* Ale on mówi: *Ale tu mam wyzwania, których tam nigdy bym nie miał. Tu mam możliwość pokonywania wyzwań albo pracowania nad technologią, której nigdy nie byłbym w stanie realizować w tak małej grze. Tu mogę wymyślać technologię, która na przykład wymaga wsparcia czterdziestu grafików itp. i tego w małym projekcie nie będzie, bo tam nie ma czterdziestu grafików, tam jest czterech. A to mnie interesuje, to mnie*

rajcuje, bo mnie rajcuje przesuwanie tych granic technologicznych jak najbardziej, a tego się w małej firmie nie da zrobić – bardzo rzadko się to udaje.

Projekty indie mają czasem bardzo fajne rozwiązania technologiczne i można się tam zrealizować, ale są one jednak bardziej rzemieślnicze, po prostu robi się gry. A duże projekty dają możliwości rozwoju, dają możliwości poznania technologii, których nigdy w małym projekcie nie zobaczymy. Już nie mówię o mock-upach i różnych takich rzeczach, ale chodzi właśnie o to, że będziemy mieć dostęp do najnowszych rzeczy. Jeśli np. Nintendo będzie miało najnowszą konsolę, to my dostaniemy tego dev-kita jako pierwsi. Zanim świat w ogóle będzie o tym myślał, to my już będziemy robić grę na tę konsolę. To jest coś, co jest poza zasięgiem małych projektów.

No dobrze, to może teraz zapytam o mity i fakty, jakie istnieją w branży GameDev. Chodzi mi o te, które powodują spadek motywacji i zaangażowania, by być jej częścią. No bo jednak już przed nagrywaniem rozmawialiśmy o wielu rzeczach, które też wyszły, więc teraz jakbyś mógł je powtórzyć albo coś jeszcze dodać.

Na pewno, nie tyle mitem, ile faktem jest, że w GameDevie zarabia się mniej niż w IT. Z punktu widzenia programisty, czyli człowieka, który łączy te dwie dziedziny, programiści z podobnym poziomem doświadczenia na podobnej pozycji, na przykład seniora, w IT będą zarabiać więcej. I to sporo więcej. Gry pobierają taki „podatek za pasję robienia gier”, jak ja to nazywam. Chcesz robić gry? Fajnie. Chcesz

programować gry? Super, ale będzie cię to kosztować. To jest ten podatek – będziesz zarabiał mniej. Dlaczego tak jest? Bo w grach wiemy, że znajdziemy ludzi z pasją do robienia gier, którzy są skłonni ten podatek zapłacić.

Branża IT jest trochę bardziej wyrachowana. Bardzo często w IT ludziom jest obojętne, co robią. Dziś robię to, jutro robię tamto. Obie te rzeczy są równie nudne, więc jakie to ma znaczenie? Ci płacą więcej, idę do nich. W grach niestety jest tak, że ktoś mówi: *Ja chcę robić gry. I bardzo chcę robić gry. Nie chcę programować front-endu. Chcę robić gry. No dobrze, proszę bardzo, zapraszamy.* Nie wiem, jaka jest teraz ta różnica, ale kiedyś była spora. Nie mówię, że dwukrotnie, ale te różnice były naprawdę widoczne. Troszkę się potem poprawiło przy boomie, ale boom objął też IT, więc IT znowu trochę odleciało. A teraz wszyscy mamy kryzys, wszyscy dostajemy po tyłku, więc trochę jest sprawiedliwie. Ale to jest niewątpliwie taki, nie tyle mit, ile fakt, że tak jest.

Drugi mit jest taki, że robienie gier jest wesołe, fajne, no bo to gry, nie? Rozrywka. To praca przy tym jest rozrywkowa. No, nie. Jest tak samo mało rozrywkowa jak praca przy filmie. To, że robimy komedię, nie znaczy, że wszyscy się śmieją podczas pracy. A powiedziałbym raczej, nikt się nie śmieje. I dokładnie tak samo jest z grami. To jest żmudna praca. Współczesne projekty growe trwają wiele lat. Już nie mówię o takich skrajnych przypadkach typu 8 lat Cyberpunka, ale rzadko kiedy da się zrobić sensowną grę. Małe gry robi się oczywiście dużo szybciej, ale taką grę średniego rozmiaru – pecetową czy konsolową – robi się półtora roku do dwóch lat. Czasem dłużej. I to nie jest jakoś tak

przesadnie rozciągnięty czas. Po prostu tyle to zajmuje. I trzeba się z tym liczyć. Trzeba się liczyć, że idziemy do projektów, w których spędzimy najbliższe dwa, trzy lata i to nie będzie bardzo wesoły czas.

