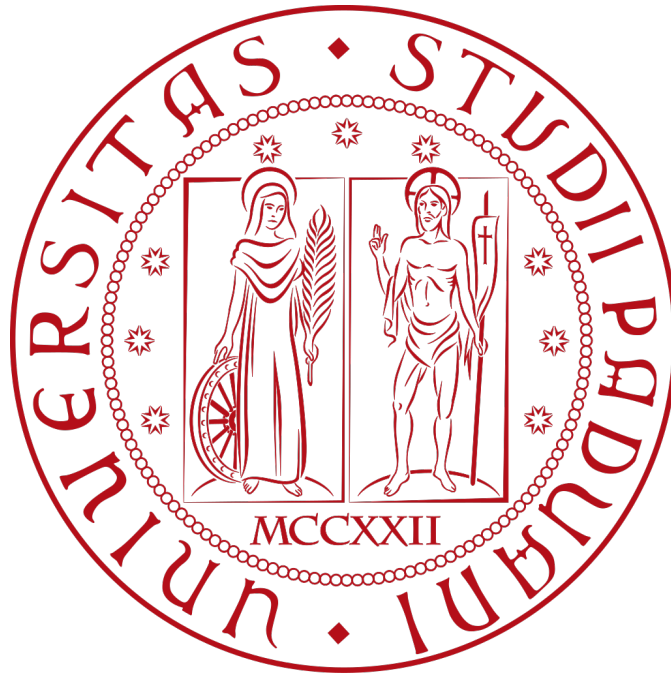




# ZUCCHETTI



# **AI TrAIning**

## **Piano di qualifica**

Filippo Sabbadin

---

## Indice

<b>1. Introduzione .....</b>	<b>1</b>
1.1. Scopo del documento .....	1
1.2. Glossario .....	1
1.3. Riferimenti .....	1
1.3.1. Riferimenti normativi .....	1
1.3.2. Riferimenti informativi .....	1
<b>2. Metriche di qualità .....</b>	<b>3</b>
2.1. Introduzione .....	3
2.2. Qualità di processo .....	3
2.2.1. Fornitura .....	3
2.2.2. Sviluppo .....	3
2.2.2.1. Codice .....	3
2.2.2.2. Grafica 3D .....	4
2.2.3. Documentazione .....	5
2.2.4. Verifica .....	5
2.2.5. Gestione della qualità .....	6
2.3. Qualità del prodotto .....	6
2.3.1. Funzionalità .....	6
2.3.2. Affidabilità .....	6
2.3.3. Efficienza .....	7
2.3.3.1. Specifiche macchina .....	7
2.3.4. Compatibilità .....	7
<b>3. Metodologie di testing .....</b>	<b>9</b>
3.1. Tipologie di test .....	9
3.1.1. Test manuali .....	9
3.1.2. Test di unità .....	9
<b>4. Cruscotto di valutazione delle metriche .....</b>	<b>10</b>
4.1. Codice .....	10
4.2. Grafica 3D .....	11
4.2.1. MPC-MTC - Model Tris Count .....	11
4.2.2. MPC-MBC - Model Bones Count .....	12
4.2.3. MPC-UIC - UV Islands Count .....	13
4.2.4. MPC-UIS - UV Islands Space .....	13

---

## Lista di immagini

<b>Figura 1</b>	<b>Modello 3D con statistiche sul numero di vertici, facce e triangoli .....</b>	<b>11</b>
<b>Figura 2</b>	<b>Modello 3D con statistiche sul numero di ossa .....</b>	<b>12</b>
<b>Figura 3</b>	<b>UV del modello del giocatore .....</b>	<b>13</b>

