

FIT3162 – Computer Science Project 2

User Guide (5%)

Due 11:55 pm Friday 11th October 2024

Bryan Daverel 32028644

Zacari Harrison 33119635

Hao-I (Benson) Lin 33049386

Charles Nierves 33120447

Emily Ung 33115958

End User Guide	3
UI components	3
Figure 1: Main UI	3
Top Utilities Bar	3
Shape Menu	4
Bottom Utilities Bar	6
Drawing Canvas	6
Manipulating the view of the drawing canvas	7
Toggling Coordinates	7
Zoom in	7
Zoom out	7
Panning the canvas	7
How to input shapes for IoU calculation	7
Manual user input	8
Random shape generation	9
Error handling & state management	11
Undo	11
Redo	11
Delete shape	11
Reset / Destroy progress	11
Save	12
Load	12
Viewing IoU metric	13
Dragging vertices	14
Dragging shapes	15
How to exit the software	16
Limitations of the software	16
Technical Guide	17
Accessing the Repo	
The repository for the project/software can be found from the following link:	
https://github.com/Beees835/IoU-Visualisation-Application/releases	17
Running Built App Releases	17
Accessing the Releases	17
Running The Windows Release	17
Running The MacOS Release	17
Building From Unity	18
Setting up Unity	18
Running a build in Unity	18
Creating a build from Unity	19

End User Guide

UI components

This section will detail all of the different components that make up the User Interface (UI)

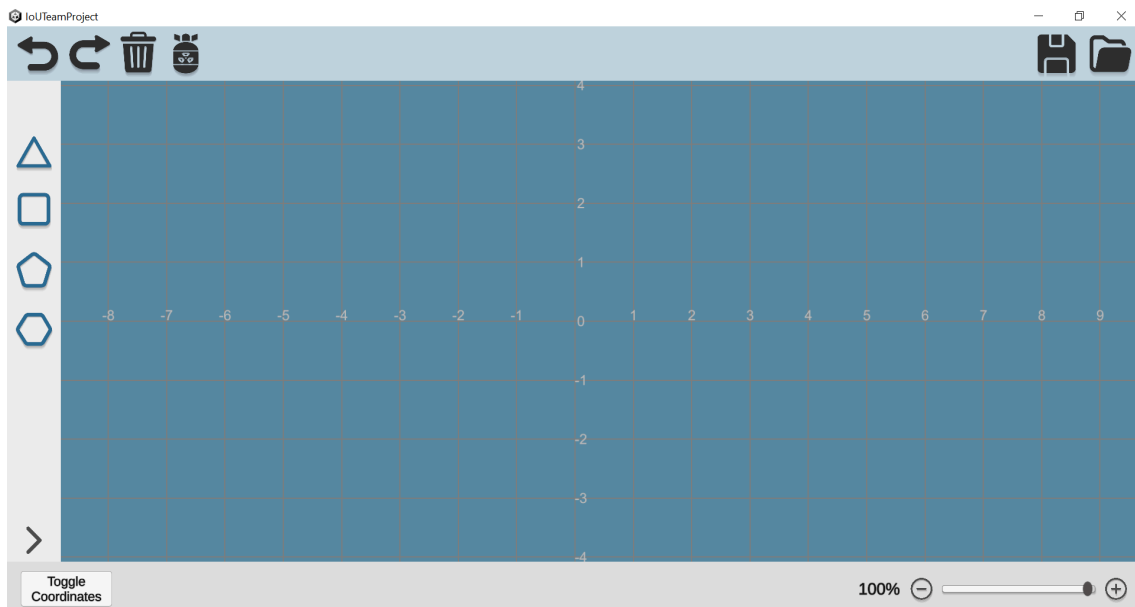


Figure 1: Main UI

Top Utilities Bar



Figure 2: Top Utilities Bar

This bar houses the buttons to manage the state of the application and its canvas.

From left to right, the buttons are:

1. Undo (Undo Icon)
2. Redo (Redo Icon)
3. Delete (Trash Icon)
4. Reset (Bomb Icon)
5. Save Canvas State (Save Icon)
6. Load Previously Saved Canvas State (Open File Icon)

Shape Menu

This menu contains the buttons and functionality for a user to be able to randomly generate a shape. It has both a collapsed version and an expanded version.

Each of the four shapes in this menu are individual buttons. These allow a user to quickly and easily randomly generate a convex polygon with small amounts of vertices.

From top to bottom, the buttons and their functionality is:

1. Triangle Button
 - a. This will randomly generate a 3-sided convex polygon
2. Quadrilateral Button
 - a. This will randomly generate a 4-sided convex polygon
3. Pentagon Button
 - a. This will randomly generate a 5-sided convex polygon
4. Hexagon Button
 - a. This will randomly generate a 6-sided convex polygon



Figure 3: Collapsed Side-Menu

The bottom button (>) will expand the Shape Menu.

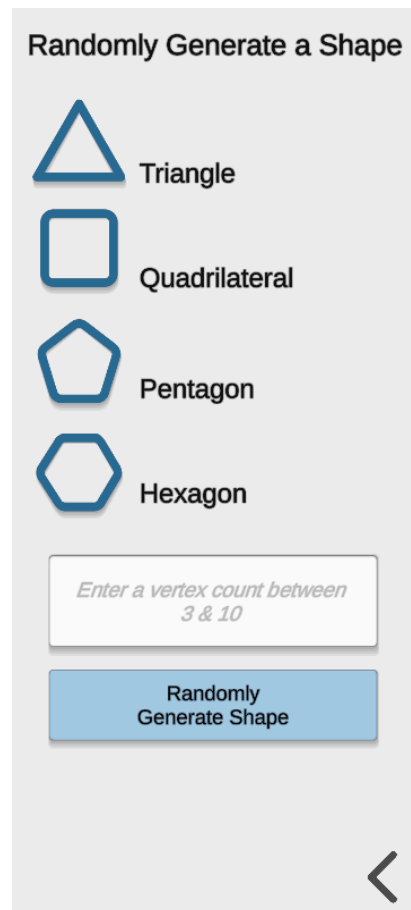


Figure 4: Expanded Side-Menu

Clicking on either the shape icon or their label will allow the user to randomly generate the corresponding shape.

