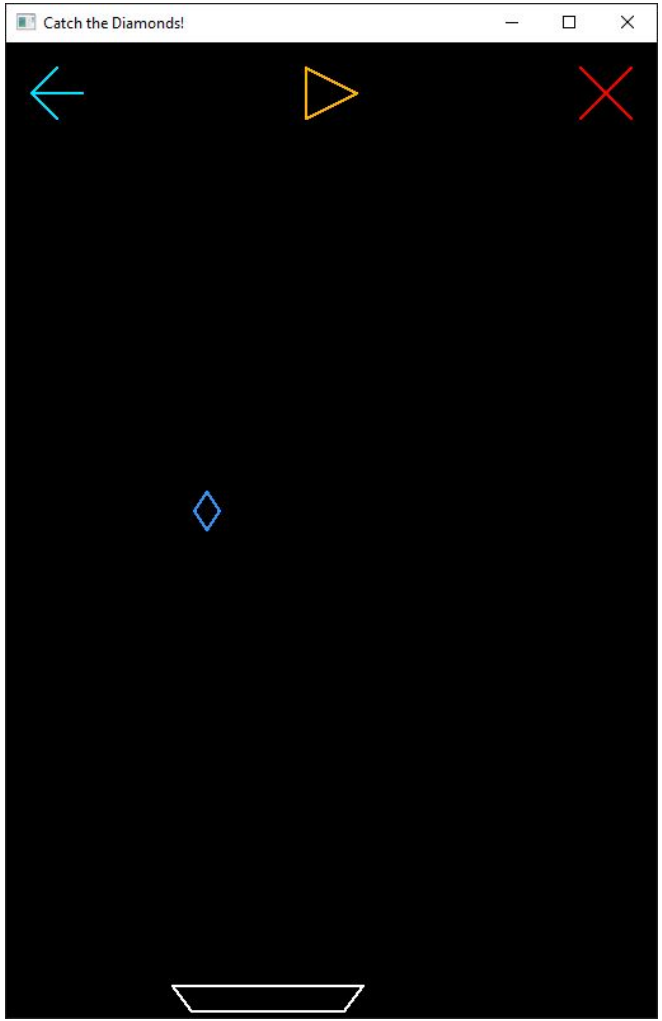
CSE423 - Computer Graphics

Lab 02 (Fall 23)



Catch The Diamonds

Game Mechanics & Implementation Hints

Do not try to solve the problem at once

Subdivide the project into individual feature chunks

Game Behaviour/Events

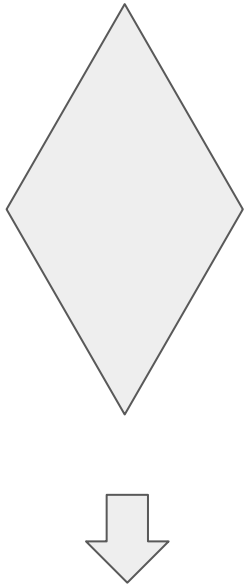
- How do we make the diamond fall spontaneously?
- What happens when user presses left/right keys on keyboard?
- How can we ensure the catcher catches the diamond when they come into contact?
- What if the catch was a failure, and the diamond sinks below the window?
- What happens when the user clicks the area of an onscreen button?
- How do we handle game over event?
- How do we handle if user wants to restart the game?
- How do we make the diamond’s speed increase after each successful catch?

Features Work plan

- Drawing mechanism (using midpoint line algorithm enhanced with eight way symmetry)
- Animation (movement of game objects per frame)
- Keyboard input (handling both ordinary & special keys)
- Mouse input (including coordinate conversion by y-axis flipping)
- Overall game logic

