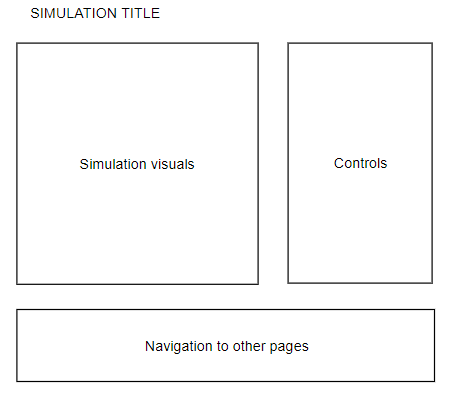
## Criterion B: Design

**Page layout**



**System variables**

|  |  |
| --- | --- |
| **Variable name** | **Type / description** |
| Time | Float / stores the time that has elapsed since the start of the simulation |
| Ball | Object / stores the ball which is the subject of the simulation |
| Width | Integer / stores the width of the simulation area in meters |
| Height | Integer / stores the height of the simulation area in meters |

**Ball object properties and methods**

|  |  |
| --- | --- |
| **Name** | **Type / description** |
| posX | Float / stores the x position of the ball |
| posY | Float / stores the y position of the ball |
| previousX | Float / stores the x position of the ball at the last frame |
| previousY | Float / stores the y position of the ball at the last frame |
| velX | Float / stores the x direction speed |
| velY | Float / stores the y direction speed |
| Speed | Float / stores the magnitude of the ball’s velocity |
| Draw() | Function / draws the ball at its current position |

**System methods**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Pseudocode** |
| Play() | Plays the simulation | IF ball.posY > 500 – 10:  ball.velY = -ball.velY \* 0.5  ball.posY = 500 - 10  ball.velX = ball.velX \* 0.5  ELSE:  ball.velY = ball.velY - 2  END IF  ball.posX = ball.posX + ball.velX  ball.posY = ball.posY - ball.velY \* 0.01  ball.draw(); |
| changeDim() | Changes the width and height of the |  |
| convertSpeed() | Calculates the magnitude and direction of the ball’s velocity | Speed = sqrt(ball.velx^2 + ball.vely^2)  Angle = arctan(ball.vely / ball.velx) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |