

## Teleport Mechanic Documentation

**Aim:** The aim for this mechanic is to have something similar to the 'Blink' ability from Dishonored. This ability creates a floating marker when the left mouse button is held down, which the player can move around. When they let go of the mouse button, the player teleports to that position. This mechanic will differ from that version in that it allows movement through walls (but not so that the player can escape the game's boundaries), and the player can manually move the marker back and forth by scrolling the mouse wheel. The mechanic is utilised by the first person character controller, meaning it features in a first person game, and the player can also walk, turn around and jump.

### Scripts:

- **blink:** Attached to the main camera (child of the First Person Controller), handles instantiation of the marker and mana. Receives a boolean from the markerBehaviour script saying whether the player can or can't teleport to the marker's position, and sends the marker's position to the teleport script. Triggers the destruction of the marker.
- **teleport:** Attached to the First Person Controller. Takes the position of the marker sent from the blink script and warps the player to that point
- **markerBehaviour:** Attached to the target prefab. Handles collisions which changes the colour of the marker and determines whether the player can teleport to the marker's position or not. Sends a boolean to the blink script saying whether the player can or can't teleport to the marker's position, and receives the order to destroy the marker from the blink script.

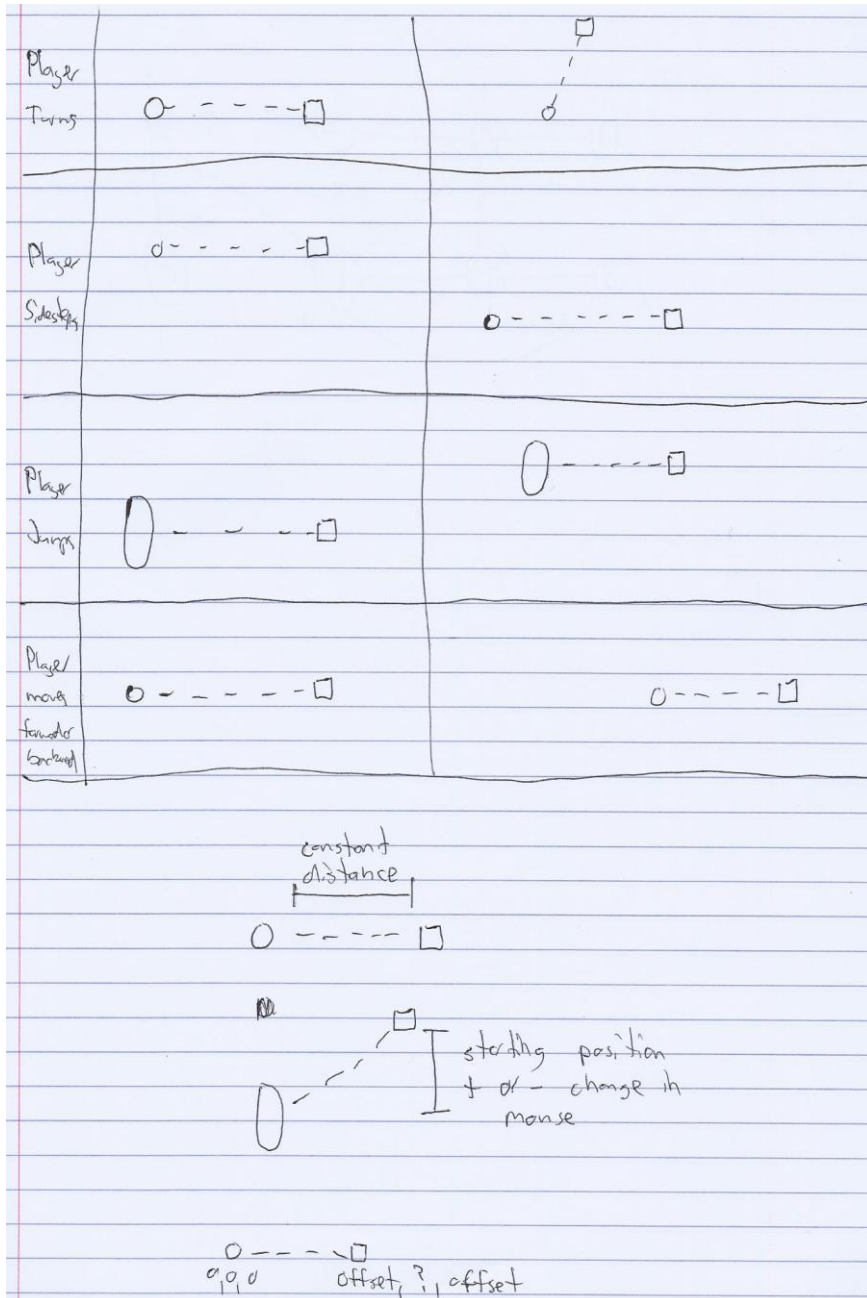
## Preliminary (pre-alpha) Testing

### Problems Discovered:

- ~~Incorrect marker positioning~~ Fixed
- ~~Marker not disappearing on mouse button being released~~ Fixed
- Player warping through walls

This stage of testing was done purely by me in order to get the mechanic in as early a functional state as possible to allow for external playtesting. As this was also the initial development of the actual mechanic, it was done over a long time period, starting from when the assignment was initially handed out, until about the 16th of June. Testing was done on a flat, grey plane.

The original approach was to have the marker move based off movements in the mouse's coordinates i.e. the mouse moving to the left would decrement the marker's x position etc. The problem with this was that with the player rotating, moving forward and backward and jumping, it would change whether the values should be incrementing or decrementing, and would mean some values would need to be constantly altered to have the marker's position remain relative to the player, and it was all becoming a bit of a mess. I drew diagrams to try and wrap my head around how the marker's position should actually work:



While doing this I realised the best way to have the marker move as desired would be to have it attached to the camera as a child, so it would move wherever the camera did, while remaining a fixed distance away. The marker would be instantiated at the parent's position, plus an offset of the parent's 'forward' property multiplied by a number. I chose 7 because it made the marker appear at what seemed a fair distance away.

Another problem was the marker not instantiating and disappearing correctly on the mouse button being pressed and released. The solution was to use `Input.GetMouseButton` instead of `Input.GetMouseButtonDown` and have a boolean checking whether the script inside the check had already been run or not. This meant that the marker would always be onscreen while the mouse was held down (and wouldn't wait for the mouse to be released for it to instantiate), and that it would only instantiate once while the mouse is held down. Without the boolean check it would constantly instantiate more markers while the mouse was held down.

The biggest problem I encountered though, was that the player could teleport through walls. Since the marker is always instantiated a set distance away, and isn't treated as a solid object, it can pass through other objects. This means the player can be teleported out of the level bounds, or get stuck inside other objects. Dishonored solves this by having the marker treated as 'solid', and is pushed back by other objects. One downside to this was that you couldn't teleport through small gaps e.g. through cracks in the floor or slightly ajar doors. I want my mechanic to be able to do things like that, so I will wait to hear what my playtesters feel before deciding on anything concrete.

## Alpha Testing

### Main Feedback Taken Into Consideration:

- Teleporting through walls isn't bad itself, but it can cause the player to fall out of bounds, which is bad
- The game should have a stamina/mana bar to force the player to be more conservative with their teleports
- The marker should be a bit further away to make the teleport feel more useful
- A shadow or something similar would be good to get an accurate feel of where the marker is in the 3D space
- Remove the mouse cursor

This stage of the testing was done on the 17th of June by six of my Facebook friends, who had the program and the following feedback sheet sent to them (overleaf) I have included summaries of the responses I received as well, for reference:

## **Teleport Mechanic Playtesting (Alpha Stage)**

Thank you for agreeing to playtest my in-progress game mechanic! Our goal for this assignment was to implement a single game mechanic/mechanism in Unity, and have it playtested at various stages. My mechanic is now in what I'd call the alpha stage, in that it's at its first playable form. It has some bugs as you may discover, but I want to test it now to see what I can fix now before it's too late, and where I should head with this mechanic.

The mechanic I have chosen to implement is a teleport ability, like the Blink ability from Dishonored. By holding down the left mouse button a marker will appear, which can be aimed with the mouse. By letting go of the mouse button, the player will then teleport to the marker's position.