---

## Lista di tabelle

<b>Tabella 1</b>	<b>Valori per misurare la qualità della fornitura .....</b>	<b>3</b>
<b>Tabella 2</b>	<b>Valori per misurare la qualità dello sviluppo .....</b>	<b>4</b>
<b>Tabella 3</b>	<b>Valori per misurare la qualità dello sviluppo .....</b>	<b>4</b>
<b>Tabella 4</b>	<b>Valori per misurare la qualità della documentazione .....</b>	<b>5</b>
<b>Tabella 5</b>	<b>Valori per misurare la qualità del processo di verifica .....</b>	<b>5</b>
<b>Tabella 6</b>	<b>Valori per misurare la gestione della qualità .....</b>	<b>6</b>
<b>Tabella 7</b>	<b>Valori per misurare la qualità del prodotto in termini di funzionalità .....</b>	<b>6</b>
<b>Tabella 8</b>	<b>Valori per misurare la qualità del prodotto in termini di affidabilità .....</b>	<b>7</b>
<b>Tabella 9</b>	<b>Componenti macchina 1 .....</b>	<b>7</b>
<b>Tabella 10</b>	<b>Valori per misurare la qualità del prodotto in termini di compatibilità .....</b>	<b>8</b>

---

# 1. Introduzione

## 1.1. Scopo del documento

Il presente documento ha lo scopo di definire le metriche di qualità che verranno utilizzate per valutare il prodotto software, e le modalità di verifica e validazione del prodotto.

## 1.2. Glossario

Per facilitare la comprensione del documento, è stato creato un glossario che contiene i termini utilizzati nel documento e le loro definizioni. I termini presenti nel glossario sono colorati di blu e seguiti da un'asterisco: [esempio\\*](#).

Il glossario è accessibile tramite il link:

<https://github.com/fliposab/ProgettoStage/blob/main/Documentazione/Glossario.pdf>

oppure consultando il rispettivo documento all'interno della stessa cartella.

## 1.3. Riferimenti

### 1.3.1. Riferimenti normativi

- Norme di progetto:

<https://github.com/fliposab/ProgettoStage/blob/main/Documentazione/Norme-di-progetto.pdf>

### 1.3.2. Riferimenti informativi

- Slide T08 - Qualità di processo:

<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T08.pdf>

- Slide T09 - Verifica e validazione:

<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T09.pdf>

- 
- Glossario:

<https://github.com/fliposab/ProgettoStage/blob/main/Documentazione/Glossario.pdf>

---

## 2. Metriche di qualità

### 2.1. Introduzione

Le metriche di qualità sono utilizzate per valutare la qualità del prodotto software, e per identificare eventuali problemi o aree di miglioramento.

### 2.2. Qualità di processo

La qualità di processo si riferisce alla qualità delle attività svolte durante lo sviluppo del prodotto software.

Le metriche di qualità di processo sono utilizzate per valutare l'efficacia e l'efficienza delle attività svolte, e per identificare eventuali problemi o aree di miglioramento.

#### 2.2.1. Fornitura

Per il processo di fornitura, vengono indicate tutte le scelte operative fatte in fase di sviluppo. Viene usato l'acronimo MPC (Minimum Predictive Capability).

In questo caso, il MPC è il valore minimo da raggiungere per essere considerato accettabile.

- **MPC-CT - Completion Time:** tempo totale previsto per completare il progetto, idealmente non deve superare quello pianificato nel Piano di lavoro.

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-CC	Completion Cost	$\leq 105\%$ EC	$\leq 100\%$ EC

Tabella 1: Valori per misurare la qualità della fornitura

#### 2.2.2. Sviluppo

##### 2.2.2.1. Codice

- **MPC-RSI - Requirements Stability Index:** indice di stabilità dei requisiti. Indica la percentuale di requisiti che sono stati modificati rispetto al totale dei requisiti. Un valore alto indica che i requisiti sono stabili e non soggetti a modifiche frequenti.

- **MPC-TD - Technical Debt Ratio:** rapporto tra il tempo necessario per risolvere i problemi tecnici e il tempo necessario per sviluppare nuove funzionalità. Un valore basso indica che il codice è ben strutturato e non presenta problemi tecnici.

