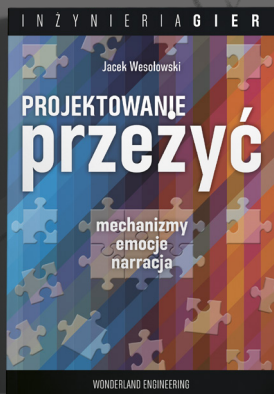


Wersja przedpremierowa

drugiego rozdziału książki z cyklu INŻYNIERIA GIER

„Projektowanie przeżyć. Mechanizmy, emocje, narracja”:



Spis treści:

wonderland-engineering.eu/2018/08/22/projektowanie-przezyc

**JAK SIĘ
BUDUJE GRY**

Jak się buduje gry

Ten rozdział pomoże Ci wykorzystać wiadomości z dalszych części książki podczas budowania własnej gry.

Gdy opowiadam o mojej pracy, słuchaczy niezmiennie szokuje, ile czasu i ludzi potrzeba do zbudowania gry komputerowej. Zupełnie się temu nie dziwię, ponieważ sam robię wszystko, by liczył się tylko utwór, czyli pewne przeżycie. Program komputerowy, który je generuje, ma po prostu działać. A kiedy coś „po prostu działa”, to upodabnia się do magii i tak samo jak czary w bajkach zdaje się pojawiać i znikać na żądanie, bez wysiłku.

W rzeczywistości budowa gry jest nie tylko pracochłonna, ale także złożona. Osoby o bardzo odmiennych specjalnościach muszą się ze sobą dogadać, więc spędzają wiele czasu na uzgodnieniach. Trzeba wciąż się upewniać, że wszystkie informacje są aktualne i dotarły do wszystkich adresatów, a wszystkie zadania są wykonywane lub czekają w kolejkach. Nawet bardzo doświadczone zespoły popełniają poważne błędy, przez co daty wydania wielu gier są przesuwane.

Samodzielny autor często nie musi niczego z nikim uzgadniać, ale w zamian sam staje się wieloma różnymi osobami, na przykład łączy role programisty i projektanta. To się wydaje proste tylko do chwili, gdy siadamy przy komputerze i rozpoczynamy pracę nad pierwszą grą. Wiele osób właśnie wtedy rezygnuje, bo odkrywa, że musi myśleć o zbyt wielu rzeczach naraz. Ale nie ma czego się bać, bo setki tysięcy autorów przeszły już przez to przed nami.

Skorzystamy teraz z ich doświadczeń i zastanowimy się:

- Co to znaczy: „zbudować grę”?
- Jak przebiega proces produkcyjny?
- Na czym polega praca iteracyjna?
- Dlaczego praca projektanta gier jest niewidzialna?
- Jak kontrolować postępy?



przypomnienie



pułapka



lektura



spostrzeżenie



ciekawość

Utwór kontra produkt

Zerknijmy do rozdziału „Czym jest gra komputerowa?” i pożyczmy z niego rozróżnienie na utwór i produkt.

Utwór to zbiór elementów wywołujących określone przeżycie. Na przykład obraz powstaje ze złożenia kształtów i kolorów, opowiadanie z opisów i dialogów, film ze scen, a gra z aktorów, rekwizytów, dekoracji i reguł.

Produkt to rzecz lub zjawisko, które musimy stworzyć po to, by dane przeżycie stało się możliwe. Kiedy myślimy o obrazie jako produkcie, powiemy, że składa się z farb na płótnie lub pikseli na ekranie. Książka to seria zadrukowanych kartek. Film to ciąg obrazów wyświetlanych w zadanej kolejności. Gra komputerowa to program uzupełniony o dane.

W dalszych rozdziałach, rozmawiając o budowaniu gier, skupimy się na wymyślaniu utworu. Będzie to z naszej strony skrót myślowy, ponieważ w grę, która została jedynie wymyślona, jeszcze nie da się zagrać. Osoba o nieskończenie giętkiej wyobraźni mogłaby odbyć całą rozgrywkę wyłącznie we własnej głowie, jednak nawet ona, jeżeli chciałaby się swoim utworem podzielić, musiałaby stworzyć produkt, czyli **wyprodukować grę**.

Jako przykład z medium, które znamy trochę lepiej, ponieważ uczono nas o nim w szkole, przywołajmy *Pana Tadeusza*. Zabierzemy się do tego przewrotnie i przyjrzymy się, jak ta książka powstawałaby dzisiaj. Adam Mickiewicz usiadłby przy komputerze i skorzystałby z edytora tekstu, który by mu na bieżąco poprawiał literówki i błędy ortograficzne. Jeżeli nie byłby zadowolony z danego wersu, przepisałby go dowolną liczbę razy, nie martwiąc się skreśleniami. Gotowy tekst wysłałby do wydawnictwa, gdzie ktoś by go przeczytał i być może zaproponowałby dalsze zmiany. *Pan Tadeusz* nie zawiera ilustracji, jednak nie możemy wykluczyć, że wydawca chciałby książkę udekorować. Z pewnością też wybrałby dla niej okładkę. Okładka nie jest częścią utworu, bo różne wydania *Pana Tadeusza* mają różne okładki, ale jest częścią produktu, bo książka bez okładki to dopiero plik kartek. Po złożeniu tekstu z ilustracjami całość zostałaby wysłana w postaci cyfrowej do drukarni, gdzie rozpoczęłoby się wykonanie licznych kopii.

Najciekawsze w powyższym przykładzie jest to, że faktyczna produkcja *Pana Tadeusza* wcale tak nie wyglądała. Za czasów Adama Mickiewicza nie było jeszcze komputerów, a pierwsza seryjna produkcja maszyn do pisania ruszyła dopiero 21 lat po jego śmierci. Autor zapisał treść odręcznie, a potem przekazał komuś, kto

– również ręcznie – przygotował wersję drukarską. Sposób, w jaki autor wymyślał utwór, mógłby z powodzeniem być dzisiaj taki sam jak 180 lat temu, ale sposób tworzenia produktu bardzo się zmienił.

Wszystko dlatego, że choć zależy nam na uwolnieniu naszych twórczych talentów i wymyśleniu jak najlepszego utworu, to jednocześnie chcemy się przy nim jak najmniej narobić. Szukamy coraz lepszych narzędzi i metod pracy, by zaoszczędzić czas i wysiłek.



Łatwo przyzwyczajamy się do działania już wypróbowanym sposobem, nawet gdy od dawna istnieją korzystne alternatywy. Metody opisane w tym rozdziale też kiedyś okazały się przestarzałe. Nie bój się podawać ich w wątpliwość.

Budowa gry, podobnie jak każdego dzieła, odbywa się w ciągłym konflikcie między pragnieniem, by była jak najlepsza, a potrzebą, by w ogóle powstała. Jeden cel często kłóci się z drugim, bo jeśli pracujemy na chybcika, po to by produkt powstał jak najwcześniej, to umykają naszej uwadze bardziej złożone elementy utworu. Jednak jeżeli zbyt mocno skupimy się na dopracowaniu wszystkiego, po to by utwór był jak najlepszy, data premiery odsuwa się w nieskończoność i w rezultacie nie mamy produktu.

Praca nowoczesnymi metodami, z użyciem dobrych narzędzi, nie tylko przynosi oszczędności, ale również sprawia, że czas, który zyskujemy, tworząc produkt szybciej, ponownie inwestujemy w wymyślanie utworu. Na przykład w jednym z dalszych podrozdziałów przekonamy się, że można równocześnie podnieść jakość i obniżyć koszty, jeżeli większość prac projektanckich wykona się na prototypie, a nie na wersji gry zbliżonej do finalnej.

Zdefiniujmy więc nasze zadanie następująco:

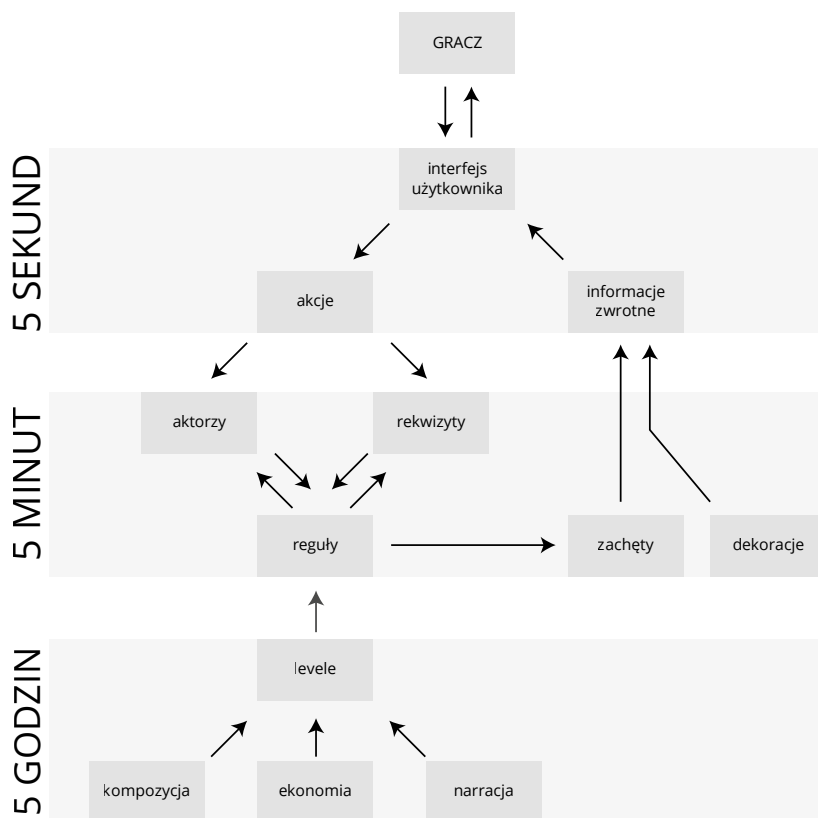
Budowa gry polega na pogodzeniu trzech rzeczy: tego, żeby gra oferowała takie przeżycie, na jakim nam zależy; tego, żeby gra w ogóle powstała i była dostępna dla odbiorców; oraz tego, by nam to zajęło możliwie najmniej czasu.

Trudność z powyższą definicją polega na tym, że nie trzeba w niej wiele zmieniać, żeby pasowała również do *Pana Tadeusza*. Przejdźmy więc teraz do omówienia

specyfiki gier komputerowych i przyjrzymy się powstawaniu jej poszczególnych elementów.

Kto buduje grę

Przypomnijmy wykres z rozdziału „Czym jest gra komputerowa?“, pokazujący elementy składowe gry komputerowej jako utworu:



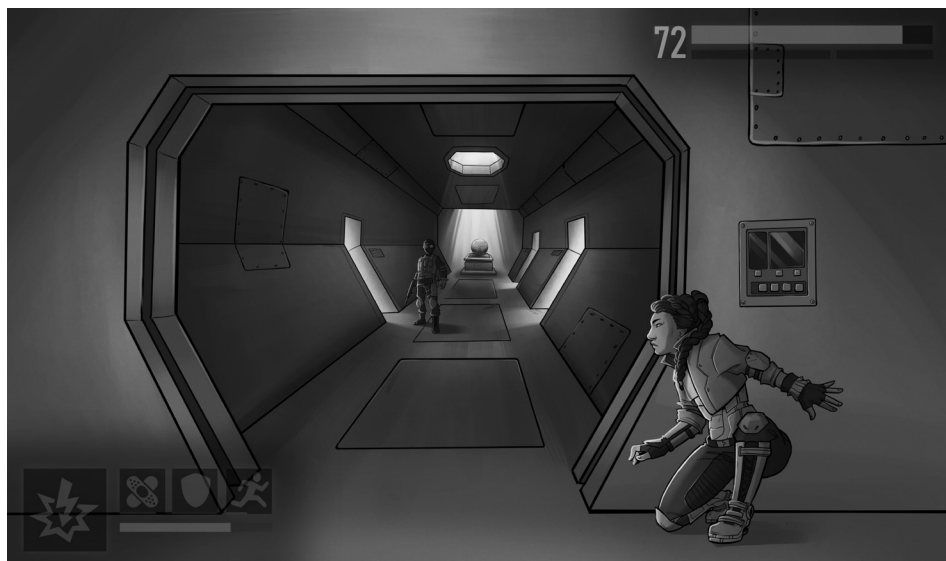
Wiemy już, że grę-produkt dzielimy na inne elementy niż grę-utwór. Części składowych produktu nie ułożymy w równie elegancki wykres, ponieważ jest ich zbyt wiele, a co gorsza większość z nich jest opcjonalna. Istnieją gry bez ani jednej ilustracji i dźwięku, gry bez menu głównego, gry bez trybów rozgrywki przeznaczonych dla samotnego gracza, gry bez fabuły, gry, w których nie przyznaje się nam punktów, gry bez samouczków i tak dalej.



Pewna gra, przy której kiedyś pracowałem, była zbudowana w taki sposób, że jej główny bohater nigdy się nie przesuwał. Zamiast tego przesuwał się cały świat wokół niego, a bohater tylko przebierał nogami. Z punktu widzenia gracza na jedno wychodziło, czyli utwór się nie zmieniał. Z punktu widzenia autorów produkt składał się z innych elementów, niż gdyby to świat gry był nieruchomy, a bohater się po nim poruszał, tak jak podpowiada zdrowy rozsądek.

Zamiast tego przyjrzyjmy się jednej z trzech przykładowych gier komputerowych, na które często będę się powoływał w tej książce. Rozmyślnie sięgnę po utwory zmyślane, ponieważ w prawdziwych grach poszczególne mechanizmy nie są tak elegancko rozdzielone i przez to trudniej byłoby się im przyglądać. W tym wypadku rzucimy okiem na grę „Emigranci”, która z całej naszej trójki jest najbardziej złożona technicznie. Akcja toczy się w trójwymiarowym otoczeniu i zawiera zarówno elementy zręcznościowe, fabularne, jak i taktyczne. Miejsca, rzeczy i postacie wyglądają i zachowują się w przybliżeniu tak jak ich rzeczywiste odpowiedniki, choć oczywiście wprowadzamy wiele uproszczeń i elementów widowiskowych.

Rozważmy następujący widok:



Co widzimy: bohaterka gry kuca przy otwartych drzwiach prowadzących do korytarza i ostrożnie przygląda się patrolującemu go przeciwnikowi. Sam korytarz jest prosty, ale ma wielokątny przekrój. Po obu stronach korytarza znajdują się wejścia do innych pomieszczeń. Na jego końcu widzimy otwarte wejście do sali,

w której stoi podest. Na podeście leży niezidentyfikowany przedmiot o kulistym kształcie. W suficie jest otwór, który najwyraźniej prowadzi do innego, znacznie mniejszego korytarza, zapewne pełniącego funkcję techniczną.



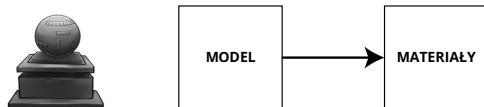
Zauważmy, że zarówno układ pomieszczeń, jak i oświetlenie sprawiają, że podest, choć znajduje się w oddali, zwraca naszą uwagę. Ktoś specjalnie o to zadbał. Gdyby jednak postanowił, że podest ma być niewidoczny, nie musiałby się dodatkowo wysilać. Wystarczyłoby, żeby zamknął drzwi.

W budowaniu gry nie chodzi o to, żeby się napracować, tylko żeby pracować z głową. Bardzo często efekt, którego szukamy, nie zależy od tego, ile wysiłku w coś wkładamy, tylko jak to robimy. Poszukiwanie właściwego „jak” to też praca, nawet jeśli na ekranie jej nie widać!

Przyjrzyjmy się więc podestowi. Po to, żeby było go widać w grze, potrzebne są dwie rzeczy: model i materiały. Model to po prostu kształt danego elementu. Przyjęło się, że modele w grach są dokładnie takimi bryłami, jakie znamy ze szkolnej matematyki, czyli składają się z wierzchołków, krawędzi i ścian, tyle że zwykle są znacznie bardziej złożone. Osoby, które je wykonują, nazywamy potocznie **grafikami 3D**. Skrót 3D pochodzi z języka angielskiego, od słowa „dimension”, które oznacza wymiar, taki jak długość, szerokość lub wysokość. Zwrot 3D oznacza, że dany grafik zajmuje się rzeczami trójwymiarowymi.

Materiały odpowiadają za to, jak wygląda powierzchnia danego przedmiotu: jaki ma kolor, w jaki sposób odbija światło, czy jest gładka, czy szorstka i tak dalej. W naszym wypadku podest jest gładki, a kula ma wypukłości, które wyglądają jak klawisze bardzo nietypowej klawiatury. Oznacza to, że do stworzenia całości będą nam potrzebne dwa różne materiały: jeden na podest, a drugi na kulę. Najprostszy możliwy materiał to po prostu malunek pokrywający powierzchnię modelu, dlatego to zadanie należy do **grafika 2D**.

Gotowy materiał trzeba jeszcze nałożyć na model. Da się to zrobić na nieskończenie wiele sposobów, na przykład można go rozciągnąć, obrócić lub pociąć na mniejsze części i poskładać z powrotem w inną całość. W ten sposób decydujemy na przykład o tym, jak duże są „klawisze” na naszej kuli. To zadanie, polegające na połączeniu elementu dwuwymiarowego z trójwymiarowym, może wykonać grafik 2D lub grafik 3D, zależnie od tego, jak podzielią się obowiązkami.