Jeszcze dodam od siebie, jeśli faktycznie robi się jakieś duże systemy, to też trwa mnóstwo czasu. I pod koniec już naprawdę dochodzi się do tego, że odlicza się godziny, dni do momentu, aż się go skończy i nie trzeba będzie już na to patrzeć.

Dokładnie to samo jest w grach – jesteśmy w 3 roku produkcji danej gry i nie możemy już na nią patrzeć. Nie możemy w nią grać i nie chce nam się w nią grać.

Z tego, co słyszałem w jednym z wywiadów, powiedziałeś, że nie przeszedłeś chyba żadnej gry, którą produkowałeś?

Dokładnie. Ja nie gram w żadną swoją grę. Po zakończeniu pracy nad grami nie chcę mieć z nimi nic do czynienia. Nie chce mi się angażować. Lubię obejrzeć, jak ktoś gra, na przykład w jakąś małą grę, ale żebym ja tak usiadł i w nią zagrał, to już mi się nie chce. Jestem zmęczony tą grą na tyle, że jeśli mam szansę robić kolejny projekt, to ta faza początkowa projektu jest tysiąc razy fajniejsza niż granie w moją poprzednią grę. Wolę więc skupić się na fazie początkowej.

No właśnie, kolejny mit. Gry robi się fajnie tylko na początku. Jest wtedy super. Wtedy wymyślamy wszystkie fajne rzeczy. Nawet jeśli patrzymy na to stricte od strony IT, czyli na przykład jesteśmy programistami i projektujemy jakieś systemy, które będą w tej grze. Fascynujemy się, jak to zrobimy. Tu zastosujemy taką rzecz, tu jakąś inną. Tutaj wzorce projektowe. Mamy też grove wzorce projektowe, które warto byłoby

znać. Jest taka publikacja *Game Programming Patterns*, która jest zresztą za darmo – jak ktoś jest ciekawy, może sobie ją po prostu ściągnąć. Jest taka strona, która się dokładnie tak samo nazywa i sobie przeczytać. One oczywiście pokrywają się z tymi klasycznymi, ale warto je znać. Kiedy więc myślimy: *dobra, tu zastosujemy to, tu zrobimy tamto, tu wreszcie zrobimy dependency injection po raz pierwszy w tym projekcie* – to wszystko wydaje się bardzo fascynujące. Ale ten długi, żmudny czas pracy: robimy, robimy, robimy. Na początku jest fajnie, bo te przyrosty są dość duże. Realizujemy systemy i one jakoś tam działają.

Odhaczamy sobie duże systemy, a potem dochodzimy do fazy końcowej (o tym właśnie przygotowywałem zajęcia dla studentów). Dlaczego faza końcowa jest tak trudna? Bo nagle okazuje się, że na początku nasza wizja tego, co trzeba zrobić – nawet od strony programistycznej – jest bardzo ogólna. Wtedy wiemy o grze niewiele. Z czasem, gdy zaczynamy ją tworzyć, ta wiedza się pogłębia. Coraz lepiej rozumiemy, jak działa nasza gra, zaczynamy grać w jej fragmenty. Kiedy dochodzimy do etapu alfy i mamy już prawie grywalną wersję od początku do końca, dopiero wtedy uświadamiamy sobie, jak wiele rzeczy pominęliśmy albo źle zaprojektowaliśmy. Trzeba więc dodać elementy, o których zapomnieliśmy, i przerobić te, które nie działają tak, jak powinny. I nagle, w momencie, gdy wydawało się, że większość pracy jest już za nami, pojawia się jej jeszcze więcej.

Wchodzą bugi, których nie sposób jest przewidzieć. Ich może być gigantyczna liczba. Więc nagle pod koniec praca się multiplikuje. Nam się wydawało, że już odhaczyliśmy wszystko, że przecież wszystkie największe rzeczy są już zrobione. I potem, co jest bardzo często

zaskakujące dla wszystkich, także programistów, jest faza kosmetyki, polishowania tej gry. Jest ona bardzo istotna dla podniesienia jakości tej gry, takiej percepcji jej jakości. Nagle okazuje się, że mamy bardzo dużo nieskomplikowanych zadań. Z punktu widzenia wyzwania programistycznego, algorytmu, czy czegokolwiek, są one bardzo proste. Ale to są zadania, których zaliczenie jest bardzo „miękkie”. Ono się opiera na feelingu. Ktoś powie: *już teraz jest dobrze*. Czyli my zmieniamy parametry algorytmu, my zmieniamy algorytm, robimy to w iteracjach. I każda iteracja: *to nie to*. No to dobra, zmieniam to. *To nie to*. Zmieniam coś tam. *No nie, trochę lepiej, ale ciągle nie to*. Zmieniam kolejną rzecz. *Trochę lepiej*. O, *już to się poprawia*. *Ale ciągle nie jest dobre*.

