

דרישות פונקציונליות ולא פונקציונליות - פרויקט המרת פריטי לבוש 2D ל-3D

דרישות פונקציונליות:

1. העלאת תמונות דו-ממדיות: המשתמש יוכל להעלות תמונה או סקיצה של פריט לבוש דו-ממדי בפורמטים נפוצים (PNG, JPG)
2. המרה אוטומטית לתלת-ממד: המערכת תבצע המרה מהירה ואוטומטית של תמונה דו-ממדית למודל תלת-ממדי תוך שמירה על הפרטים והדיוק של העיצוב
3. תצוגה וסיבוב מודל 3D: המשתמש יוכל לצפות במודל התלת-ממדי שנוצר, לסובב אותו, להתקרב ולהתרחק בצורה אינטואיטיבית
4. הדמיה על אוטארים ודמויות: המערכת תאפשר 'להלביש' את המודל התלת-ממדי על אוטארים או דמויות קיימות במערכת
5. שילוב ב-VR וב-AR: המערכת תאפשר ייצוא המודלים לסביבות מציאות מדומה (VR) ומציאות רבודה (AR) בפורמטים תואמים
6. התאמה אישית: היכולת לבצע שינויים במודל, כגון צבעים, מרקמים, והתאמת המידות לפרופורציות של הדמות או האוטר
7. הורדה ושיתוף: המשתמש יוכל להוריד את המודל התלת-ממדי שנוצר בפורמטים תואמים (FBX, OBJ) ולשתף אותו עם צוותים אחרים
8. ניהול ספריית פריטים: אפשרות לשמור פריטים תלת-ממדיים בספרייה אישית או ארגונית ולצפות בהם מאוחר יותר
9. שילוב במערכות צד שלישי: תמיכה בחיבור API עבור שילוב המערכת עם כלים קיימים בתחום המשחקים, האופנה והמסחר האלקטרוני

דרישות לא פונקציונליות:

1. ביצועים: זמן המרה ממוצע עד 5 דקות לתמונה באיכות סטנדרטית.
2. דיוק באיכות התלת-ממד: המערכת תבטיח שמירה על הפרטים הקטנים של הפריט הדו-ממדי במעבר לתלת-ממד
3. אבטחת מידע: כל המידע המועלה יאוחסן בצורה מאובטחת עם הצפנה (AES ו-HTTPS). מנגנון אימות משתמשים מאובטח
4. שימושיות: ממשק המשתמש יהיה אינטואיטיבי וידידותי למשתמשים חדשים וללא ניסיון בתחום התלת-ממד. המערכת תכלול הסברים וכלים עזר אינטראקטיביים
5. תאימות: המערכת תתמוך בפלטפורמות מרכזיות (Windows, Mac, Web). התאמה למגוון מכשירים כולל טאבלטים ומחשבים ניידים. ייצוא המודלים לפורמטים נפוצים לשימוש במשחקים ובסביבות VR/AR
6. התאמה לסביבת פיתוח חיצונית: המערכת תאפשר חיבור למערכות CAD ותוכנות דוגמת Blender או Unity

דרישות פונקציונליות ולא פונקציונליות לחלק הפרויקט שלנו

דרישות פונקציונליות:

חיפוש ואיסוף מאגרי נתונים

מה המערכת עושה:

חיפוש מאגרי נתונים ופתוחים שכוללים מודלים תלת-ממדיים לבגדים, כמו מאגרי Fashion3D ו-SewingPatterns. המאגר צריך לכלול:

מודלים תלת-ממדיים איכותיים עם נתוני פוליגונים, מפות טקסטורות (textures), משקולות (weights) ורפלקשן (reflection).

רישיונות ברי-שימוש המאפשרים אינטגרציה בפרויקט הגדול ללא הפרת זכויות יוצרים.

חשיבות: איסוף נתונים איכותיים חיוני לאמינות ולאפקטיביות של שאר המערכת.

