**Alpha Features**

Director

Dynamic sprites

Input for player movement

Collision Handling

Camera Movement

Player

Platforms

Spawning

Class name: **Game**

Stereotype: Director

Inherits: **arcade.Window**

Attributes:

* **platforms\_list** – arcade.SpriteList(). Contains all the platforms that are active in the game. Initialized in init().
* **all\_sprites** – arcade.SpriteList(). Contains all the sprites that are active in the game. Intialized in init().
* **dynamic\_sprites** – arcade.SpriteList(). Contains all the sprites that are active and whose position in the game is relative to the player. Initialized in init().
* **paused** – boolean initialized as False. Determines if the game is paused or not. Initialized in init().
* **player** – Player(). The player of the game. Initialized in setup().

Methods:

* **setup**(self) – sets up the game by initializing the players, the platforms, and their position in the game. Once players and platforms are initialized, they are stored in their respective lists.
* **on\_key\_press**(self, symbol[int], modifiers[int]) – handles user keyboard input. Q to quit the game. W/A/S/D to move the player up/left/down/right. Argument symbol is for which key was pressed while argument modifiers is for which modifier was pressed. *Refer to tutorial.*
* **on\_key\_release**(self, symbol[int], modifiers[int]) – undo player movement update when keys are released. Argument symbol is for which key was pressed while argument modifiers is for which modifier was pressed. *Refer to tutorial.*
* **on\_update**(self, delta\_time[float]) – update the positions and statuses of all game objects. If paused, do nothing. Will also check for collisions and camera panning. *Refer to tutorial.*
* **on\_draw**(self) – draw all game objects. *Refer to tutorial.*
* **handle\_collision**(self) – checks if player has collided with any platform or enemy and updates the game state accordingly. Will also limit the players movement within the boundaries of the screen.
* **pan\_camera**(self) – checks if player will move pass the midpoint of the screen. If so, every moving sprite will move to the left equal to how much the player will have moved beyond the midpoint. Player’s position can never go beyond the midpoint of the screen.

Class name: **Player**

Stereotype: Information Holder

Inherits: **arcade.Sprite**

Attributes:

* *built-in arcade.Sprite() attributes*

Methods:

* *built-in arcade.Sprite() methods*

Class name: **Platform**

Stereotype: Information Holder

Inherits: **arcade.Sprite**

Attributes:

* *built-in arcade.Sprite() attributes*

Methods:

* *built-in arcade.Sprite() methods*
* **respawn**(self) – checks if current instance of platform is outside the game screen. If so, move this instance of platform outside of the game screen to the right, and in a random value of y within the game screen.

**Game constants**

WIDTH: 1000

HEIGHT: 700

NUMBER\_OF\_PLATFORMS: 3