It's recommended that in the beginning startup window you set the resolution to be your monitor's resolution, and at the moment you need to press Alt+F4 in order to quit (or the Mac equivalent).

Now, on with the questions! Please answer these questions after playing around with the program for a minute or two.

### **What kind of game do you think this mechanic would be used for?**

Responses included Action adventure, Stealth, RPG, platformer and puzzle game

### **Did you encounter any problems using this mechanic?**

All responses mentioned being able to teleport outside the level. People said that while glitching through the walls/floors was bad, teleporting through walls was a cool idea that I should keep. One person mentioned that since the marker doesn't have a shadow, or something to that effect, it can be hard to judge exactly where you're going to teleport.

I decided to keep the ability to warp through walls, while preventing the player from warping into out of bounds areas. I also decided giving the marker a shadow that indicated its position would be a good idea.

### **On a scale of 1-10 (with 1 being barely moving at all, and 10 being extremely oversensitive) how did you find the aiming sensitivity?**

Responses ranged from 4-8, with the average being 5.75

I decided messing with the mouse sensitivity wasn't worth my time since people seemed ok with it anyway.

### **On a scale of 1-5 (with 1 being too close and 5 being too far away) what did you think of the marker's distance away from the player?**

Four people gave a 2, person gave a 2.5, one person gave a 3

I originally decided to move the marker further out, but while doing this realised allowing the player to 'zoom' the marker in and out would make everyone happy while opening up new gameplay opportunities.

**At the moment, the player can teleport indefinitely with no limits. This means that with good timing you can effectively float in the air indefinitely. Would you think a game keeping the mechanic this way would be fun, or would you prefer a form of stamina/mana bar that restricted usage? Why/why not?**

Half of the testers said that a mana bar would make the game better, as it would force the player to be more careful about their usage of the teleport, and would give the game an element of challenge. One person however, suggested that the limit could be done differently, and gave an example of the game Sonic Heroes, which let you float and ascend with no limit, but horizontal movement while in this mode cost stamina. Two other people suggested that having no limits on the teleport would make the game more unique, and makes the game more fun since you can 'fly' all over the place.

I decided to enforce a mana limit in order to give the game some challenge, as I felt this would appeal to more people, and would make level design incorporating the mechanic a lot easier to design.

