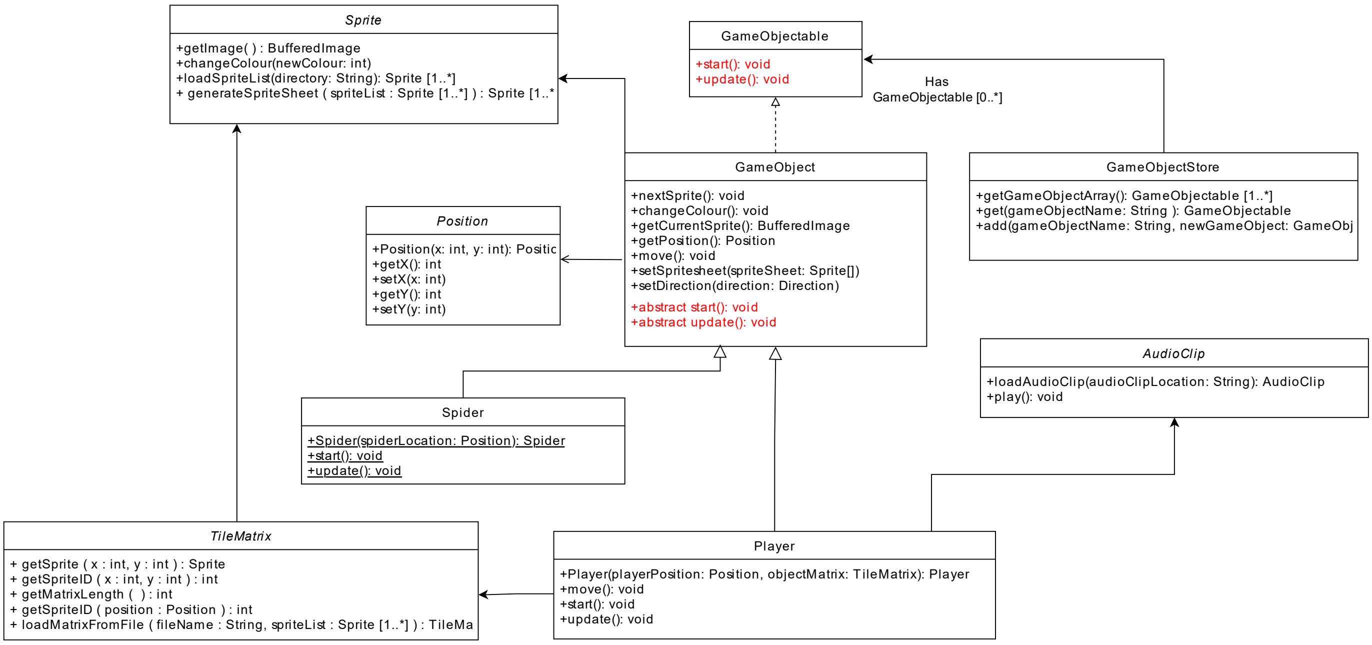
Declan Kelly - G00378925

UML diagram


Sprite
+getImage( ) : BufferedImage
+changeColour(newColour: int)
+loadSpriteList(directory: String): Sprite [1..*]
+ generateSpriteSheet (spriteList : Sprite [1..*]) : Sprite [1..*
GameObjectable
+start(): void
+update(): void
Has
GameObjectable [0..*]
GameObject
+nextSprite(): void
+changeColour(): void
+getCurrentSprite(): BufferedImage
+getPosition(): Position
+move(): void
+setSpritesheet(spriteSheet: Sprite[])
+setDirection(direction: Direction)
+abstract start(): void
+abstract update(): void
GameObjectStore
+getGameObjectArray(): GameObjectable [1..*]
+get(gameObjectName: String): GameObjectable
+add(gameObjectName: String, newGameObject: GameObj
Position
+Position(x: int, y: int): Positic
+getX(): int
+setX(x: int)
+getY(): int
+setY(y: int)
AudioClip
+loadAudioClip(audioClipLocation: String): AudioClip
+play(): void
Spider
+Spider(spiderLocation: Position): Spider
+start(): void
+update(): void
TileMatrix
+ getSprite (x : int, y : int) : Sprite
+ getSpriteID (x : int, y : int) : int
+ getMatrixLength ( ) : int
+ getSpriteID (position : Position) : int
+ loadMatrixFromFile (fileName : String, spriteList : Sprite [1..*]) : TileMa
Player
+Player(playerPosition: Position, objectMatrix: TileMatrix): Player
+move(): void
+start(): void
+update(): void

To avoid clutter `Direction`, `GameView`, `Position`, `Renderer` and `Runner` are absent from this diagram, since they are not key to the design.

Design

1. Tile matrices have been moved into their own files found in `./resources/tilemaps/`, load a new tile matrix using `TileMatrix.loadMatrixFromFile`.
2. The `Player` and `Spider` exhibit common functionality (both having a `Position`, `Direction` and `Sprite` associated with them), they now both inherit from the `GameObject`.
3. To be a `GameObject` you must implement the `start` and `update` methods, start being called on initialisation and update called once per a frame, this pattern is consistent with game engines such as Unity.
4. The `GameView.Builder` can be used to assist with the setup of the `GameView`, this is used by the `Runner`.
5. The `AudioClip` cannot be initialised directly, only with the use of the factory static method `AudioClip.loadAudioClip`.

Extras

1. The `AudioClip` class is used to implement the feature of playing sound effects in the game.
2. When you press the **X** key, the `move` method in the `Player` class is executed, playing the move sound.
3. The objective of the game is to locate the treasure chest. When the treasure chest is found, a celebratory sound will be played.