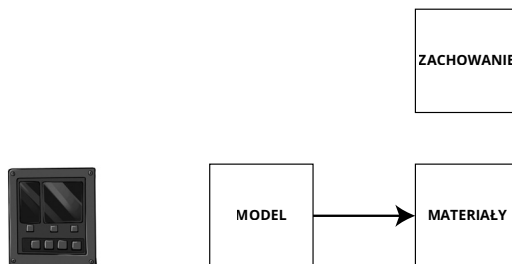


Grafik 2D i grafik 3D mogą być tą samą osobą. Łączenie ról jest szczególnie popularne w małych zespołach.

Gdyby rozpatrywać naszą grę jako utwór, podest byłby typową dekoracją: nic nie robi sam z siebie, nie można nic z nim zrobić, po prostu jest i ładnie wygląda. Jednak kiedy produkujemy grę, nie pracujemy nad „dekoracją”, tylko nad „modelem” i „materiałami”. To, że po złożeniu w całość pełnią one funkcję dekoracji, wynika z naszej umowy z samymi sobą. Ktoś w naszym zespole musiał się zastanowić nad tym, do czego w „Emigrantach” potrzebne są podesty, czyli wykonać pracę, której nie widać na ekranie.

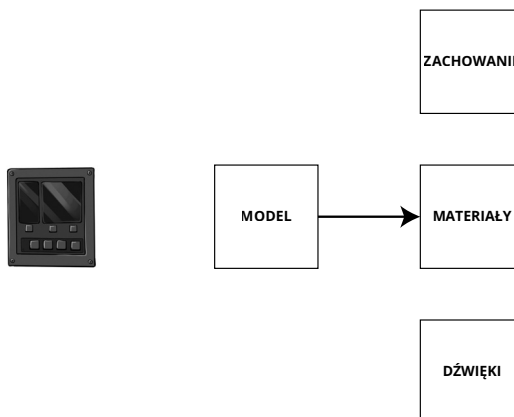
Przyjrzyjmy się teraz elementowi nieco bardziej skomplikowanemu, mianowicie panelowi z przyciskami i wyświetlaczem widocznemu na pierwszym planie, na prawo od bohaterki. On również ma kształt i powierzchnię, więc trzeba przygotować model i nałożyć na niego materiał. Mimo to panel dekoracją nie jest, ponieważ można go użyć. Kiedy ktoś, na przykład bohaterka, wprowadzi prawidłowy kod, na wyświetlaczu pojawi się potwierdzenie dostępu, a drzwi prowadzące do korytarza otworzą się lub zamkną.

Skąd panel „wie”, jak zareagować na jego użycie? Ktoś stworzył program komputerowy, w którym opisał wszystkie rzeczy, które można zrobić z panelem, i wszystkie reakcje panelu na to, co się z nim dzieje. To niezwykle ważne zajęcie powierza się, oczywiście, **programiście**. Zauważmy od razu, że programista, tworząc program opisujący zachowanie panelu, na ogół nie musi dużo wiedzieć o tym, jak panel wygląda. Nad wyglądem rekwizytu i nad jego zachowaniem można pracować osobno.



Nic nie stoi na przeszkodzie, by kilka rekwizytów, które wyglądają zupełnie różnie, opisać tym samym zachowaniem. W takim wypadku programista wykonuje pracę tylko raz, za to graficy mają tyle zadań, ile jest rekwizytów. Widzimy na tym przykładzie, że ilość pracy, której wymaga produkcja danej gry, nie zależy od tego, ile jest w niej elementów łącznie, tylko ile jest elementów, które na tyle różnią się od pozostałych, że trzeba je zrobić od nowa. W dalszej części tego rozdziału przekonamy się, że możemy zmniejszyć liczbę różniących się od siebie elementów gry, posługując się w tym celu parametryzacją.

Nasz rekwizyt nie jest jeszcze gotowy. Kiedy bohaterka naciśnie przycisk na panelu, nie tylko zmieni on wygląd, na przykład zaświeci się, ale również wyda krótki dźwięk. Podobne zdarzenie nastąpi, gdy bohaterka zatwierdzi wprowadzoną kombinację, przy czym dźwięk potwierdzenia dostępu i odmowy będą zapewne różne. Wszystkie dźwięki przygotuje dla nas **inżynier dźwięku**.



Przy okazji: dlaczego dźwięk odmowy jest inny od dźwięku potwierdzenia? Po to by graczowi było wygodniej. Jeden dźwięk dobierzemy tak, żeby kojarzył się z sukcesem, a drugi, by zasugerować, że coś nie wyszło. Potrzebujemy w zespole kogoś, kto zadecyduje wcale nie o tym, że dźwięków ma być mniej

lub więcej, lecz że mają to być dźwięki określonego rodzaju, przywodzące właściwe skojarzenia.

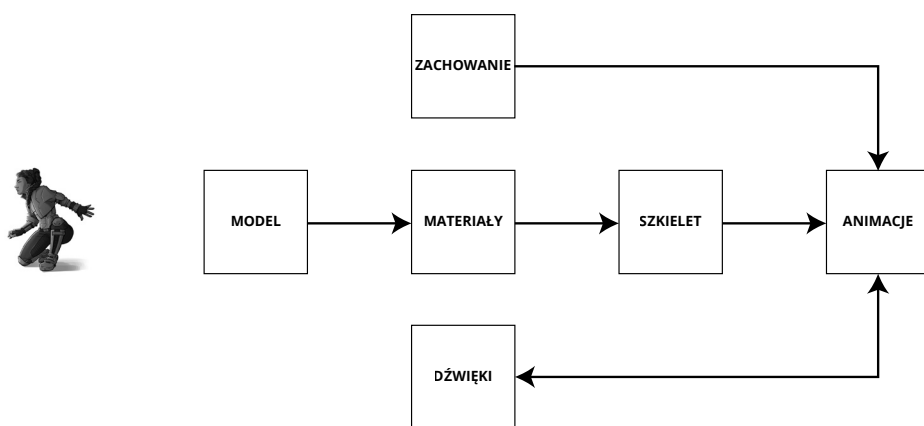
Omówmy teraz występujące w „Emigrantach” postacie. Nie są one ani dekoracjami, ani rekwizytami, lecz aktorami. Skupmy się na bohaterce. Ona też ma model pokryty materiałami i dysponuje pewnymi zachowaniami, na przykład kuca, chodzi, skacze i wychyla się zza rogu. Wymagają one od niej wykonywania skomplikowanych ruchów całym ciałem. Dla gracza jest bardzo ważne, by te ruchy były wyraziste, czytelne i przekonujące, ponieważ patrząc na nie, dowiaduje się, co bohaterka w danej chwili robi. Jeżeli rozgrywka ma być wygodna, to odróżnienie na przykład kucnięcia od uniku nie może zajmować więcej niż ułamek sekundy.

W grach komputerowych bardzo często coś się rusza, ale zależnie od tego, jakie dokładnie są to ruchy, będziemy je tworzyć różnymi metodami. Ruchy proste, takie jak przesunięcie, obrót lub rozciąganie, łatwo opiszemy programem komputerowym. Dlatego zwykle potraktujemy je jak część zachowania danego elementu, a dodanie ich będzie zadaniem dla programisty. Jednak gdy zależy nam na znacznie bardziej złożonych ruchach, wykonywanych przez człowieka lub zwierzę, stworzenie programu, który by je opisał, okazałoby się koszmarnie czasochłonne i niewygodne. Lepiej przygotować taką animację w osobnej aplikacji służącej specjalnie do tego celu. Naturalnie zajmie się tym animator, a ściślej mówiąc – **animator 3D**, ponieważ w tym wypadku wprawiamy w ruch trójwymiarowy model naszej bohaterki. Techniki produkcji tradycyjnej animacji dwuwymiarowej są inne, dlatego traktujemy je jak osobną specjalność.

Model naszej bohaterki nie jest jeszcze gotowy do animowania. Wspomniałem już, że to złożona bryła, ale nie powiedziałem, jak bardzo. Wiele zależy od tego, czy chcemy, by gra wyglądała raczej realistycznie, czy raczej umownie, ale w takiej grze jak „Emigranci” postać może liczyć kilka–kilkanaście tysięcy wierzchołków. Dla porównania, znany nam ze szkoły prostopadłościan ma ich tylko sześć.

Animator, który chciałby ręcznie poruszać każdym wierzchołkiem modelu bohaterki, pracowałby bardzo wolno. Dlatego pomaga sobie dodatkowym, bardzo uproszczonym modelem, który ze względu na charakterystyczny wygląd nazywamy szkieletem. W pierwszym kroku, z użyciem specjalnego narzędzia, animator określa, która część szkieletu odpowiada której części modelu. Następnie animuje się już tylko szkielet, a przełożenie jego ruchów na ruchy modelu jest automatyczne.

Animacja zależy i od zachowania, i od dźwięków. Z zachowania postaci wynika, jakie animacje są potrzebne, bo na przykład postać, która nie walczy, nie potrzebuje animacji ciosów i uników. Dźwięki są potrzebne, gdyż bez nich animacja nie jest kompletna, na przykład widzimy, że postać idzie, ale nie słyszymy jej kroków. Jednocześnie to właśnie z przebiegu animacji wynika, jakie dźwięki będą potrzebne. Na przykład w większości gier postać ma kilka animacji, które wykonuje w chwilach, gdy nic nie robi. Jeżeli animator postanowi, że taka animacja przedstawia drapanie się po głowie, to będzie mu potrzebny inny dźwięk, niż gdyby zamiast tego postać kasłała.



Skoro jednak dźwięki zależą od animacji, a animacje od dźwięku, to od czego zacząć? Z reguły od tej strony, od której będzie nam wygodniej. Robimy po prostu tyle, ile da się zrobić, a w trakcie pracy odkrywamy, czego dokładnie nam brakuje. W powyższym przykładzie animator najpierw wymyśliłby, że animacja przedstawia kasłanie, potem zgłosiłby się do inżyniera dźwięku z prośbą o odgłos kasłania, w oczekiwaniu na niego zająłby się inną animacją, na przykład kucaniem, a po otrzymaniu brakującego elementu wróciłby do pracy nad kasłaniem.

Bardzo często wygląda to jeszcze trochę inaczej. Animator wykonuje wstępną wersję animacji kasłania i prosi inżyniera dźwięku o dostarczenie odpowiedniego odgłosu. Gdy go otrzymuje, często odkrywa, że animacja do niego nie pasuje, bo na przykład słychać, że postać odkasłuje trzy razy, a w animacji widać, że robi to tylko raz. Może to wynikać z tego, że animator i inżynier dźwięku się nie dogadali, ale znacznie częściej wynika z tego, że dopiero po spróbowaniu kilku różnych rzeczy zrozumieli, jak dana animacja powinna przebiegać. W takim wypadku mogą uzgodnić, że albo inżynier dźwięku wykona nowy odgłos, albo animator zmieni

animację. W obu wypadkach pewną już wykonaną część zadania trzeba wyrzucić do kosza i zrobić od nowa.

Na razie się tym nie martwimy. Nie urodził się jeszcze taki autor, który zdołałby całą grę wymyślić od razu w idealnej postaci. Tak jak Adam Mickiewicz w drugim werse *Pana Tadeusza* być może próbował różnych rymów do „zdrowie”, tak samo my przyglądamy się pierwszej wersji danego elementu i odkrywamy, że inna wersja byłaby dużo lepsza. Wykonujemy pewną próbę „w ciemno” i dzięki niej dowiadujemy się, co właściwie trzeba zrobić! Dopiero w kolejnym wykonaniu pojawia się okazja, by tę nowo zdobytą wiedzę wykorzystać. Taki tryb pracy nazywamy iteracyjnym, a każde kolejne wykonanie danego zadania nazywamy **iteracją**.

Zdecydowana większość elementów gry powstaje w trybie iteracyjnym: powtarzamy dane zadanie kilka razy, a za każdym razem dowiadujemy się, co trzeba zrobić, żeby wynik kolejnej iteracji był jeszcze lepszy.

Do iteracji jeszcze wrócimy, a tymczasem dokończmy produkcję bohaterki. Musimy teraz dodać do niej **sterownik**, czyli program odpowiadający za odbieranie bodźców, które wpływają na zachowanie postaci. W przypadku bohaterki, której działaniem kieruje gracz, sterownik będzie odbierał informacje o ruchu myszy lub gątek pada i naciskaniu klawiszy bądź przycisków. Na tej podstawie program ustali, którą akcję gracz zamierza wykonać, i uruchomi przypisane do niej zachowanie.

Choć bohaterką steruje gracz, a przeciwnikami komputer, to różnica między nimi jest mniejsza, niż się wydaje. Przeciwnik składa się z tych samych elementów co bohaterka, wyjąwszy sterownik. Sterownik przeciwnika, zamiast śledzić naciśnięcia przycisków, wykrywa sytuacje zachodzące w grze, takie jak to, że bohaterka znalazła się w zasięgu wzroku danego przeciwnika i jednocześnie nic jej nie zastrasza. Programista buduje sterownik tak, by przeciwnik „wiedział”, że w takiej sytuacji ma na przykład wezwać pomoc lub nakazać bohaterce, żeby się przedstawiła. Zbiór technik programistycznych służących temu, żeby postaci same z siebie „wiedziały”, jak mają postępować, nazywamy potocznie **sztuczną inteligencją**.



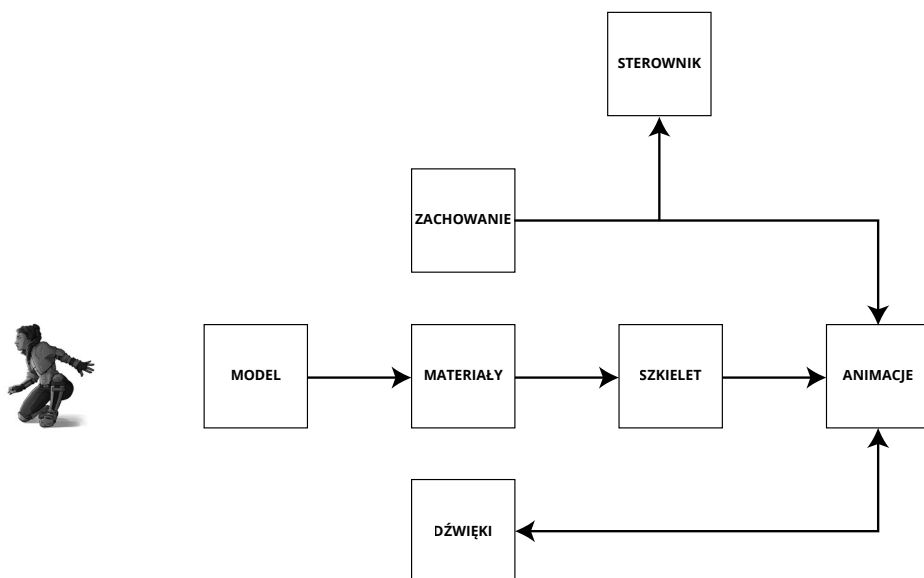
Przyjrzyj się ponownie ilustracji z gry.

Bohaterka wystaje zza ściany, a mimo to przeciwnik jej nie widzi.

Czy tak powinno być?



Nasza bohaterka prawdopodobnie różni się od przeciwników również tym, jakie zachowania ma do dyspozycji. Na przykład w wielu grach chcemy umożliwić graczowi skradanie się, ale nie chcemy dawać tej możliwości jego wrogom, ponieważ zależy nam, żeby to gracz ich zaskakiwał, a nie oni gracza. Za to wyposażymy ich w specjalną czynność polegającą na oglądaniu się za siebie. Gracz zwykle jej nie potrzebuje, bo może zamiast tego po prostu obrócić postać o 180 stopni. Im mniej akcji, tym prostsze sterowanie! Za to postać obsługiwana przez komputer musi otrzymać od programisty bardzo precyzyjną listę rzeczy, które może zrobić, ponieważ komputer, w odróżnieniu od człowieka, nie umie się domyślać i nie zdefiniuje „oglądania się” na własną rękę. Zauważ, że tej różnicy w zachowaniu nie widać na ekranie. Gracz może przejść całą grę i nie zorientować się, że reguły nie są jednakowe dla wszystkich. Ponownie ktoś ustalił za jego plecami coś, czego nie widać na pierwszy rzut oka.



Podsumujmy: grafik 3D wykonuje model, grafik 2D szykuje materiały, a szkieletem i animacjami zajmuje się animator 3D. Programista odpowiada za zachowanie i sterownik. Dźwiękami zajmuje się inżynier dźwięku.

Okazało się, że możemy zbudować od zera całą postać, i to taką, którą gracz może sterować, nie angażując do tego zadania projektanta.

A może coś nam umknęło?

Zadania projektanta

Gdy na poprzednich stronach przedmiotem naszych rozważań były: dekoracja, rekwizyt i aktor, kilka razy rzuciło nam się w oczy, że ważną rolę w grze pełni coś, czego nie widać:

- Drzwi do pomieszczenia na końcu korytarza są otwarte, a mogłyby być zamknięte;
- Kula na podeście ma wypustki przypominające klawisze. Ktoś postanowił, że nie da się ich nacisnąć;
- Dźwięki panelu sterującego drzwiami są dobrane tak, by od razu było wiadomo, czy dana kombinacja jest poprawna, czy nie;
- Bohaterka dysponuje innymi zachowaniami niż jej przeciwnicy;
- Bohaterka wychyla się zza ściany i widzimy jak na dłoni, że przeciwnik powinien ją zauważyć. A jednak – nie zauważa.