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-RSI	Requirements Stability Index	$\geq 80\%$	100%
MPC-TD	Technical Debt Ratio	$\leq 15\%$	$\leq 5\%$

Tabella 2: Valori per misurare la qualità dello sviluppo

### 2.2.2.2. Grafica 3D

Le seguenti metriche vengono applicate principalmente al [modello 3D\\*](#) principale del giocatore, visto che gli altri modelli utilizzati sono semplicemente cubi o sfere:

- **MPC-MTC - Model Tris Count:** un modello 3D è costituito da un certo numero di triangoli e vertici. Ovviamente, maggiore è il numero di triangoli, più risorse sono richieste per il [rendering\\*](#) di esso, rendendo inoltre più complicate successive modifiche. Un numero giusto (né troppo alto né troppo basso) è consigliato per garantire una buona qualità del modello ed evitare possibili complicazioni.
- **MPC-MBC - Model Bones Count:** il movimento del modello tramite ossa è un processo che consuma risorse della CPU. Visto che la CPU rappresenta un potenziale [bottleneck\\*](#) visto che è molto più lenta rispetto alla GPU, è necessario minimizzare il numero di ossa nell'armatura del modello. Il numero minimo dovrebbe essere: 12 per gli arti, mani e piedi + 4 per la spina dorsale + 16 per le ossa [IK\\*](#).
- **MPC-UIC - UV Islands Count:** avere un numero minore di [UV Islands\\*](#) permette una gestione della [texture\\*](#) più semplice e prestazioni leggermente migliori, visto che i vertici ai bordi dell'isola UV vengono renderizzati due o più volte.
- **MPC-UIS - UV Islands Space (in percentuale):** percentuale della texture occupata e utilizzata dalle UV Islands. Uno spazio maggiore utilizza più pixel dell'immagine e garantisce una maggiore qualità.

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-MVC	Model Tris Count	$\leq 5.000$	$\leq 3.000$
MPC-MBC	Model Bones Count	$\leq 40$	$\leq 32$
MPC-UIC	UV Islands Count	$\leq 30$	$\leq 15$
MPC-UIS	UV Islands Space	$\geq 40\%$	$\geq 50\%$

Tabella 3: Valori per misurare la qualità dello sviluppo



---

## 2.2.3. Documentazione

- **MPC-IG - Indice di Gulpease:** indica la complessità nella lettura di una frase o documento. Considera come variabili il numero di parole, di frasi e di lettere.

Formula dell'indice di Gulpease:

$$89 + \frac{(300 * \text{numero di frasi}) - (10 * \text{numero di lettere})}{\text{numero di parole}}$$

- **MPC-CO - Correttezza ortografica:** indica il numero di errori ortografici presenti nella documentazione.

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-IG	Indice di Gulpease	≥40	≥60
MPC-CO	Correttezza ortografica	3	0

Tabella 4: Valori per misurare la qualità della documentazione

## 2.2.4. Verifica

- **MPC-CCO - Code coverage:** quantità di codice eseguito durante i test. Viene utilizzato per valutare la qualità dei test e garantire che il codice sia stato adeguatamente testato. Un alto livello indica che il codice è stato eseguito in molti contesti e scenari diversi con diverse parti di codice. In altre parole, indica quanto codice è stato sottoposto ai test.
- **MPC-TSP - Test superati in percentuale:** indica la proporzione di test automatizzati o manuali che sono stati eseguiti con successo rispetto al totale dei test previsti. Viene espressa come una percentuale e serve a misurare quanto dell'applicazione in fase di sviluppo è stato verificato con successo tramite i test. Una percentuale alta di test superati indica che il sistema è stabile e che la maggior parte delle funzionalità funzionano come previsto. In altre parole, indica quanti test sono stati superati.

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-CCO	Code coverage	≥90%	100%
MPC-TSP	Test superati in percentuale	100%	100%

Tabella 5: Valori per misurare la qualità del processo di verifica

---

## 2.2.5. Gestione della qualità

- **MPC-SQM - Satisfaction of Quality Metrics:** misura la quantità di metriche soddisfatte. Questo valore viene calcolato come la somma delle metriche di qualità soddisfatte diviso il numero totale di metriche di qualità.