I to jest kilka iteracji. Tych iteracji może być wiele. Może uda się za pierwszym razem, a może będzie ich dwadzieścia. I każde z tych małych, drobnych zadań to jest kilka czy kilkanaście iteracji – nie wiesz, ile. A tych zadań jest bardzo, bardzo wiele. A tu trzeba zmienić animację, tu literki źle wskakują, tutaj jeszcze coś innego. Tu coś się źle rusza, tu za szybko, tu za wolno. I tego jest coraz więcej. I oczywiście można powiedzieć: *Aa, zostawmy to, jakoś to będzie*. *Da się grać? Da się grać*. *To spoko*. Ale wiemy, że każdy z tych drobnych detali wpływa na jakość gry, na percepcję jakości, więc wysiłek w nie włożony bardzo się opłaca. Tylko że, tego jest bardzo dużo! *A przecież mieliśmy już prawie gotową grę*. *Mieliśmy ją już mieć skończoną*. *A my ciągle mamy taką gigantyczną listę*. I to też zaskakuje. *Dlaczego pod koniec musimy tak bardzo pracować? Przez cały rok robiliśmy wielkie zadania, wszystko dobrze działało, a my teraz musimy przysiąść*.

Realnie rzecz biorąc w tej fazie trzeba „przycrunchować”. To jest tak, że, *no, kurczę, już jesteśmy tak blisko, przecież nie rzucimy tego w cholerę. Tylko zepniemy się, przycrunchujemy, przysiadziemy, zrobimy to, zrobimy więcej, więcej, więcej, więcej.* Tym bardziej że my widzimy, że ten cel, czyli ta gotowa gra jest już bardzo blisko i że to się opłaca. Jakby ktoś powiedział na początku, *ej, no, przysiadźmy teraz więcej. No nie, przecież mamy jeszcze dwa lata, to co my się teraz będziemy tutaj szczypać?* Ale jak już jesteś blisko, to jednak nawet nie chodzi o to, że jakiś zły manager przychodzi i każe ci siedzieć po godzinach. Bardzo często jest tak, że *jeszcze to, jeszcze to, jeszcze to zmienię, jeszcze to, bo będzie fajniej.* No i tak się dzieje. Potem kończymy tę grę i zastanawiamy się, czy było warto? Czy powinienem tak siedzieć? A jeśli jeszcze okaże się, że gra nie odnosi takiego sukcesu, jak się spodziewaliśmy, mimo że włożyliśmy w nią tyle pracy i serca, to pojawiają się wątpliwości. Myślisz: *Czy było warto? Czy nie dałem się zrobić w balona? Miał być bonus, a go nie ma?* I zaczynasz się zastanawiać, czy to wszystko miało sens.

Powiem ci, że jak tak o tym opowiadasz, to – nie chcę być okrutny – ale od razu przychodzi na myśl Cyberpunk. Tam chyba jednak zabrakło trochę czasu.

Oczywiście, że zabrakło. Ale jestem stuprocentowo przekonany, że to była świadoma decyzja, narzucona z góry. To nie jest tak, że oni nie wiedzieli, co robią.

No pewnie, trudno było tego nie zauważyć.

Tak, doskonale wiedzieli. I właśnie to jest dobry przykład. W dużych

projektach tak się dzieje bardzo często. Siedzisz, robisz swoje i widzisz, że coś wymaga poprawek. Wiesz, co trzeba zmienić i jak to zrobić. To nie jest żadna niespodzianka, że gracze później narzekają – my już wcześniej dokładnie wiemy, co nie działa. Nagle ktoś przychodzi i mówi: *nie, wypuszczamy tak, jak jest*. I masz wtedy ochotę wywrócić stół. Zwłaszcza jeśli przez ostatnie cztery lata włożyłeś w ten projekt mnóstwo pracy. To potrafi być naprawdę frustrujące. Tutaj wracamy do tego aspektu dużych projektów. To jest tak, że ta decyzja zapadnie ponad tobą. Nie masz nic do gadania. Co więcej, nie będziesz znał prawdziwych powodów. Bo prawdziwe powody to mogą być sute bonusy dla zarządu czy coś takiego. I wtedy myślisz: *jak oni to mogli zrobić?* No mogli. Dostali 60 milionów. I wtedy myślisz, *kurde, to może na ich miejscu też bym tak zrobił?*

Mówimy tu głównie o negatywach. To może teraz coś pozytywnego — powiedz, skoro jesteś w branży już jakieś 30 lat, to dlaczego wciąż w niej zostajesz, mimo tych wszystkich trudnych momentów?

Wiesz co, sam się czasem nad tym zastanawiam. Miewam wątpliwości, oczywiście. Ale potem przychodzi ta myśl: *kurde, przecież ja niczego innego nie umiem robić*. Zacząłem tworzyć gry jeszcze na studiach, mając dwadzieścia kilka lat, i od tamtej pory niczym innym się nie zajmowałem.

To nie zabrzmiało zbyt pozytywnie.

Ale taka jest prawda. Skoncentrowałem się na robieniu gier i ja te gry robiłem. Brałem udział w budowaniu tej branży. Nie chcę mówić, że ona

leży na moich barkach, bo nie leży, ale byłem pionierem, przecierałem szlaki. I nigdy nie zastanawiałem się, po co to robię. Kiedy naszły mnie te wątpliwości, było już trochę za późno. Okazało się, że nic innego nie umiem robić. Nie nadaję się nawet do IT. Umiem programować, ale jestem średnim programistą, ponieważ nie programuję non stop – to rdzewieje. Człowiek zapomina pewne rzeczy, które robi się z automatu. Umiem coś zakodować – poradzę sobie z tym, ale nie jestem biegłym programistą. Byłem nim na początku kariery. Naprawdę non stop się programowało, bo wtedy gry to było głównie programowanie. Owszem, była grafika i tak dalej, ale generalnie to wszystko kręciło się wokół kodu. Dziś praca nad grą to może w 30–50% kod, reszta to są wszystkie inne rzeczy. Nie jestem dobrym programistą, więc nie jest tak, że mogę nagle przyjść do IT i zająć się front-endem. To już odjechało. I nagle okazuje się, że jedyną rzeczą, którą umiem robić są gry. Pozostało mi przetrwać do emerytury lub śmierci, cokolwiek nastąpi wcześniej. I mam wrażenie, że przekroczyłem już punkt, z którego nie ma odwrotu. Nieraz się zastanawiam, czy istnieje dla mnie jakaś inna droga. Może powinienem zrobić kurs na wózek widłowy — bo nigdy nie wiadomo, kiedy taka umiejętność może się przydać.

Ale ciekawe jest to, że kiedy cię słucham — pomijając już fakt, że miało to brzmieć pozytywnie, a nie do końca tak wyszło — to myślę sobie, że naprawdę wiele osób podziwia twoją drogę i to, co osiągnąłeś. Wiem, że to trochę brzmi jak marketing, ale ta pierwsza komercyjna gra na PC, współpraca z Electronic Arts, praca nad Wiedźminem, wszystkie projekty, które po drodze powstały — to przecież duże osiągnięcia. A mimo to w twoich słowach słyszę wątpliwości.

Wątpliwości pewnie ma każdy i nie ma w tym nic złego. Ja po prostu lubię robić gry. Czy chciałbym robić coś innego? Nie, nie chciałbym. Ale życie jest, jakie jest. Czasem człowiek się zastanawia, czy to był najlepszy wybór, bo nie każda gra kończy się sukcesem. Powiedziałbym wręcz, że większość gier nim nie jest. I to jest coś, z czego wiele osób nie zdaje sobie sprawy. Sprawdziłem to zresztą dziś — potrzebowałem danych na zajęcia. W tym roku na Steamie ukazało się ponad 12 tysięcy nowych gier. Ich łączny szacowany przychód to około 3,6 miliarda dolarów. A teraz zgadnij — jaka jest szacowana mediana przychodów jednej gry?

Nie wiem, nie mam pojęcia.

200 dolarów. Czyli 6 tysięcy z tych gier nie zarobiło nawet 200 dolarów. Sporo gier po prostu nie odnosi sukcesu.

Ale wiesz, też się zastanawiam, co to znaczy na Steamie? Czy te gry faktycznie istnieją i są grywalne?

Tak, to są gry, które miały premierę, zostały faktycznie wydane. Łącznie na Steamie jest teraz około 100 tysięcy tytułów. Jasne, wiele z nich to tzw. Al-slopy i inne projekty, które zasłużenie nie zarobiły tych 200 dolarów, ale dane i tak pokazują skalę. Z tych 12 tysięcy gier tylko około 6 tysięcy zarobiło więcej niż 200 dolarów. Większość z 3,6 miliarda dolarów przychodów przypadła kilku procentom tytułów – może kilkuset grom. Jest tam sporo gier polskich, sporo indie, ale nie każda gra jest sukcesem.

Szanse, że w małym projekcie trafi się na grę, która okaże się sukcesem, są naprawdę niewielkie. Każdemu zdarzają się porażki. Ja

sam miałem studio, które musiałem zamknąć, bo zbankrutował wydawca. To były jeszcze czasy dystrybucji pudełkowej, więc bez wydawcy nie dało się funkcjonować. Musiałem zamknąć firmę, przeprowadzić się do Warszawy — nie pochodzę stąd, mieszkam tu od 20 lat, po prostu przeprowadziłem się za pracę. Czasem gra odnosi większy sukces, czasem mniejszy, a czasem jest klapą. Przyczyny są różne. Można zrobić dobrą grę, która się nie sprzeda — przez słaby marketing, brak szczęścia... Sukces w tej branży zawsze ma w sobie trochę przypadku. Więc to naturalne, że co jakiś czas człowieka dopadają wątpliwości. Mnie też dopadają. Ale wciąż lubię robić gry.

Traktuję to trochę jak podatek, który płacę za możliwość robienia tego, co lubię. Poza tym, nic innego nie umiem, więc w pewnym sensie nie mam wyboru. Ale naprawdę — nie chciałbym robić nic innego. Trzymam się tego, szukam okazji, uczę o grach, bo uważam, że to świetna dziedzina. Tylko zależy mi na tym, żeby mówić o tym realistycznie. Nie chcę, żeby ludzie się rozczarowywali przez mity czy złudzenia, że w GameDevie wszystko jest piękne, wszystko ładnie pachnie, jest kolorowo. Nie jest. To nie jest pesymizm, to realizm.

To może dopytam, bo wspomniałeś o AI. Jak obecnie sztuczna inteligencja wpływa na branżę GameDev? Czy branża jest na AI bardziej odporna niż inne specjalizacje?

Moim zdaniem — tak, jest odporna. Choć jeszcze rok temu wydawało się, że nadchodzi przełom, na razie go nie widać. Oczywiście, niektórzy próbują wykorzystywać AI do generowania różnych rzeczy i wrzucają to na

rynek, ale często spotyka się to z negatywną reakcją graczy, którzy uznają, że wszystko tworzone przez AI jest złe. A to nieprawda — to po prostu narzędzie.

AI nas jeszcze nie zastąpi. Może kiedyś, ale nie tak szybko, jak się wydawało. Coraz więcej osób zauważa, że to narzędzie, które jest bardzo przydatne w wielu aspektach. Ja sam często programuję z pomocą Claude’a. Pomaga mi. Jeśli zadanie jest proste, efekt jest dobry. Ale kiedy pojawia się większy stopień skomplikowania, AI się gubi natychmiast. Kod działa, ale nie robi tego, czego chcę. I wtedy wchodzę ja — i nagle okazuje się, że włożyłem tyle wysiłku, że trudniej się wycofać. I wtedy myślisz: *gdybym to zrobił sam, już by działało, a tak straciłem trzy dni i nic z tego*. Wracam i proszę go o jakąś rzecz, on mi to generuje i to działa. Wygeneruje kod, bierze pod uwagę moje zmiany, tu jest debugowa informacja. Ja sam dodałbym to już za trzecim podejściem, żeby to działało. A tu już wszystko jest gotowe, więc to jest super.

Z grafiką jest trudniej. AI potrafi tworzyć ciekawe obrazy, ale nie generuje jeszcze asetów, które można bezpośrednio użyć w grze. W 2D — owszem, można zrobić ikonki czy ilustracje, ale w 3D nadal brakuje narzędzi, które pozwolą wpisać „zrób domek na kurzej stopce w takim kolorze” i od razu wrzucić gotowy model do gry. Jeszcze tego nie ma. Widzę, że postępy były bardzo szybkie, ale teraz, mimo że co chwila pojawiają się nowe próby, to każdorazowo jest to reakcja: *O! Już fajnie wygląda, robimy, puszczamy!* – ale nie, to jeszcze nie to. Podobnie jest z kodem. AI szybko wyprodukuje jakiś prosty kod, ale kiedy każesz mu stworzyć kod złożony, on po prostu nie działa. Jest do niczego i trzeba

go napisać samemu. Jeżeli ktokolwiek mówi, że zwolniono kogoś z powodu AI, to szanujmy się. Owszem, są dziedziny, które bardziej „oberwały”, jak na przykład lokalizacja, czyli tłumaczenia. Osoby zajmujące się tłumaczeniem są już chyba świadome, że to maszyny będą tłumaczyć.

Oczywiście, człowiek wciąż jest potrzebny na etapie QA, czyli do sprawdzenia testów lokalizacyjnych i tak dalej. Teksty trafiają do maszyny i wyskakują w dziewięćdziesięciu sześciu językach, powiedzmy, w miarę przyzwoicie. Natomiast jeśli chodzi o generowanie fajnych dialogów, to człowiek wciąż pisze je lepiej. Jak się okazało, jeśli potrafimy wykorzystać tę maszynę jako narzędzie, ona nam pomaga i jest dużym ułatwieniem, ale nas nie zastępuje. Wydaje mi się, że nie nastąpi to w najbliższych latach. Jesteśmy bardzo daleko od momentu, w którym napiszesz: *zróbmy grę o niebieskowłosym jeżu, który będzie biegał i zbierał to i tamto* i wyskoczy nam Sonic. Nie, jeszcze raz – nie. Jesteśmy od tego przeraźliwie daleko. Więc nie obawiałbym się.