Dorzućmy jeszcze kilka spostrzeżeń:

- Bohaterka i przeciwnicy poruszają się z określoną prędkością, niekoniecznie tą samą. Od tych prędkości zależy na przykład, czy bohaterka może łatwo uciec przeciwnikowi, który ją goni;
- Przeciwnicy są uzbrojeni. Ktoś podjął decyzję o tym, czy są gotowi użyć broni przeciwko bohaterce, czy też noszą ją tylko na postrach;
- Otwór w suficie kusi nas jako bezpieczna kryjówka. Mimo że sufit jest bardzo wysoko, bohaterka umie do niego doskoczyć. Na ekranie tego nie widać i gracz nie domyśli się tego na własną rękę. Dla Ciebie w tej chwili też jest to niespodzianka. Ale zaręczam, że to prawda.

To są wszystko bardzo ważne sprawy – tak ważne, że czytasz właśnie książkę, która w całości ich dotyczy. Właśnie takie decyzje jak powyższe sprawiają, że przeżycie grania w daną grę działa tak, a nie inaczej. Gdyby drzwi do pomieszczenia na końcu korytarza były zamknięte, gracz nie wiedziałby, dokąd zmierza. Gdyby bohaterka nie mogła dobiec do bocznych drzwi niezauważona, musiałaby z przeciwnikiem walczyć. A gdyby dało się do czegoś użyć kuli na podeście, to pewnie kryłaby się za tym zagadka lub zwrot akcji.

Żadnej z tych decyzji w grze nie widać. Bez względu na to, czy drzwi mają być otwarte, czy zamknięte, i tak ktoś musi przygotować ich model i materiał, ktoś musi opisać ich zachowanie, a później ktoś musi wstawić je do gry. Jeżeli gra nie będzie działać w dany, konkretny sposób, to po prostu będzie działać w jakiś

inny. Tak czy inaczej napracujemy się przy jej produkcji i powstanie kompletny produkt. Możemy więc odnieść wrażenie, że ustalanie owego „jak” jest proste lub nieważne.

Na podobnej zasadzie, czytając *Pana Tadeusza*, widzimy tylko słowa, które ktoś zapisał jedno za drugim. Nie widzimy wysiłku włożonego przez autora w to, by opowieść toczyła się w taki sposób, a nie inny. A jednak, o ile cenimy twórczość Mickiewicza, to przecież nie za jego umiejętność kaligrafii!

Nie jest nam wszystko jedno, jak skomponowane są elementy gry. Jeżeli zestawimy je ze sobą w pierwszy z brzegu, przypadkowy sposób, to gra będzie przedstawiać graczowi przypadkowe przeżycie. Prawdopodobnie będzie ono mało interesujące. Prawie na pewno nie będzie tym przeżyciem, na którym nam od początku zależało. Na przykład możemy do woli obiecywać sobie, że nasza gra będzie szybka i widowiskowa, ale jeżeli nie zastanowimy się, jak przełożyć to zamierzenie na konkretny wygląd lub działanie poszczególnych elementów, to może się bardzo łatwo okazać, że akcja toczy się za wolno, a gracz nie wie, o co chodzi.

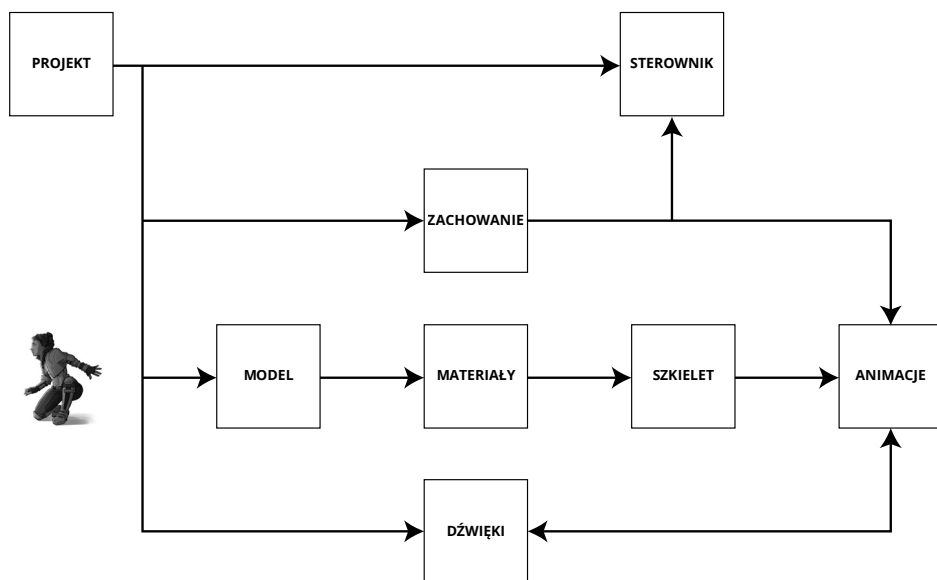
Podczas budowania gry to właśnie **projektant** wykonuje niewidzialną pracę polegającą na ustaleniu, jak poszczególne elementy współgrają ze sobą, tak aby powstało określone przeżycie. Inaczej mówiąc:

Projektant ustala reguły i przebieg gry.

W tym rozdziale nie będziemy jeszcze przyglądać się uważnie ani regułom, ani przebiegowi gry. Przyda nam się jednak spostrzeżenie, że niektóre reguły dotyczą gry jako całości, na przykład „w tej grze można skakać”, a inne konkretnych elementów, na przykład „ta postać skacze na wysokość trzech metrów”. W szczególności możemy wyodrębnić te reguły, które dotyczą konkretnego aktora, takiego jak bohaterka „Emigrantów”. Oto garść przykładów:

- lista zachowań, którymi bohaterka dysponuje,
- lista rekwizytów, których może użyć,
- opis reakcji na oddziaływania, którym podlega ze strony innych aktorów,
- opis sterowania, na które reaguje,
- opis informacji zwrotnych, które są przekazywane graczowi.

Kompletny zbiór reguł, które dotyczą danego elementu, nazwiemy jego **projektem**.

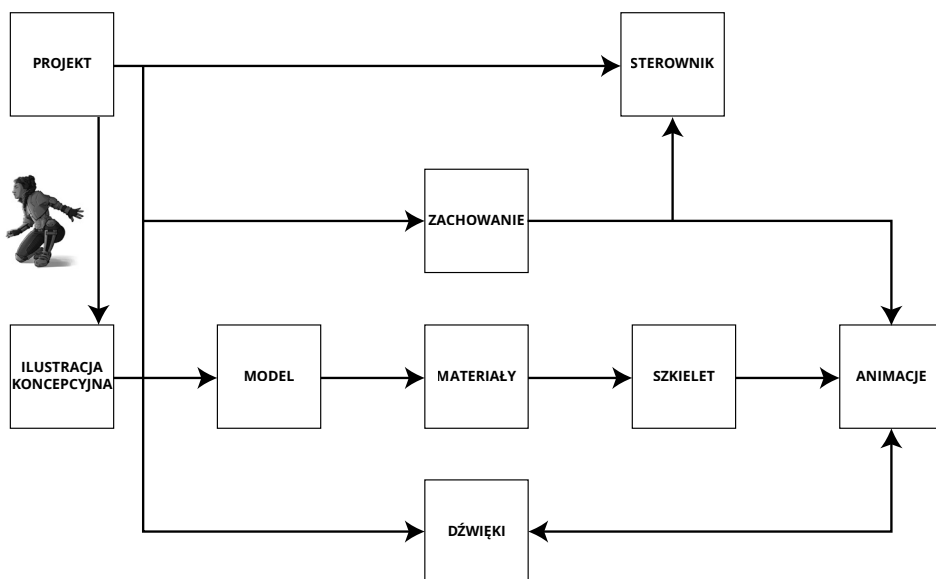


Projekt wpływa na wszystko po trochu, ponieważ nawet jeśli poszczególne elementy postaci powstają osobno, to z punktu widzenia gracza tworzą pewną całość. To właśnie projekt spina je ze sobą. Na przykład wspomniałem wcześniej, że bohaterka „Emigrantów” umie bardzo wysoko skakać. Dopowiem teraz, że zgodnie z jej projektem chodzi w specjalnych, antygrawitacyjnych butach. Gdyby je zdjęła, musiałaby poruszać się tak samo jak wszyscy. Z punktu widzenia gracza jest więc ważne, czy postać nosi antygrawitacyjne buty, kalosze, czy też może jest boso! Powinno to być widoczne na pierwszy rzut oka, czyli trzeba to uwzględnić w wyglądzie bohaterki. W rezultacie projekt wpływa na model, materiały, a nawet na animacje. Ba! antygrawitacyjne buty podczas skoku prawdopodobnie wydają nietypowy dźwięk!

Projekt postaci powstaje bez względu na to, czy go wyrazisz wprost, na przykład zapiszesz w dokumencie, czy też wymyślisz w locie, na przykład podczas programowania zachowań danego rekwizytu lub aktora. Podobnie jak w wielu innych wypadkach, projektant, programista i grafik mogą być tą samą osobą, ale nadal są to różne funkcje. Tak jak wymyślenie wersu „Litwo, ojczyzno moja” nie jest tym samym, co zapisanie go piórem na papierze, tak samo sformułowanie reguły, że postać odziana w odpowiednie buty skacze na duże wysokości, nie jest tym samym, co zaprogramowanie czynności skakania.

Projektuj świadomie. W przeciwnym razie decyzje projektanckie podejmą się same i będą byle jakie.

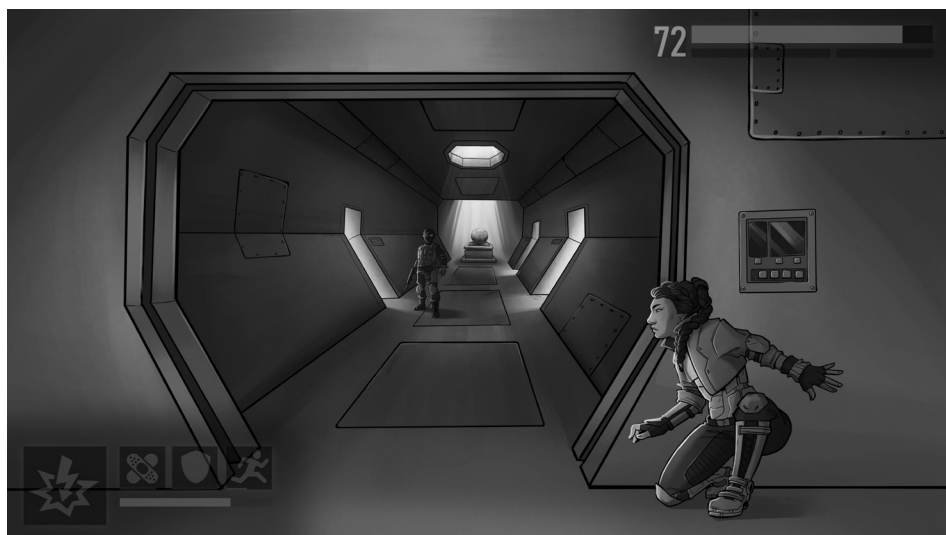
Powszechną praktyką jest rozdzielenie projektu na część interaktywną i część prezentacyjną. Kluczową umiejętnością projektanta jest operowanie elementami interaktywnymi, czyli na przykład tym, jak postać się porusza, jakie ma umiejętności, jakich elementów otoczenia może używać i na co musi uważać. Wyglądem postaci zajmuje się zwykle grafik 2D, który przygotowuje tak zwaną **ilustrację koncepcyjną**. Przedstawia ona dany element gry i ma za zadanie uchwycić nie tylko jego oczywiste cechy, takie jak kształt i kolor, ale również funkcję w grze, w tym zwłaszcza jego „nastrój”. Więcej na ten temat powiemy w rozdziałach „Narracja” i „Kompozycja”. Zauważ, że nie ma potrzeby, by ilustrację koncepcyjną wykonywać tą samą techniką, co grę, bo i tak jej w grze nie umieścimy. Ilustracja koncepcyjna jest częścią projektu i opisuje, jak mają działać model, materiały, szkielet i animacja.



Projektowanie przebiegu rozgrywki

W poprzednim podrozdziale pojawiła się definicja: projektant to ktoś, kto ustala reguły i przebieg gry. W krótkim zdaniu udało nam się połączyć dwie bardzo odmienne, ale ściśle powiązane funkcje.

Spójrzmy jeszcze raz na widok z „Emigrantów”, by przyjrzeć się elementowi rozgrywki, który dotąd nie przykuł naszej uwagi, mimo że zajmuje około 80% powierzchni ekranu.



Mam na myśli sam korytarz wraz z przylegającymi do niego pomieszczeniami. Ich układ ma kolosalne znaczenie z punktu widzenia gracza. Gdyby po bokach nie było drzwi, bohaterka nie mogłaby się za nimi schować. Gdyby na korytarzu nie stali przeciwnicy, pobiegłaby bez przeszkód na jego drugi koniec. Gdyby z kolei sam korytarz nie był prosty, gracz musiałby uważać na to, czy aby za najbliższym rogiem nie kryje się przeciwnik.

Inaczej mówiąc, taki, a nie inny układ otoczenia tworzy pewną sytuację, z którą w danej chwili mierzy się gracz, odczuwając być może takie emocje jak ostrożność, zagrożenie lub ciekawość. W przypadku zwrócenia na siebie uwagi przeciwników mogłaby też nastąpić konfrontacja. Właśnie to mamy na myśli, gdy mówimy w tej książce o przeżyciach.

W poprzednim akapicie, choć mówiłem o otoczeniu, to nie uniknąłem wzmianki o przeciwnikach. Otoczenie, w którym toczy się rozgrywka, tworzy pewną całość

ze wszystkimi rekwizytami i aktorami, którzy się w niej znajdują. Same ściany, bez przeciwników, byłyby przestrzenią do zwiedzania, czyli czynności zupełnie innej od tego, czym zajmuje się nasza bohaterka. Gdyby zaś nie było korytarza ani pomieszczeń, a wszyscy przeciwnicy znajdowali się na otwartej przestrzeni, toby nas po prostu od razu zauważyli i to by był koniec zabawy. Ciekawi są dopiero przeciwnicy wśród ścian, za drzwiami, oświetleni lub skryci w cieniu, hałasujący lub siedzący cicho, patrzący w stronę bohaterki lub zwróceniu do niej plecami i tak dalej.

Za każdym razem, gdy decydujemy, że w grze stanie się konkretna rzecz, taka jak kłótnia lub pościg, musimy ją zaaranżować, używając do tego przygotowanych wcześniej elementów rozgrywki. W tym celu te elementy, zwłaszcza aktorzy, muszą oddziaływać na siebie nawzajem, na przykład zauważać jeden drugiego, otwierać i zamykać drzwi czy omijać ściany. W przeciwnym razie nie mamy do czynienia z rozgrywką, lecz z inscenizacją: gracz wkracza pomiędzy poruszające się według zadanego scenariusza kukietki, ale nic się z tego powodu nie zmienia.

Źródłem przeżycia w grze jest wzajemne oddziaływanie jej elementów, a nie same elementy. Nie stworzysz dobrej gry, po prostu wrzucając do jednego worka jak najwięcej „fajnych” rzeczy.

Przyglądając się naszemu korytarzowi, zauważymy dwa ciekawe podobieństwa do elementów omówionych w poprzednich podrozdziałach. Po pierwsze, korytarz i pomieszczenia to też jest pewna trójwymiarowa bryła, czyli model, tyle że dużo większa i bardziej skomplikowana. Po drugie, pewne elementy otoczenia, takie jak podest z kulą, drzwi, panel i przeciwnik, są powtarzalne. One już zostały wyprodukowane, a teraz możemy wstawić je do otoczenia tyle razy, ile chcemy.

Do budowania otoczeń, w których toczy się rozgrywka, używamy odrębnych narzędzi. Zależy nam na tym, by operować na poszczególnych elementach, czyli dekoracjach, rekwizytach i aktorach, a nie na ich składowych, takich jak modele i zachowania. Jeżeli wstawiamy do korytarza przeciwnika, to interesuje nas cały przeciwnik, a nie tylko kolor jego kurtki. Osobę, która wykonuje to zadanie, obchodzi przede wszystkim to, czy zbudowana przez nią sytuacja, w której przeciwnik stoi akurat w drzwiach drugiego pokoju po prawej, a nie po lewej lub na wprost, jest interesująca dla odbiorcy. Mniej ważne jest dla niej, jaki algorytm steruje przeciwnikiem albo ile materiałów zostało użytych do pokrycia modelu.

Taką osobę nazywamy w tej książce **projektantem treści**, choć na co dzień zetkniesz się raczej z angielską nazwą tej funkcji, która brzmi „level designer”. Jej głównym narzędziem pracy jest „edytor leveli”, który służy przede wszystkim do montowania rozgrywki z gotowych elementów, przygotowanych wcześniej w innych narzędziach.

Różne zespoły różnie podchodzą do tego zadania. Niektóre edytory zupełnie dosłownie ograniczają nasze możliwości do wklejania gotowców i zmieniania niektórych ich cech. Aby stworzyć w takim edytorze korytarz z przyległymi pokojami, należałoby poprosić grafika 3D, żeby nam dokładnie takie otoczenie wyprodukował w swoim programie, a następnie wkleić je do edytora. Bardziej zaawansowane narzędzia pozwalają nam tworzyć proste bryły samodzielnie. W takim wypadku projektant treści sam buduje korytarz i pokoje, nadając im bardzo uproszczone kształty, pozbawione detali takich jak progi, gzymsy czy meble. Taką wersję otoczenia nazywamy **makieta**.