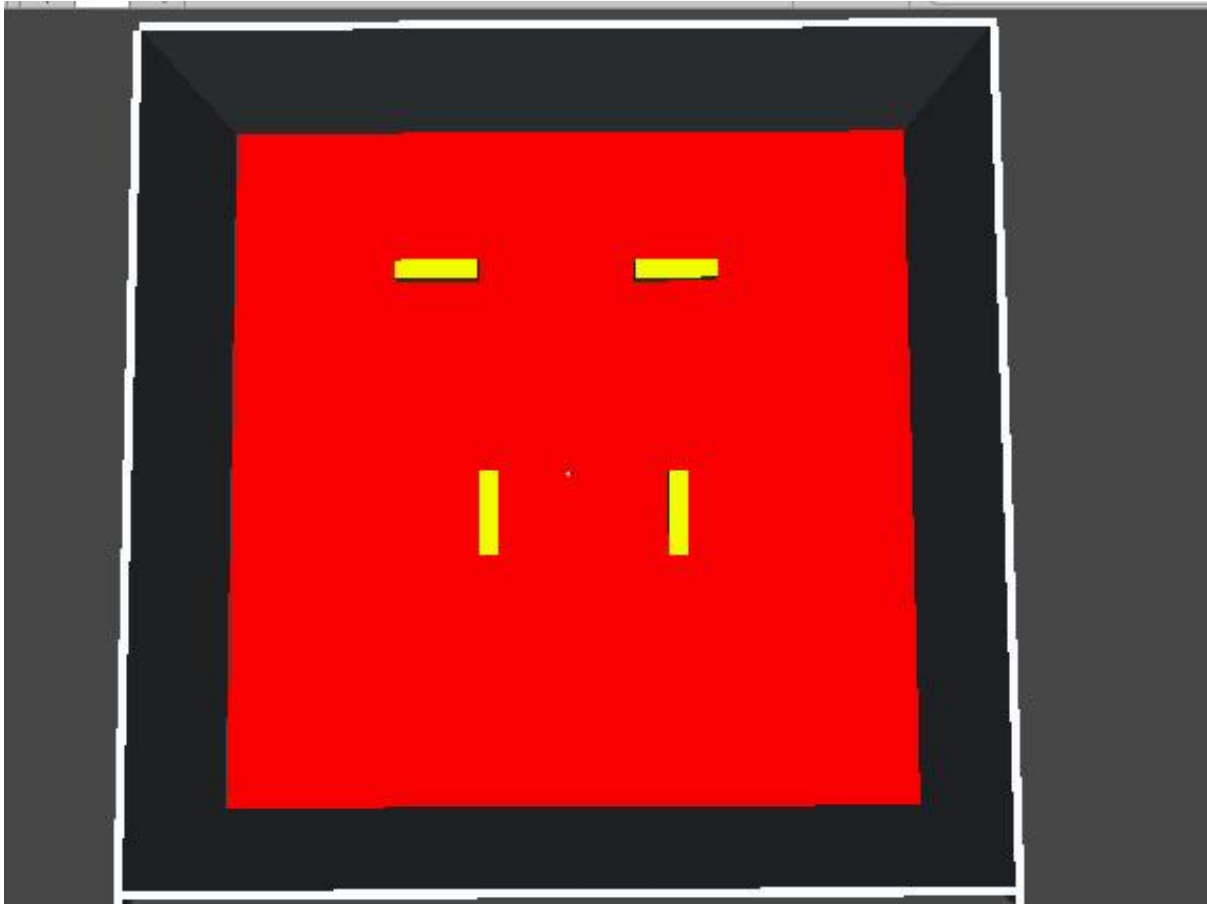
**Finally, feel free to list any miscellaneous comments you have after testing**

Most people left this blank. One person said that the mouse cursor not aligning with the marker was disorienting, so I decided to hide the mouse cursor.

Thanks again for volunteering, this will be a big help!

At this stage, the player could move, create a marker when the mouse was held down, and teleport to the marker's position when the mouse was released. However, the player could teleport inside objects and through the walls and floor as this was still the early, alpha version. I wanted testing done as early as possible though, so this wasn't a problem.

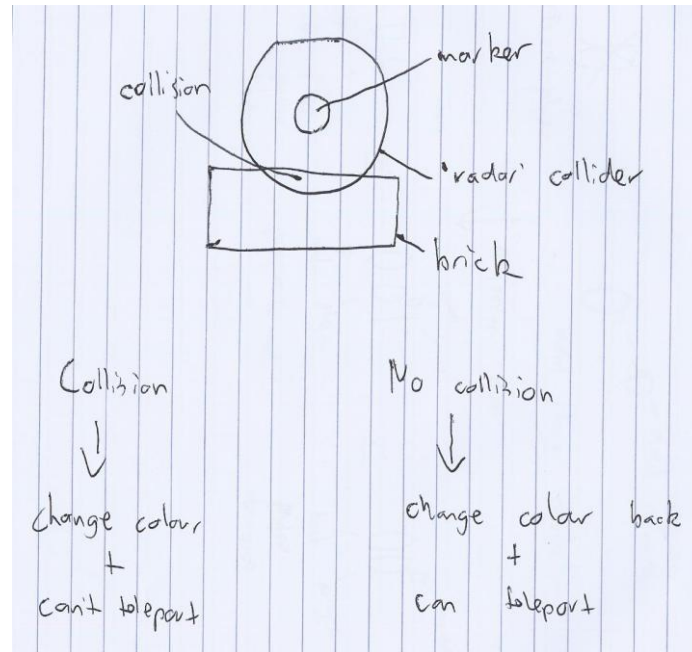
Testing was done in this 'level'. The floor is a red plane, surrounded by grey walls. This was done to differentiate them from each other and make the play space more clear to the player. Big, yellow blocks were also scattered throughout the level, which allowed players to test teleporting on top of, and through objects.