בניית מבנה נתונים

מה המערכת עושה:

עיצוב סכמה למסד הנתונים שתכלול:

מידע על הפריט: סוג (חולצה, מכנסיים), מיקום חלקים בדמות.

מידע גרפי: כמות הפוליגונים, סוגי פוליגונים (triangles, quads), מפות טקסטורות.

אפשרות להוספת מאפיינים נוספים בעתיד, כמו מידע פיזיולוגי או פרטים על ההדמיה (simulation).

בחירת טכנולוגיה: אבחר מסד נתונים מתאים (כמו PostgreSQL עם PostGIS) שמאפשר ניהול נתונים מרחביים ותמיכה בתלת-ממד.

חשיבות: מבנה נתונים טוב יבטיח התאמה לתהליכים עתידיים וקלות שימוש למשתמשי המערכת.

איחוד וניהול נתונים

מה המערכת עושה:

כתיבת סקריפטים לניקוי נתונים שיטפלו בבעיות כמו:

ערכים כפולים במסד הנתונים.

נתונים חלקיים או חסרים.

לאחר הניקוי, השלמת תהליך איחוד נתונים שמטרתו ליצור מאגר אחיד ועדכני המשלב נתונים ממספר מקורות שונים.

חשיבות: תהליך זה מבטיח שהמערכת תעבוד על נתונים עקביים, ללא שגיאות או חוסרים.

פיתוח חבילה לניהול ואימון מודלים

מה המערכת עושה:

פיתוח API פנימי לשליפה ושימוש בנתונים.

יצירת DataLoader ב-Python:

טעינה יעילה של נתונים באצוות (batches).

הוספת יכולות augmentation (כגון סיבובים, שינוי קנה מידה, החלפת טקסטורות) לשיפור אימון המודלים.

אופטימיזציה לטעינה מקבילית, למשל באמצעות prefetching.

שילוב DataLoader בפלטפורמת האימון המרכזית של הפרויקט הגדול, כך שכל איטרציה מקבלת נתונים מתאימים ויעילים.

חשיבות: DataLoader יעיל מבטיח ניצול טוב של המשאבים ומסייע באימון מודלים איכותיים.

ויזואליזציה וניתוח נתונים

מה המערכת עושה:

בניית דאשבורד פנימי להצגת נתונים חיים על מאפייני המודלים:

חיתוכים לפי פרמטרים (לדוגמה: סינון לפי כמות הפוליגונים או סוגם).

מעקב אחרי התקדמות העבודה על הנתונים והמודלים.

חשיבות: דאשבורד מספק תמונת מצב ברורה ומאפשר ניתוח נתונים בזמן אמת.

דרישות לא פונקציונליות:

ביצועים

איך המערכת עובדת:

המערכת תטען נתונים בצורה מהירה, גם בעומסים גבוהים. טעינת נתונים באצוות (batches) מבטיחה עיבוד מהיר, תוך ניצול מיטבי של המשאבים.

חשיבות: ביצועים גבוהים חיוניים לשילוב עם שאר חלקי המערכת בפרויקט הגדול.

דיוק ואיכות נתונים

איך המערכת עובדת:

המערכת תזהה ותסיר נתונים כפולים או פגומים באופן אוטומטי. בנוסף, ייושם מנגנון בדיקה תקופתי לאיכות הנתונים.

חשיבות: נתונים איכותיים מבטיחים שהמודלים יאומנו בצורה אפקטיבית וללא בעיות.

תאימות

איך המערכת עובדת:

התאמת הטכנולוגיה לפורמטים נפוצים (FBX, OBJ) ותמיכה בכלים הנדרשים לפרויקט הגדול.

חשיבות: תאימות מבטיחה שהמערכת שלך תשתלב בצורה חלקה בפרויקט הגדול.

שימושיות

איך המערכת עובדת:

הממשק של ה-API וה-DataLoader יהיה ידידותי למשתמשים, והדאשבורד יציע חוויית משתמש אינטואיטיבית.

חשיבות: שימושיות טובה תקל על צוותי הפיתוח והשימוש במערכת.