Underneath, there is a user input box that will allow the user to enter a custom vertex count. Only 12.9numbers between 3 & 10 (inclusive) can be entered. This is accompanied by the "Randomly Generate Shape" Button, which must be pressed to confirm the action of generating the shape.

Here is an example of an error message when an incorrect vertex count is entered:

Invalid input: Please enter a number between 3 and 10

Figure 5: Sample Error Message

The button in the bottom right corner (<) will allow the user to collapse the Shape Menu.

Bottom Utilities Bar



Figure 6: Bottom Utilities Bar

The bottom utilities bar contains the controls to change the view of the canvas.

From left to right:

1. “Toggle Coordinates” button
2. The number percentage tells us how zoomed in the drawing canvas is
3. Zoom out button
4. Zoom slider
5. Zoom in button

This is also where warning/error messages will appear in case of invalid behaviour/user input. For example:



Figure 7: Bottom Utilities Bar with Error Message

Drawing Canvas

This is where the shapes can be drawn/randomly generated. It is the main component of the UI and is sandwiched between the Top Utilities Bar and the Bottom Utilities Bar. It is **this colour**.

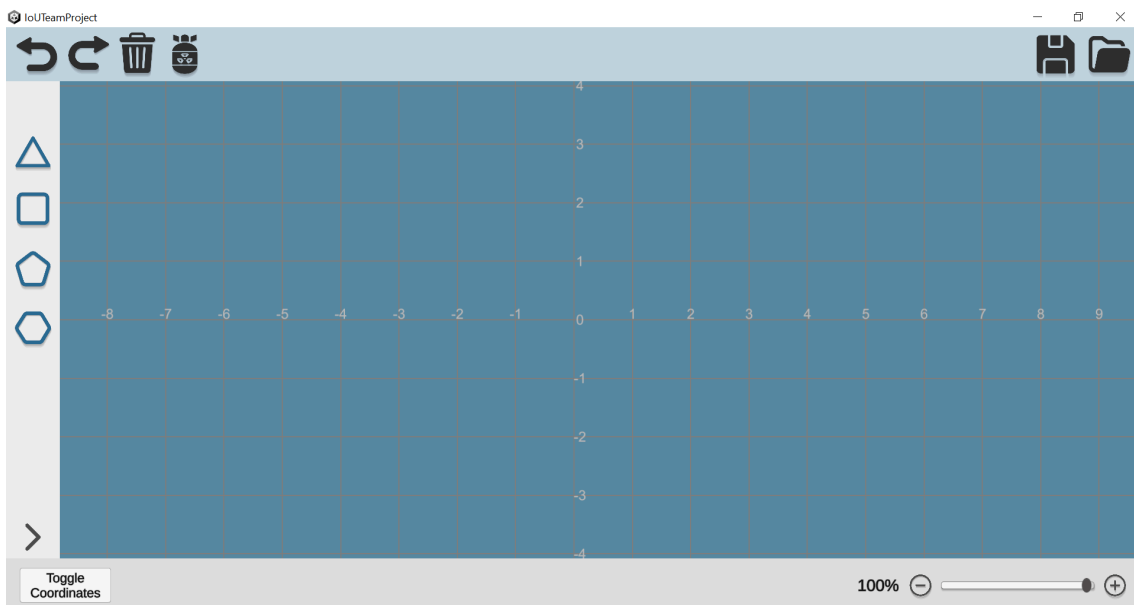


Figure 8: Sample Canvas

Manipulating the view of the drawing canvas

Toggling Coordinates

Clicking the “Toggle Coordinates” button allows the user to turn off/on the coordinate labels on each of the vertices

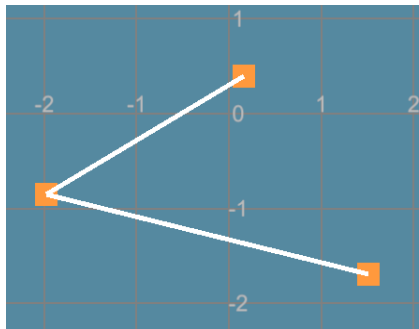


Figure 9: Coordinates example (Disabled)

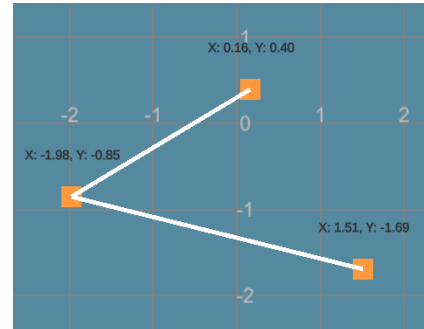


Figure 10: Coordinates example (Enabled)

Zoom in

There are a few different techniques to zoom in on the canvas:

- Pressing the zoom in button
- Moving the zoom slider towards the right
- Pinch and zoom
- Using the mouse scroller

Zoom out

There are a few different techniques to zoom out on the canvas:

- Pressing the zoom out button
- Moving the zoom slider towards the left
- Pinch and zoom
- Using the mouse scroller

Panning the canvas

In case the user wants to view a different section of the drawing canvas, the user can pan by holding down the right click button and dragging the mouse.

How to input shapes for IoU calculation

As the aim of this application is to showcase the Shoelace Algorithm for Intersection over Union (IoU) calculations, users must be able to input different shapes to view the IoU metric. IoU has applications in image detection, where the IoU needs to be calculated between two different shapes.

Users must draw/generate two shapes to be able to see the IoU calculations. These two shapes can be drawn using any combination of manual user input and random shape generation.

Manual user input

- Users can manually click on the canvas to define vertices which will make up a shape.