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-SQM	Satisfaction of Quality Metrics	≥85%	100%

Tabella 6: Valori per misurare la gestione della qualità

## 2.3. Qualità del prodotto

### 2.3.1. Funzionalità

- **MPD-RO - Copertura requisiti obbligatori:** indica la percentuale di requisiti obbligatori coperti dal prodotto. Un valore del 100% indica che tutti i requisiti obbligatori sono stati implementati.
- **MPD-OP - Copertura requisiti opzionali:** indica la percentuale di requisiti opzionali coperti dal prodotto. Un valore del 100% indica che tutti i requisiti opzionali sono stati implementati.

Metrica	Nome	Valore accettabile	Valore ottimo
MPD-RO	Copertura requisiti obbligatori	100%	100%
MPD-OP	Copertura requisiti opzionali	≥50%	100%

Tabella 7: Valori per misurare la qualità del prodotto in termini di funzionalità

### 2.3.2. Affidabilità

- **MPD-CC - Code coverage:** indica la percentuale di codice coperto dai test. Un valore alto indica che il codice è stato testato in modo approfondito e che è meno probabile che contenga errori.

Metrica	Nome	Valore accettabile	Valore ottimo
MPD-CC	Code coverage	≥80%	100%

Tabella 8: Valori per misurare la qualità del prodotto in termini di affidabilità

## 2.3.3. Efficienza

- **MPD-TF - Target FPS:** indica gli [fps\\*](#) obiettivo da mantenere durante l'esecuzione del gioco.
- **MPD-LS - Lag Spikes:** indica il numero di [lag spikes\\*](#) che occorrono durante l'esecuzione del gioco. Non vengono contati durante un [caricamento\\*](#) tra un livello e un altro e vengono contati solo durante i livelli.

### 2.3.3.1. Specifiche macchina

Molto importante è specificare le componenti della macchina su cui viene testato il gioco, dato che macchine diverse offrono prestazioni diverse:

Componente	Dettagli
<a href="#">CPU*</a>	AMD® Ryzen 5 4500U
<a href="#">GPU*</a>	Integrata alla CPU
<a href="#">RAM*</a>	8GB DDR4
<a href="#">SSD*</a> / <a href="#">HDD*</a>	Almeno 300 MB liberi
Sistema Operativo	Pop_OS

Tabella 9: Componenti macchina 1

In sintesi, la macchina su cui viene testato il gioco offre prestazioni sulla fascia media-bassa, quindi si ritiene che se il gioco offre delle buone prestazioni sulla macchina di testing, offrirà in media buone prestazioni su tutte le macchine con un sistema operativo supportato.

## 2.3.4. Compatibilità

- **MPD-SOS - Supported Operative Systems:** numero di sistemi operativi supportati, si punta a supportare almeno Windows 11 e Ubuntu.
- **MPD-SL - Supported Languages:** numero di lingue supportate, si mira ad avere almeno 2 lingue supportate: italiano e inglese.

---

<b>Metrica</b>	<b>Nome</b>	<b>Valore accettabile</b>	<b>Valore ottimo</b>
MPD-SOS	Supported Operative Systems	$\geq 2$	$\geq 3$
MPD-SL	Supported Languages	$\geq 2$	$\geq 2$

Tabella 10: Valori per misurare la qualità del prodotto in termini di compatibilità

---

## **3. Metodologie di testing**

### **3.1. Tipologie di test**

#### **3.1.1. Test manuali**

#### **3.1.2. Test di unità**

I test di unità sono utilizzati per verificare il corretto funzionamento delle singole unità del prodotto software, come ad esempio le funzioni o i metodi.

I test di unità sono eseguiti durante lo sviluppo del prodotto, e sono utilizzati per identificare eventuali problemi o errori nelle singole unità del prodotto.

