

As for the submission itself, it is a breakout-style game with a convex bat and a bouncy ball, where the objective is to shoot the ball through green stars. The only controls are keys O and P, moving the bat left and right, respectively. The angle at which the ball is going to bounce off the bat depends on the location where the ball hits, as well as the velocity of the bat at that moment. Once a green star is hit, the ball's direction changes randomly. There is only one life, if the player allows the ball to fall off the bottom of the screen, the score is zeroed.

The code:

```
10 BORDER 6: CLS : LET s=0: LET x=RND*31: LET b=INT (x*28/31): LET
y=RND*21
20 LET u=2*(RND-.5): LET v=2*(RND-.5): PRINT AT RND*21,RND*31; INK
4;"*";#1;AT 1,0;s;
30 LET m=(INKEY$="p" AND b<28)-(INKEY$="o" AND b>0)
40 PRINT #1;AT 0,b-(m<0);" " AND m>0;" ████████";" " AND m<0;: LET b=b+m
50 IF (u<0 AND x<0.5) OR (u>0 AND x>30.5) THEN LET u=-u
60 IF v<0 AND y<0.5 THEN LET v=-v
70 IF y<20.5 OR v<0 THEN GO TO 100
80 LET d=x-b-1.5: LET u=u+.2*d+.3*m: LET v=-v*(2-ABS d)-.4
90 LET r=1/(ABS u+ABS v): LET u=r*u: LET v=r*v: IF ABS d>2 THEN RUN
100 PRINT AT y,x;" "; LET x=x+u: LET y=y+v: LET h=ATTR (y,x)=60:
PRINT AT y,x;"©";: LET s=s+h: GO TO 30-10*h
```

Explanation:

Line 10 is the initialization, runs only in the beginning and after every live lost. It turns the border yellow, clears the screen, resets the score (s), places the ball in a random location (x,y) on the screen and the bat (b) underneath it, so that there is no immediate loss of life.

Line 20 sets the velocity vector (u,v) of the ball randomly, puts the green star at a random location and prints the score. It runs only in the beginning, after a life lost or a star scored.

Line 30 reads the keyboard and calculates the velocity (m) of the bat to be -1, 0, or +1, depending on which keys are pressed and whether it has already hit the edge of the playing field.

Line 40 draws the bat and erases the remainders of its previous locations, if it has moved. Also, it updates its location (b) given its velocity (m).

Line 50 bounces the ball off the left and right side of the playing field.

Line 60 bounces the ball off the top of the playing field.

Line 70 checks whether the ball hit the bottom and skips line 80, if it has not. Note that originally line 70 and 80 were in one line, but I split them to fit into the 80 character limit as defined above.

Line 80 calculates the horizontal offset (d) between the center of the bat and the ball. Then it updates the horizontal and vertical velocities of the ball according to this offset.

Line 90 normalizes the velocity vector of the ball. If the bat-ball offset is too large, the ball is considered lost, the program is restarted.

Line 100 removes the ball from the screen, updates its position with the velocity, checks whether (h) there is a green star under the new position and places the ball in the new position. Then it updates the score if a star was hit or not and returns to either line 20 or 30, respectively.