Zauważmy teraz, że prawie wszystkie detale są powtarzalne. Na przykład w typowym budynku większość okien i drzwi wygląda jednakowo, a w typowym biurze meble pochodzą z jednego zestawu. Nie zdziwimy się też, jeżeli w całej grze na stołach będą leżały jednakowe talerze i kubki, a na parapetach identyczne doniczki z paprotkami. Gracz zazwyczaj nie przygląda się dekoracjom wystarczająco dokładnie, by zauważyć, że nie są one tak różnorodne jak w rzeczywistości. W wielu wypadkach powtarzalność elementów jest mu wręcz na rękę. Na przykład jeżeli wszystkie panele sterujące drzwiami wyglądają tak samo, to gracz nie musi ich szukać, bo rzucają się w oczy z daleka. Więcej na ten temat powiemy w rozdziałach „Bodźce” i „Gromadzenie wiedzy o świecie gry”.

Powtarzalność detali jest źródłem kolosalnej oszczędności czasu. Projektant treści prosi grafików, żeby mu przygotowali po jednej sztuce każdego detalu, który będzie mu potrzebny przy budowie danego otoczenia. Następnie wkleja każdy element tyle razy, ile potrzebuje. Początkowo powierzamy tę czynność projektantowi treści, a nie grafikowi, ponieważ każdy element gry, nawet jeśli jest tylko dekoracją, wpływa na przeżycie. Na przykład okno, przez które do pokoju wpada oślepiające światło słoneczne, może utrudniać graczowi zauważenie potwora czającego się w kącie.

Rolą projektanta treści jest sprawienie, żeby gra była ciekawa, a nie ładna.

Ponieważ rozgrywka jest ważniejsza niż dekoracje, to bardzo często projektant najpierw ustala przebieg rozgrywki, nie przejmując się przesadnie dekoracjami,

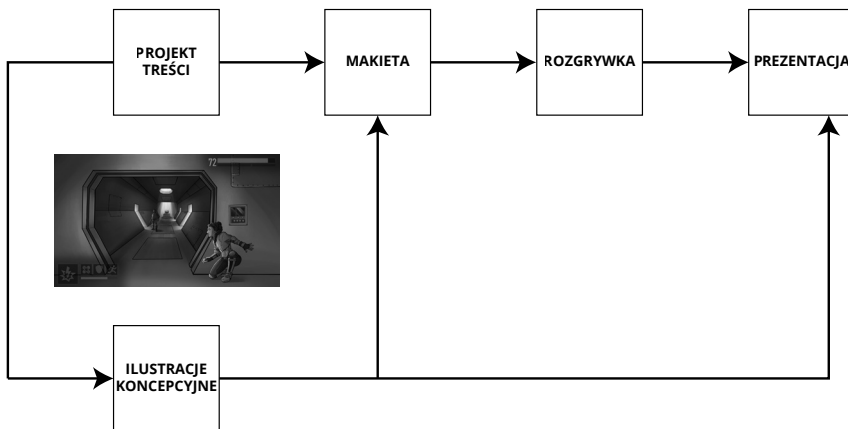
a później grafik poprawia po nim wygląd, w miarę możliwości nie zmieniając nic w rozgrywce. Na przykład gdyby projektantowi zależało na tym, żeby w pokoju było tylko jedno okno, przez które wpada oślepiające światło, to umieściłby je w grze, a następnie poinformowałby grafika, że dalszych okien nie należy dodawać. Grafik być może uznałby, że to brzydko wygląda, więc umówiłby się z projektantem, że doda kolejne okna, ale wszystkie załoni. Podobne rozmowy projektant toczy z innymi specjalistami zajmującymi się prezentacją, na przykład z inżynierem dźwięku, muzykiem i autorem dialogów.



Nie każda dekoracja jest rzeczą, którą widać! Dźwięki otoczenia, takie jak szum wiatru w tle, to też dekoracje. Oczywiście do wyprodukowania odgłosu wiatru nie jest potrzebny model ani materiał.

Tok pracy nad przebiegiem gry jest więc następujący:

- zaczynamy od projektu, czyli opisu tego, co ma się w grze wydarzyć,
- w ramach projektu tworzymy ilustracje koncepcyjne,
- budujemy makietę, czyli wstępną, maksymalnie uproszczoną wersję otoczenia,
- do makiety wstawiamy aktorów, rekwizyty i kluczowe dekoracje, w ten sposób tworząc rozgrywkę,
- działającą już grę uzupełniamy o pozostałe dekoracje i nadajemy jej ostateczny wygląd.



Dlaczego ilustracje koncepcyjne są nam potrzebne już na etapie budowy makiety, skoro właściwy wygląd gry ustalamy na samym końcu? Dlatego że

podstawową funkcją makiety jest ustalenie relacji przestrzennych: czy pomieszczenia są wysokie, czy niskie, czy wszystko jest pod ręką, czy też trzeba się nabiegać, czy krajobraz jest płaski, czy górzysty, czy wszystko toczy się na jednej płaszczyźnie, czy może w wielu wymiarach i tak dalej. Ilustracja koncepcyjna pokazuje te rzeczy. Na przykład gdyby w projekcie zostało zapisane, że bohaterka podróżuje statkiem kosmicznym, projektant treści mógłby w pierwszym odruchu stworzyć ciasne wnętrze przypominające lądowisk księżycowy. Gdyby dopiero później dowiedział się, że na ilustracji koncepcyjnej widać przestronny mostek, wygodne kajuty i pojemną ładownię, w której mieści się mały samolot, musiałby całą dotychczasową pracę nad tą częścią gry wyrzucić do kosza.

Gry bywają bardzo rozległe, a otoczenie, w którym toczy się rozgrywka, może mieć powierzchnię liczoną w kilometrach kwadratowych. Dlatego nie budujemy całego otoczenia naraz. Tak samo jak typowy film jest serią scen, a książka składa się z rozdziałów i podrozdziałów, tak samo w prawie każdej grze da się wydzielić fragmenty w znacznym stopniu niezależne od siebie nawzajem. Na przykład bohaterka „Emigrantów” podróżuje po kosmosie w statku kosmicznym, a od czasu do czasu ląduje nim na wybranej planecie, gdzie przeżywa przygody. W praktyce wygląda to tak, że bohaterka spędza trochę czasu na pokładzie statku, ląduje, załatwia swoje sprawy na planecie, odlatuje, znowu przebywa przez pewien czas na statku, znowu ląduje, znowu zwiedza planetę i tak dalej. Są w tej sekwencji momenty, gdy pewne zdarzenia rozstrzygają się i możemy sobie pozwolić na chwilę przerwy. Na przykład tak dzieje się tuż przed odlotem: wszystkie sprawy na planecie zostały już załatwione, a bohaterka dopiero za chwilę zaplanuje kolejną podróż. Ten moment stanowi naturalne zakończenie jednego wątku i rozpoczęcie następnego.

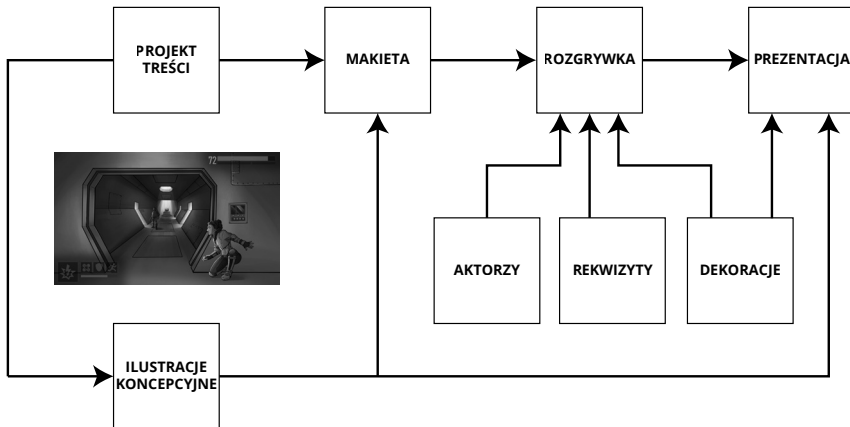
Zauważ, że jeżeli na danej planecie spotkamy przeciwników, to w większości przypadków oni raczej nie będą nas gonić po całym kosmosie, lecz zostaną w domu. Nie stanie się nic złego, jeżeli, przy zachowaniu pewnej ostrożności, potraktujemy zarówno statek bohaterki, jak i każdą planetę jak osobną całość. Będzie je oczywiście spinał scenariusz i może się nawet zdarzyć, że daną planetę odwiedzimy wiele razy, ale większość elementów z danego fragmentu rozgrywki możemy ustalić, nie wiedząc, co się dzieje w pozostałych odcinkach. Taki częściowo niezależny odcinek, który ma łatwy do wskazania początek i koniec, nazywamy **levellem**. Od tego słowa pochodzi angielski termin „level designer” i nazwa narzędzia „edytor leveli”.



Projektowanie i budowa leveli to tak obszerny temat, że napisałem o nim osobną książkę: „Level design dla początkujących”. Jej treść uzupełnia wiadomości z tej książki. Powtarzam się tylko w miejscach, w których jest to konieczne dla zachowania przejrzystości.

Podział gry na levele jest bardzo ważny, jeżeli zależy nam, by gra powstawała szybko. Skoro levele są częściowo niezależne od siebie, to możemy budować je niezależnie, a dopiero na końcu połączyć. W czasie gdy jeden grafik 3D tworzy drzwi, drugi panel, a trzeci kulę na podeście, jeden projektant może budować korytarz z przyległymi pokojami, a drugi zająć się statkiem kosmicznym.

Powiedziałem wcześniej, że level powstaje z gotowych elementów rozgrywki. Czy to znaczy, że najpierw musimy zrobić wszystkie elementy, a dopiero po nich levele? Spójrzmy jeszcze raz na wykres, ale tym razem dorysujmy do niego miejsca, w których będziemy potrzebować dekoracji, rekwizytów i aktorów:



Aktorzy i rekwizyty będą nam potrzebne do zmontowania rozgrywki, ponieważ to są te części gry, z którymi gracz może coś zrobić. Pozostałe dekoracje przydadzą nam się dopiero na samym końcu, gdy będziemy ustalać wygląd levelu. Wniosek: bardzo dużą część pracy projektant treści wykona od razu, nie czekając na grafików i programistów. Co więcej, skoro wygląd liczy się dopiero na końcu, to elementy, z których korzysta projektant, nie muszą od razu wyglądać ładnie, a jedynie robić to, co do nich należy. Na przykład jeżeli przeciwnik biega, skacze, strzela i odzywa się do bohaterki, to może jednocześnie wyglądać jak pudełko z napisem „przeciwnik”, dzięki czemu nie musimy czekać, aż powstaną wszystkie animacje. Więcej na ten temat powiemy w rozdziale „Narzędzia projektanta”, omawiając prototypowanie.

Projektant treści kontra projektant systemu

Mamy już dwa wątki budowy gry komputerowej. W jednym z nich tworzymy poszczególne elementy rozgrywki: dekoracje, rekwizyty i aktorów. W drugim powstają levele. Widzimy też, że gra powstaje dopiero, gdy te dwa wątki ze sobą scalimy, to znaczy umieścimy elementy rozgrywki w levelach. To trudniejsze zadanie, niż wydaje się na pierwszy rzut oka.

Rozważmy następujący przykład. Bohaterka „Emigrantów” chodzi w specjalnych butach, które umożliwiają jej skakanie na duże wysokości. To bardzo widowiskowa umiejętność, ale z punktu widzenia gracza może nie istnieć, dopóki mu się do czegoś nie przyda. Levele muszą być zbudowane tak, by, po pierwsze istniały takie miejsca, gdzie da się doskoczyć tylko przy użyciu tych butów, a po drugie, by istniały powody, dla których warto to zrobić. W przykładowej sytuacji, przytaczanej w tym rozdziale, w suficie korytarza znajduje się otwór, za którym bohaterka może się ukryć przed patrolującym korytarz przeciwnikiem. Jest więc i miejsce, i powód.

Nie możemy tak po prostu wymyślić magicznych butów do skakania. Wprowadzając je do gry, czynimy szereg ukrytych założeń: że przeciwnicy będą się przechadzać podczas patroli, że nie będą zbyt często spoglądać w górę, a przede wszystkim: że levele w naszej grze będą rozbudowane wzwyż, po to by zachęcać gracza do częstego przemieszczania się w pionie.

Możemy nasze rozumowanie odwrócić i wyobrazić sobie, że to projektant treści wymyślił taki level, w którym trzeba schować się przed przeciwnikami pod sufitem. Teraz to on wywiera wpływ na resztę gry, bo właśnie założył, że bohaterka ma możliwość dostania się pod sufit. Czy robi to w ten sposób, że założy specjalne buty i wykona wysoki skok, czy też raczej wespnie się wewnątrz ciastnego szybu wentylacyjnego? Projektant treści musi znać dokładną odpowiedź.

A z trzeciej strony: wiemy już, że poszczególne elementy rozgrywki wpływają na siebie nawzajem. Być może przeciwnicy dysponują specjalnym urządzeniem, które psuje skoczne buty. Być może reguły gry są takie, że bohaterka nie musi wcale przeciwników unikać, bo może z nimi porozmawiać lub przeciwnie – zastrzelić. Stąd, zanim dodamy nowy element, musimy się zastanowić, czy nie popsuje nam on działania już istniejących elementów. Na przykład jeżeli podajemy bohaterce broń, która przenosi dowolną osobę o pięć godzin naprzód

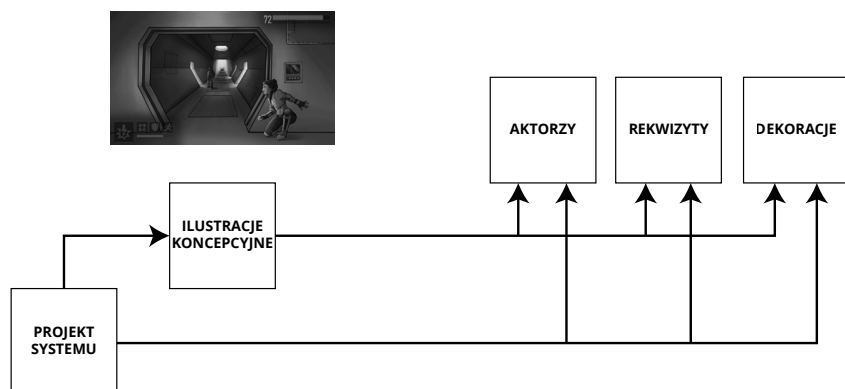
w czasie, to w praktyce gracz będzie mógł każdego przeciwnika pozbyć się natychmiastowo, nieodwołalnie i bez wyrzutów sumienia. Zanim przeciwnik wróci do gry, bohaterka będzie już trzy planety dalej, a w międzyczasie nie będzie musiała ani się skradać, ani uciekać. Urządzenie do przenoszenia przeciwników w czasie uczyni większość innych elementów rozgrywki niepotrzebnymi i dlatego projektant nam go nie udostępni.

Wszystkie elementy rozgrywki tworzące daną grę tworzą pewien system. Dlatego osobę, która zadba o to, by wszystkie dekoracje, rekwizyty, aktorzy i reguły pasowały do siebie nawzajem i działały jako spójna całość, nazywamy w tej książce **projektantem systemu**. Na jego zadaniach skupimy się w dalszych rozdziałach. Anglojęzyczny odpowiednik tego terminu, czyli „system designer”, jest stosowany, jednak znacznie częściej w tym samym znaczeniu sięga się po termin ogólniejszy, „game designer”, czyli „projektant gry”. Jest to o tyle nieścisłe, że projektant treści, czyli „level designer”, również projektuje grę. Po prostu skupia się na innych jej częściach niż projektant systemu.

Projektant systemu dba o to, by wszystkie elementy rozgrywki pasowały do siebie, a nie o to, by każdy z nich był „fajny”.

O ile praca projektanta treści jest namacalna, bo w jej wyniku powstaje konkretny level i konkretna rozgrywka, o tyle projektant systemu ma zajęcie dużo bardziej abstrakcyjne. Interesuje go, jakie są wszystkie możliwe sytuacje z użyciem butów, wszystkie możliwe zależności między bohaterką a przeciwnikami i wszystkie możliwe levele, które da się zbudować w danej grze. Projektant systemu często pełni niewdzięczną rolę polegającą na mówieniu innym osobom w zespole „nie”. Nie zwiększymy wysokości skoku, bo wtedy bohaterka zacznie uderzać głową o sufity. Nie wyostriamo wzroku przeciwników, bo wtedy nie będzie się dało ich ominąć albo będzie trzeba zbudować wszystkie levele od nowa. Nie wyposażymy kuli na podeście w możliwość naciskania „przycisków”, które ją pokrywają, ponieważ zależy nam na tym, by gracz możliwie najdłużej zastanawiał się, do czego ona służy.

Zauważmy, że to oznacza, że jest trochę inaczej, niż nam się wydawało we wcześniejszych podrozdziałach: nie istnieje osobny projekt postaci, przeciwnika, panelu, drzwi i tak dalej, ponieważ wszystko wiąże się ze wszystkim. Praca projektanta systemu wpływa na wszystkie elementy po trochu.



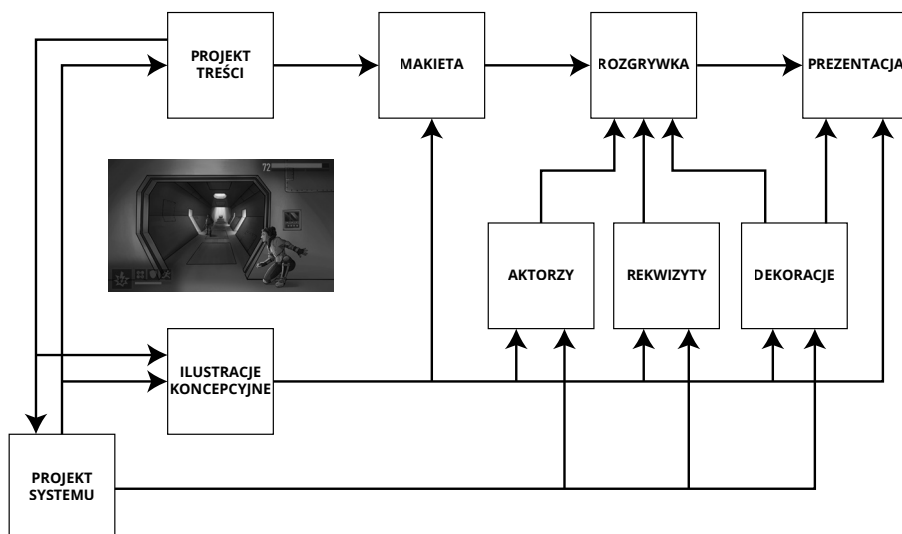
Oczywiście jeżeli spróbujemy myśleć o wszystkich elementach jednocześnie i pracować nad nimi równocześnie, to będzie tego za dużo i wszystko nam się rozpadnie. Nie uda nam się też wykonać tej pracy szybko. W praktyce jak najbardziej projektujemy najpierw jeden element, potem drugi, potem trzeci i tak dalej. Musimy jednak pamiętać, że tworzą one system, i zadbać o to, by każda kolejna rzecz, którą dodajemy, pasowała do już istniejących.



Ilustracje koncepcyjne są częścią projektu gry, ale prawie zawsze zajmuje się nimi inna osoba niż ta, która ustala reguły. To nie przypadek: reguły dotyczą mechaniki i dynamiki, a wygląd to już warstwa prezentacji.

Praca projektanta treści i projektanta systemu stykają się po raz pierwszy właśnie na etapie projektu, czyli zamysłu. W idealnym wypadku rozmowa jest partnerska: projektant systemu proponuje pewien zbiór reguł, a projektant levelu zastanawia się, do czego może je wykorzystać. W tym samym czasie projektant levelu zauważa rzeczy, których z danym zestawem reguł nie da się zrobić, a które mogłyby się przydać. Wtedy projektant systemu często zmienia reguły lub je uzupełnia. Im więcej takich rozmów uda się przeprowadzić, zanim powstaną konkretne dekoracje, rekwizyty, aktorzy i levele, tym lepiej, bo mniej będzie trzeba później poprawiać lub wyrzucać do kosza. Dlatego choć może nas bardzo kusić, by już pierwszego dnia prac nad grą rzucić się w wir produkcji, czyli rysowania, modelowania, programowania i tak dalej, to warto okazać cierpliwość i przeznaczyć nieco czasu na wymyślanie.

Gdy scalimy nasze wcześniejsze wykresy, pokazujące przebieg prac nad levelemi i prac nad systemem rozgrywki, otrzymamy następujący widok:



Po uzgodnieniu spójnego projektu gry projektant systemu przystępuje do prac nad poszczególnymi elementami rozgrywki, a projektant leveli rozpoczyna budowę makiet. Zauważ, że kolejność przygotowania elementów ma znaczenie. Te z nich, które są potrzebne do stworzenia levelu, którym zajmiemy się najpierw, powinny powstać najszybciej. Tu również odnosimy korzyść z wcześniejszych uzgodnień, bo jeśli wiemy, które elementy są potrzebne w którym levelu, to wiemy też, w jakiej kolejności rozpocząć ich budowę.



Levelle powstają zwykle w innej kolejności niż ta, w której gracz je później przechodzi.

Najpierw tworzymy levelle ze środka gry, a dopiero potem te z początku i końca, ponieważ początek i koniec najmocniej zapadają w pamięć i zależy nam, żeby były wykonane jak najlepiej. To oznacza, że najlepiej zająć się nimi pod koniec produkcji, gdy większość elementów gry jest już dopracowana.

Produkcja, preprodukcja i faza koncepcyjna

Wiemy już, że makieta levelu służy do ustalania relacji przestrzennych. Zwykle jednak bardziej interesuje nas czas, bo gracza nie obchodzi, ile metrów przeszedł, tylko ile sekund na to przeznaczyl. Jeśli gra będzie za szybka, to może być trudno za nią nadążyć, a jeśli za wolna, to gracz może się nudzić. Dlatego zależy nam,

by po zbudowaniu makiety levelu po prostu przejść się po niej w takim tempie, w jakim w ostatecznej wersji będzie się poruszać bohaterka.

Problem w tym, że w takim razie musimy najpierw coś wiedzieć o bohaterce. Nie jest nam potrzebne, by wyglądała tak jak w ostatecznej wersji, ale musimy chociaż ustalić, jaka jest jej właściwa prędkość. Wbrew pozorom nie dzielimy tu włosa na czworo, bo gracz spędzi na chodzeniu około 80% czasu. Jeżeli tak podstawowa czynność jak chodzenie będzie uciążliwa, to cała gra się rozsypie. W następnej części książki, „Pięć sekund”, przekonamy się, że właśnie takie drobne zdarzenia jak przejście z jednego punktu do drugiego tworzą różnicę między grą angażującą a nużącą.

Rozważmy inny przykład. Wiemy już, że w „Emigrantach” występują przeciwnicy, którzy zauważają bohaterkę dopiero, gdy podejdzie dostatecznie blisko. Jeżeli korytarz, w którym stoi przeciwnik, jest odpowiednio długi, to bohaterka może w nim stanąć wyprostowana, na samym środku, a mimo to pozostać niezauważona. Podobny efekt osiągniemy, jeżeli korytarz będzie miał zwykłą długość, ale za to zmniejszymy zasięg wzroku przeciwników. Może nam to być potrzebne na przykład wtedy, gdy uznamy, że korytarze w naszej grze są za długie, a gracz spędza zbyt wiele czasu na przemierzaniu ich. Decyzja o tym, by przeciwnicy nie widzieli bohaterki, będzie wtedy z naszej strony świadomym uproszczeniem. Jednak w innej grze, opartej na innych regułach, mogłaby okazać się usterką, na przykład dlatego, że będzie to gra o pojedynkach snajperskich, w której możliwość podejścia bardzo blisko przeciwnika będzie niepożądana.



W wielu grach, w których można się skradać, przeciwnicy są krótkowidzami i zauważają gracza dopiero z odległości około 30 metrów. W przeciwnym razie levely musiałyby być dużo rozleglejsze lub bardziej skomplikowane. Za to w pozornie podobnych grach, w których się tylko walczy, wrogowie mają często sokoli wzrok, właśnie po to, by nie dało się im uciec.

Widzimy na tych przykładach, że nawet podczas budowy makiety musimy już coś wiedzieć o najważniejszych elementach rozgrywki. Podobnych zależności jest bardzo dużo. Jeżeli w grze wszyscy przeciwnicy poruszają się pieszo, to potrzebujemy innych leveli, niż gdyby część z nich prowadziła samochody lub pilotaowała samoloty. Jeżeli w grze można rozmawiać, to będą potrzebne inne otoczenia, niż gdyby można było tylko walczyć. Jeżeli przeciwnicy mogą do nas strzelać, to potrzebujemy otwartych przestrzeni przecinanych przeszkodami terenowymi,

a jeśli mogą nas tylko uderzyć pięścią, to bardziej nam się przydadzą levele, w których łatwo można na kogoś wyskoczyć zza rogu.

Część zależności możemy ustalić z góry, ale bardzo wiele z nich musimy wypróbować. Na przykład projektując buty bohaterki, założymy z góry, że pozwolą jej one na wysokie skoki. Pytanie tylko: jak wysokie? Jeżeli przesadzimy i ustalimy za dużą wysokość skoku, to może się okazać, że każdy skok zabiera zbyt wiele czasu. Gracz będzie się wtedy niecierpliwić, bo następną akcję podejmie dopiero, gdy wyląduje. Jeżeli z powodu tej niedogodności zmniejszymy wysokość skoku, to suity będzie trzeba odpowiednio obniżyć, po to by gracz wciąż doskakiwał we wszystkie właściwe miejsca.

Zanim więc na dobre zabierzemy się za budowę wszystkich leveli, musimy przynajmniej w przybliżeniu ustalić najważniejsze reguły. To oznacza, że musimy je nie tylko wymyślić, ale również sprawdzić, czy zaproponowany przez nas system działa w zgodzie z naszymi zamierzeniami.

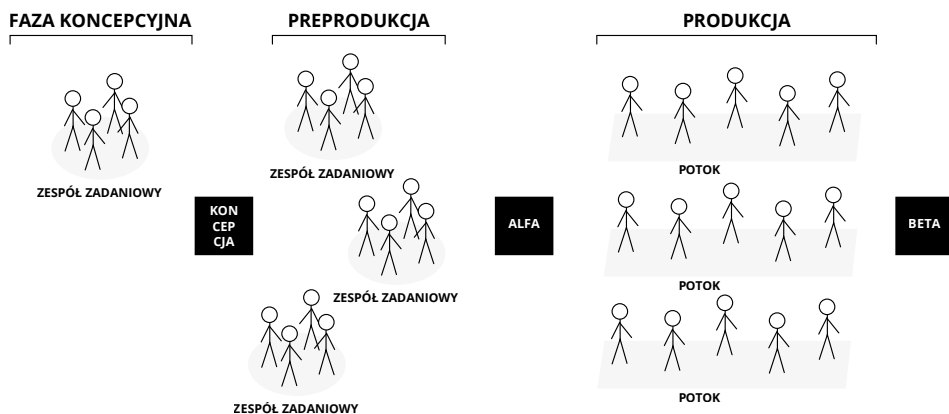
W przypadku bardziej złożonych gier okazuje się, że owe „zamierzenia” nie są wcale oczywiste. Składają się one z tak dużej liczby elementów, że trzeba podejść metodycznie do sprawdzania, czy wszystko do siebie pasuje. Zazwyczaj polega to na tym, że zapisujemy po kolei wszystkie rzeczy, które chcemy umieścić w grze, i sprawdzamy, czy po przeczytaniu takiej listy wszystko nadal brzmi sensownie, czy też nasze pomysły są od Sasa do Lasa. W ten sposób powstaje **koncepcja gry**. O koncepcji powiemy więcej w części czwartej, „Pięć godzin”.

Pod tym względem medium interaktywne nie różni się od tradycyjnego. Na przykład kiedy przystępowałem do pisania tej książki, spisałem najpierw jej konspekt, czyli wymieniałem w punktach, jakie wiadomości znajdują się w każdym z rozdziałów. W ten sposób upewniłem się, że się nie powtarzam. Podobnie produkcję filmu poprzedza zwykle przygotowanie scenariusza, po to by upewnić się, że kolejne sceny tworzą logiczny ciąg.

Czy da się tworzyć grę metodą radosnej improwizacji, bez planowania? Da się, ale trzeba liczyć się z tym, że improwizacja okaże się bez sensu i będzie trzeba ją wyrzucić. W praktyce im bardziej skomplikowana gra i większy zespół autorów, tym większe ryzyko związane z improwizacją i większy pożytek ze sporządzenia koncepcji. Również w małych zespołach często okazuje się, że niektóre elementy są bardziej skomplikowane lub nietypowe od pozostałych i przez to łatwiej się w nich pogubić. W takich wypadkach mały zespół działa podobnie jak duży, choć zwykle zadowala się prostszymi narzędziami. Na przykład prawie każdy zespół

tworzy ilustracje koncepcyjne, ale nie każdy potrzebuje storyboardów i projektów narracyjnych.

W tym rozdziale omawiamy najbardziej ogólny przypadek, dlatego wzorujemy się na dużych projektach, w których uczestniczy po kilkadziesiąt lub kilkaset osób. W takim wypadku budowa gry przechodzi przez trzy główne etapy: fazę koncepcyjną, preprodukcję i produkcję.



W **fazie koncepcyjnej** niewielki zespół uzgadnia między sobą najważniejsze fakty dotyczące gry: o co będzie w niej chodziło, jakie będą jej najważniejsze reguły, jaki będzie jej nastrój, o czym będzie opowiadać fabuła, jak gra będzie wyglądać, jak będzie brzmieć i tak dalej. Osoby z tego zespołu często później kierują całym przedsięwzięciem od strony twórczej. Jeżeli zespół jest mały, to po prostu wszyscy jego członkowie uczestniczą w fazie koncepcyjnej. Jej rezultatem jest dokument koncepcyjny, którego przykładową postać omówimy w rozdziale „Narzędzia projektanta”.



Większość autorów marzy o stworzeniu wielu różnych gier. Dlatego mamy odruch wrzucania do jednej koncepcji wszystkich pomysłów, które nas ekscytują. Tak właśnie zaczynają się przedsięwzięcia, których nigdy nie udaje się dokończyć lub które powstają przez 10 lat tylko po to, by rozczarować. Dlatego Twoja gra powinna mieć motyw przewodni, który mieści się w jednym zdaniu.

W **fazie preprodukcji** konfrontujemy zamierzenia z rzeczywistością, czyli tworzymy pierwsze wersje wszystkich elementów rozgrywki. Następnie – i to jest najważniejsza część naszej pracy – drobiazgowo sprawdzamy, czy działają tak,

jak chcemy. Skupiamy się na wszystkim, co nie działa, budzi wątpliwości lub czego po prostu nie wiemy. Czy dana wysokość skoku bohaterki na pewno jest najlepsza? Czy przeciwnicy powinni poruszać się trochę szybciej? Co się stanie, jeżeli przeciwników będzie dwóch, trzech, pięciu lub dziesięciu? Czy drzwi powinno się obsługiwać klamką, przyciskiem, czy jednak powinny otwierać się i zamykać automatycznie? Czy uwzględnić w grze fakt, że bohaterka, poruszając się, robi hałas? Czy w levelach powinno być raczej jasno, żeby gracz wszystko dobrze widział, czy raczej ciemno, żeby czuł się bardziej osaczony?

Fazę preprodukcji zacznij od zrobienia listy wszystkich rzeczy, o których już wiesz, że znajdują się w grze. Nie musi być kompletna, bo na pewno w trakcie prac coś do niej dopiszesz lub usuniesz. Musi jednak być to tak zwana lista priorytetowa, to znaczy taka, w której na górze znajdują się rzeczy najważniejsze. Uszereguj ją, zaczynając od tych elementów, bez których gra nie może się obejść. Zakończ na tych, które pełnią funkcję ozdobną lub są „miłymi dodatkami”.

W „Emigrantach” na samym szczycie znajdzie się prawdopodobnie sama bohaterka, bo bez niej nie ma w ogóle gry. Na drugim miejscu będzie otoczenie, to znaczy ściany i sufit, bo bez nich nie będzie można ukrywać się przed przeciwnikami. Na trzeciej pozycji umieścimy samych przeciwników. Mniej ważne, ale również potrzebne będą drzwi i panele. Na końcu listy znajdzie się podest z kulą, który nic nie robi.

Nie możemy tworzyć elementów rozgrywki w dowolnej kolejności. Na przykład nie warto zabierać się za tworzenie drzwi, dopóki w grze nie ma nikogo, kto mógłby je otworzyć. Z tego powodu nie mogę podać Ci dokładnego przepisu na budowę gry, bo to od Ciebie zależy, z jakich elementów ona się składa i jakie są pomiędzy nimi zależności. Listę priorytetową wymieniającą rzeczy, które trzeba dodać, czyli tak zwany **backlog**, musisz stworzyć samodzielnie.

W przypadku gdyby okazało się, że nie masz wystarczająco dużo czasu, pomocników lub pieniędzy, żeby zbudować grę złożoną ze wszystkich elementów, które chcesz w niej zamieścić, lista priorytetowa pozwala Ci zrezygnować najpierw z tych, których brak najmniej zaszkodzi grze. Są to po prostu te elementy, które znalazły się na końcu listy.

Niektórzy autorzy ulegają złudzeniu, że wszystko, nad czym zastanawiają się w danej chwili, jest najważniejsze. To im przeszkadza w sporządzeniu listy priorytetowej, bo każdy nowy pomysł umieszczają na samej górze. W takim wypadku najlepiej pozwolić sobie na dzień lub dwa zwłoki, a po upływie tego czasu zastanowić

się, czy dany pomysł wciąż nas ekscytuje tak samo jak na początku. Odpowiedz sobie na pytanie: z których elementów już obecnych na liście możesz zrezygnować, zanim zrezygnujesz z nowego pomysłu? Na przykład: czy w naszej grze warto wprowadzać nową, ciekawą broń, jeżeli w zamian miałoby nie być w niej głównej bohaterki? Oczywiście nie i dlatego bohaterka powinna być wyżej na liście.

Lista rzeczy do dodania może bardzo łatwo zrobić się bardzo długa. Często oznacza to, że musimy powiększyć zespół, bo w przeciwnym razie pracy będzie za dużo. Dlatego zespół w fazie preprodukcji często pracuje w podgrupach, z których jedna zajmuje się bohaterką, druga przeciwnikiem, trzecia drzwiami i tak dalej.

