

Żeby dało się zrobić interaktywną makietę osiedla 3D na stronie WWW (z możliwością klikania w poszczególne budynki), grafik / 3D designer powinien przygotować **konkretny zestaw plików i danych**. Poniżej masz **checklistę**, którą możesz mu po prostu przekazać.

1 Model 3D osiedla (najważniejsze)

Formaty (web-friendly):

-  **GLTF / GLB** (*najlepszy wybór – standard webowy*)
- ewentualnie: FBX (jeśli dev będzie konwertował)

Wymagania techniczne:

- **Każdy budynek jako osobny obiekt (mesh)**
 np. **Building_A**, **Building_B**, **Building_C**
 - **Spójny pivot (origin)** – najlepiej na poziomie gruntu
 - **Rozsądna liczba polygonów (low / mid poly)** – pod WWW
 - **Brak zbędnych detali niewidocznych z kamery**
-

2 Materiały i tekstury

Tekstury:

- Albedo / Base Color
- Normal Map (*opcjonalnie, ale mile widziane*)
- Roughness / Metallic (jeśli PBR)

Formaty:

- PNG / JPG
- Rozmiary zoptymalizowane pod web (np. 1K–2K, nie 8K)

Ważne:

- Jeden budynek = możliwość **podmiany materiału** (np. hover / klik / sold out)
-

3 Podział logiczny pod interakcję (to często pomijane!)

Grafik powinien:

- ♦ **Nazwać obiekty jednoznacznie**
- ♦ Nie scalać wszystkiego w jeden mesh
- ♦ Oddzielić:
 - budynki
 - drogi
 - zieleń
 - elementy tła (drzewa, ogrodzenia)

📌 Dzięki temu developer:

- wykryje kliknięcie (**raycasting**)
 - podświetli konkretny budynek
 - przypisze mu dane (np. liczba mieszkań)
-

4 Widoki / kamera (opcjonalnie, ale bardzo pomaga)

Grafik może przygotować:

- 📷 **Domyślne ujęcie kamery**
- 📷 Alternatywne widoki (np. z lotu ptaka, zbliżenie)
- 🎯 Punkty focusu na konkretne budynki

(np. zapisane jako notatka lub scena w pliku 3D)

5 Dane opisowe do integracji (bardzo przydatne)

W osobnym pliku (np. **JSON / Excel / CSV**):

Dla każdego budynku:

ID budynku

Nazwa

Liczba mieszkań

Status (wolne / sprzedane)

Link do podstrony

Kolor highlightu

 ID musi **zgodzić się z nazwą obiektu w modelu 3D**

6 Mockupy / referencje wizualne (dla devów i biznesu)

Grafik może dorzucić:

-  Screenshoty makiety (desktop / mobile)
 -  Referencję kolorów hover / active
 -  Jak ma wyglądać:
 - hover
 - klik
 - stan „sprzedane”
-

7 Czego NIE robić (ważne ostrzeżenia)

-  Jednego zlepionego modelu całego osiedla
-  Milionów polygonów „jak do renderu”
-  Braku nazw obiektów

- ✗ Tekstur 8K bez sensu
 - ✗ Eksportu tylko do JPG / PNG bez modelu 3D
-

8 Przykładowy stack po stronie WWW (dla kontekstu)

Developer zwykle użyje:

- **Three.js / Babylon.js**
 - GLTF / GLB
 - Raycasting do klikania
 - Overlay HTML (tooltipy, modale)
-

9 Gotowa checklista do wysłania grafikowi

- ✓ Model 3D osiedla w GLTF/GLB
 - ✓ Każdy budynek jako osobny obiekt z nazwą
 - ✓ Tekstury zoptymalizowane pod web
 - ✓ Możliwość zmiany materiału budynku
 - ✓ Opcjonalne ujęcia kamery
 - ✓ Lista ID budynków + dane opisowe
-

Jeśli chcesz, mogę:

-  przygotować **krótką specyfikację PDF dla grafika**
-  rozpisać to pod **konkretny silnik (Three.js / Babylon.js)**
-  dopasować checklistę pod **stronę dewelopera (sprzedaż mieszkań)**

Daj znać 

w jakich programach może to zrobić grafik ?

Grafik może przygotować **interaktywną makietę osiedla 3D pod WWW** w kilku **sprawdzonych programach**. Poniżej masz **konkretną listę + kiedy który wybrać**, żeby efekt końcowy nadawał się do klikalnej wizualizacji na stronie.



Najczęściej polecane programy (WWW-friendly)

● Blender ★ (najlepszy wybór)





Dlaczego:

- darmowy i bardzo popularny
- **świetny eksport do GLTF / GLB**
- pełna kontrola nad:
 - nazwami obiektów
 - materiałami
 - hierarchią sceny
- idealny pod **Three.js** / **Babylon.js**

Do czego idealny:

- osiedla deweloperskie
- klikalne budynki
- low / mid poly pod web





Dlaczego:

- standard w architekturze
- świetny do precyzyjnych brył budynków
- dobre narzędzia do dużych scen

Uwaga:

- płatny
- eksport do GLTF często wymaga **konwersji** (np. przez Blender)

Cinema 4D





Dlaczego:

- bardzo czytelna hierarchia obiektów
- łatwe zarządzanie sceną
- lubiany przez motion / web designerów

Uwaga:

- eksport GLTF OK, ale czasem wymaga testów

Programy wspierające (często jako źródło geometrii)

SketchUp





Dlaczego:

- szybkie modelowanie architektoniczne
- często używany przez architektów

Ważne:

- **nie finalnie pod WWW**
- model zwykle:
 - trafia do Blendera
 - jest optymalizowany
 - eksportowany do GLTF



Do czego:

- źródło danych BIM
- dokładne bryły budynków

Ważne:

- modele są **zbyt ciężkie**
 - wymagają uproszczenia przed WWW
-

Game engines (tylko wyjątkowo)

Unity / Unreal Engine





Kiedy TAK:

- konfigurator mieszkań
- spacer 3D
- bardzo zaawansowana interakcja

Kiedy NIE:

- zwykła strona WWW (ciężkie, overkill)

Rekomendacja końcowa (praktyczna)

✓ Najlepszy workflow:

Architekt: Revit / SketchUp

↓

Grafik 3D: Blender

↓

WWW: GLB + Three.js

🏆 Jeśli grafik ma wybrać jeden program:

👉 Blender

Co grafik MUSI umieć (niezależnie od programu)

- eksport do **GLTF / GLB**
 - naming obiektów
 - low-poly / optymalizacja
 - PBR materials
 - praca „pod interakcję”, nie tylko pod render
-

Jeśli chcesz, mogę:

-  przygotować **oficjalną rekomendację „Blender required”**
-  rozpisać **pipeline dla grafika + developera**
-  dać **checklistę eksportu GLTF (krok po kroku)**

Daj znać 