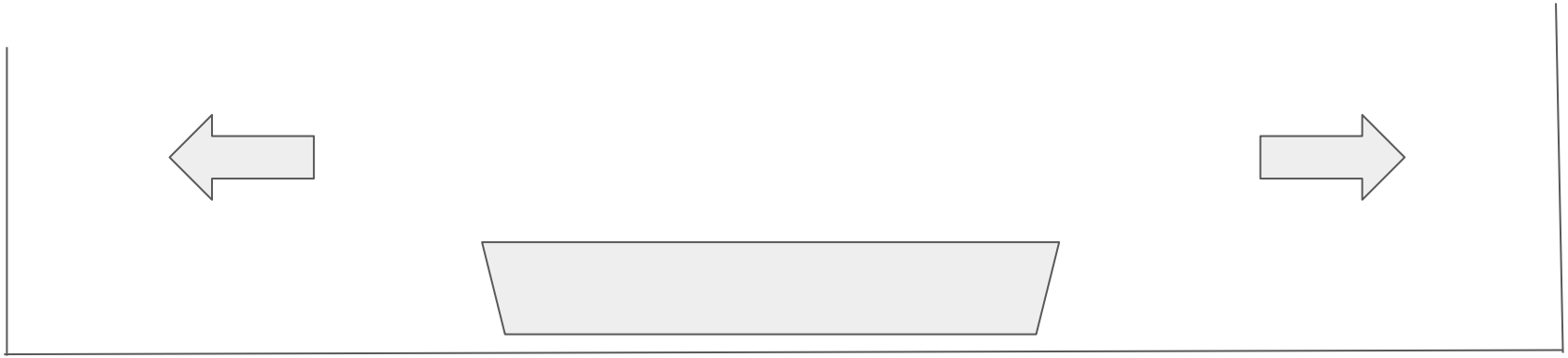
How do we make the diamond fall spontaneously?

- The diamond doesn’t move horizontally, only vertical axis
- Decrease the vertical coordinate of it, should be done every frame
- The decrease is dependent on the falling speed
- The speed also increases if there is a successful catch
- Make sure there is a speed cap



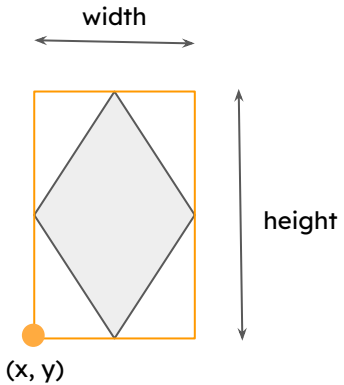
What happens when user presses left/right keys on keyboard?

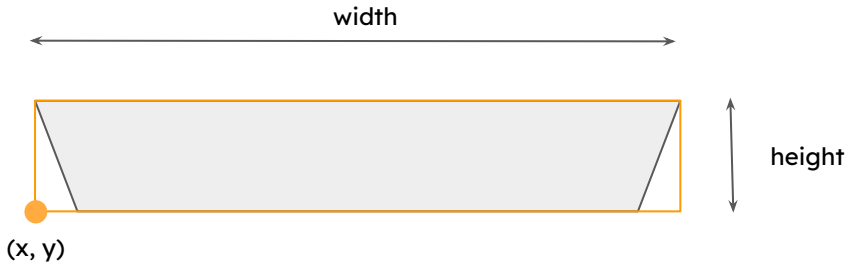
- The catcher only moves horizontally, so only the horizontal coordinate value of catcher gets affected
- Decreases when left key is pressed, increases for right key
- Make sure the catcher doesn't go outside the window while you move it
- Make use of window width and catcher width smartly to tackle this



How can we ensure the catcher catches the diamond when they come into contact?

- Imagine both the catcher and diamond have their respective axis-aligned bounding boxes (AABB).
- AABB of an object has the same coordinates of the object itself, so they move together.
- Four properties (x, y, width, height). Width and Height are constant.
- We can check for collision/intersection of these two AABBs. A collision indicates a successful catch.