Fakt, że próbuje się powiązać te kryzysowe wydarzenia, głównie zwolnienia, z AI i czasem tłumaczy się je AI, jest moim zdaniem wyłącznie wymówką. Łatwiej jest powiedzieć: *śłuchajcie, wyrzuciliśmy was, bo AI was zastąpi*, niż przyznać, że ktoś w Excelu wyzerował kolumny i powiedział: *do widzenia, nie zgadza nam się budżet*. Słyszałem też taką teorię i myślę, że jest w niej trochę prawdy, iż jest to swoisty komunikat dyscyplinujący. Chodzi o to, żeby przekazać: *śłuchajcie, jeśli będziecie się stawiać, na przykład żądać pieniędzy, to AI was wszystkich wyeliminuje*. Ktokolwiek się na tym zna, wie, że tak nie będzie. Ale komunikat jest, presja jest.

Moim zdaniem ma to pewien sens. Ten kryzys nie jest bez powodu. Koszty tworzenia gier bardzo wzrosły, głównie z powodu podwyżki płac. Wzrost ten nie był tyle wymuszony, ile był efektem nadmiaru pieniędzy i inwestycji w branży. Wszyscy potrzebowali ludzi, więc ich zatrudniali. Był to rynek pracownika – każdy mógł powiedzieć: *będę pracował z Wysp Kanaryjskich za tyle i tyle, i nie macie nic do gadania*. Teraz pieniędzy zabrakło, odpłynęły, i niezbędna jest pewna racjonalizacja. Wiem, że to nie jest komunikat, który ktokolwiek chce słyszeć: „Słuchajcie, teraz będziecie zarabiać mniej”. Ale opcja jest taka: będziecie zarabiać mniej albo nie będziecie zarabiać w ogóle, bo przestaną was zatrudniać. Wszyscy zrozumieli, że tak gigantyczny wzrost kosztów nie ma sensu. Spowodowało to kryzys w Stanach, ponieważ część pracy przeniosła się gdzie indziej – tam, gdzie pracownik jest wyjątkowo drogi.

W Polsce wciąż mamy tę przewagę płacową, ale jednak trochę doszusowaliśmy do stawek zachodnich co powoduje, że jest znowu odpływ do tańszych krajów. Nawet w naszej branży widać, że w tej chwili inwestorzy, którzy finansują produkcję gier, zaczynają szukać zespołów poza Polską. Kiedyś polskie zespoły uchodziły za tanie. Dziś na przykład Turcja jest bardzo dużym konkurentem, bo tam robi się gry przeraźliwie tanio. W porównaniu do Turków nam wydawało się, że my mamy takie „bieda projekty”.

I jeśli gdzieś przy tym wszystkim mówi się o AI to jeszcze nie jest tak dobrze. Natomiast ja jestem entuzjastą technologii jako takiej, w szczególności AI, bo chcę mieć narzędzia, które pozwolą mi robić rzeczy taniej – to jest oczywiste – ale przede wszystkim pozwalają mi robić te rzeczy, których ja sam nie umiem robić. Nie chodzi o to, że nie chcę

zatrudnić grafika, bo jest mi żal i nie chcę dawać grafikom pieniędzy. Bardzo często realia projektów są takie, że chcę zrobić jakiś projekt i w zasadzie jestem w nim tylko ja, pracujący w cudzysłowie za darmo. Grafik byłby dla mnie gigantycznym obciążeniem, ponieważ mam zbyt mały budżet. Więc jeśli pojawi się narzędzie, które pozwoli mi generować grafikę, którą uznam za odpowiednią do tego projektu i która będzie spełniała moje artystyczne wymagania, to przecież dla mnie będzie tylko lepiej. Co mam powiedzieć? Że mi się to nie podoba?

Celem generalnie jest to, aby powstawały gry, a nie żeby dawać ludziom zatrudnienie. Wielu z nas ma z tym problem – ja bardzo często muszę uświadamiać moim kolegom, że celem firm game devowych czy firm IT nie jest dawanie zatrudnienia ludziom z branży, tylko generowanie zysku i zarabianie pieniędzy. Ludzie są tylko środkiem do tego celu, a nie odwrotnie. Każdy, kto prowadził biznes, dobrze o tym wie. Wie, że rządzi tym tak zwany rachunek ekonomiczny i nie chodzi tu o sentymenty. Ja bardzo lubię pracować z ludźmi, lubię pracować w zespole, lubię zatrudniać. Jak tylko mam na to pieniądze, chętnie to robię. Ale jeśli ich nie mam, no to trudno. Sięgam wtedy po narzędzia, po maszynę.

