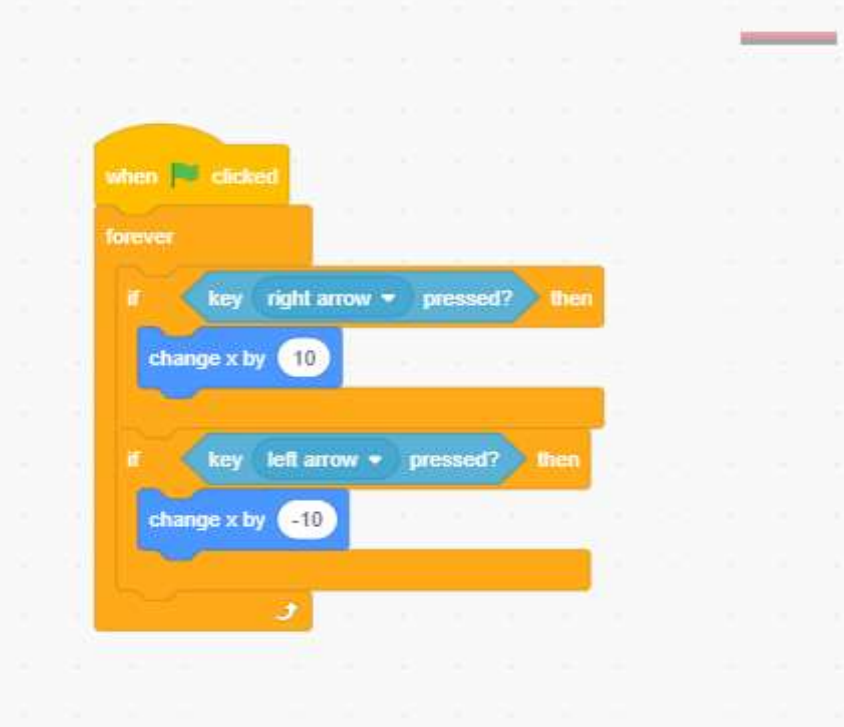
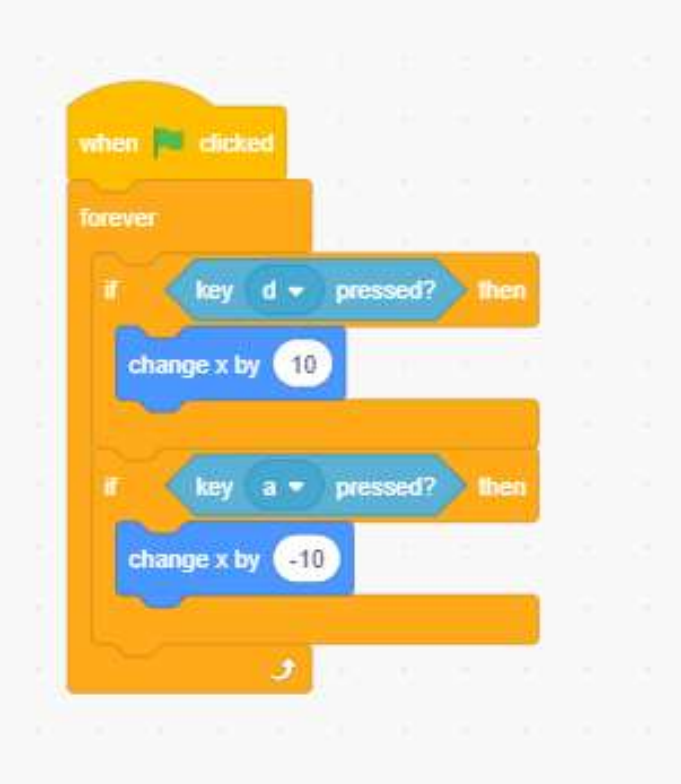
PIN POOM BALL

REDBAR:

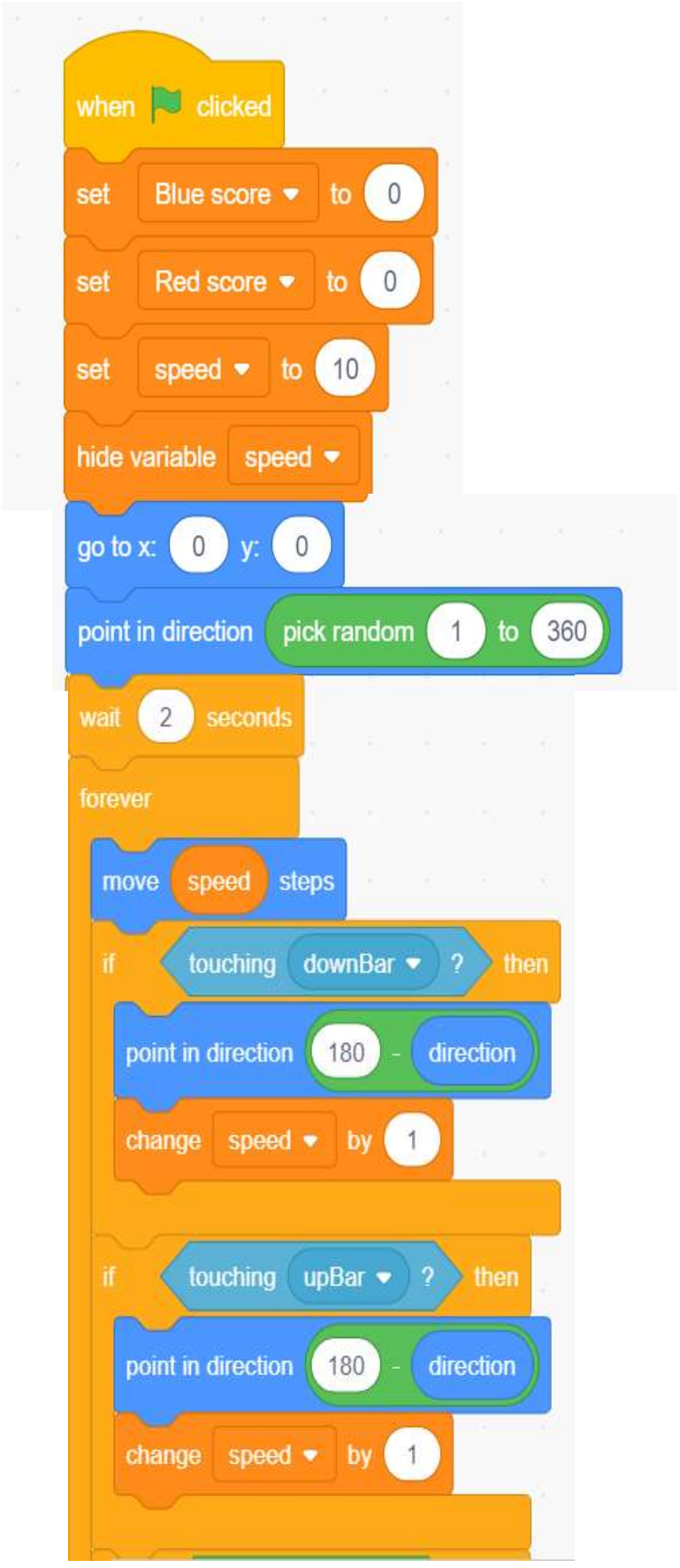


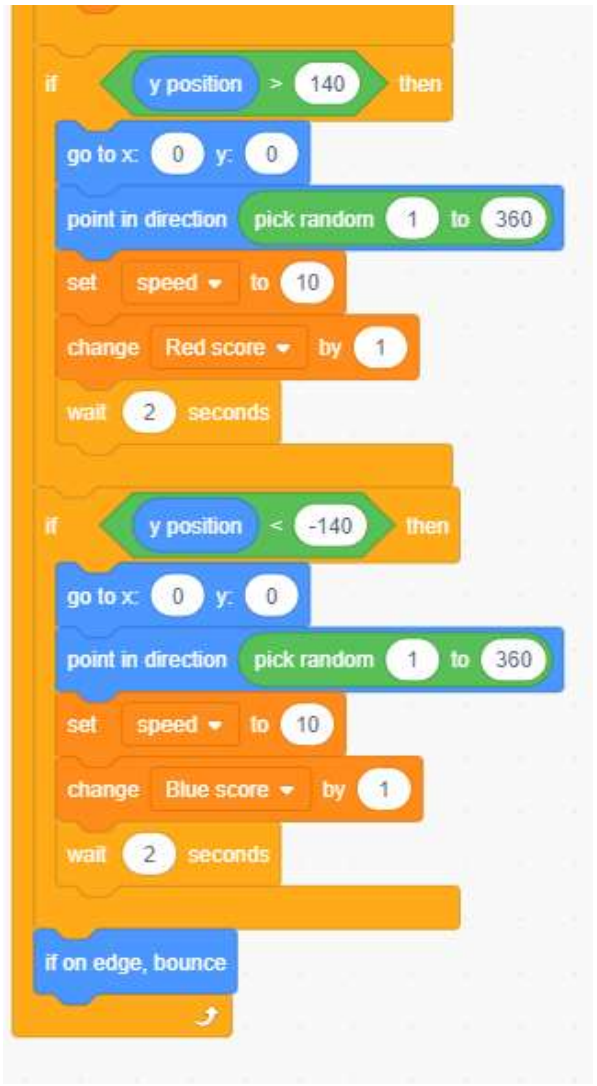
BLUEBAR:



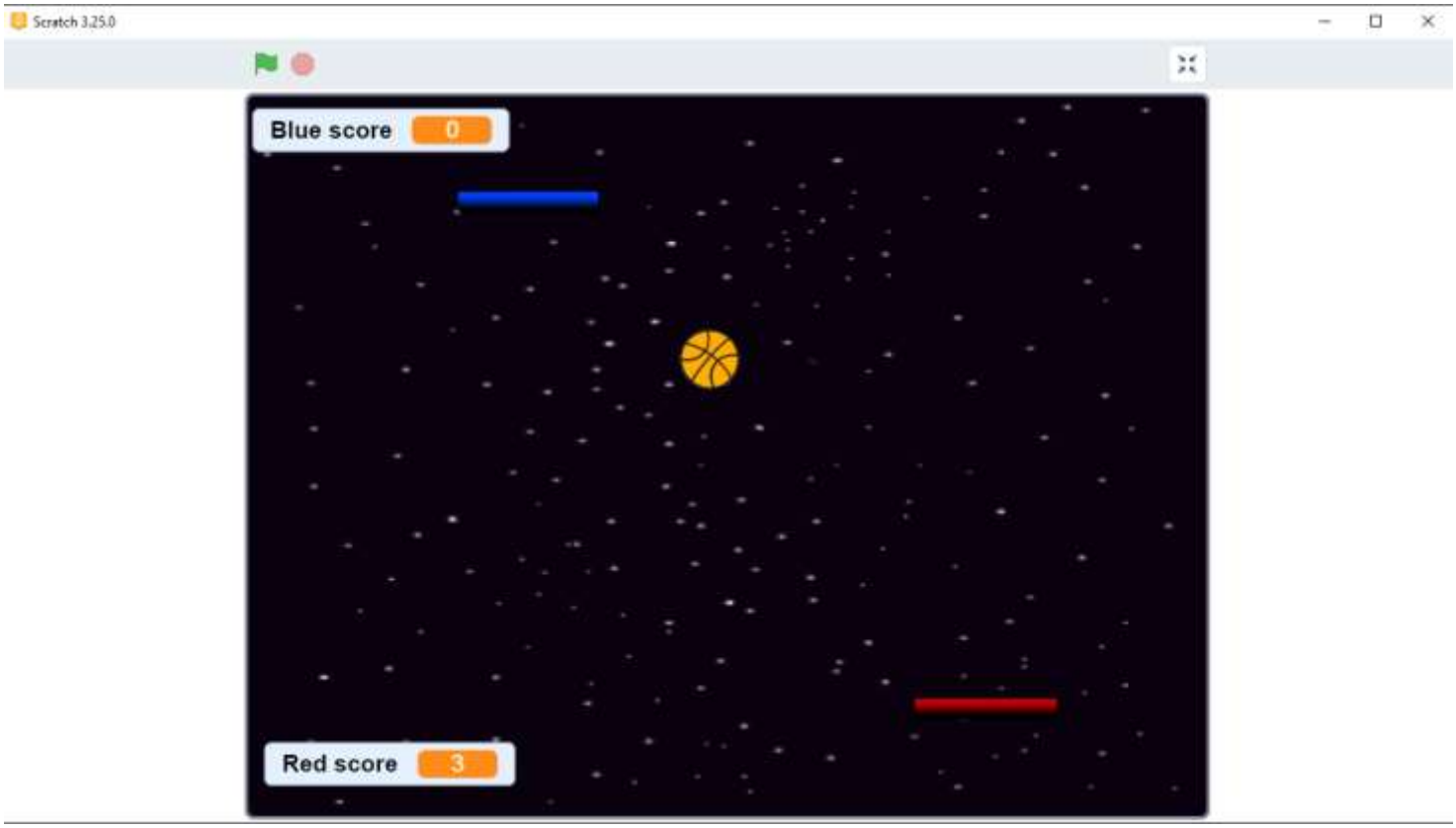
PIN POOM BALL

BASKETBALL:





PIN POOM BALL



PIN POOM BALL

PROCEDURE:

Welcome back.
It's time to finish up the scheme by implementing scoring and the resetting of the ball when it passes
either one of the platforms.
So let's get on with it.
I'm going to start by resetting the ball in the dead center of the screen and pointing it in a random direction when it passes, either the blue bar on to the top.
So when it goes upwards up until maybe here or it goes down up until maybe here.
All right.
So we will implement a small if condition that if the Y coordinate, that is the vertical coordinate of the ball is greater than something 160 or something like that or less than -160.
Then it will reset to the center of the screen and pointing it in a random direction.
So I'm going to go to control.
And I'm going to bring in the if block.
And in the diamond shaped hole for the F block.
I'm going to go to operators and I'm going to bring in this diamond shaped, greater than and in the
first space of the diamond shaped block, I'm going to go to motion and I'm going to bring in the Y
position.
That is the vertical position of the Sprite.
And in the right space, I'm going to put in 140.
So if the Y coordinate of the basketball is greater than 140, so up until here, we will reset the ball to its original position, the center of the screen X zero and Y zero, and pointed in another random direction.
So I'm going to.
Detach the forever block.
Right click on the first blue block.
Go to X and Y.
Hit duplicate.
So this gives me the two blocks to reset the ball and I'm going to put them inside the if block.
I'm going to reattach the forever block again.
So if I move the basketball right up top of the stage, so the coordinate is 150, 155 and click this detached if block the ball will reset to the center of the screen.
See now I'm going to right click on this.
If like it, I'm going to duplicate it because I'm going to replicate this condition for the down position
as well.
So if the Y position in this case is less than -140, which means down here, something like that, I'm going to do the same thing, reset the ball to the center of the screen and pointed in a random direction.
So I'm going to take the second diamond shaped green block and I'm going to remove it and I'm going