### **Alpha to Beta Progression Period**

This was the period in between my alpha testing, and the next round of testing I did later, from the 18th-19th of June. I set about working on the feedback I got from the initial testing. Here's how I solved each problem:

## Teleporting out of bounds:



I gave the marker a capsule collider, set to be a trigger about the size of the player, and gave it the rigidbody component. This means that if the marker collided with something, it meant the player couldn't fit there. It would change colour depending on whether the player could (blue) or couldn't (red), and sent a message to the camera (which has the teleport script attached) that the player could/couldn't teleport. Originally I had a problem with my OnTriggerExit function, in that it set the player as able to teleport whenever it exited a collision, even if it was still colliding with something else. By adding a variable that counted the number of collisions, it would then only send the message when ALL of the collisions were exited.

In keeping with my feedback (and deciding it would be fun to implement), I wanted to give the player the ability to warp through walls so long as it didn't take them out of bounds. When conceptualising the idea I realised that if the marker is behind a wall then players won't know if they can teleport there or not. I found a shader that changes the colour of objects if they're behind another object, and added it to my marker. If it was behind another object it would either be green (if the player can teleport there) or maroon (if the player can't teleport there).

The last remaining problem here was that if the player could teleport through walls, they could also teleport through the level bounds and floor. To circumvent this, I placed empty colliders around the bounds of the level, which trigger the marker as being inside an object. At first they didn't overlap, they were just as long as the walls and floors, which meant you could still break through the boundaries by finding the points where they intersected and teleporting through there. I adjusted their sizes so that there was plenty of overlap and this was fixed.

## Mana Bar

This was a simple fix by taking the OnGUI code from the practicals, and replacing the number of bricks/health remaining to instead refer to the amount of mana remaining

```
void OnGUI ()
```

```

{
    GUI.Box (new Rect (10, 10, Screen.width / 2 / ((float)manaInit / manaCurr), 20), "Mana: " + manaCurr);
}

```

Then, before the player teleports, it checks to see if they have more than 30 Mana left. If they do, 30 mana is subtracted and the player teleports. If not, they remain where they are.

### Marker distance

I increased the marker's offset to be the parent's forward property multiplied by 10 in order to address the feedback stating that it was too close. However, when I tried teleporting around, it felt odd to not be able to warp to close by locations as well. I decided to make it so that when the player is holding down the mouse button, they could move the marker forward and backwards by scrolling the mouse wheel. Now whenever the mouse wheel is scrolled, the offset value is incremented or decremented, so long as it's not higher than 20 or lower than 1. The marker's position is then adjusted accordingly.

```

if (Input.GetAxis ("Mouse ScrollWheel") < 0) {
    if (offset > 1) {
        offset--;
        target.transform.position = transform.position + (transform.forward *
offset);
    }
}
if (Input.GetAxis ("Mouse ScrollWheel") > 0) {
    if (offset < 20) {
        offset++;
        target.transform.position = transform.position + (transform.forward *
offset);
    }
}
}

```

This originally meant the player could warp out of bounds again, because my boundary meshes weren't big enough, so I adjusted them accordingly.

### Marker shadow

This was something I hadn't considered until I got feedback, but agreed that it was important in order to get a good idea of where the marker is actually placed in the world. I spent lots of time trying to get the marker's shader to cast a shadow on the world, but it just wasn't working. Eventually I found that I just had to add *FallBack "VertexLit"* before the end of the shader code. Now by rotating the directional light to be perpendicular to the level, the marker would cast a shadow directly down onto the ground it was above.



### **Remove the mouse cursor**

Fixing this was simple, I just had to add `Screen.showCursor = false;` to the Start function of the teleport script.

## **Beta Testing**

### **Feedback Taken Into Consideration:**

- A bug prevented players from teleporting if they tried teleporting while the marker was red/maroon

This took place on the 19th of June, and I used the same testers as I did for the previous testing. This was so I could get a better idea of whether the changes I had made fixed their original problems with the first code. I sent them the following sheet (overleaf), and I have again added a summary of answers:

## **Teleport Mechanic Playtesting (Beta (potentially final!) Stage)**

Thanks again for helping me with playtesting for my assignment, the first batch of responses were really helpful! My mechanic is now in the Beta stage, although this will probably be its final stage as well, as everything seems to be working. My goal for this round of testing is for you to try and change that- try your hardest to break the game!