W miarę jak dodajemy do gry kolejne elementy, powstaje jej wersja alfa, czyli pierwsza wersja, w której wszystko działa w przybliżeniu tak, jak ma działać w wersji ostatecznej. Gra może jeszcze zupełnie nie nadawać się do grania, na przykład dlatego, że nie ma w niej niektórych leveli, a te, które są, nie mają wszystkich dekoracji i zachęt. Jednak cały jej system już jest. Brakuje tylko treści.

Zdecydowana większość treści powstaje w **fazie produkcji**. Nadal dodajemy do gry bardzo wiele elementów, głównie w warstwie prezentacji, ale nie musimy już sprawdzać na każdym kroku, czy każdy z nich działa tak, jak powinien, ponieważ tę pewność dała nam wersja alfa. W fazie produkcji zakładamy, że jeśli dany element działa w określony sposób, to widocznie tak miało być, i dostosowujemy do tego dalsze kroki. To dokładnie odwrotnie niż w preprodukcji! Gdyby wtedy przeszkadzała nam wysokość skoku bohaterki, wolno byłoby nam ją zmienić. Podczas produkcji zakładamy, że jest ona dokładnie taka, jak ma być, więc jeśli odkrywamy, że skacząc na taką wysokość, bohaterka uderza głową w sufit, to znaczy, że trzeba podnieść sufit, a nie zmienić wysokość skoku.

To nas pod wieloma względami ogranicza – ale to dobrze, bo to znaczy, że w razie problemów mamy mniej opcji do wypróbowania, czyli szybciej znajdziemy rozwiązanie. Zamiast żonglować wszystkimi elementami gry jednocześnie, udało nam się zająć nimi po kolei: najpierw system, czyli zachowanie bohaterki, przeciwnika, drzwi, panelu i tak dalej, a potem treść, czyli levele, które montujemy z gotowych elementów. Dzięki temu, że na większość pytań i wątpliwości udało nam się odpowiedzieć już w fazach koncepcyjnej i preprodukcji, w fazie produkcji możemy pracować „w ciemno”: po prostu bierzemy to, co mamy do dyspozycji i nie kombinujemy, co by tu jeszcze dodać lub zmienić.

Oszczędzamy na tym mnóstwo czasu, przede wszystkim dlatego, że nie musimy już wszystkiego ze wszystkimi uzgadniać. Gdyby na tym etapie zmieniała

się wysokość skoku bohaterki, to co prawda w naszym levelu pozwoliłoby nam to uniknąć podnoszenia sufitów, ale we wszystkich innych, już zbudowanych, mogłoby się okazać, że bohaterka nie sięga w miejsca, w które powinna móc się dostać. Decyzję tego typu należałoby uzgodnić ze wszystkimi projektantami treści, a potem jeszcze wdrożyć, czyli zbudować wszystkie levele od nowa, czyli zmarnować bardzo dużo pracy. Dlatego narzucamy sobie dyscyplinę: czas na ustalenie wysokości skoku bohaterki był fazie preprodukcji, a po jej zakończeniu kłamka zapadła.

Budowa treści gry wymaga zwykle znacznie więcej pracy niż przygotowanie jej systemu, dlatego zespół w fazie produkcji jest największy. Najczęściej właśnie na początku produkcji popełniane są błędy w zarządzaniu polegające na złym oszacowaniu tego, ile właściwie jest do zrobienia. Dlatego opłaca nam się podejść do budowy gry metodycznie. Jeżeli faza koncepcyjna i preprodukcja nie zostały zaniedbane, to wiemy dość dokładnie, ilu potrzebujemy aktorów, rekwizytów, dekoracji i leveli. Wiemy też, z czego najłatwiej będzie nam zrezygnować, w razie gdyby zabrakło nam czasu.

Wynikiem produkcji jest wersja beta, czyli taka, w której jest już cały system i cała treść. Gra może jeszcze mieć usterki, ale w praktyce jest już w przybliżeniu gotowa. Gratulacje dla autorów!

Opisany powyżej model jest wyidealizowany. W praktyce bardzo często prace nad różnymi częściami gry biegną w różnym tempie, a gdy jeden element jest jeszcze w fazie koncepcyjnej, inny może już być wyprodukowany. Najważniejsze dla nas jest to, żeby żadnej fazy nie pominąć. Powszechnym błędem jest pomijanie preprodukcji elementów, których sposób działania wydał się autorom „oczywisty”, podczas gdy w rzeczywistości między wersją „oczywistą” a tą, której gra faktycznie potrzebowała, były drobne, ale istotne różnice.

Na przykład w pewnej grze, przy której pracowałem, zespół stworzył robota, który wydawał się bardzo groźny i silny, a mimo to walka z nim była nudna. Dopiero po bliższym przyjrzeniu się odkryliśmy, że za każdym razem, gdy robot zostanie trafiony, zatrzymuje się. Nie było to zamierzone zachowanie, a jedynie coś, co wydało się programiście ciekawe, dlatego dodał taką funkcję na własną rękę. Jednak w praktyce oznaczała ona, że robot był zupełnie niegroźny, bo broń gracza była szybkostrzelna i wystarczyło raz wycelować, a potem strzelać do skutku do unieruchomionego celu. Żeby to naprawić, wyposażyliśmy robota w możliwość szarżowania pomimo uderzeń pocisków. Na szczęście nie musieliśmy w tym celu

zmieniać jego modelu ani animacji, a jedynie usunąć niepotrzebne zachowanie. Upiekło nam się, bo jeszcze nie zdążyliśmy wybudować żadnych leveli, które opierałyby się na tym, że robot działa właśnie tak, a nie inaczej!

Praca iteracyjna

Prawie każdą wystarczająco skomplikowaną czynność wykonuje się na raty. Na przykład tekst tej książki, nim trafił w Twoje ręce, został wielokrotnie wygładzony i uściślony, a następnie dwukrotnie trafił do korektora, który naniósł własne poprawki. Niektóre fragmenty pisałem po kilka razy, szukając właściwego tonu i układu treści. Wykonałem kilka **iteracji**, czyli powtórzeń pewnej czynności opartych na pewnej dodatkowej informacji. W tym wypadku dodatkową informacją było to, że po każdym powtórzeniu czytałem tekst i coś mi się w nim nie podoobało. Nowa wersja tekstu nie powstawała niezależnie od poprzedniej, tylko na podstawie tego wrażenia. Dlatego po każdym kolejnym powtórzeniu uzyskiwałem nieco lepszy tekst książki niż w poprzednim.

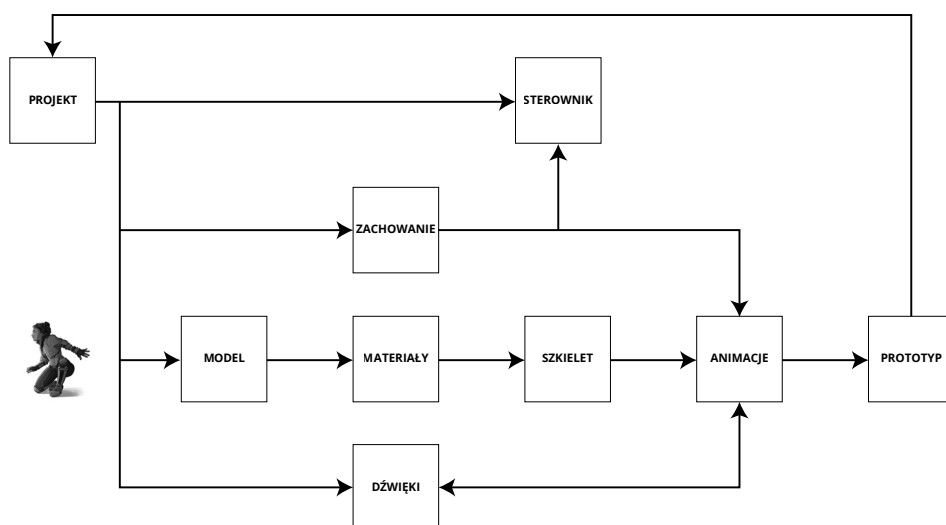
Bardzo często – właściwie prawie zawsze – po prostu nie wiemy, jaki jest najlepszy możliwy projekt danego elementu. Musimy wykonać próbę „w ciemno”, dzięki której zdobędziemy dodatkowe informacje. Dlatego tworzymy dany element na raty, w kolejnych iteracjach, czyli pracujemy iteracyjnie.

Praca iteracyjna działa dobrze, gdy spełnia dwa warunki:

- Każda iteracja opiera się na poprzedniej: dodajemy rzeczy, dopracowujemy je, uzupełniamy szczegóły, poprawiamy usterki, ale nie zastępujemy tego, co już mamy, czymś zupełnie innym, chyba że poprzednia wersja okazała się ślepym zaułkiem. Taki sposób pracy nazywamy metodą kolejnych przybliżeń.
- Każda iteracja daje nam wiedzę o tym, jak powinna wyglądać następna iteracja. Nie próbujemy różnych rzeczy na chybił-trafił, tylko szukamy tropu wiodącego do najlepszego rozwiązania.

W idealnym wypadku tylko w pierwszej iteracji działamy zupełnie w ciemno. Zdarza się wtedy, że proponujemy kilka równorzędnych i bardzo odmiennych wariantów, na przykład buty do skakania i plecak odrzutowy, który też pozwala poruszać się w pionie. Po pierwszej iteracji warto jak najszybciej wybrać konkretny wariant i na nim się skupić.

Żeby spełnić wymienione przed chwilą warunki, musimy narzucić sobie pewną dyscyplinę. Przede wszystkim: unikamy zbędnej pracy. Zakładamy, że wersja, którą tworzymy, jest tymczasowa i staramy się wykonać ją jak najszybciej i najtaniej. W tym celu pomijamy wszystkie jej elementy, które w danej iteracji nie są potrzebne. Tymczasową, roboczą wersję danego elementu rozgrywki, umieszczoną w próbnym levelu, nazywamy **prototypem**. Proces poddawania wersji roboczych kolejnym próbom, które pomagają nam w odpowiedzi na pytanie, czy dany element działa tak, jak powinien, nazywamy prototypowaniem. Omówimy je szerzej w rozdziale „Narzędzia projektanta”.

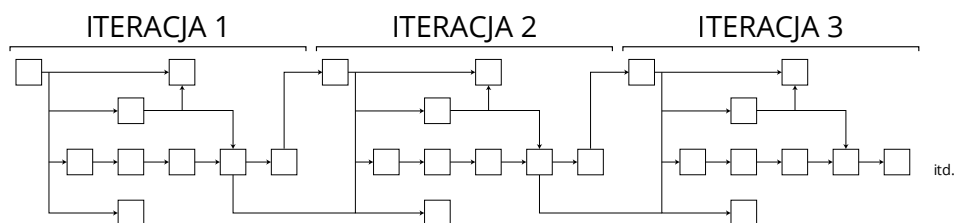


Prototypowanie jest oczywiście iteracyjne, dlatego że najczęściej na skutek wykonanego eksperymentu znajdujemy jakiś problem, który trzeba naprawić. W takim wypadku tworzymy nową wersję tymczasową i ponawiamy eksperyment. Wynik eksperymentu dostarcza nam informacji zwrotnej, która wpływa na projekt danego elementu, na przykład w ten sposób, że dochodzimy do wniosku, że warto trochę zmienić jego działanie. Zmiana w projekcie wpływa na jedną lub więcej części składowych: model, materiał, zachowanie, dźwięki i tak dalej.

Po uzupełnieniu wykresu zależności o prototyp widzimy, że pracujemy w pętli. Prototyp dostarcza informacji zwrotnej, która pośrednio wpływa na wszystkie elementy. Na przykład decyzja o zmianie maksymalnej wysokości skoku może pociągnąć za sobą zarówno przeróbki w zachowaniu bohaterki, bo to w nim opisana jest czynność o nazwie „skakanie”, jak i w jej animacji, bo to ona decyduje o tym,

jak skakanie wygląda. Po naniesieniu tych zmian znowu prototypujemy i być może otrzymujemy nową informację zwrotną, która pozwala wykonać kolejną iterację.

Powyższy wykres pokazuje nam, co od czego zależy, ale ukrywa przed nami bolesną prawdę o tym, ile pracy wiąże się z budową gier i jak bardzo musimy oszczędzać czas. Żeby się z nią zderzyć, wprowadźmy przeróbkę, polegającą na tym, że usuniemy pętle, a jeżeli dane zadanie trzeba wykonać drugi raz, to narysujemy je ponownie. Nasz wykres zmieni się przez to nie do poznania:



Zadań przybywa w bardzo szybkim tempie, bo w najgorszym możliwym wypadku każda nowa iteracja oznacza konieczność zbudowania całego elementu od nowa. Zważywszy, że iteracji bardzo łatwo może być nie trzy, lecz trzydzieści, musimy koniecznie zadać sobie pytanie, w jaki sposób zmniejszyć ilość pracy o około 99%. W przeciwnym razie nasza gra nie powstanie za naszego życia.

***Staraj się budować, opierając się na tym, co już masz.
Zaczynanie wszystkiego za każdym razem od początku
jest bardzo kosztowne.***

Skracanie iteracji

Sposobów na oszczędności podczas pracy iteracyjnej jest kilka.

Pierwszy sposób polega na tym, że stosujemy *placeholders*, czyli **zamienniki**. Wspominałem już kilka razy, że nasza bohaterka nie musi od początku wyglądać tak jak w wersji ostatecznej. Zamiast tego użyjemy zastępczego modelu, być może niepodobnego nawet do postaci ludzkiej, bo jest nam on potrzebny tylko po to, żeby się coś ruszało na ekranie. Kiedy przejdziemy do prac nad kolejną postacią, na przykład nad przeciwnikiem, jako zamiennika możemy użyć.. modelu bohaterki! Przeciwnicy przez pewien czas będą wyglądali tak samo jak bohaterka,

ale nam, jako autorom, nie przeszkodzi to w prototypowaniu. W podobny sposób stosujemy zastępcze materiały, dźwięki, animacje i każdy inny element, który jest nam akurat potrzebny, ale jeszcze nie chcemy zużywać na niego czasu. Zamienimy go na wersję docelową dopiero wtedy, gdy będzie nam to potrzebne do kolejnego prototypu. Dzięki temu każda iteracja będzie znacznie szybsza, ponieważ większość pracy po prostu ominiemy.

Drugi sposób polega na tym, że eksperymenty prowadzimy w miarę możliwości w taki sposób, by **zacząć od najbardziej ryzykownych**. W ten sposób ograniczamy liczbę iteracji, bo jeśli odkryjemy, że nasz ryzykowny element rozgrywki zupełnie się nie sprawdza i nie warto go poprawiać (to bardzo częsta sytuacja!), to wyrzucimy go, nie tworząc w ogóle większości jego składników.

Po to właśnie powstają ilustracje koncepcyjne. Najbardziej ryzykowną rzeczą w powstawaniu modelu postaci lub rekwizytu jest jego wygląd, a nie techniczne szczegóły jego budowy. Korzystamy więc z faktu, że do określenia wyglądu nie jest nam potrzebny sam model, a wystarczy rysunek. Szybki szkic postaci zajmie wprawnemu grafikowi 2D dwie godziny. Bardziej dopracowane studium – kilka dni. Za to praca nad modelem i materiałami może zająć nawet trzy miesiące, jeżeli wygląd postaci jest skomplikowany. Skoro zależy nam na czasie, to opłaca nam się na przykład wykonać 10 szybkich szkiców, dwa z nich wytypować jako podstawę pod dopracowane studium, a z tej dwójki wybrać do realizacji jeden model i zestaw materiałów. Zajmie nam to łącznie około 75 dni, podczas gdy stworzenie 10 pełnych modeli wraz z materiałami zajęłoby około 650 dni.

Najwięcej ryzyka wiąże się zwykle z dodawaniem do rozgrywki nowych zachowań. We wcześniejszym podrozdziale wspominałem, że gdy wprowadzamy do gry akcję, to może się okazać, że jest ona niepotrzebna, albo odwrotnie, czyni inną akcję niepotrzebną. Podobnie kiedy wyposażamy przeciwników w nową umiejętność, mogą oni okazać się nie do pokonania, albo przeciwnie – nie umieją z danej umiejętności korzystać. Z drugiej strony, jeżeli wprowadzisz do gry gadżet, który robi coś ciekawego, i odkryjesz, że w rezultacie w rozgrywce pojawiły się nowe, ciekawe sytuacje, to już na tym etapie dowiesz się, że dany gadżet się sprawdza, na długo zanim zacznie ładnie wyglądać i wydawać satysfakcjonujące odgłosy. Dlatego najczęściej staramy się najpierw wypróbować dany element w przykładowych sytuacjach, a dopiero potem zadbać o jego prezentację.

Trzecim sposobem na oszczędność pracy jest **parametryzacja**. Założmy, że w „Emigrantach” można przeciwnika przestraszyć. Niektórzy przeciwnicy są

tchórzliwi i płoszą się byle czym, a inni twardo robią swoje, nawet gdy ich koledzy dawno zwiali. W pierwszym odruchu możemy przygotować dwa różne zachowania: „chojrak” i „tchórz”. Prawdopodobnie szybko odkryjemy jakieś niedociągnięcie, na przykład takie, że rozgrywka nie jest tak ciekawa, jak nam się wydawało, ale możemy ją poprawić, jeżeli „chojrak” będzie trochę ostrożniejszy, a „tchórz” trochę śmielszy. Jeżeli taka poprawka będzie od nas wymagała stworzenia obu zachowań od nowa, to będziemy z tym mieć sporo pracy.

