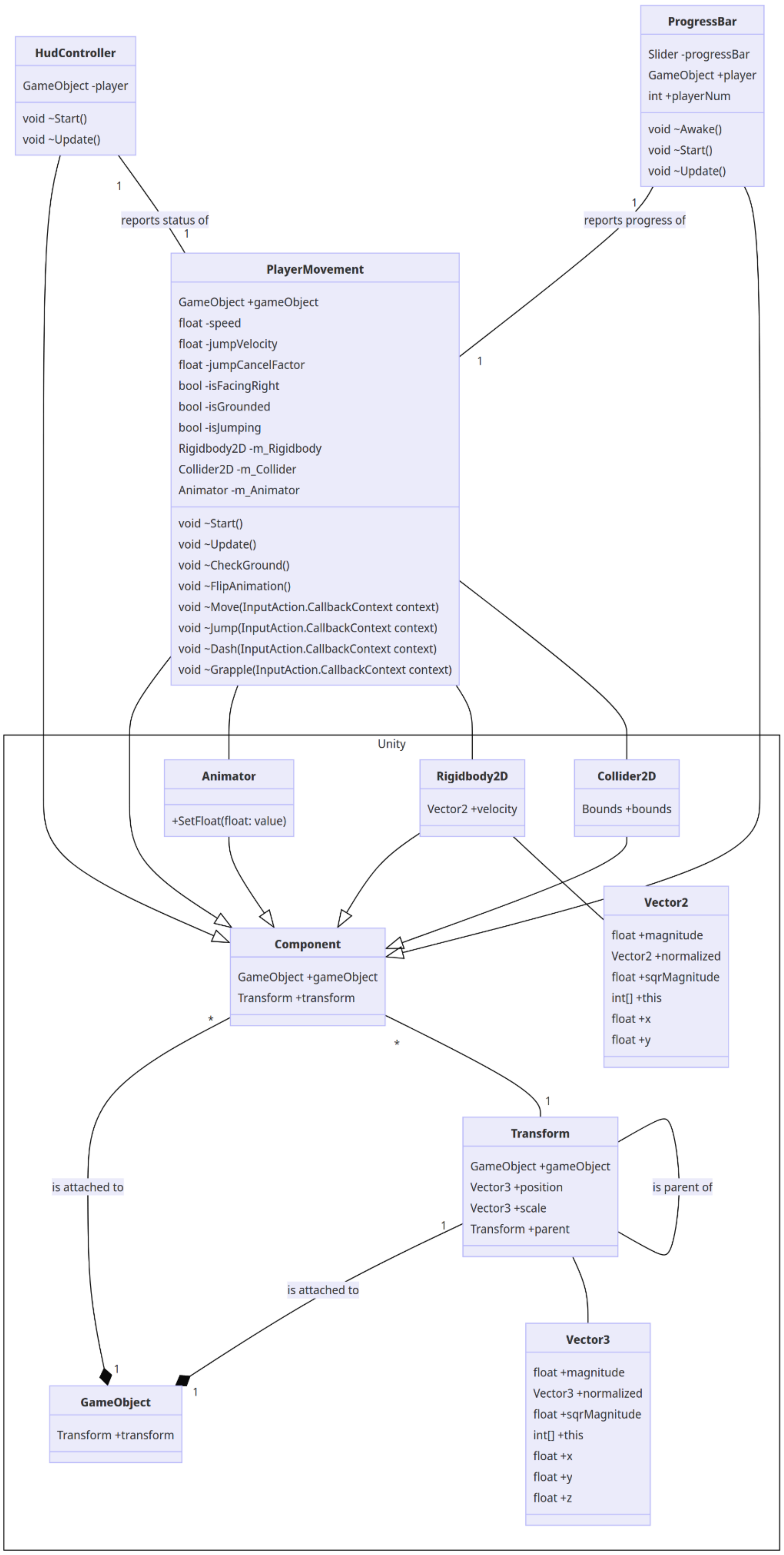
Double Dash Dodge and Deceive aims to allow the user to control an avatar in a virtual world to compete against another player in a race. To do this it needs to take input from the user to move the player character in a PlayerMovement script. This input is processed and sent to associated Rigidbody2D and Animator components attached to the player GameObject to reflect the change in state.

The game also needs to inform the user about the current state of the game via a heads up display (HUD). The HUD is controlled by a HudController and ProgressBar script, which hook into a specific GameObject with a corresponding PlayerMovement script representing a player. This gives a one to one multiplicity between HudControllers and PlayerMovement scripts, as well as a one to one multiplicity between ProgressBars and PlayerMovement scripts.

In the Unity game engine, behaviour scripts like PlayerMovement are components which can be attached to GameObjects. This means that PlayerMovements, HudControllers, and ProgressBars all inherit from the Component class.

Each GameObject has a single associated Transform which describes its position in three-dimensional space, and so there is a one to one multiplicity between GameObjects and Transforms. Additionally GameObjects may have any number of Components, but each Component can only be attached to a single GameObject; therefore, there is a many to one multiplicity between Components and GameObjects. Components also store a reference to the Transform of the GameObject they are attached to, and so there is also a many to one multiplicity between Components and Transforms.

Transforms use a Vector3 object to store both their position in cartesian coordinates, as well as their scale along their intrinsic axes. They also store a reference to their parent Transform, which they inherit movement and rotation from.


HudController
GameObject -player
void ~Start()
void ~Update()
ProgressBar
Slider -progressBar
GameObject +player
int +playerNum
void ~Awake()
void ~Start()
void ~Update()
1
reports status of
1
1
reports progress of
1
PlayerMovement
GameObject +gameObject
float -speed
float -jumpVelocity
float -jumpCancelFactor
bool -isFacingRight
bool -isGrounded
bool -isJumping
Rigidbody2D -m_Rigidbody
Collider2D -m_Collider
Animator -m_Animator
void ~Start()
void ~Update()
void ~CheckGround()
void ~FlipAnimation()
void ~Move(InputAction.CallbackContext context)
void ~Jump(InputAction.CallbackContext context)
void ~Dash(InputAction.CallbackContext context)
void ~Grapple(InputAction.CallbackContext context)
Unity
Animator
+SetFloat(float: value)
Rigidbody2D
Vector2 +velocity
Collider2D
Bounds +bounds
Vector2
float +magnitude
Vector2 +normalized
float +sqrMagnitude
int[] +this
float +x
float +y
Component
GameObject +gameObject
Transform +transform
*
*
1
Transform
GameObject +gameObject
Vector3 +position
Vector3 +scale
Transform +parent
is parent of
is attached to
1
is attached to
1
1
GameObject
Transform +transform
Vector3
float +magnitude
Vector3 +normalized
float +sqrMagnitude
int[] +this
float +x
float +y
float +z