The teleport mechanic has now evolved so that it allows you to teleport through walls, but not inside solid objects or out of the level's bounds. If the marker is blue or green, you're able to teleport to that spot, with a green marker signifying that it's currently behind another object. If the marker is red or maroon, then you aren't able to teleport to that spot, with maroon signifying the marker is behind another object. You can now move the marker forwards and backwards by scrolling the mouse wheel while holding down the left mouse button. There is a limit on how far forward and back the marker can be moved though.

You'll also notice the mana bar at the top of the screen. This shows the amount of mana you currently have available, and if you have less than 30 mana then you can't teleport. Mana regenerates over time.

The testing level itself has changed quite a bit from when you last saw it. I've added in a few little structures for you to toy around with that should be interesting to toy around with using the teleport.

Again, it's recommended that in the beginning startup window you set the resolution to be your monitor's resolution, and at the moment you need to press Alt+F4 in order to quit (or the Mac equivalent). The game also only works with mouse and keyboard only, not gamepads.

Please answer the following questions after you're done with the program

**On a scale of 1-10 (with 1 being extremely worse, 5 being about the same, and 10 being much better) how does this version of the mechanic compare to the last one?**

Responses ranged from 8-10, with 9 being the average.

**Did you encounter any problems using this mechanic?**

One person discovered that if you tried teleporting into an out of bounds area (i.e. the marker is red or maroon) then it would disable your teleporting ability until the marker moved in and out of another object. I would need to fix this.

Some people thought being able to teleport into the house was a bug, so in a full game implementing this mechanic it would need to be stressed that being able to teleport into, and hide in, structures like these is part of the game.

**Did you find the limits on how far and close you could place the marker were fair? If not, why not?**

All but one person said that the furthest limit felt fair- long enough to justify teleporting, but short enough to not feel like the player was cheating or breaking the game. The one person who disagreed wanted the range to be longer. I feel like the far limit is far enough, and the majority of testers agreed, so I will keep it the way it is.

The testers said that the back limit was fair, but a couple of people commented that it didn't make sense because no matter what, you can't teleport to that location, which makes sense. One person said that this made it an easy way to cancel a teleport when you decided you didn't want to, which I hadn't thought of, so I will keep it this way for that reason.

### **Do you think the mana limits are fair? If not, what would you change?**

The consensus here was that it was hard to judge this without seeing the full game. They said the mana limit forced them to be conservative with the teleport, but could get in the way of their traversal of the obstacles in the test level. One person said the idea of mana was a silly limit for a single ability, and would make more sense if the player had more abilities.

My vision was that a full game utilising this mechanic would have mana potions that could restore your mana in a bind, and that the player would indeed have more powers available, but I deemed both of these ideas outside the scope of this project, which was to implement a single mechanic.

### **Did you feel the marker adequately showed the position you would teleport to? If not, what could be done to alleviate this?**