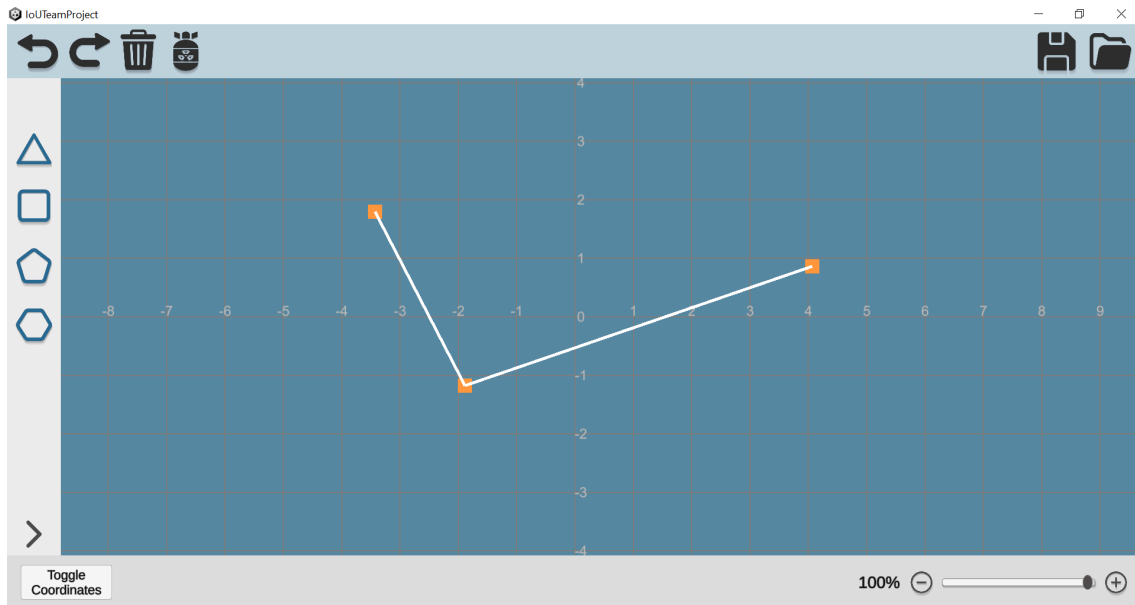


Figure 11: Canvas with Unfinished User-Defined Shape

- In order to close the shape (once the user has defined at least 3 vertices), the user must click on the first vertex again to close the shape. This will tell the system that when the user next clicks on the drawing canvas, the system will start defining the second shape.

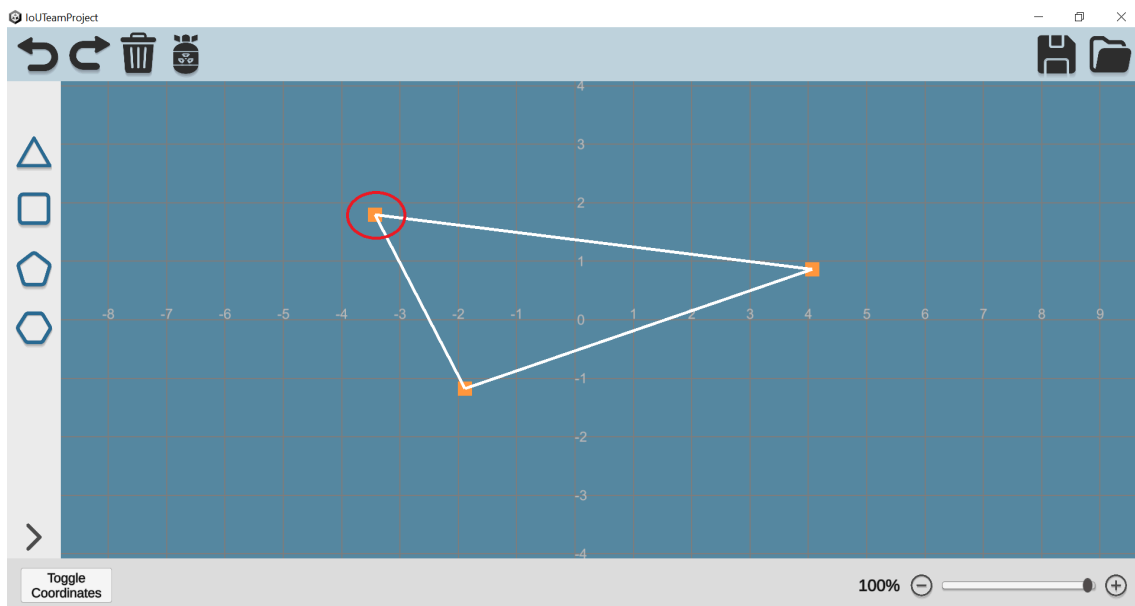


Figure 12: Canvas with Closed User-Defined Shape

- If the user accidentally clicks on a section of the canvas that will cause a concave shape, the user is alerted with a red vertex and an error message on the bottom utilities bar

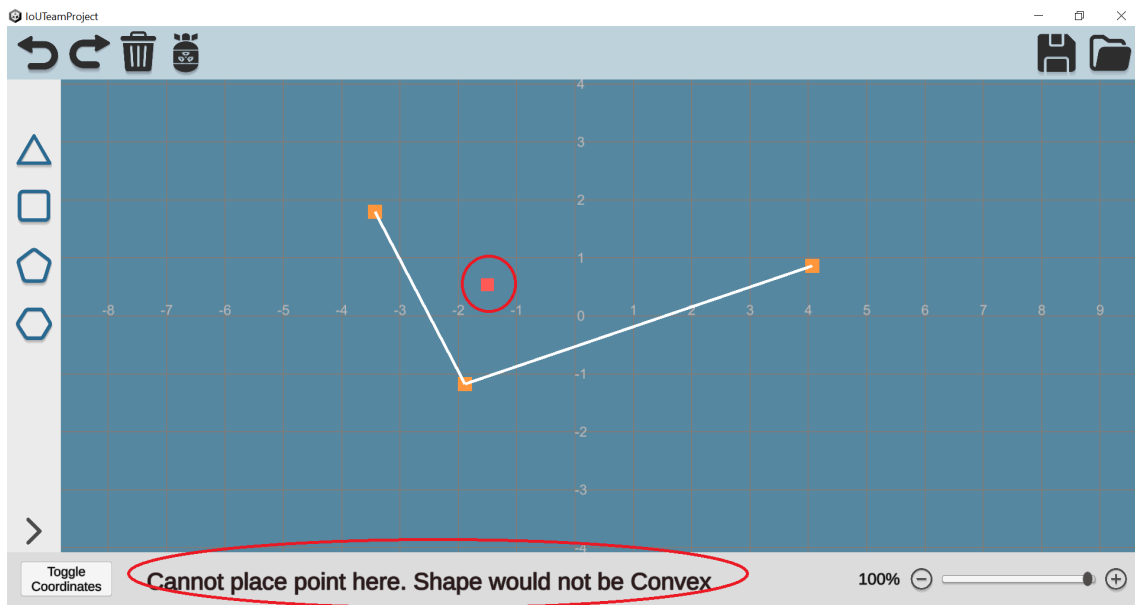


Figure 12: Canvas with Sample Error Message

Random shape generation

For the convenience of the user, 4 different types of convex polygons have their own button. Pressing any of these will place the relevant randomly generated shape onto the canvas.

For example, pressing the Quadrilateral button will place a randomly generated convex quadrilateral onto the screen:

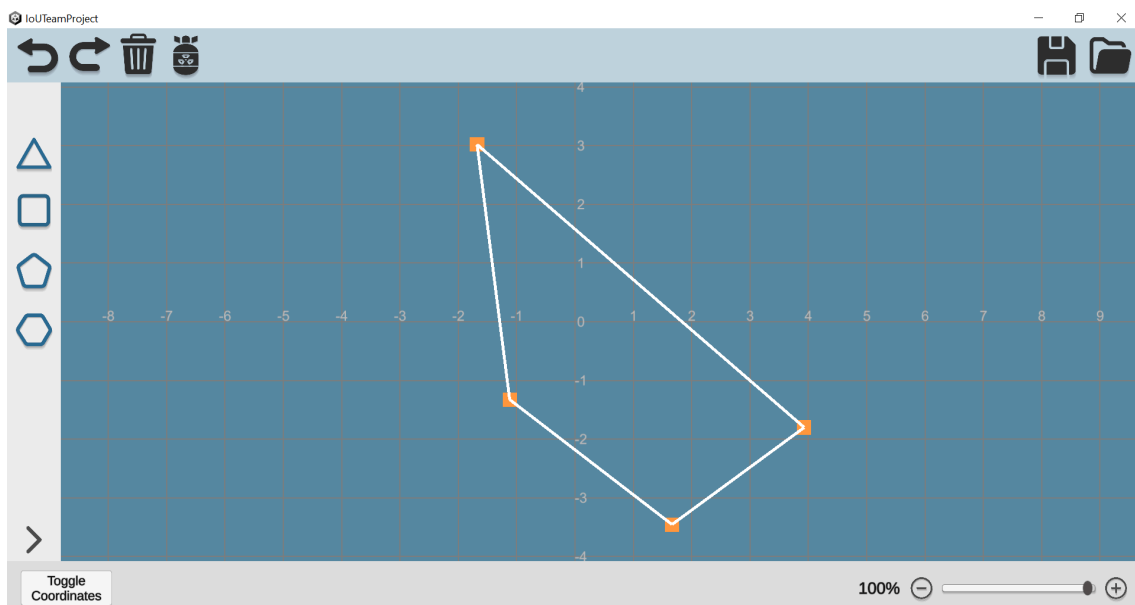


Figure 13: Canvas with Random Shape

In case the user wants to randomly generate a shape with any number of vertices from 3 to 10, they can enter the number in the input box with the placeholder "Enter a vertex count between 3 & 10." To confirm this and randomly generate the shape, the user must press the **Randomly Generate Shape** button.

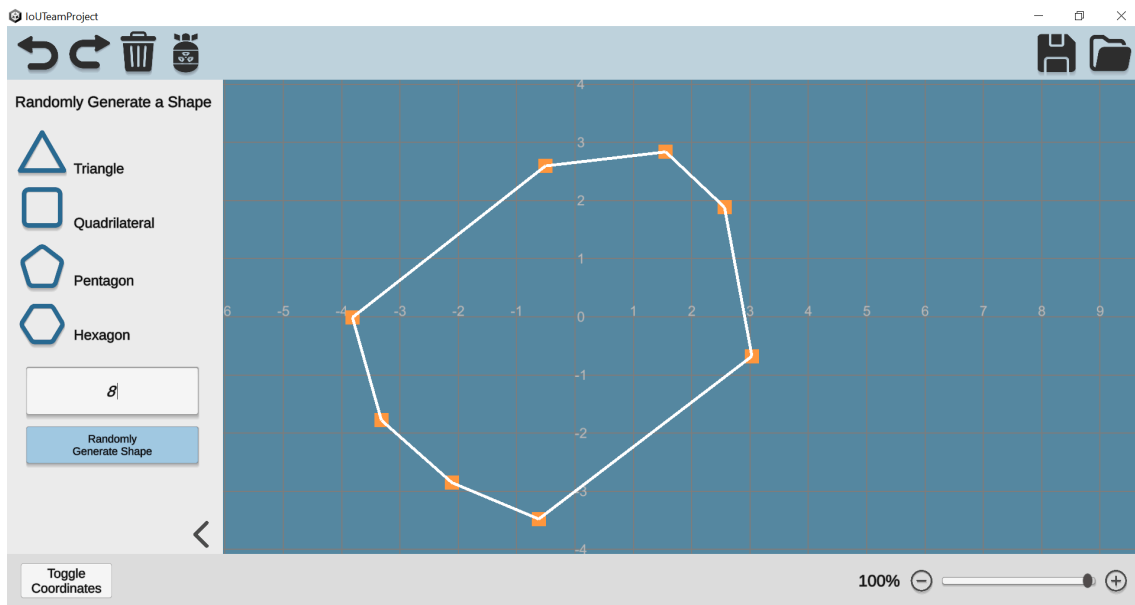


Figure 14: Canvas with Custom-Sized Random Shape

The following error message will appear in case the user attempts to input an invalid number/input:

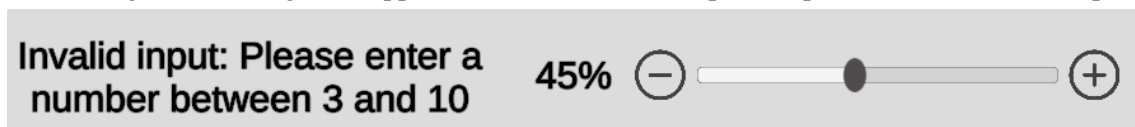


Figure 15: Bottom Bar with Invalid Shape-Size

Shapes cannot be randomly generated while a manually drawn shape is in progress (incomplete). The following error message will be shown in this event:

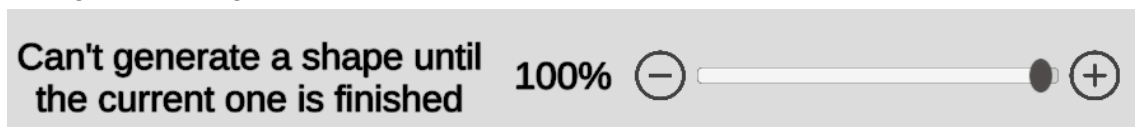


Figure 16: Bottom Bar with Auto-Gen Shape Error

Error handling & state management

Undo

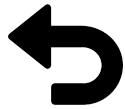


Figure 17: Undo Icon

This will undo the last action (drawing a line, generating a shape, etc)

This can be done by either pressing the Undo Button or using CTRL/CMD + Z keybind

Redo



Figure 18: Redo Icon

If the user has just undone an action, clicking redo will redo that action and display it on the screen (e.g. an undone line can be redone and it will reappear on the screen).

This can be done by either pressing the Redo Button or using CTRL/CMD + SHIFT + Z keybind.

Delete shape



Figure 19: Delete Icon

Once there are two shapes on the canvas, a user can select a shape by double clicking one of its vertices. This will highlight the shape in **orange**. Then, the user can delete the selected shape.