---

## **4. Cruscotto di valutazione delle metriche**

### **4.1. Codice**

---

## 4.2. Grafica 3D

### 4.2.1. MPC-MTC - Model Tris Count

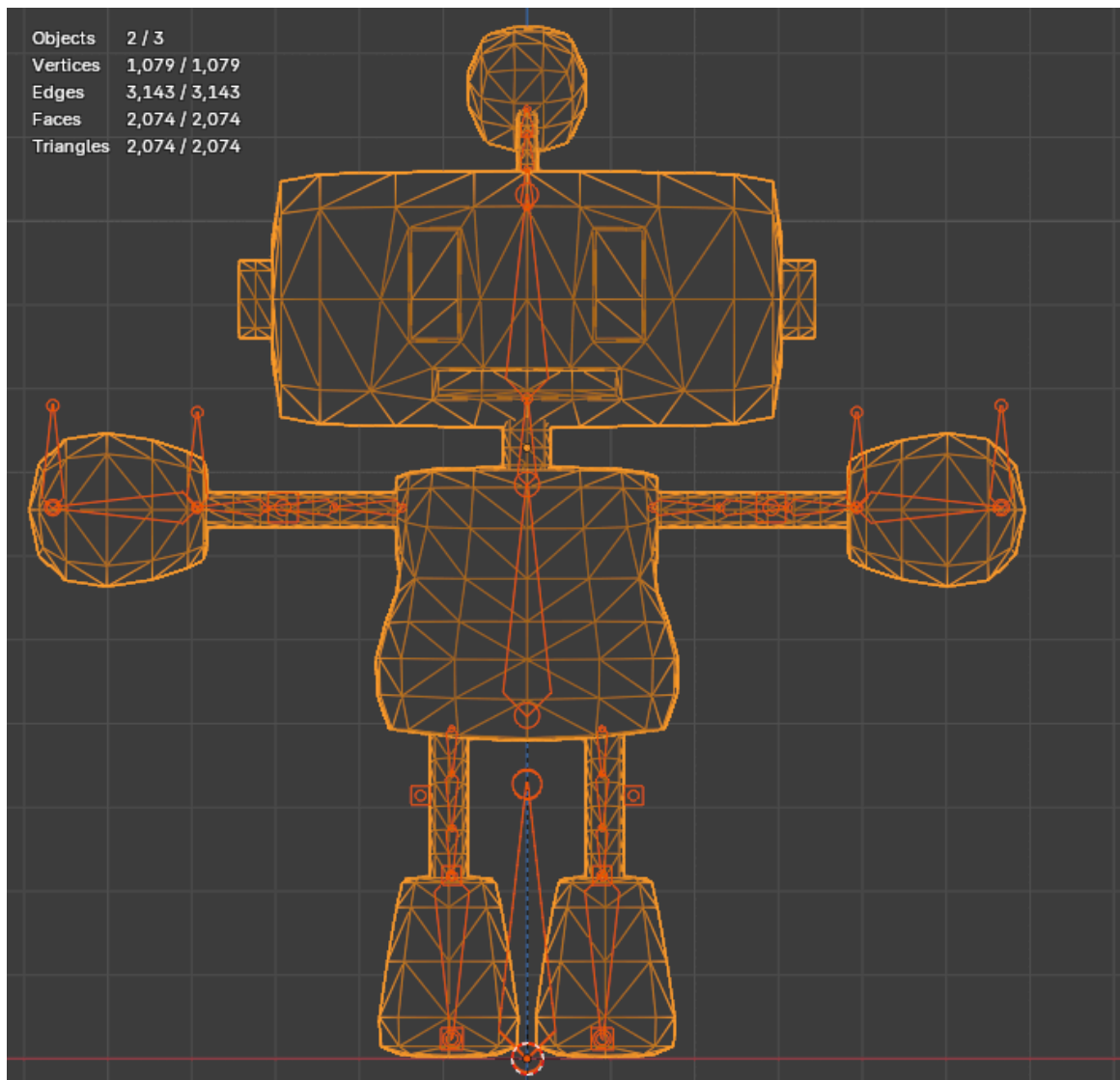


Figura 1: Modello 3D con statistiche sul numero di vertici, facce e triangoli

Dall'immagine si può vedere che il numero totale di triangoli del modello ammonta a 2.074. Ogni volta che il modello presentava forme simili a un cilindro, si è cercato di mantenere otto facce laterali.