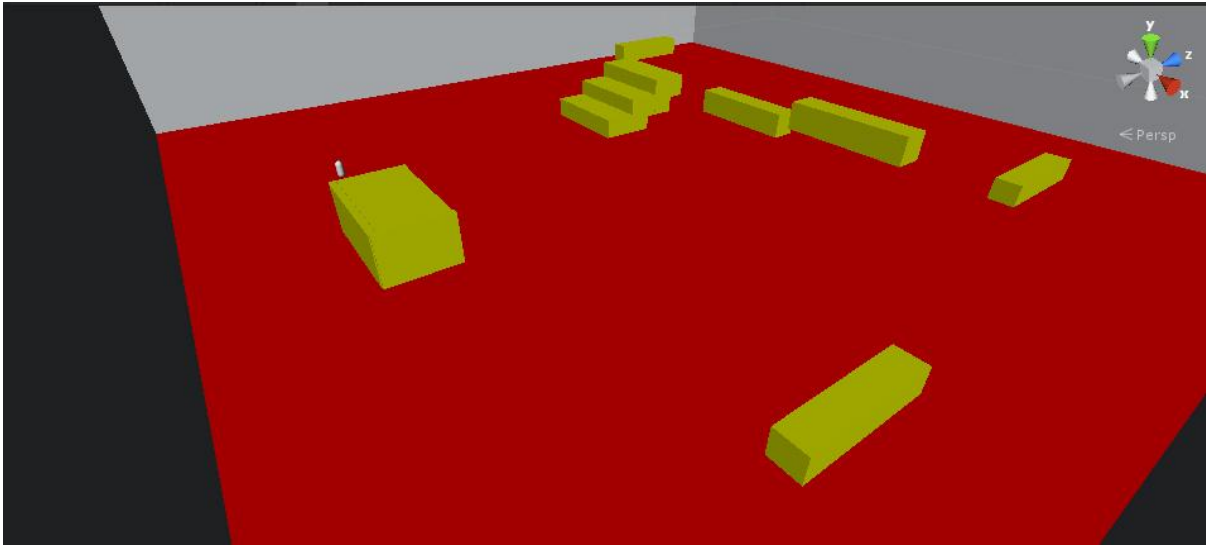
Most people felt that the changes to the marker made it really clear where you'd be teleporting, but one person mentioned that if you're at the wrong angle then you can't see the shadow under the marker. There's not really much I can do about that though, unfortunately.

### **Finally, feel free to list any miscellaneous comments you have after testing**

A couple of people commented on the shadow being added to the marker as a great idea

Thanks again for volunteering, this will be a big help!

So long as no bugs were found, this would be the final test. Unfortunately a bug was found, which I have now corrected, and the mechanic seems to be totally functional now. The test took place in the same space as before, but with different 'obstacles'



In addition to a couple of blocks scattered around the map, there were now stairs the player could jump up (or teleport up), and some floating platforms that the player could teleport to. There was also what appears to be a big brick, but it is actually a hollow 'house' of sorts, that encouraged the testers to try warping through walls as much as possible, to see if there were bugs.

At this (essentially) final stage, the player can hold down the mouse to make the marker appear, and while doing this they can scroll the mouse wheel to move the marker back and forth. They can teleport through walls and objects, but not if it would get them stuck in an object or outside the level's bounds.

The feedback suggested that the mechanic was much better now, and only one bug was found- namely that if the player tried to teleport somewhere that they couldn't teleport to, it would disable the player's ability to teleport. The only way they could fix this was if they moved the marker in and out of another object. I found the cause of the problem was that the marker only sends a message to its parent when it enters or exits a collision- not when it's instantiated. This meant that when the marker is instantiated, the boolean saying whether the player can teleport or not will have the same value as when the mouse was last released. If you're teleporting correctly nonstop this won't be an issue, but if you release the mouse in an out of bounds area, it means that this value will remain false until the marker collides with something again.

## Final Testing

### Problems Discovered:

- ~~Marker could instantiate off-centre if the player is moving the camera around when they push the left mouse button~~ Fixed

On the 20th of June I tested out my program some more now that it was effectively in its final stage and all the problems and suggestions raised in the feedback were resolved. Everything was working perfectly, except when I was spinning around and pressed the mouse button, the marker appeared off-centre. I fixed this by moving the line of code that repositions the marker out of the mouse

scrolling code, and into the code that just checks if the mouse is held down. This means that the marker will always be placed in the right position, no matter what action the player is taking.

On the 21st of June when I was trying things out again, I discovered you could still warp outside the level if you zoomed the marker out to the maximum length and moved into the corner. The intersection between the 'wall' boundaries was sufficient for the marker at its original length, but now that it can be zoomed out I need to place more boundaries. I duplicated the walls, and rotated them so they would stretch outwards from the corners. This made for an out of bounds area that the marker couldn't escape from.

## **Conclusion**

Due to no further bugs arising from my, or my playtesters', testing I believe that my mechanic is now fully functional, with no errors. It functions the way I envisioned it to, and took feedback from playtesting into consideration when developing the mechanic. Since I playtested early on, and during the middle of the process, I got plenty of good feedback in time to implement it, and let it shape the development of my idea.