This can be done by pressing either the backspace or delete keys.

Reset / Destroy progress



Figure 20: Reset Icon

This will reset the state of the application and clear all of the user's progress. All shapes will be deleted and this cannot be undone.

This can be done by either pressing the Reset Button or using CTRL/CMD + R keybind.

Save



Figure 21: Save Icon

This will save the current state of the canvas. It can be used to save the state of the canvas before destroying it.

This can be done by either pressing the Save Button or using CTRL/CMD + S keybind.

Load



Figure 22: Load Icon

This will open the most recently saved state of the canvas. Loads can be done after closing and reopening the application, provided there has been a recently saved state.

This can be done by either pressing the Load/Open Button or using CTRL/CMD + O keybind.

Viewing IoU metric

Once two shapes have been drawn/inputted, the relevant IoU information will appear on the top right corner of the screen. The intersection between the two shapes will be highlighted for easy viewing as well.

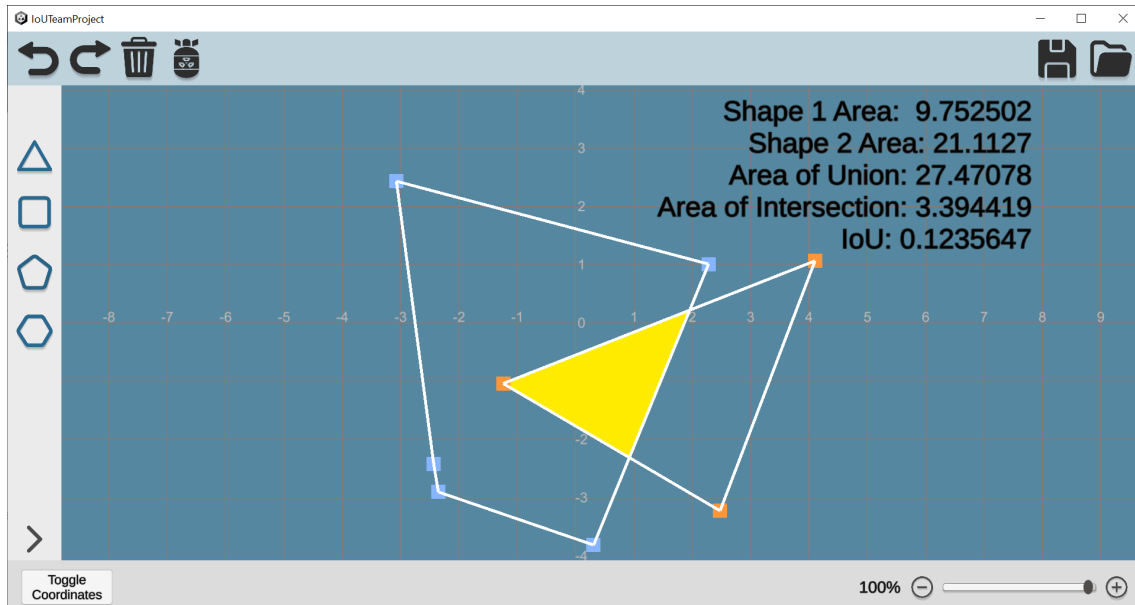
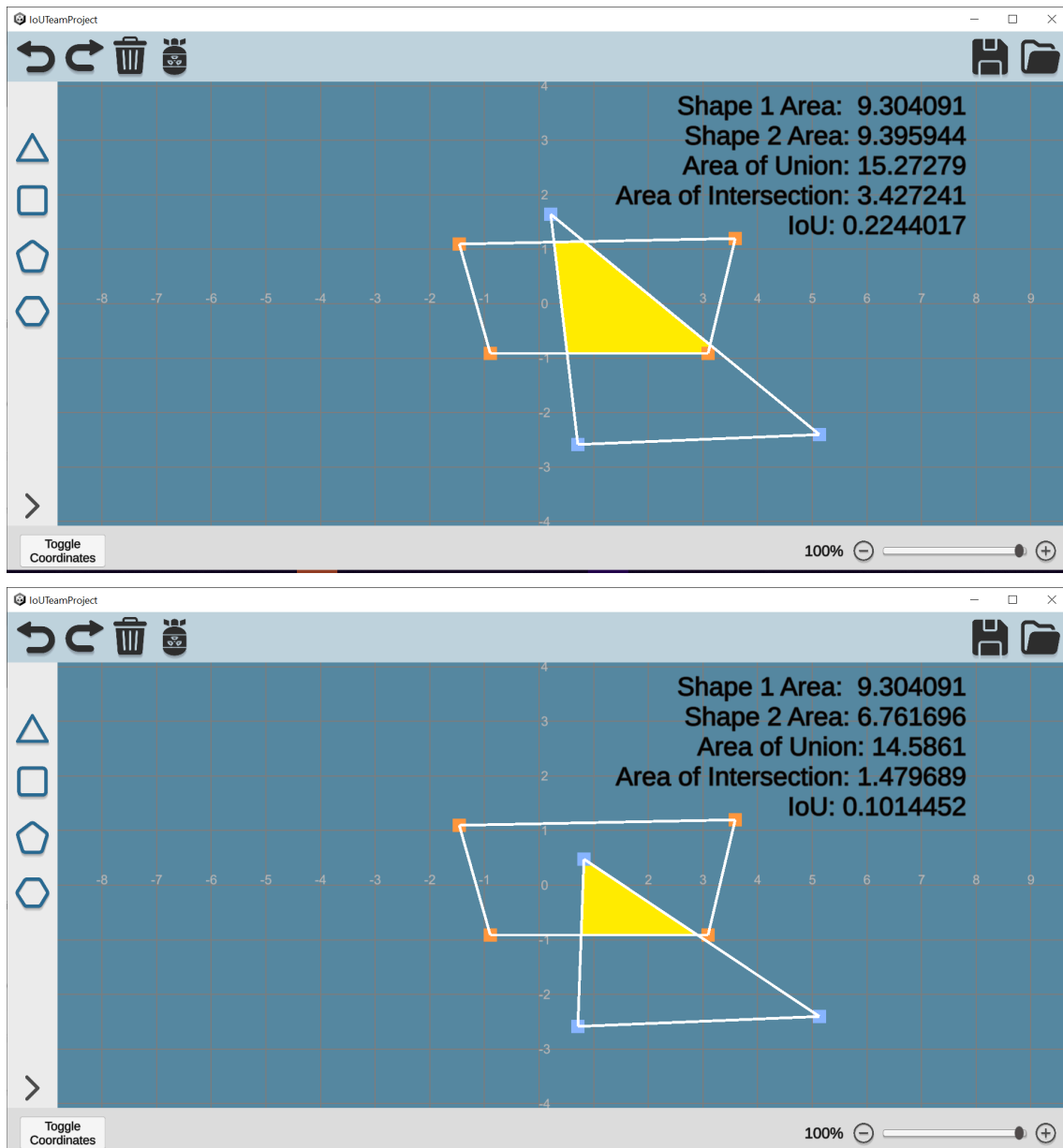