---

### 4.2.2. MPC-MBC - Model Bones Count

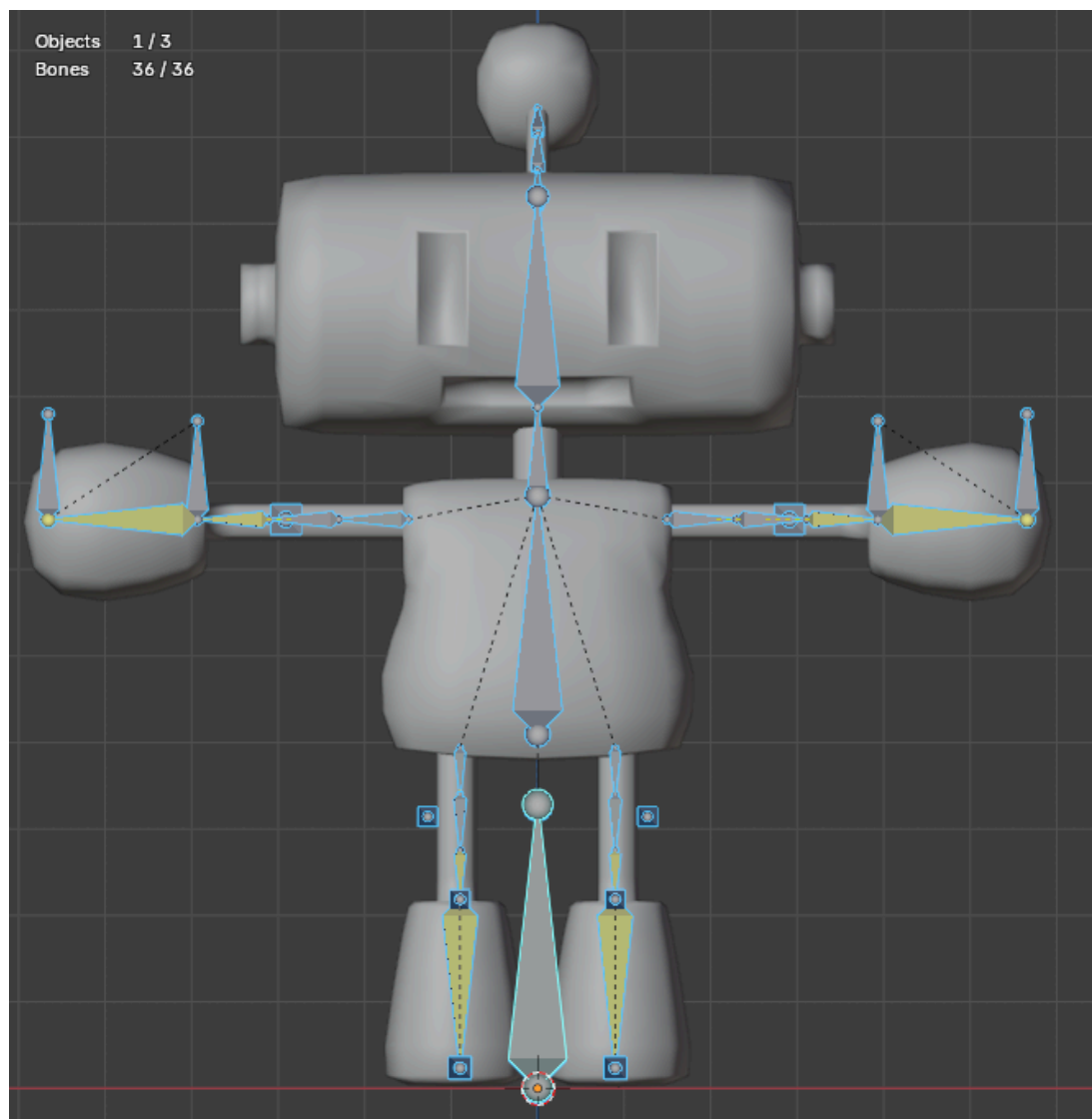


Figura 2: Modello 3D con statistiche sul numero di ossa

Dall'immagine si può vedere che il numero totale di ossa (ossa IK incluse) ammonta a 36. Sono state richieste più ossa per dare un effetto «curva» agli arti del modello. Inoltre sono state aggiunte tre ossa per animare l'antenna sulla testa.