PIN POOM BALL

to bring in another one.
So I'm going to go to operators and I'm going to pick in the less than operator.
I'm going to go to motion.
Scroll down.
Take the rounded y position and in the second space, I'm going to fill in -140.
Okay.
And then I'm going to click on this block to make sure that the ball can reset to its original position.
All right.
Now, what I'm going to do is I'm going to take these two if blocks and I'm going to put them right
before if on edge bounce.
So I'm going to put them inside the forever loop.
So if an edge bounce is going to be the last block.
All right.
So the code here has gotten quite big.
So the basketball now has a lot of conditions and moves ten steps.
It checks if it touches either the down bar or the upper.
And it also checks if it's too far up or too far down, in which case it will reset to its original position.
So if I hit the flag again, the ball should reset to the center of the screen on its own.
See.
So it bounces off of the edges of the screen.
It also bounces off of the platforms.
But if it's, in any case, too far up or too far down on the stage, it will reset to its original position and go in another direction.
All right.
So this is cool.
All right.
So the ball can reset to its original position.
It can bounce off of the edges of the stage.
And it can also bounce off of the platforms, which is very, very nice.
There is one last thing that we need to implement, and that is scoring.
For that, we will need to learn something completely new and that is variables.
So I want you to go two variables, this dark orange section from the code and I want you to click on
this button, make a variable.
I'm going to name this variable read score.
Check that for all sprites checkbox.
And I'm going to make another variable called Blue Score again for all sprites and click.
Okay.
So what is a variable?
Well, a variable is just a piece of information that can be changed in time.
And we will programmed this variable to be changed when the basketball either goes too far up or too
far down.

PIN POOM BALL

If it goes too far up, we will increase the red score.
If it goes too far down, we will increase the blue score.
And that is basically what a variable is.
So I will need that when the flag is clicked.
We need the blues score to be zero as well as the read score because that's the beginning of the game.
And when the Y position of the basketball goes too far up, we will increase the Reds score by one.
And if it's too far down, we will increase the blues score by one.
So here's what I'm going to do.
I'm going to take this orange set, blue score to zero.
And I'm going to pull it into the open space.
I'm going to duplicate it.
And in the second block I'm going to select Read Score.
So we are setting both the blue score and the red scores to zero.
And I'm going to pull them both below when flag clicked.
So this is the first thing that the computer will do.
When the flag is clicked, we will reset the blue and red scores to zero.
Now, when the white position is bigger than 140, that means the basketball is too far up.
We need to increase the red score by one.
So I'm going to pull this orange change score by one.
And I'm going to pull it inside the if y position greater than 140.
And obviously, I'm going to select Red Square because that's what's going to change.
By one, if the white position is bigger than 140, then the red score deserves a point.
Then I'm going to do the same for the blue score by pulling the orange block in the other.
If we position less than -140 and this time change blue score by one is correct because the blue scored
a point.
And then just for visual reasons, I'm going to put in these dashboards for the red score to be in the
red space and the blue score to be on top in the blue space.
Okay.
And that is basically the game.
Let's hit on the green flag and let's play a little bit.
So we have blue scoring like crazy.
That's because we aren't actually playing.
So we have a fully functional game where both players can.
Catch the ball and make it reflect into one another.
All right.
So we have to play a game.
They can actually score points now.
All right.
So I stop the game and I have to confess that I did play in the meantime a little bit.
So it's quite fun.
But there are a couple of things that I would like to improve on this game before we wrap up, because

PIN POOM BALL

there are a number of things that I'm annoyed about.
First of all, notice that when the ball starts in the middle of the screen and scores a point for one of the players, it usually scores multiple points because the players don't have the time to adjust.
Let me show you.
So if I hit the flag and one of the players, let's say, scores a point.
Then the ball restarts so, so quickly.
See that the players don't have the time to adjust.
So I would like to make the ball wait for a couple of seconds here on the center of the screen so that
the players have the time to prepare.
All right.
So one of the things that I would like to improve here is to make the ball wait for a couple of seconds.
So here's what I'm going to do in the basketball sprite.
Of course, I am going to add a weight block when the ball resets to its original position.
So I'm going to go to control the orange section and I'm going to bring this very first weight for 1 seconds.
BLOCK And I'm going to pull it right before the forever loop, and I'm going to make the wait time to
2 seconds.
So when the game starts, the ball will stop in the center of the screen for 2 seconds and give the players time to adjust.
And I'm going to do the same when the ball restarts in its original position.
So here in this if condition and in the last condition as well.
So I'm going to pull this weight 1 seconds block right here in the if y position greater than 140.
Make the wait time, 2 seconds as well and bring it again in the last if condition and as before, wait
for 2 seconds.
So let's see the difference now.
So if I hit the flag again, notice that the ball waits for a couple of seconds before it starts shooting.
And again, when one of the players scores a point, which is exactly what we wanted.
The second thing that I would like to add to this game is a little bit of challenge.
Because if the ball moves at the same speed, it can get pretty boring after you play with your friends
for 5 minutes or something.
So let's add a little bit of fun.
Let's say that when the ball touches one of the platforms, it might increase its speed, so go a little faster, which makes it an extra challenge for the other player to catch it.
If they do catch it, then it becomes a greater challenge for the other player to catch the ball because
it starts rolling at ever greater speeds.
So I would like to add this challenge by implementing an extra variable for the ball and make it go
ever faster when it touches one of the platforms until it goes so fast that one of the players has to lose.