To teraz kolejne pytanie – skoro mówimy o tym rachunku ekonomicznym, to uważam, że jeżeli jesteś lepszym programistą, to można coś zrobić lepiej, szybciej i ten rachunek ekonomiczny jest lepszy. Jaki materiał polecisz osobie, która chce być lepszym programistą GameDev?

To jest trudne pytanie – to zależy, w jakiej dziedzinie chcesz się

specjalizować. Jeśli chcesz zająć się kwestiami silnikowymi, głównie związanymi z programowaniem grafiki, to są takie książki z serii *Graphics Programming Gems* – jest kilka części (1, 2, 3, 4, 5, 6 – nie wiem, który numer jest najnowszy). To są potężne, obszerne tomy. Jest też bardzo dobre wydawnictwo, którego tytułu nie pamiętam, bo to przede wszystkim nie jest moja dziedzina. Ja otwieram taką książkę i na trzecim wzorze już mówię: *O, dziękuję, straciłem kontakt*. Ale jest bardzo dobra książka o architekturze silników. Wiem o niej tylko dlatego, że jednym ze współautorów – bo to jest dzieło zbiorowe – jest Michał Iwanicki. To jest nasz człowiek, który kiedyś pracował przy Wiedźminie, a potem przeszedł ścieżkę kariery w firmach AAA. Był w Naughty Dog, a teraz od wielu, wielu lat jest dyrektorem technicznym w Activision. Jest tam jeden rozdział, który on pisał, więc mi ją pokazywał i cieszył się, że wyszła. To są strasznie low-levelowe i skomplikowane rzeczy, ale po takie właśnie sięga się, jeśli to nas interesuje.

Jeśli natomiast chcemy iść w rozwiązania gameplayowe, nie wiem, czy istnieje coś takiego jak odpowiednik *Gameplay Programming Gems* – szczerze mówiąc, nie jestem pewien. Ale tak jak mówiłem, tutaj idziemy w kierunku tych podstaw, czyli jeśli coś robimy, a jakiś temat nas zainteresuje, to działamy. Na przykład kupujemy sobie dobry kurs na Udemy, czy na jakiegokolwiek innej platformie. W promocji można go kupić nawet za dwadzieścia dolarów. Te kursy są oceniane, mają swoje gwiazdki.

Sporadycznie kupuję całe bundle kursów, nie dlatego, że od razu chcę je przerabiać, ale po to, żeby mieć je pod ręką, gdy przyjdzie mi ochota. Biorę na przykład kursy z Unity – bo ten silnik dość dobrze znam – tylko

po to, żeby sprawdzić, czy są rzetelnie prowadzone. I muszę powiedzieć, że te, które mają dobre oceny, są dobrze przygotowane i dobrze prowadzone. Więc bierzemy sobie takie kursy, kupujemy na wszelki wypadek bundle, na przykład Humble Bundle często oferuje zestawy poświęcone GameDevowi – są tam kursy z różnych dziedzin, a to kosztuje dwadzieścia dolarów. Kupujemy taki kurs. Jak będziemy mieć chwilę, to przerobimy sobie tam programowanie C++ w Unrealu i jedziemy z koksem. To są minimalne inwestycje i bardzo łatwo dostępne.

Oczywiście już nie chcę mówić hardcore'owo – możemy „spiracić” dowolną liczbę książek. Wchodzimy sobie na Anna's Archive i wpisujemy temat. Dostajemy wszystkie książki świata poświęcone danemu tematowi. Nie chcę dyskutować nad etycznością – jestem takim piratem z krwi i kości, wychowałem się w latach 90-tych i nic innego się wtedy nie robiło tylko ściągało się nielegalnie materiały. Ja generalnie uważam, że jeśli ktoś chce uzyskać dostęp do wiedzy i spiracić książkę, to tylko lepiej dla niego.

Nie wiem, czy są w tej chwili jakieś miejsca w sieci, które się odwiedza i gdzie toczy się rzetelna dyskusja. Moim zdaniem zepsuły się te wszystkie internetowe miejsca spotkań, głównie pokoleniowo. Ubolewam nad tym, bo my, te starsze pokolenia, spotykaliśmy się kiedyś na forach dyskusyjnych. Tam dyskutowało się na różne tematy i one żyły całkiem nieźle. Obecnie, z powodu zmiany pokoleniowej, one wszystkie praktycznie poumierały. Bardzo żałuję, bo teraz zastanawiam się, gdzie taki młody człowiek, który chciałby kogoś popytać, ma iść. Umierają portale społecznościowe typu Facebook, a te grupy facebookowe nigdy nie działały dobrze, nigdy nie były dobrym miejscem do dyskusji. Z

