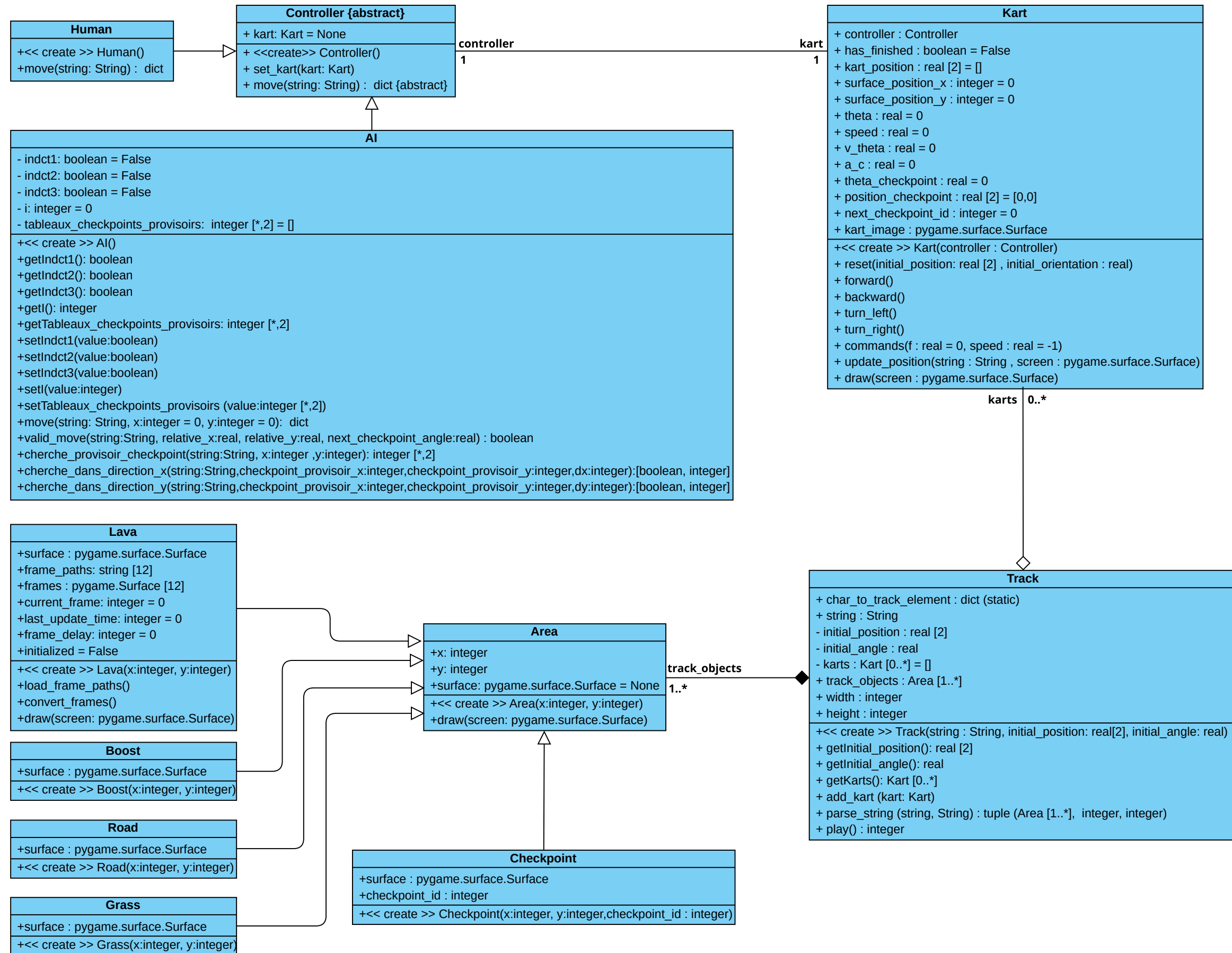


UML Diagram



Relation entre la classe Controller, et les classes Human et AI : (Relation d'héritage)

Nous avons créé une classe Controller qui possède un constructeur initialisant la variable d'instance "kart" de type Kart à None. En outre, elle comporte une méthode set_kart permettant de définir un objet de type Kart comme la valeur de la variable d'instance "kart". Cette méthode est appelée au niveau du constructeur de Kart, établissant ainsi une association significative entre la classe Controller et Kart. La classe Controller est déclarée abstraite car elle contient une méthode abstraite "move" qui sera redéfinie au niveau des classes filles Human et AI, lesquelles héritent de Controller.

Relation entre la classe Kart et la classe Controller : (Relation d'association)

La classe Kart possède un attribut de type Controller (Human ou AI), avec la particularité que chaque objet de Kart est associé à un unique objet de Controller. Comme mentionné précédemment, la méthode "set_kart" est appelée dans le constructeur de Kart pour permettre à l'objet de Controller d'accéder à l'objet Kart. Ainsi, la relation entre les deux classes est caractérisée comme une relation d'association.

Relation entre la classe Kart et la classe Track : (Relation d'agrégation)

La classe Track peut utiliser des objets de type Kart pour les manipuler à l'aide de méthodes et attributs spécifiques à Kart tels que "controller," "reset," "draw," etc. Les objets de type Kart peuvent subsister indépendamment de Track, c'est-à-dire que la destruction de Track n'entraîne pas automatiquement la destruction de Kart. La relation entre ces deux classes est une relation d'agrégation.

Relation entre la classe Area et la classe Track : (Relation de composition)

La classe Track possède un attribut "track_objects" qui est rempli à l'aide de la méthode "parse_string." Cette méthode permet la création d'objets de différentes classes telles que Grass, Road, Boost, etc., qui héritent de la classe Area. Ainsi, on peut affirmer que les objets créés par Track sont de type Area. Étant donné que l'existence de ces objets dépend de l'existence d'un objet Track, c'est-à-dire que la destruction de Track entraîne la destruction des objets de type Area, la relation entre ces deux classes est une relation de composition.

Relation entre les classes (Grass, Road, Boost, Lava, Checkpoint) et la classe Area : (Relation d'héritage)

La classe Area possède une méthode "draw" qui est utilisée par toutes les classes filles. De plus, son constructeur est utilisé dans les constructeurs des classes filles en surchargeant uniquement l'attribut "surface," car chaque attribut "surface" est spécifique à une classe fille. La relation entre ces classes est clairement une relation d'héritage.