PIN POOM BALL

All right.
So here's what I'm going to do.
I'm going to go to this variables section, the dark orange.
Now I'm going to make an extra variable.
And I'm going to name speed.
And this will be a private variable for this sprite only.
This is for the basketball sprite I'm going to hit.
Okay.
And I'm going to set the speed of the basketball to be the number of steps that we now move the ball
by, which is ten.
So I'm going to.
Bring this orange set block and instead of blue score, I'm going to set the speed to be ten.
All right.
And for the ball, instead of moving ten steps, we're going to move it by speed steps.
So speed will be the number of steps that the ball will move on the stage.
So until now, nothing has actually changed.
We created a variable whose value is ten and we're moving the ball by speed steps, which is ten steps
as it was before.
But the nice thing about this variable is that we can change it with time.
So when the ball touches one of the platforms, we might want to increase the speed.
So let's say that in these if blocks in, if touching down more and if touching upper, we might want
to increase the speed.
So I'm going to.
Add this orange change blocks or change instead of blue score, I'm going to select speed.
So change speed by one.
That's that is we're increasing the speed of the ball and I'm going to pull it again in the other if touching upper condition.
And again we're going to select speed.
So if the ball touches either the down bar or the upper, it will change its speed by one that it will increase the speed.
So the next time it will touch one of the bars, instead of speed equals ten, speed will become 11.
So the ball will move even faster on the screen.
Before we test, let's hide this speed variable for the basketball because I don't really want it on the stage.
I only care about the scores.
So let's bring this orange hide variable.
I'm going to select speed and I'm going to pull it right after the other orange blocks blow when flat
clicked.
All right, so when we hit the green flag, the speed variable will be hidden from the stage.
All right, let's test it.
All right, we have the ball.
And we're playing.

PIN POOM BALL

And if you notice.
The ball is accelerating, so it's moving ever faster.
So this is the fun of the game.
All right.
It's going so fast until Red has lost.
But I'm not sure if you noticed, but Blue scored two times because the ball shot towards the red zone
so fast that the red player did not have time to adjust.
That is because we are not resetting the ball speed back to ten.
When one of the players scores a point.
So we need to fix that.
So again, in the basketball sprites code, I'm going to play in the orange set and I'm going to select speed.
210.
And I'm going to pull it where the ball resets to its original position.
When the ball resets to its original position, it also needs to adjust its initial speed.
So I'm going to pull it right here under the if white position greater than 140.
And I'm going to pull it again.
Under the if one position less than -140.
So set speed to ten.
All right.
So this makes sure that the ball resets to its original speed when it restarts the round of the game.
So if I hit the flag again and try to play a couple of times.
Let me actually try to hit the ball.
But it goes so fast that I don't have time to adjust.
Okay, so we see the ball accelerating and when the new round starts, it restarts with its original speed, which is cool.
This is exactly what we wanted.
All right.
So at this point, we have a fully functional game.
And I have to say, I am so proud of you.
You created your first full game that you can play with your friends now and you've learned a lot.
We've learned about coordinates and directions and degrees in motion and variables and so much more.
And I'm really excited to show you more games and applications in this course based on what we learn
here.
In the meantime, have fun with the game that you created and modify it as you like.