Zamiast tego stworzymy uogólnione zachowanie: „przeciwnik, który czasem może się przestraszyć”. Będzie ono działało w ten sposób, że od czasu do czasu sprawdzi, jaki jest dla danego przeciwnika parametr „odwaga”. Jeżeli będzie wysoki, przeciwnik będzie się bał rzadko. Jeżeli niższy – to częściej. Wszystko, czego w tym wypadku potrzeba, by zmienić „tchórza” w „chojraka”, to podać nową wartość parametru. Podobny zabieg da się zastosować do prawie każdego elementu rozgrywki.



W czasach gdy większość gier była złożona ze zwykłych, płaskich rysunków, a nie z modeli, chętnie oszczędzano wysiłek, sięgając po warianty kolorystyczne. Dwa lub więcej elementów różniło się od siebie tylko kolorem, na przykład jeden domek miał czerwone dachówki, drugi niebieskie, a trzeci zielone. Pozwalało to stworzyć wrażenie różnorodności małym kosztem, ponieważ podmiana jednego koloru na inny wymagała tylko zmiany parametru, a nie rysowania całego domku od nowa.

Czwarty sposób oszczędzania czasu to inwestowanie w narzędzia pracy, tak aby **prawie każdą zmianę dało się wprowadzić natychmiast**. Na przykład jeżeli parametr, taki jak „odwaga”, jest wpisany na stałe do programu komputerowego, to projektant, który chce dostroić zachowanie przeciwnika, musi podejść do programisty, przeszkodzić mu w pracy, poprosić o zmianę parametru, a wtedy programista musi przepisać odpowiedni fragment programu, przygotować nową wersję gry, którą projektant musi od niego pobrać, uruchomić, sprawdzić, czy już jest dobrze, odkryć, że wciąż nie do końca jest tak, jak powinno, podejść do programisty ponownie.. to nie jest zmiana natychmiastowa.

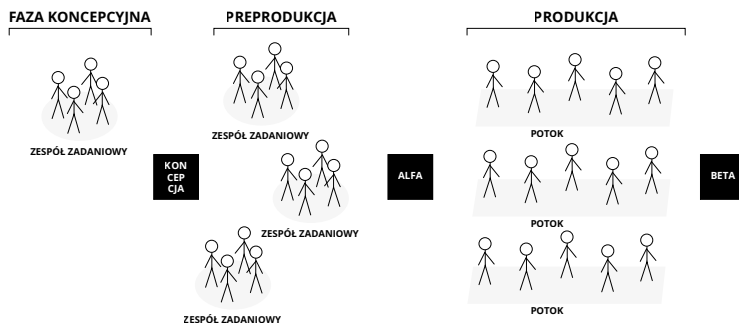
Ale co będzie, jeśli zamiast tego w edytorze leveli zrobimy specjalne okienko, w którym można podejrzeć przeciwnika i wszystkie jego parametry, a następnie po prostu przesunąć odpowiedni suwak? Dla programisty to trochę dodatkowej pracy, bo trzeba rozbudować edytor, ale za to projektant nie musi już zwracać mu głowy, a każdą zmianę wykonuje i sprawdza w minutę.

Piątym sposobem na oszczędności jest **powierzanie prototypów projektantom**. Zarówno potoczne wyobrażenie o pracy projektanta, jak i, przyznam uczciwie, wszystko, co napisałem do tej pory, tworzy niebezpieczne złudzenie. Mianowicie może się wydawać, że projektant to ktoś, kto wykonuje dużo bardzo twórczego myślenia, które notuje w pisemnym projekcie, a następnie nadzoruje, jak reszta zespołu wciela jego piękny zamysł w czyn. To nieprawda właśnie dlatego, że ten zamysł wcale nie jest taki piękny, lecz przeciwnie, kryje się w nim wiele niedociągnięć. Jednocześnie projektant jest tą osobą w zespole, która ma najlepsze pojęcie o tym, jak dany element powinien działać, czyli najlepiej wie, jak naprawić swoje własne błędy i jak ich szukać. Dlatego to projektant powinien wykonać większość prototypów i poddać próbie swoje własne pomysły.

Sprawny projektant systemu nie boi się „grzebać w bebecach”, włącznie z tym, że nie tylko buduje próbne levele, ale często także sam programuje lub tworzy własne zamienniki. Pracuje wtedy znacznie szybciej, niż gdyby miał co chwilę prosić kogoś o pomoc.

Zespół zadaniowy i potok

Jeżeli tylko budujesz grę w gronie większym niż tylko Ty sam, stajesz przed pytaniem, jak podzielić się zadaniami i jak wymieniać się informacjami. Większość zespołów wypracowuje z czasem własny styl współpracy, jednak prawie zawsze wiąże się on z wykorzystaniem dwóch podstawowych form organizacji. Przypomnijmy ilustrację z wcześniejszego podrozdziału:



Znalazły się na niej dwa terminy, które teraz wyjaśnimy: zespół zadaniowy i potok.

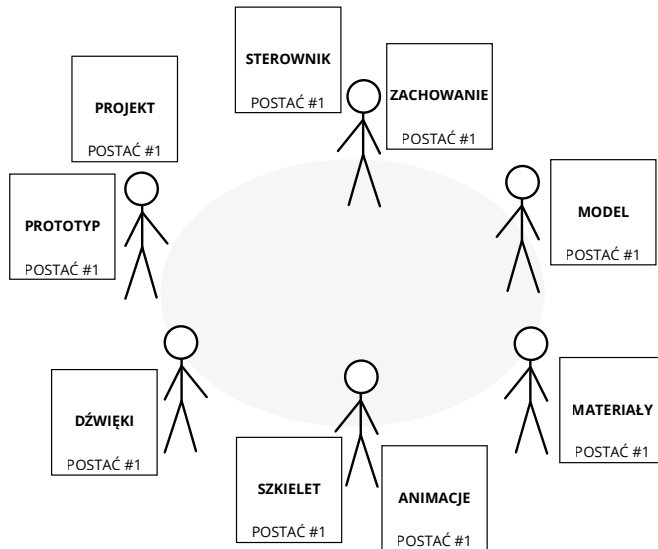
Zespołem zadaniowym nazywamy grupę autorów, którzy pracują nad danym elementem gry **równocześnie**, konsultując między sobą wszystkie podejmowane czynności lub znaczną ich część. Na przykład w zespole zadaniowym pracującym nad nową postacią animator będzie cały czas w kontakcie z grafikiem 3D, po to by uzgodnić szczegóły anatomiczne modelu, oraz z dźwiękowcem, po to by zgrać brzmienie animacji z jej wyglądem. Projektant będzie dużo rozmawiał z grafikiem 2D o temperamencie i przyzwyczajeniach postaci, po to by uchwycić je w jej wyglądzie. Z programistą wymieni dużo uwag o jej zachowaniach. Programista z kolei może mieć prośby do wszystkich po trochu, ponieważ często zdarza się, że dwa bardzo podobne pomysły bardzo różnią się trudnością zaprogramowania, a drobna zmiana projektu pozwala zaoszczędzić bardzo dużo czasu.

Zespół zadaniowy dużo ze sobą rozmawia, a w jego pracy najważniejsze jest, żeby rozmowy nie zajmowały zbyt wiele czasu w porównaniu z budową samej gry. Dlatego taki zespół nie może być za duży. Przyjmuje się na ogół, że nie powinien przekraczać ośmiu osób.

Kiedy budujemy małą grę, wszystkich jej autorów można zorganizować w jeden zespół zadaniowy. To bardzo wygodne. Przy bardziej skomplikowanych przedsięwzięciach zespołów musi być więcej, a wtedy powierza im się konkretne zadania. Ważne jest wtedy, żeby zadanie było zamknięte, po to by zespół nie musiał zbyt często prowadzić uzgodnień z kimś, kto do niego nie należy. Dobrym przykładem zadania spełniającego ten warunek jest opracowanie konkretnego elementu rozgrywki, reguły lub levelu. Zespół zadaniowy tworzy również wstępną koncepcję gry, ponieważ co prawda obejmuje ona grę jako całość, ale za to nie jest bardzo szczegółowa.

Zespoły zadaniowe sprawdzają się najlepiej wtedy, gdy nie wiemy, jaki jest pożądaný rezultat naszych działań, czyli w fazach koncepcyjnej i preprodukcji. Gdy dużo eksperymentujemy, w naszym pomysle na dany element rozgrywki często coś się zmienia i trzeba szybko się zorientować, czy daną zmianą czegoś nie psujemy. Na przykład jeśli projektant uzna, że postać powinna poruszać się o połowę szybciej, musi to uzgodnić z animatorem, który dopasuje szybkość ruchu do tempa przebiegania nogami. Jeżeli takie tempo miałyby sprawić, że postać zacznie wyglądać niepoważnie lub nienaturalnie, to dużo szybciej będzie o tym porozmawiać, niż przekonać się dopiero, gdy animacja powstanie. Zespół zadaniowy oszczędza najwięcej czasu, gdy pracuje nad rzeczami skomplikowanymi, które mają wiele wzajemnie zależnych składowych.

Organizując zespół zadaniowy, pamiętaj o tym, że każdy jego członek ma prawo głosu w każdej sprawie, ale to nie znaczy, że ma w niej ostatnie słowo. Kompetencje zawsze powinny być jasno przypisane: ta osoba odpowiada za rozgrywkę, ta za wygląd, ta za wydajność, a ta za dźwięki. Wymiany uwag są bardzo cenne i gwarantuję Ci, że Twój współpracownik podsunie Ci wiele dobrych pomysłów, które Tobie nie przyszyłyby do głowy. Jednocześnie rozmowy bardzo łatwo wymykają się spod kontroli i można ugrzęznąć w niekończących się uzgodnieniach, zwłaszcza gdy dwóm lub więcej osobom w zespole bardzo zależy na rzeczach, które się wzajemnie wykluczają. Ktoś musi wtedy rozstrzygnąć spór, czyli mieć ostatnie słowo. Nie musi to być za każdym razem ta sama osoba, inaczej mówiąc, zespół zadaniowy nie musi mieć szefa, ale zawsze musi być wiadomo, kto ma ostatnie słowo w danej sprawie.



Potok, dla odmiany, w ogóle nie jest zespołem. Osoby współpracujące w potoku prawie nie muszą ze sobą rozmawiać, ponieważ przekazują sobie zadania na podobieństwo taśmy produkcyjnej: pierwsza osoba przygotowuje pierwszą składową, następna coś do niej dodaje, podobnie następna i tak dalej. Inaczej mówiąc, opracowują one dany element rozgrywki **po kolei**. Każda z tych osób może należeć do innego zespołu i bardzo często tak właśnie się dzieje, ponieważ w pracy potokowej grupuje się ludzi według specjalności, a nie według zadań. Graficy 2D są wtedy w jednym zespole, graficy 3D w drugim, animatorzy w trzecim i tak dalej.

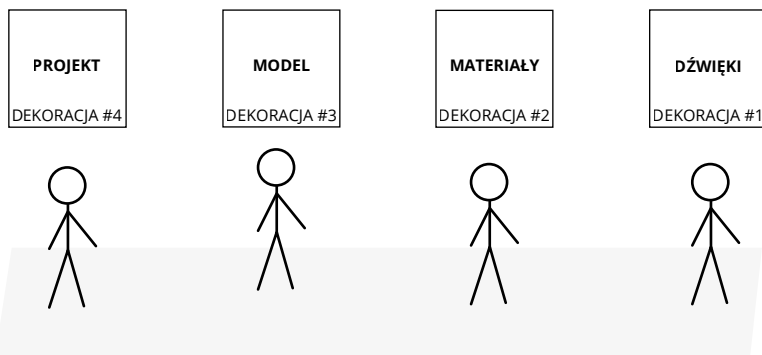


Im większa łączna liczba osób pracujących nad daną grą, tym większa specjalizacja poszczególnych zespołów. Na przykład w gronie liczącym kilkadziesiąt osób graficy 3D specjalizujący się w postaciach mogą należeć do innego zespołu niż graficy 3D specjalizujący się w maszynach.

Potok działa najlepiej wtedy, gdy nie ma już wielu niewiadomych lub gdy części składowe danego elementu nie są od siebie wzajemnie zależne. Dobrym przykładem jest budowa levelu: kiedy projektant treści ustali już przebieg rozgrywki, przekazuje go dekoratorom, po to by dany obszar upiększyli. Dekorator w takim wypadku przyjmuje zastaną rozgrywkę jako pewnik, którego nie wolno mu podważać, ale za to sam nie tłumaczy się z tego, jak opracował wygląd levelu. W trybie potokowym łatwo przygotowuje się również same dekoracje, ponieważ nie są one interaktywne, czyli nie wpływają na inne elementy, więc nie trzeba w nich dużo sprawdzać ani uzgadniać.

Pewna liczba uzgodnień jest wciąż konieczna ze względu na to, że tak skomplikowanej pracy nieuchronnie towarzyszą błędy, usterki i nieporozumienia. Jednak w porównaniu z pracą zespołu zadaniowego członkowie potoku prawie się nie spotykają. Często nie pracują w tym samym pomieszczeniu ani nawet w tym samym budynku.

Potok jest szybszy od zespołu zadaniowego, ponieważ w całości skupia się na budowie, z pominięciem uzgodnień. Dużo łatwiej jest też zorganizować go tak, by nikt na nikogo nie czekał. Właśnie w tym celu autorów grupuje się według specjalności. Kiedy postać budowana w trybie potokowym opuszcza zespół grafików 3D i trafia do zespołu animatorów, można przypisać ją temu animatorowi, który akurat ma najmniej do zrobienia. W zespole zadaniowym, który musi być możliwie najmniej liczny, prawdopodobnie byłby tylko jeden animator, więc w przypadku, gdy byłby akurat zajęty, nowe zadanie trafiłoby do kolejki.





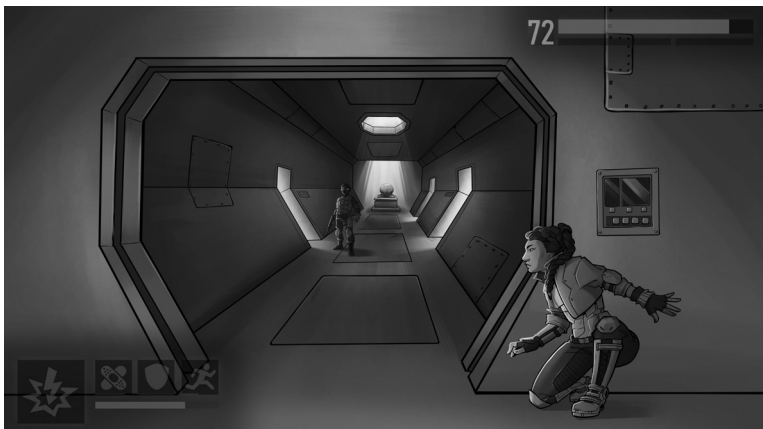
Kiedy pojawia się nowe zadanie do wykonania, współpracownicy mogą naciskać na Ciebie, by zostało ono wykonane najpierw, nawet kosztem tego, co robisz w tej chwili. To droga do szaleństwa, bo kiedy przyjdzie jeszcze jedno zadanie i też otrzyma priorytet, będziesz mieć na głowie już dwie niedokończone rzeczy. Jeżeli nie masz bardzo dobrego powodu, by postąpić inaczej, wykonuj zadania w takiej kolejności, w jakiej przychodzą. Dzięki temu wszystko zostanie zrobione i żadna rzecz nie będzie czekać bardzo długo.

Potok nie umożliwia uzgodnień, więc nie sprawdza się w preprodukcji, czyli tam, gdzie trzeba dużo eksperymentować i dopasowywać do siebie poszczególne elementy. Jednocześnie, ponieważ potok pozwala pracować szybko, bardzo przydaje się w produkcji, gdy prawie wszystko zostało już uzgodnione, ale zdecydowana większość pracy wciąż na nas czeka.

Zauważ, że zarówno w preprodukcji, jak i w produkcji organizujemy pracę tak, by zaoszczędzić czas. Wybieramy w nich różne formy organizacji, dlatego że wykonujemy różne czynności. **Nie ma jedynego słusznego rozwiązania organizacyjnego.** Zawsze bierz pod uwagę charakter danego zadania. Czy jest ono proste, ale czasochłonne, czy szybkie, ale skomplikowane? Czy wymaga współpracy osób o podobnych specjalnościach, czy różnych? Czy w razie pomyłki będzie trzeba robić wszystko od nowa, czy wystarczy drobna korekta?

Szczegółowy przypadek: interfejs

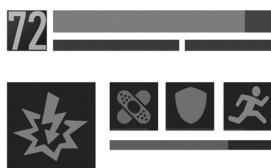
Spójrzmy ostatni raz na przykładowy widok z „Emigrantów”. Jest na nim kilka elementów, których jeszcze nie omówiliśmy.



W prawym górnym rogu ekranu widzimy wskaźnik zdrowia bohaterki i wskaźnik energii zasilającej jej gadżety. W lewym dolnym rogu widzimy zbiór ikon przedstawiających informacje związane z ekwipunkiem.

Żaden z tych elementów nie znajduje się w świecie gry, lecz jest nałożony na jego widok. Bohaterka nie może ich dotknąć, a jeśli się nad tym chwilę zastanowimy, to zdamy sobie sprawę, że nawet nie wie o ich istnieniu. Wszystkie te elementy istnieją tylko po to, by sam gracz czerpał z nich informacje o stanie rozgrywki.