TikToka też niczego się nie dowiemy, to jest pewne. Fajnie jest oczywiście być na Twitterze (czyli obecnie X), poszukać sobie ludzi z branży i obserwować, co wrzucają. Czasem udostępniają bardzo fajne rzeczy. Wrzucają linki do ciekawych publikacji albo jakieś white papy. Oczywiście już nie chcę mówić hardcore'owo. Hardcore'owo to możemy „spiracić” dowolną liczbę książek. Wchodzimy sobie na Anna's Archive i wpisujemy, co tam chcemy. Dostajemy wszystkie książki świata poświęcone danemu tematowi. Nie chcę dyskutować nad etycznością – jestem takim piratem z krwi i kości, wychowałem się w latach 90-tych i nic innego się wtedy nie robiło tylko piraciło. Ja generalnie uważam, że jeśli ktoś chce uzyskać dostęp do wiedzy i spiracić książkę, to tylko lepiej dla niego.

to już są bardziej zaawansowane treści. Trzeba tylko wyszukać takie źródła i regularnie śledzić, co publikują. Od tych ludzi można wyciągnąć sporo fajnej wiedzy.

Co do publikacji – chciałbym cię zapytać o to, gdzie możemy cię znaleźć w sieci. Wiem, że mówiłeś, że rzadko udzielasz wywiadów, ale bywasz na wydarzeniach. Może jeśli pojawi się więcej obserwujących, to zachęcą cię do częstszego publikowania?

Ja mam o tyle problem z publikowaniem, że w życiu już wiele publikowałem, zanim stało się to modne. Jeżeli jednak ktoś jest ciekawy, co miałem do powiedzenia, chce zobaczyć jakieś moje wykłady czy wypowiedzi, to wszystkie linki do praktycznie wszystkich wideo są na mojej stronie. Strona nazywa się miasik.net. Każdy może tam wejść –

nie musi mnie wyszukiwać na YouTube – znacznie prościej jest tam kliknąć, bo ja staram się wyszukiwać te nagrania, które są ze mną i które uważam za wartościowe i wrzucam je właśnie na tę stronę.

Jestem obecny na większości eventów branżowych, więc jak ktoś ma okazję i chce, to może wpaść, może mnie zaczepić. Ale tak naprawdę może też pójść na sto innych wykładów, które będą równie ciekawe jak to, co ja mówię, więc polecam. Pewnie najbliższy będzie ten game jam, o którym mówiliśmy. Kto chce, może wpaść na game jam, żeby podzielać przy grze, ale możemy też pogadać – na pewno będzie ku temu wiele okazji. Tyle mogę zaoferować. Można mnie też wygooglować i znaleźć jakieś złote myśli albo wywiady papierowe, które się ukazały.

Czasem zastanawiam się, czy nie powinienem spróbować zostawić tej wiedzy i doświadczenia, jakie mam, w jakiejś trwalszej formie niż wideo na YouTube, bo nigdy nie wiadomo, co z nim będzie. Ale z wiekiem robię się leniwy. Kiedyś miałem więcej zapału do robienia różnych rzeczy, teraz jestem trochę bardziej leniwy. I tak wydaje mi się, że jestem stosunkowo aktywny. Staram się uczestniczyć w wielu przedsięwzięciach. Jak ktoś mnie gdzieś zaprasza, bardzo chętnie biorę udział, więc to nie jest tak, że nic nie robię. Jestem dość aktywny, ale robię się leniwy z wiekiem i coraz rzadziej mi się chce robić różne rzeczy.

To ja powiem tylko tyle: tym bardziej musicie się pojawić w Sierpcu, bo potem nie wiadomo, czy Maćkowi będzie się chciało gdzieś pojawiać dalej. A tu już się zadeklarował, więc zapewne będzie.

Zapraszam wszystkich do Sierpca. Wpadnijcie na game jam. Tak

naprawdę nie ma obowiązku jamowania. Można wpłacić wpisowe, dać datek i nic nie robić. Nikt was z tego nie będzie rozliczał ani wytykał palcami. Mamy cel, a ten cel jest szczytny. Ale fajnie jest też wpaść na przykład w sobotę i pogadać, jeśli będzie taka potrzeba. Ja nie będę jamował, więc będę miał dużo czasu, żeby porozmawiać z każdym chętnym.

Super, to Maćku, w takim razie dziękuję za dzisiejszą rozmowę i podzielenie się swoimi doświadczeniami. Mam nadzieję, że przyjdzie ci ochota, żeby je spisać, bo to też na pewno byłoby bardzo ciekawe. Widzimy się w Sierpcu.

Do zobaczenia.

Dzięki, trzymaj się, hej.