## מכללת הדסה, החוג למדעי המחשב תכנות מונחה עצמים ופיתוח משחקים סמסטר ב,' תשפ"א

**תרגיל 4**

תאריך אחרון להגשה:

הנביאים – יום א,' ,23/05/21 בשעה 23:59

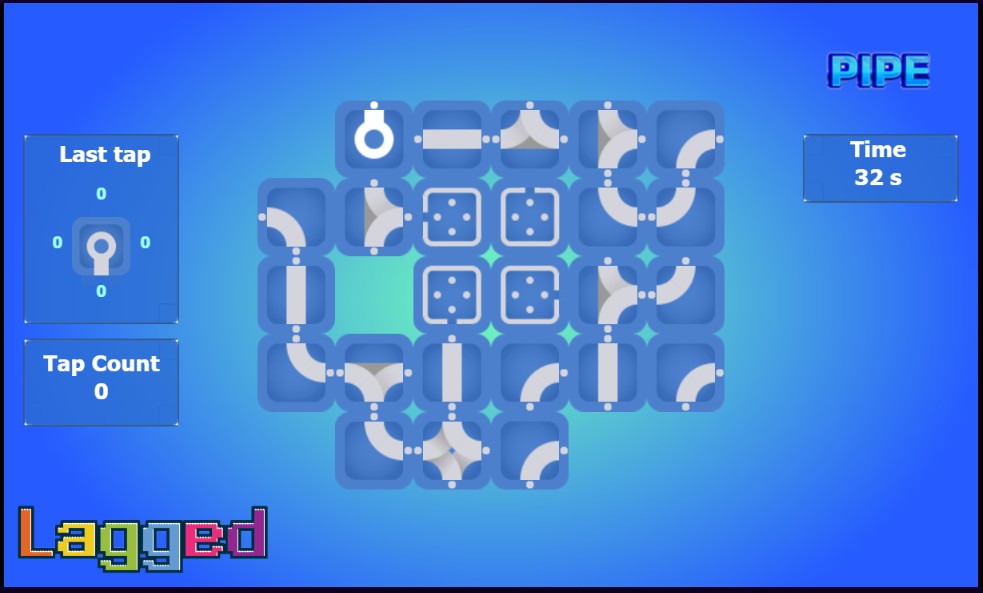
שטראוס גברים – יום ג,' ,18/05/21 בשעה 23:59

שטראוס נשים – יום ג,' ,18/05/21 בשעה 23:59

# מטרת התרגיל:

שימוש במבני נתונים, איטרטורים ואלגוריתמים בסגנון STL וריענון השימוש ב־.SFML

# תיאור כללי:



בתרגיל זה נממש את המשחק .Pipe

ניתן למצוא הדגמה של המשחק כאן: [ttps://www.crazygames.com/game/pipe](https://www.crazygames.com/game/pipe) .[h](https://www.crazygames.com/game/pipe)

# תיאור המשחק:

כללי

לפי

לפתור

צריך

שהוא

"חידה"

לוח עם

לשחקן

מוצג

בהם

הפאזל,

משחקי

מז'אנר

זה הוא

משחק

המשחק.

לשחקן מוצג לוח אריחים עם "צינורות," "מקור מים" ו־"כיור" )היעד של המים.(

### מטרת המשחק:

יותר בעזרת הצנרת. אין חובה לחבר את כל מקורות המים

אחד או

מים

למקור

כל ה"כיורים"

לחבר את

לצנרת, ואין חובה להשתמש בכל חלקי הצנרת.

### כללי המשחק:

כללי המשחק די פשוטים, כפי שניתן לראות בדוגמה. השחקן מחבר את מקור המים עם הכיור בעזרת "סיבוב" האריחים. השחקן יכול ללחוץ על כל אריח: לחיצה על הכפתור הימני של העכבר מסובבת את האריח נגד כיוון השעון ב־90 מעלות, ולחיצה על הכפתור השמאלי מסובבת את האריח עם כיוון השעון ב־90 מעלות. כל האריחים יכולים להסתובב למעט הכיור. כאשר מסובבים ומחברים צינור אחד למקור מים )או מחברים אותו לצינור אחר המלא במים,( הצינור מתמלא במים )וכל מה

שמחובר אליו,( וכאשר מנתקים אותו, הוא מתרוקן )וכל מה שמחובר אליו.(

# פירוט הדרישות:

כשאנחנו מממשים את המשחק, אנחנו לא צריכים לפתור אותו, זה תפקיד השחקן, אבל אנחנו צריכים להציג לוח "פתיר," לממש את כללי המשחק, לדאוג "למלא" במים את הצינורות המחוברים למקור מים, ולדאוג כמובן לזהות את סיום השלב, כלומר למלא מים עד הכיור, ולעבור לשלב הבא )או להציג מסך של "שלבים" כמו

להחליט בשאלה מה הצורה המתאימה מבחינה אלגוריתמית לטפל

נצטרך

שכך,

מכיוון

המקורי.(

במשחק

בבעיה זו )של זיהוי הצנרת המחוברת למקור מים,( וכן בשאלה מה מבנה הנתונים )או מבני הנתונים( שיעזור

למימוש נכון של המשחק.

אתם נדרשים לממש משחק שנעים לשחק בו, שהכל הולך חלק, ושדברים לא "נתקעים."

5 נקודות מהתרגיל )לא בונוס,(! יוקצו לטובת הנראות של התרגיל. כלומר משחק שנראה מצויין יקבל את כל

ה־5 נקודות. משחק "פשוט" "יתומחר" בהתאם.

## פרטי המשחק:

.1 במשחק שלכם יהיו שלושה שלבים כשהלוח שונה בין השלבים. .2 כאשר השחקן מסיים שלב, נציג הודעה על ההצלחה ונעבור לשלב הבא )או נציג מסך של "שלבים"

המשחק

על סיום

הודעה

נציג

השלישי,

את השלב

מסיים

השחקן

כאשר

המקורי.(

במשחק

כמו

בהצלחה.

.3 אין חובה לממש תפריט או מסך "שלבים" או לוח שיאים כמו במשחק המקורי. רק שה־X בפינה יצא

מהמשחק.

.4 אין חובה להציג שעון כמו במשחק המקורי.

.5 חובה להציג את מספר הלחיצות שנעשו עד עתה. .6 **אם השלבים שלכם קבועים:** טעינת השלבים צריכה להיעשות מקובץ באופן שאפשר יהיה לשנות את

בקובץ בודד או בקבצים נפרדים, בדרכים

להשתמש

ניתן

הקוד.

מחדש את

לקמפל

בלי

השלבים

שבהן הצענו לעשות זאת בעבר: למשל, שם הקובץ יהיה מורכב ממילה וממספר סידורי, או קבצים נפרדים וקובץ "playlist" שמכיל רשימה של כל הקבצים לטעינה. הקובץ יהיה בפורמט שעליו תחליטו ויש צורך לתעד אותו היטב ב־.Readme כמו כן עליכם לצרף קובץ כזה )או קבצים כאלה, אם לא כל

השלבים באותו הקובץ( לקוד שאתם מגישים. הצירוף חייב להיעשות באופן כזה שאחרי קומפילציה,

יועתקו

הקבצים

)כלומר,

בעצמה

הקובץ/הקבצים

תזהה את

והתוכנית

את הקוד,

להריץ

אפשר

אוטומטית על ידי CMake לתיקיית התוכנית, בעזרת פקודות .(configure\_file

מידע שאתם למשל יכולים להכניס לקובץ: מידע על גודל הלוח, מידע על כל אריח, או אולי להכין

### של השכנים של כל אריח בזמן ריצה מזכה

**חישוב**

האריחים.

השכנויות של

את רשימת

מראש

### בבונוס 2) נקודות.(

.7 אפשרות אחרת, מורכבת יותר, היא יצירה של השלבים באופן אקראי בכל פעם **ללא עזרה מקובץ** **חיצוני כלל**. האקראיות צריכה להיות בסוג האריחים, בכמות מכל סוג, ובכיוון שלהם. אין צורך שגודלהמסך של השלב שנוצר יהיה אקראי )אבל כדאי שבקלות תוכלו לשנות אותו בקוד שלכם.( האפשרות

יש צורך לוודא תוך כדי יצירה שהמסך שנוצר ניתן לפתרון. **מימוש יצירת**

בעיקר כי

מורכבת

הזו

### השלבים באופן כזה מזכה בבונוס נוסף 4) נקודות.(

**זיהוי תקלות וטיפול בשגיאות:**

במידה ובחרתם במשחק עם שלבים קבועים: בניתוח תוכן הקובץ, על התוכנית להתמודד עם קלט לא חוקי, למשל, קובץ לא קיים או שהפורמט לא מתאים למצופה לפי מה שהגדרתם. במקרים כאלה, תיזרק חריגה (exception) ופונקציית ה־main צריכה לתפוס אותה, להדפיס הודעה מתאימה לקובץ בשם log.txt )בתיקייה

הנוכחית( ולסיים את ריצת התוכנית.

## הערות למבנה הנתונים:

אפשרות די טבעית למימוש המשחק היא בעזרת גרף ואלגוריתם מתאים לטיול בגרף, כמו BFS או DFS

חלקי )כדי לזהות מה מחובר ומה לא,( אבל אולי ייתכנו אפשרויות טובות אחרות.

אם תממשו מבנה נתונים כזה, כדאי לבנות אותו יחסית גנרי, לא רק מוכוון למטרה הספציפית שלנו כרגע, אבל

לקחת את הדברים לכיוון גנרי מידי, לפי העיקרון שקיים בחלק מהשיטות של

לא לנסות

גם להיזהר

צריך

You Ain’t

תיבות של

ראשי

,YAGNI

המכונה:

(XP, Extreme Programming

)למשל,

תוכנה

הנדסת

...Gonna Need It

## הערות למימוש הגרפי ב־:SFML

כדי לממש את האריחים, אפשר להשתמש בתמונות שתמצאו ברשת או אפשר לממש בעזרת אובייקטים של

מלבנים ועיגולים של .SFML

אם החלטתם לממש בעזרת אובייקטים של ,SFML חשוב לציין 2 הערות:

.1 זה לא נחשב למשחק שנראה "טוב" )אלא אם כן, תצליחו "להפתיע" אותנו (: .( .2 אסור להשתמש ב"התנגשויות" של אובייקטים של ,SFML על מנת למצוא את הצינורות המלאים מים

)המחוברים למקור המים.(

# הערות לתיכון התוכנית:

שימו לב שיש בתרגיל למעשה שלושה חלקים:

.1 מבנה הנתונים שאנחנו נעזרים בו, האיטרטורים והאלגוריתמים.

.2 המימוש של הלוגיקה של המשחק.

.3 המימוש הגרפי.

כדאי ומומלץ להפריד בין מימוש מבני הנתונים, מימוש הלוגיקה ומימוש הגרפי, גם אם בסופו של דבר ברור

שהכול מתחבר יחד לכדי תוכנה אחת שמפעילה את כל החלקים.

כרגיל, את ההחלטות שתחליטו תתעדו כראוי בקובץ ה־.Readme בפרט יש צורך לתעד את ההחלטות על

והשימוש בספרייה

האלגוריתמים

הנתונים,

מבני

לגבי

מים

למקור

הצנרת

חיבור

ובדיקת

הלוח

ייצוג

אופן

הסטנדרטית, במה בחרתם להשתמש ולמה העדפתם מבנה נתונים או אלגוריתם מסוים על פני האחרים.

## קובץ ה־:README

יש לכלול קובץ README שיקרא README.docx README.doc, או README.txt )ולא בשם אחר.(

הקובץ יכול להיכתב בעברית ובלבד שיכיל את הסעיפים הנדרשים.

קובץ זה יכיל לכל הפחות:

.1 כותרת.

.2 פרטי הסטודנט: שם מלא כפי שהוא מופיע ברשימות המכללה, ת"ז.

.3 הסבר כללי של התרגיל.

התפקיד של כל אחד מהם

בתוכנית, מה

השונים

האובייקטים

מהם

קצר

הסבר

:(design)

.4 תיכון

וחלוקת האחריות ביניהם ואיך מתבצעת האינטראקציה בין האובייקטים השונים. .5 רשימה של הקבצים שנוצרו ע"י הסטודנט, עם הסבר קצר )לרוב לא יותר משורה או שתיים( לגבי

תפקיד הקובץ.

.6 מבני נתונים עיקריים ותפקידיהם.

.7 אלגוריתמים הראויים לציון.

.8 באגים ידועים.

.9 הערות אחרות.

יש לתמצת ככל שניתן אך לא לוותר על אף חלק. אם אין מה להגיד בנושא מסוים יש להשאיר את הכותרת

ומתחתיה פסקה ריקה. תכתבו ב־README כל דבר שרצוי שהבודק ידע כשהוא בודק את התרגיל.

## אופן ההגשה:

ששמו

לקובץ

להלן,

מה שיצוין

למעט

לתרגיל,

הקשור

כל קובץ

יש לדחוס

להגשה:

הקובץ

,exN\_firstname\_lastname.zip כאשר N הוא מספר התרגיל ו־firstname\_lastname הוא השם המלא

התרגיל הראשון: .(ex1\_albert\_einstein.zip במקרה של הגשה

כך את

יגיש

איינשטיין

אלברט

)לדוגמא

עם ,exN\_firstname1\_lastname1\_firstname2\_lastname2.zip

התבנית

לפי

יהיה

הקובץ

בזוג, שם

שמות המגישים בהתאמה )ללא רווחים; כלומר, גם בשמות עצמם יש להחליף רווחים בקו תחתי, כפי המודגם

לעיל.( כמו כן, במקרה של הגשה בזוג, **רק אחד** מהמגישים יגיש את הקובץ ולא שניהם.

לפני דחיסת תיקיית הפרויקט שלכם יש למחוק את הפריטים הבאים:

● תיקייה בשם ,out אם קיימת

● תיקייה בשם .vs

.vs לפעמים

התיקייה

.(VS

בעזרת

פותחים

שאנחנו

)זו

הראשית

בתיקייה

נמצאות

האלה

התיקיות

שתי

מוסתרת, אבל אם תפתחו את קובץ ה־zip שיצרתם, בוודאי תוכלו למצוא אותה ולמחוק אותה. ככלל אצבע, אם קובץ ה־zip שוקל יותר ממ"ב אחד או שניים, כנראה שלא מחקתם חלק מהקבצים הבינאריים

המוזכרים.

### וודאו כי קובץ ה־zip מכיל תיקייה ראשית אחת, ורק בתוכה יהיו כל הקבצים ותתי התיקיות של הפרויקט.

את הקובץ יש להעלות ל־Moodle של הקורס למשימה המתאימה.

**הגשה חוזרת:** אם מסיבה כלשהי סטודנט מחליט להגיש הגשה חוזרת יש לוודא ששם הקובץ זהה לחלוטין

לשם הקובץ המקורי. אחרת, אין הבודק אחראי לבדוק את הקובץ האחרון שיוגש.

כל שינוי ממה שמוגדר פה לגבי צורת ההגשה ומבנה ה־README עלול לגרור הורדת נקודות בציון.

מספר הערות:

.1 שימו לב לשם הקובץ שאכן יכלול את שמות המגישים.

.2 שימו לב שעליכם לשלוח את תיקיית הפרוייקט כולה, לא רק את קובצי הקוד שיצרתם. תרגיל שלא

יכלול את כל הקבצים הנדרשים, לא יתקבל וידרוש הגשה חוזרת )עם כללי האיחור הרגילים.(

המלצה כללית: אחרי שהכנתם את הקובץ להגשה, העתיקו אותו לתיקייה חדשה, חלצו את הקבצים שבתוכו ובדקו אם אתם מצליחים לפתוח את התיקייה הזו ולקמפל את הקוד. הרבה טעויות של שכחת קבצים יכולות

להימנע על ידי בדיקה כזו.

**בהצלחה!**

**תכנון:**

נעשה מחלקה File Handler שתתעסק עם כל הקבצים. היא תקרא מהקובץ ותיצור ווקטור של סטרינגים מקובץ פתור, ותבלגן אותו על הדרך. אחרי זה היא תעבור תא תא ותשלח את המידע לשני המבני נתונים – הגרף וההצגה על המסך. נעשה BFS, אם הבלגון לא גרם לכך שיהיה משהו פתיר – נשאיר את זה ככה. אחרת, נבלגן את זה מחדש ונשלח את זה שוב למחלקה.