Figure 23: Sample Intersection

Dragging vertices

Once there are two complete shapes on the canvas, the user can click on a vertex and drag it to a different location if they wish. This will update the intersection highlighting and IoU metrics in real time. Users will not be able to drag the vertices to make the shape non-convex.



Figures 24 & 25: Sample Vertex Drag

Dragging shapes

A shape can be selected by double clicking on one of its vertices. This will highlight the shape in **orange**. Once selected, the shape can be dragged around by holding down the cursor and moving the mouse.

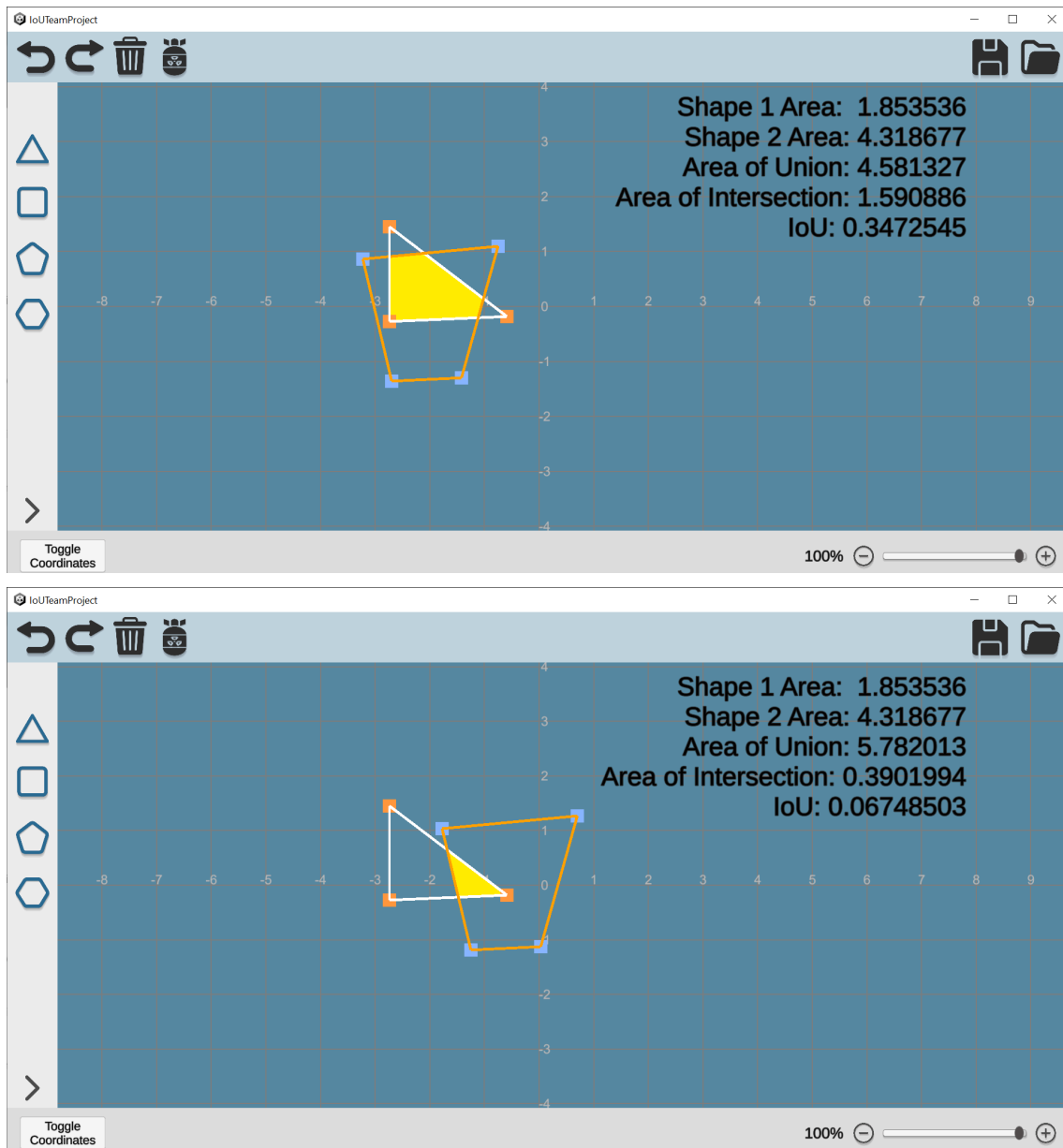


Figure 26 & 27: Sample Shape Drag

This will update the intersection highlighting and IoU metrics in real time.

How to exit the software

This can be done using ALT + F4 Keybind or pressing the close window button.

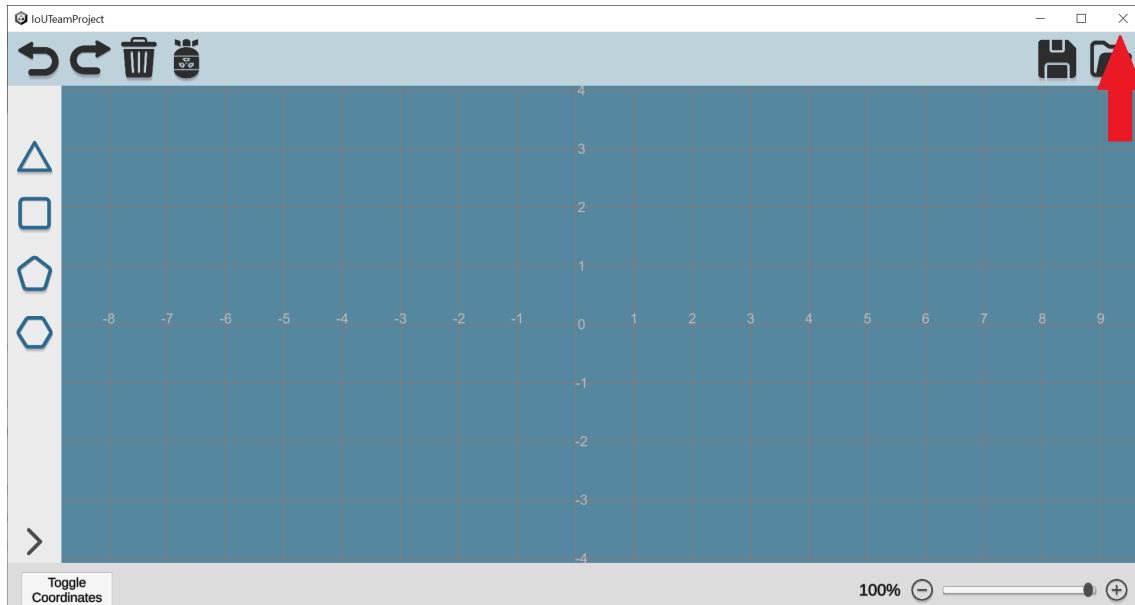


Figure 28: Close Button

Limitations of the software

IoU cannot be calculated for more than 2 shapes, hence the user will not be able to draw more than two shapes on the canvas. Appropriate error messages of “Too many shapes already” or “Cannot add new points when the two shapes are defined” will appear if the user tries to draw a third shape.

This was done as the software’s aim is to showcase IoU for image detection, which is only ever calculated between two shapes.

Technical Guide

Accessing the Repo

The repository for the project/software can be found from the following link:

<https://github.com/Beees835/IoU-Visualisation-Application/releases>

From which the source code can be opened and the releases can be found.

Running Built App Releases

Accessing the Releases

The release versions can be accessed from the following link:

<https://github.com/Beees835/IoU-Visualisation-Application/releases>

Or by clicking on release from the github repository link provided above, from here, the source code can be freely downloaded as well, on top of the two release versions for MacOS and Windows within a zipped file.