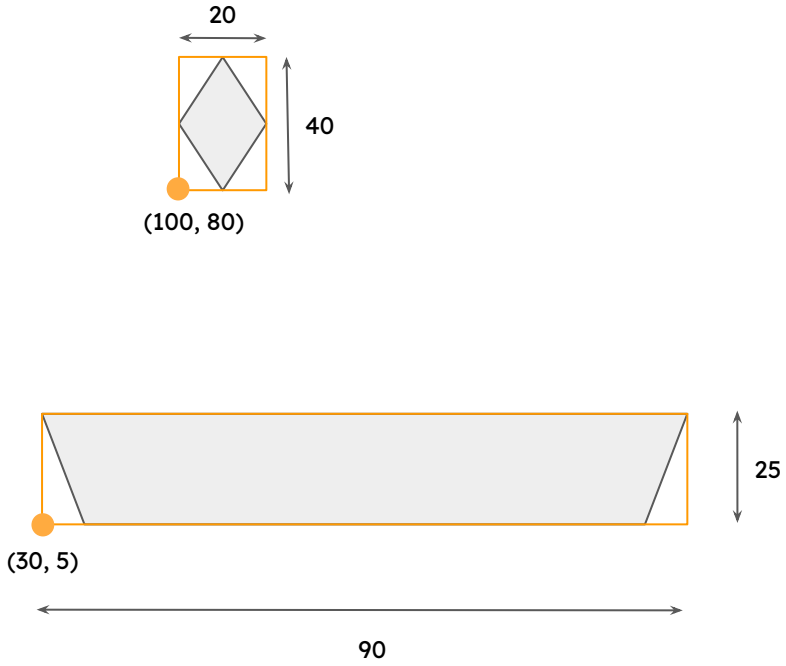
AABB Collision detection algorithm

```
bool hasCollided(AABB box1, AABB box2) {
    return box1.x < box2.x + box2.width  &&
           box1.x + box1.width > box2.x  &&
           box1.y < box2.y + box2.height &&
           box1.y + box1.height > box2.y;
}
```

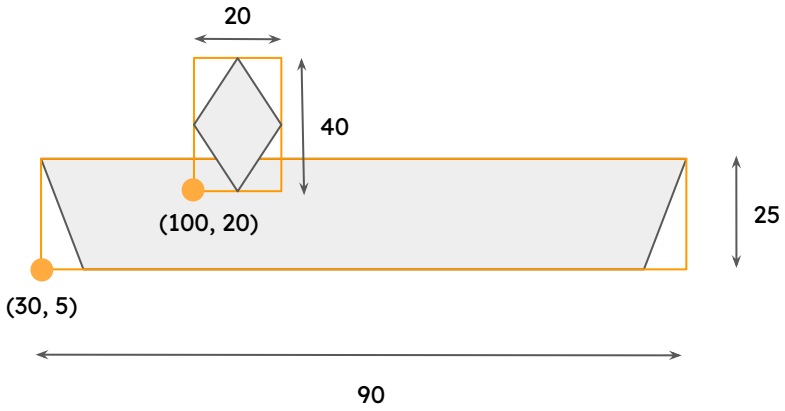
Can be called like this (every frame):

```
catched = hasCollided(diamond_aabb, catcher_aabb)
```

Collision Example



No collision, no catch



Collided, successful catch

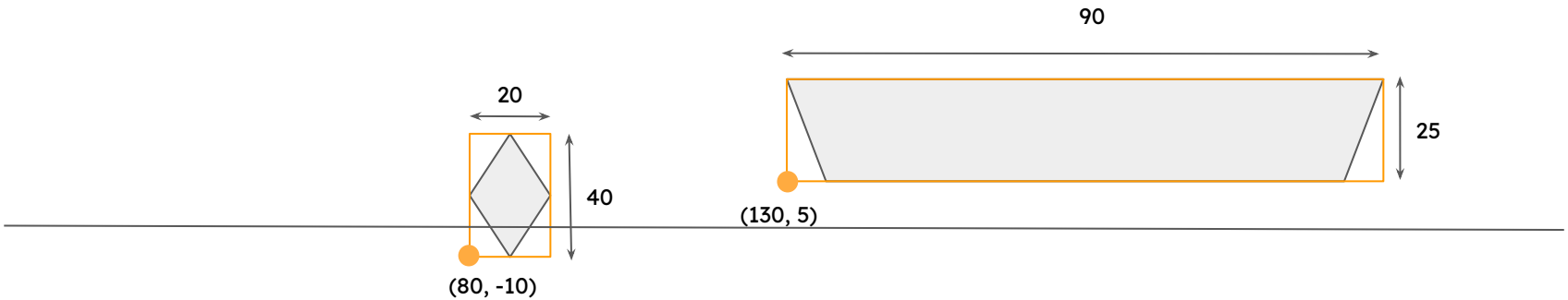
What happens when there is a successful catch?

- Increment score by 1, increment diamond speed by a certain amount
- Print score
- Set the diamond vertical position to the top of screen, and horizontal position to a random value between 0 and window width. Also set diamond color to a random one.

There is actually only one diamond that gets repositioned and recolored after every successful catch, giving the illusion of infinite diamond falling.

What if the diamond sinks below window?