Inaczej mówiąc, są częścią **interfejsu użytkownika**.



Interfejs ma dwie strony. Jedną z nich widzimy na ekranie: to właśnie wskaźniki przekazujące graczowi pewną informację zwrotną. Drugiej strony w tej książce nie pokażemy, ale potrafimy ją opisać, mianowicie tworzą ją urządzenia wejściowe, z których gracz korzysta, by wprowadzić do gry swoje akcje. Jako autorzy gry zazwyczaj nie musimy przejmować się samymi urządzeniami, bo zakładamy po prostu, że gracz już nimi dysponuje. Jednak bardzo dużo uwagi poświęcamy zaprogramowaniu ich w odpowiedni sposób, na przykład zależy nam, by dany klawisz lub przycisk pada odpowiadał za skoki, a inny – za chodzenie.



Często pozwalamy graczowi samodzielnie określić znaczenia przycisków.

To bardzo poprawia wygodę korzystania z gry, na przykład umożliwia dostosowanie jej do potrzeb osób z ograniczeniami ruchowymi.

Z formalnego punktu widzenia elementy interfejsu są rekwizytami. Przejawiają pewne zachowania, takie jak automatyczne uaktualnianie odczytów, które pozwala graczowi na bieżąco oceniać, ile energii zostało mu do dyspozycji i czy bohaterka doznała obrażeń. Elementy interfejsu składają się też z tych samych części co inne rekwizyty. Na przykład w „Emigrantach” ikony i wskaźniki widoczne na ekranie są dwuwymiarowymi rysunkami uzupełnionymi o odtwarzane we właściwych sytuacjach, ostrzegawcze dźwięki.

Mimo to nie traktujemy ich jak zwykłych rekwizytów, ponieważ:

- są w większości unikatowe, to znaczy pojawiają się w grze tylko w jednym, wybranym miejscu,
- są elementami użytkowymi,
- nie oddziałują na inne elementy rozgrywki, tylko bezpośrednio na gracza,
- są swego rodzaju filtrem, który narzuca graczowi określone rozumienie całej rozgrywki.

Zastanówmy się chwilę nad ostatnim punktem. Na czym polega to filtrowanie? Na przykład na tym, że autor interfejsu wybiera, które informacje są dla gracza łatwo dostępne, a które nie. W „Emigrantach” gracz cały czas ma do dyspozycji informację o tym, ile zdrowia pozostało głównej bohaterce, ale nie ma pojęcia, jak się czują przeciwnicy. Mogłoby być odwrotnie i łatwo sobie wyobrazić, że rozgrywka miałaby wtedy zupełnie inny nastrój.

Gra, która na bieżąco informuje nas o ubytkach zdrowia bohaterki, każe nam się o nią martwić. Gra, która zamiast tego alarmuje o utracie zdrowia przeciwników, robi nam niemalże wyrzuty sumienia za to, że ich ranimy. Więcej na ten temat opowiemy w części drugiej, „5 sekund”.

W odróżnieniu od innych rekwizytów interfejs musi być jak najszybciej dobrze wykonany i działać w sposób możliwie zbliżony do docelowego. Jeżeli ma usterki, to nie możemy grać w grę zgodnie z intencją autorów, czyli nie możemy sprawdzić, czy gra działa tak, jak powinna, co uniemożliwia nam dalszą pracę.

Ponieważ interfejs pełni w grze tak ważną funkcję, to wkłada się w jego dopracowanie dużo czasu i wysiłku. Bardzo często zespół, któremu przydziela się zbudowanie interfejsu, pracuje tylko nad nim przez kilka-kilkanaście miesięcy. Oczywiście jest to zespół zadaniowy.

Kontrola postępów

Największym wrogiem autorów gier, zwłaszcza pracujących „po godzinach”, jest zniechęcenie. Nawet stosunkowo mała gra ma dużo elementów, a jej budowa zajmuje od kilku miesięcy do kilku lat. Bardzo łatwo stracić orientację w tym, jaką następną rzecz trzeba zrobić. Można też odnieść wrażenie, że przedsięwzięcie nigdy się nie skończy. Gdy testerzy lub gracze zaczynają hurtowo zgłaszać usterki, ciężar pilnych zadań do wykonania może przygnieść.

Dlatego autorzy pomagają sobie narzędziami i dobrymi praktykami, które:

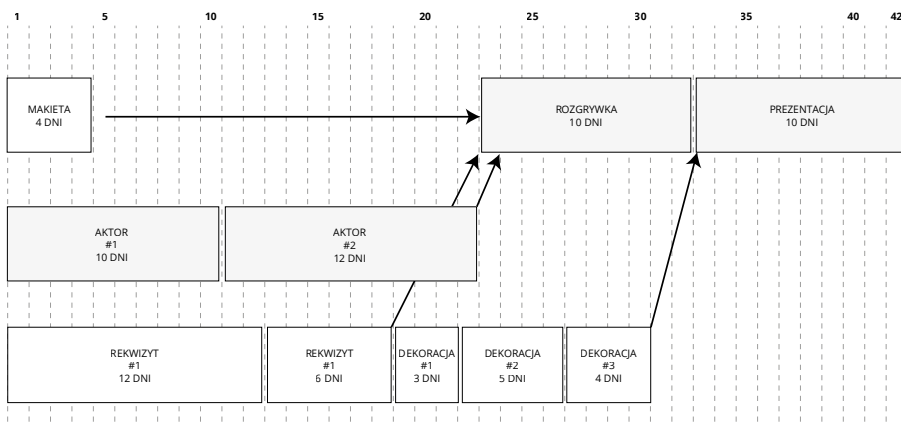
- odciążają ich pamięć,
- dzielą dużą pracę na małe części,
- umożliwiają szczelny obieg informacji, czyli taki, w którym nic się nie gubi,
- odciążają od myślenia o szczegółach organizacyjnych.

Podstawowym narzędziem do uszczelniania obiegu informacji jest lista. Jeśli nie chcesz zapomnieć o żadnym zadaniu, które masz do wykonania, to je po prostu zapisz. We wcześniejszym podrozdziale wspomniałem już o **liście priorytetowej**, czyli takiej, na której rzeczy ważniejsze są wymienione przed mniej ważnymi. Taką listę zawsze warto stworzyć i na bieżąco aktualizować w fazie preprodukcji.

Innym rodzajem listy jest **harmonogram**. W nim również wymieniamy rzeczy do zrobienia, ale tym razem w takiej kolejności, w jakiej będziemy je wykonywać. Harmonogram przydaje się najbardziej w fazie produkcji, ponieważ powinno się dodawać do niego tylko rzeczy, co do których mamy całkowitą pewność, że je zrobimy. Takiej pewności podczas preprodukcji nie mamy: dany element być może będzie trzeba zrobić jeszcze raz albo też poprawić w nieznanej z góry liczbie iteracji. Na skutek przeprowadzonych prób część elementów może okazać się nieprzydatna i wtedy je wyrzucimy, a na ich miejsce być może wymyślimy nowe. Może też zwyczajnie zabraknąć nam czasu. W takim wypadku harmonogram mówi nam: „no trudno, nie udało się zbudować gry w terminie”, co oznacza, że od początku nie można było mu wierzyć. Lista priorytetowa mówi: „po prostu wyrzucić rzeczy, które się nie zmieściły, bo one nie były dość ważne”. To jest uczciwsze, bo właśnie taka była od początku nasza umowa, że rzeczy z końca listy mogą wypaść.

Za to harmonogram pozwala nam na śledzenie zależności czasowych. Część zadań da się wykonać niezależnie od siebie, na przykład kilka różnych dekoracji może powstawać w tym samym czasie, rękami różnych osób o podobnych umiejętnościach. Inne zadania są wzajemnie zależne, na przykład nie możemy zbudować rozgrywki w danym levelu, dopóki nie działają niezbędni dla niej aktorzy i rekwizyty. Podczas produkcji takie informacje bardzo się przydają, bo na przykład wynika z nich, kiedy do zespołu warto dodać nowe osoby. Harmonogram umożliwia również ustalenie tak zwanej **ścieżki krytycznej**, czyli takiego ciągu zadań, że opóźnienie któregośkolwiek z nich spowoduje opóźnienie całego przedsięwzięcia. To właśnie ścieżka krytyczna decyduje o tym, ile czasu zajmie zbudowanie Twojej gry.

Spójrzmy na przykładowy, bardzo uproszczony harmonogram:



W naszym zespole produkcja przebiega trzema ścieżkami: pierwszy zespół pracuje nad lewalem, drugi nad aktorami, a trzeci nad rekwizytami i dekoracjami. Gdyby zależało nam, by dwaj aktorzy, których mamy w tym przykładzie, powstawali w tym samym czasie, należałoby zorganizować dodatkowy zespół.

Niektóre zadania od siebie zależą, a inne nie. Do rozpoczęcia prac nad rozgrywką potrzebni są nam aktorzy i rekwizyty, ale nie dekoracje. Tych będziemy potrzebować dopiero do prac nad wyglądem levelu. Jest nam wszystko jedno, którym z dwóch aktorów drugi zespół zajmie się najpierw, ale za to pierwszy zespół pracuje w zadanej kolejności: najpierw makieta, potem rozgrywka, potem prezentacja. Trzeci zespół też uporządkował swoje zadania rozsądnie, bo zaczął od rekwizytów, które będą potrzebne wcześniej niż dekoracje.

Dzięki harmonogramowi takie decyzje są oczywiste. Bez niego mogłoby się zdarzyć, że trzeci zespół zacząłby na przykład od rzeczy najłatwiejszej, czyli od którejś dekoracji, zamiast od najbardziej pilnej, czyli w tym wypadku od rekwizytu.

Zauważ teraz ciekawą zależność czasową. Aktorzy powstają tak długo, że po zakończeniu prac nad makieta zespołu tworzącego level musi na nich poczekać. Gdyby udało się przyspieszyć prace nad aktorami, to cała gra powstałaby szybciej, ale tylko do momentu, gdy praca ta zaczęłaby zajmować mniej czasu niż praca nad rekwizytami. Od tego momentu rozgrywka czekałaby już na rekwizyty, a nie na aktorów, czyli to rekwizyty, a nie aktorzy, byłyby teraz na ścieżce krytycznej. Jeżeli zależałoby nam na przyspieszeniu prac, musielibyśmy wtedy poszukać oszczędności w rekwizytach. Być może któryś z nich nie jest nam potrzebny?

Jeżeli nie mamy możliwości przyspieszenia prac, to jest jeszcze jedna opcja: możemy dać pierwszemu zespołowi więcej do roboty, na przykład zlecić przygotowanie makiet następnych leveli. Jest szansa, że nie będzie nas to dodatkowo kosztowało, ponieważ nie zatrudnimy nowych osób, tylko zapełnimy przestoje.

Zauważ, że w czasie, gdy pierwszy zespół pracuje nad makietą, jego członkowie mogą na przykład zrobić sobie przerwę i pójść na urlop, ponieważ makietę nie znajduje się na ścieżce krytycznej. Jeżeli prace nad nią trochę się opóźnią, termin ukończenia budowy całej gry się nie zmieni.

Z dekoracjami mamy podobną sytuację. Kiedy zespół, któremu je powierzamy, upora się z rekwizytami, okazuje się, że ma dużo wolnego czasu. Gra nie będzie powstawała dłużej, nawet jeśli urozmaicimy dekoracje i stworzymy dwa lub trzy dodatkowe elementy. Dlatego dekoracje też nie są na ścieżce krytycznej: jeżeli w zespole zajmującym się nimi ktoś zachoruje i prace się opóźnią, nic złego się nie stanie. Z kolei zespół produkujący level może co prawda iść na urlop po wykonaniu makiety, ale za to podczas prac nad rozgrywką i dekoracjami musi się zmobiliżować, bo każde opóźnienie wpłynie na czas budowy całej gry. Po to właśnie jest harmonogram, żeby pokazywać Ci takie zależności.

Harmonogram działa pod warunkiem, że informacje w nim zawarte są wiarygodne. Nie używaj go do dekretowania rzeczy, których nie możesz zagwarantować, takich jak czas pracy nad zadaniem, które nie jest dobrze opisane. O to, ile czasu zajmie wykonanie danego elementu, pytaj osoby, które później będą faktycznie dany element wykonywać. Pamiętaj, że one też zgadują, tylko trochę lepiej niż Ty. Zupełnie normalne jest, że pomyłają się o 10–20 procent, dlatego zrób zapas. Unikaj tak zwanych pobożnych życzeń: to, że wolisz, żeby prace udało się zrobić na dany termin, nie znaczy, że istnieje taka możliwość. Jeżeli współpracownicy mówią, że nie zdążą, to nie zdążą. Nie kłóć się z tym.



Kiedy pracy jest tyle, że nie starcza czasu, naiwni autorzy wpadają na pomysł, żeby zostać w pracy dłużej. To nie działa, ponieważ zmęczony człowiek pracuje wolniej i popełnia więcej błędów. Wydaje mu się, że haruje, a w rzeczywistości traci czas. Dlatego zryw „po godzinach” nigdy nie powinien trwać dłużej niż tydzień. Po każdym intensywnym okresie pamiętaj, żeby wziąć trochę dodatkowego wolnego.

W trakcie preprodukcji zgadywanie, ile czasu zajmą nam poszczególne zadania, zwykle się nie udaje. Jest po prostu za dużo niewiadomych. W takich wypadkach

przystawiamy się na tak zwany **timeboxing**, czyli „pracę w pudełku czasowym”. Polega to na tym, że wyznaczamy sobie nieprzekraczalny termin i zakładamy, że to, co będziemy mieć do dyspozycji w tym dniu, będzie naszą wersją ostateczną. Na przykład kiedy prototypujemy skoczne buty bohaterki, ograniczamy w ten sposób liczbę iteracji. Jeżeli do takiego a takiego dnia uda się sprawić, by buty działały dobrze, to świetnie: mamy nowy element rozgrywki i odtąd używamy go w takiej wersji, jaką udało nam się stworzyć. Jeżeli się nie uda, to decydujemy, czy dodajemy czas, czy też wyrzucamy buty z gry. Za to na pewno nie powiemy po prostu: „jak buty będą działać, to wtedy będzie zrobione”, dlatego że wtedy buty powstawałyby na święte nigdy. Konkretny termin tworzy zobowiązanie, które da się wyegzekwować, bo kiedy dany termin przekraczamy, rozumiemy, że to jest nasze uchybienie, a kiedy go przesuwamy, to wiemy, ile to będzie kosztowało.

Zespoły zadaniowe często w związku z tym pracują w **sprintach**. Sprint to krótki odcinek czasu, zwykle nie dłuższy niż miesiąc, z nieprzekraczalnym terminem, w którym na początku wyznaczamy sobie pewne cele, takie jak sprawienie, żeby antygrawitacyjne buty dobrze działały. Te cele dzielimy w miarę możliwości na drobniejsze kroki, takie jak „narysowanie ilustracji koncepcyjnej”, „przygotowanie pierwszej wersji modelu”, „zaprogramowanie zachowania”, „dodanie obsługi przycisku skakania” czy „przeróbki w sterowniku bohaterki”. Ponieważ w zespole zadaniowym jest tylko kilka osób i zadań nie ma dużo, a sprint jest z założenia krótki, to cały czas widzimy, czy nadążamy.

Pomaga nam w tym **scrum**, czyli odbywane raz dziennie, zwykle rano, piętnastominutowe spotkanie, podczas którego wszyscy członkowie zespołu po kolei deklarują, co zrobili poprzedniego dnia, co zamierzają zrobić dzisiaj, co im przeszkadza w pracy i jakiej pomocy potrzebują. Dzięki temu wszyscy w zespole są na bieżąco z pracą całego zespołu. Scrum działa pod warunkiem, że faktycznie wszyscy mówią krótko. Dyskusje ucina się i zostawia na potem. Jeżeli ktoś zaczyna łąć wodę, na przykład dlatego, że wstydzi się przyznać, że się z czymś spóźnia, należy mu przerwać.

W skrócie

- Elementy rozgrywki produkuje się ze składników takich jak modele, materiały, dźwięki, zachowania i sterowniki. Pracuje się nad nimi na bardzo odmienne sposoby.
- Bardzo wiele w grze zależy od tego, jak ją budujemy, a nie ile pracy w nią wkładamy. Czasami wystarczy zamknąć jedne drzwi, by level zmienił się nie do poznania.
- Projektant ustala reguły i przebieg gry, czyli rzeczy, których nie widać na pierwszy rzut oka.
- Budowa gry dzieli się na fazę koncepcyjną, preprodukcję i produkcję. Ten podział służy temu, by najbardziej ryzykowne i skomplikowane decyzje podjąć wtedy, gdy błąd najmniej nas kosztuje.
- Większość elementów powstaje w trybie iteracyjnym. Po każdej iteracji analizujemy rezultaty i zastanawiamy się, jak można dany element poprawić. Unikamy wyrzucania go do kosza i zaczynania od nowa.
- Złożone zadania z dużą liczbą zależności powierzamy zespołom zadaniowym. Zadania powtarzalne i pozbawione ryzyka wpuszczamy w potok.
- Harmonogram jest Twoim najlepszym przyjacielem, pod warunkiem że umieszczasz w nim tylko rzeczy, które możesz zagwarantować.
- Nie ma w tym rozdziale ani słowa o testach. Wrócimy do nich w rozdziale 4.