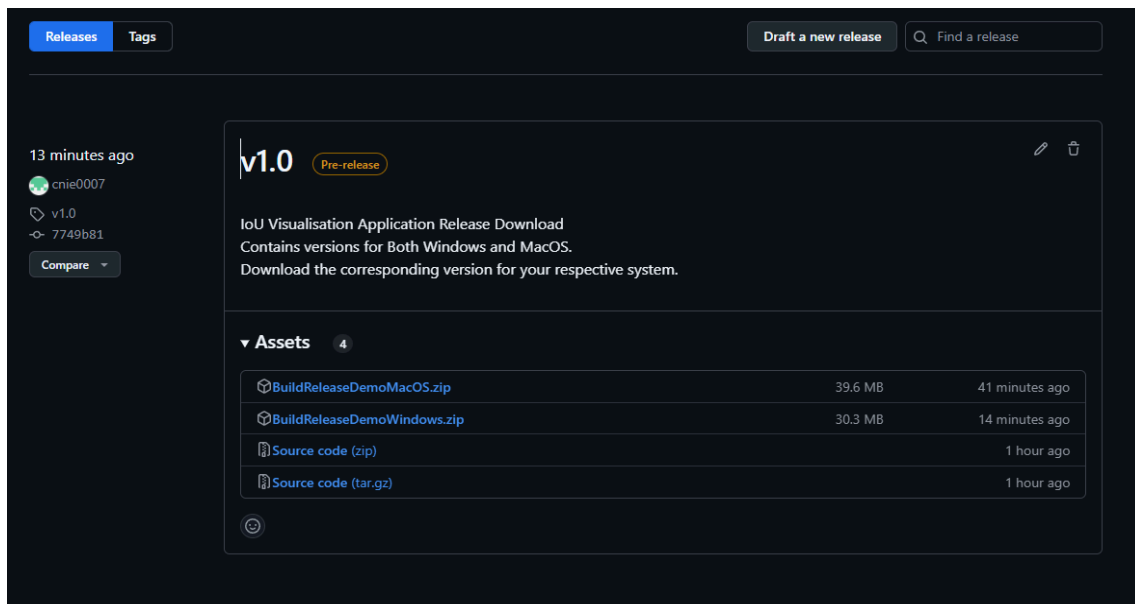


Figure 29: Git Repository Release Window

Running The Windows Release

In order to run the windows release, first download the latest version of the zip file available for windows, and upon extracting, the program can then be accessed by running the IoUTeamProject.exe

Running The MacOS Release

In order to run the MacOS release, first download the latest version of the zip file available for MacOS, and then double click the zip file, this will then extract the IoUTeamProject.app, which would then run the software.

Building From Unity

Setting up Unity

The repository contains the source code for the Unity project. The source code can be opened within Unity (which can be downloaded here <https://unity.com/download>), by installing Unity with editor version 2022.3.40f1 (the version that the program was created in), reducing the chances of any errors from happening when opening the project. Upon creating a fork or downloading the project, the file can be opened by pressing the add button on UnityHub, and finding the folder with which the project is stored in.

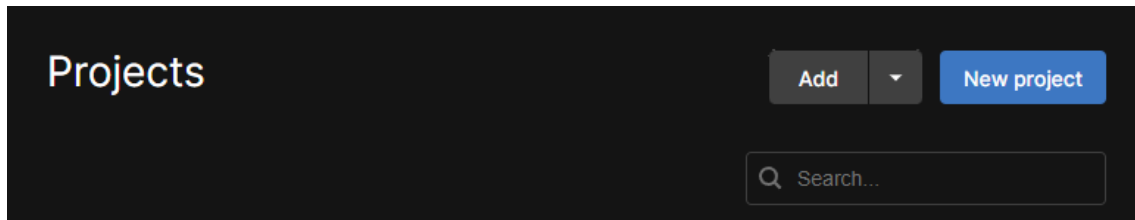


Figure 30: Unity Hub Project Manager

Running a build in Unity

The build can be run within unity by first finding the Sample Scene within Assets > Scenes and double clicking on it, setting it as the current scene, and afterwards, clicking the play button at the top of the scene bar.

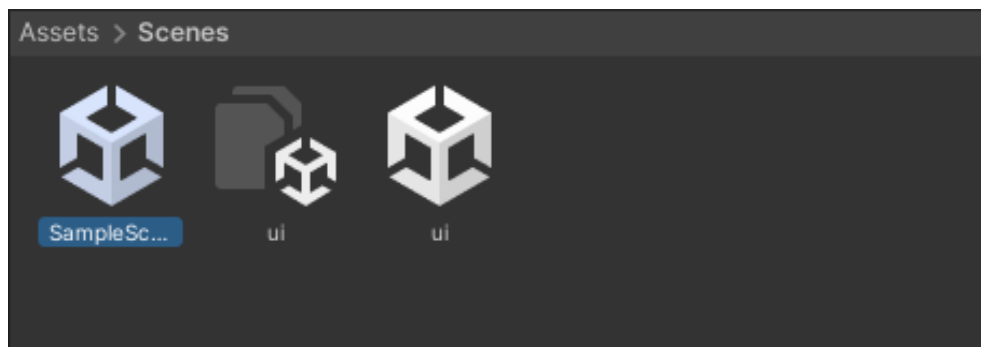


Figure 31: Unity Game Engine Scene Display

Creating a build from Unity

From Unity, the project can be built by going into the menu bar and heading into File > Build Settings... and building it into the desired environment.

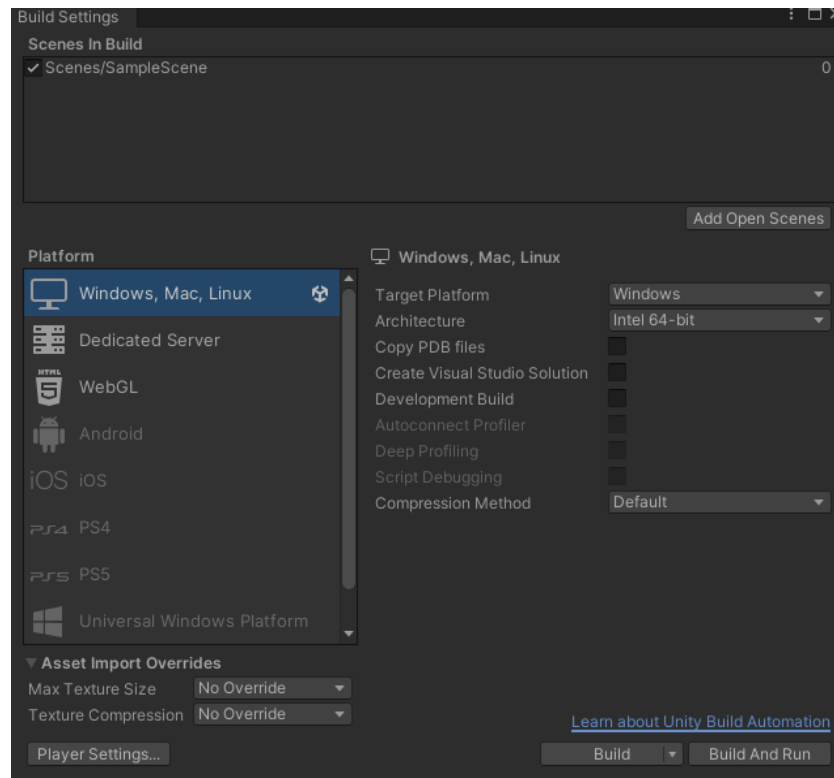


Figure 32: Unity Game Engine Build Window