- Game Over happens
- If there was a successful catch, the sinking won't happen.
- But how to detect this sinking? Figure it out by yourself.
- Whatever it is, this sinking checking also needs to be done every frame, just like catch detection



When the Game is Over

- Print game over status along with the score
- Reset the score, speed of diamond
- Vanish the diamond (A.K.A. do not draw it)
- Set the color of catcher to red, and disable the catcher’s horizontal movement

How to detect click over on-screen buttons?

- Buttons also have their own AABBs

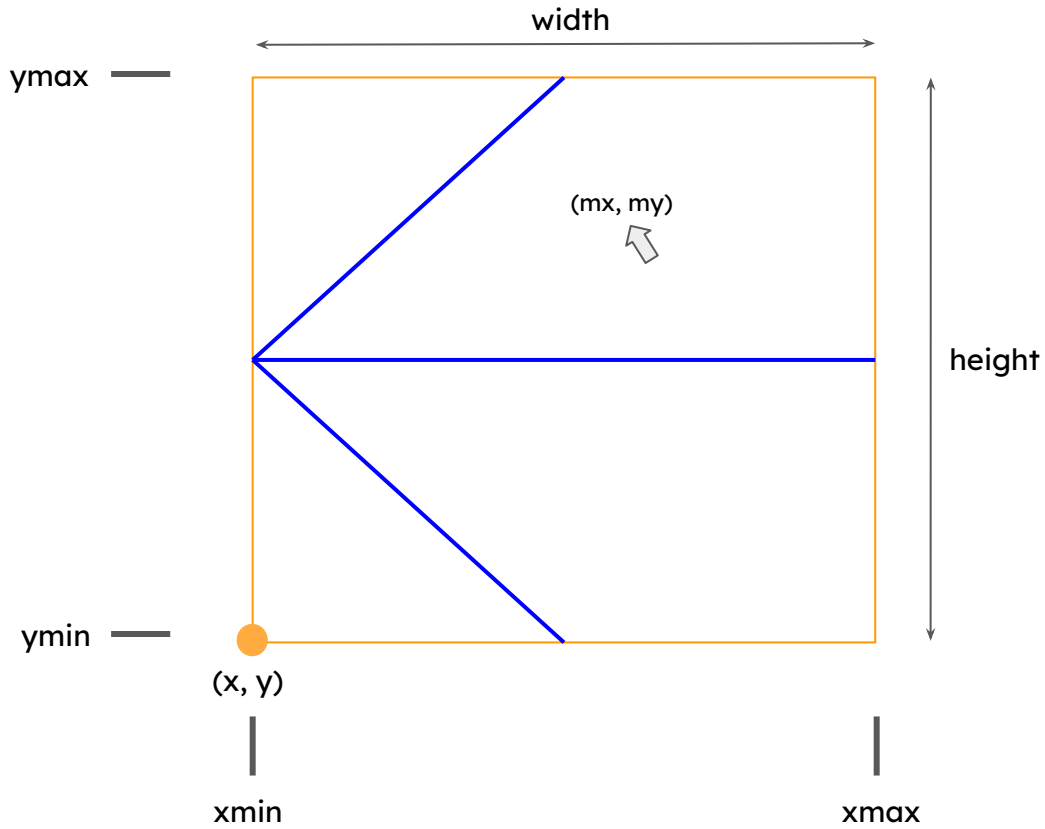
- Calculate axial bounds:

```
xmin = x
ymin = y
xmax = x + width
ymax = y + height
```

- A successful click is registered when the mouse coordinates fall in the axial ranges:

$$xmin \leq mx \leq xmax$$
$$\text{and}$$
$$ymin \leq my \leq ymax$$

(mx, my) are converted (y-flipped) coordinates from mouse input. Obtained from mouse listener



Buttons Miscellaneous

- When user presses the arrow shaped on-screen button, a new game starts. This resets the score, diamond speed & color. The diamond also gets repositioned just like for a successful catch event. Remember, new game doesn't care about whether previous game was running, paused, or game over.
- When user presses the play/pause shaped on-screen button, it toggles the play/pause state. Diamond falling will freeze, and user won't be able to move the catcher. Also make sure to change the button's shape depending on play/pause state.
- When user presses the cross shaped on-screen button, it simply terminates the game. Doesn't matter the state of the game.

Thank You