---

### 4.2.3. MPC-UIC - UV Islands Count

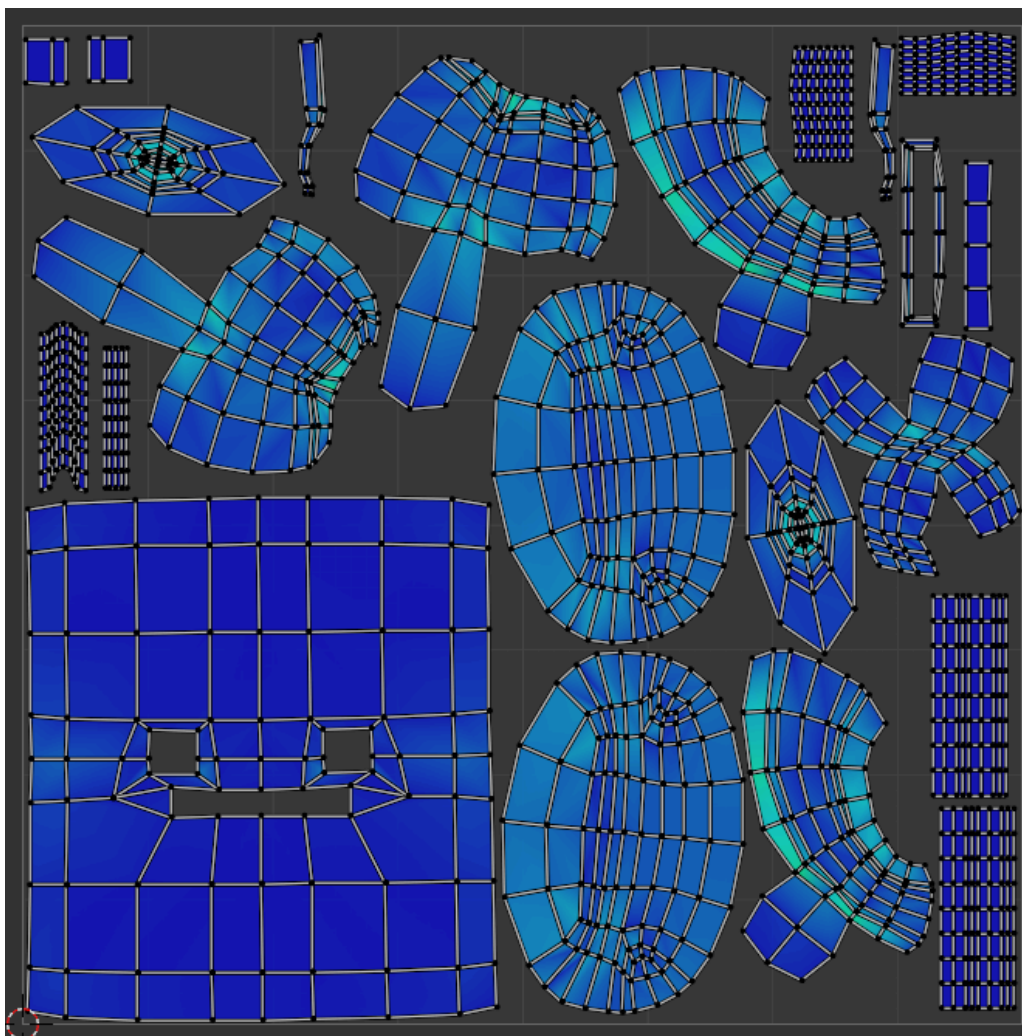


Figura 3: UV del modello del giocatore

Dall'immagine è possibile vedere anche il numero di isole UV (22) e l'allungamento delle facce: in blu le facce con un basso allungamento, mentre in azzurro-verde quelle che via via sono allungate.

### 4.2.4. MPC-UIS - UV Islands Space

Sempre dall'immagine sopra si può vedere la disposizione delle isole UV in una possibile immagine quadrata. Lo spazio occupato è pari al 68%.

Non offrendo la disponibilità di misurare lo spazio UV occupato nativamente u blender, è stato utilizzato uno script di Python esterno.