

GITHUB

<https://github.com/BroZor/PyramidGamesAcademy>

ASSETY

<https://freesound.org/people/OwlStorm/sounds/151229/>

<https://freesound.org/people/Valenspire/sounds/699352/>

<https://pixabay.com/sound-effects/forest-wind-and-birds-6881/>

<https://www.unrealsensei.com/asset/ue5beginner>

<https://www.unrealengine.com/marketplace/en-US/product/stylized-pbr-nature-pack>

ODPOWIEDZI NA PYTANIA

1. How would you define object-orientedness? What does it mean for a language to be object-oriented?
Programowanie obiektowe to takie w którym używamy obiektów klas, te obiekty klas komunikują się między sobą. Każdy obiekt ma własne zmienne, funkcje i odpowiada za inną część kodu.
2. What is inheritance?
Dziedziczenie pozwala jednej klasie na używanie metod, lub zmiennych z drugiej klasy. Pozwala też robić klasy abstrakcyjne które mogą być używane jako „wzór” którego mają używać klasy dziedziczące po takiej.
3. What is the difference between a class and an interface?
Interfejs może implementować tylko metody, dodatkowo wszystko w interfejsie jest publiczne. Natomiast w klasie możemy implementować metody własne, dziedziczone, czy zdefiniowane w implementowanym interfejsie.
4. How many interfaces can a class implement in C++?
Można implementować nieskończenie wiele interfejsów.
5. What are the access modifiers in C++?
Modyfikatory dostępu ograniczają dostęp do zmiennych i metod klas:
private – tylko klasa główna może korzystać z metod i klas
protected – klasa główna i klasy dziedziczące mogą korzystać z tych metod i zmiennych
public – metody i zmienne mogą być wykorzystane wszędzie w kodzie
6. What is polymorphism? What does it mean to say that inheritance is polymorphic?
Polimorfizm to określenie na zmianę metody wirtualnej dla danej klasy dziedziczącej. Dziedziczenie jest polimorficzne wtedy gdy klasa Parent implementuje metody które są wirtualne, a klasa Child która dziedziczy po klasie Parent implementuje te metody i zmienia ich zachowanie dla własnych potrzeb. Przez takie zachowanie się klas można stworzyć dużo klas dziedziczących po jednej klasie które będą wykorzystywały te same metody
7. Is using public fields in classes a sensible practice?
Ustawianie publicznych metod lub zmiennych w klasie nie jest dobrą praktyką, wszystkie metody i zmienne w klasie powinny być w jakiś sposób chronione przed

niechcianą zmianą. Dlatego wykorzystanie modyfikatorów dostępu typu `protected` i `private` jest dobrym sposobem żeby się uchronić od niechcianych zmian w kodzie.

8. How does a list differ from an array?

Długość listy może być łatwo zmieniana i nie potrzebujemy odgórnie znać jej wielkości, natomiast tablica często wymaga byśmy znali jej wielkość przy implementacji.

Dodatkowo lista ma dużo więcej metod pozwalających na operacje w nich.

Żeby przejrzeć listę należy użyć iteratora i bez jego użycia, nie ma możliwości tak jak w tablicy zajrzenia w indeks `n`.

9. What are delegates?

Delegat to zmienna wskazująca na funkcję, najbardziej powszechnym przykładem byłaby `lambda`.

10. Are you familiar with any design patterns?

Tak, Singleton, Factory, Strategy, Builder.

11. In what scenarios would you choose to use Blueprints over C++ in Unreal Engine 5, and vice versa? Please provide specific examples of where one is more advantageous than the other.

Blueprinty są łatwiejsze do opanowania i dzięki nim można szybko zrobić prototyp programu. Natomiast C++ pozwala nam na dogłębne spojrzenie w kod programu i zdecydowanie które elementy kodu mają pozostać, a które nie. Pozwala nam to na optymalizację programu na wyższym poziomie niż byłoby to możliwe z blueprintami. C++ pozwala też wykorzystać większą inwencję twórczą podczas programowania własnych funkcji.

Blueprintów użyłbym w sytuacji gdy chciałbym na szybko przetestować działanie pewnej funkcji w grze, a potem mógłbym taką funkcję zoptymalizować w C++.

C++ użyłbym gdybym musiał operować na bitach, lub robić większe obliczenia do gry.

12. What is an Actor class?

Klasa aktor to główna klasa po której dziedziczą wszystkie obiekty w grze.