**Player.cs**

Description: This script handles the spells’ and abilities’ effects after the target is chosen.

Methods that you would likely want to modify when customizing your game:

|  |  |
| --- | --- |
| **Method** | **Description** |
| void OnMouseDown () | This method is used when the player needs to choose a target that can be “this player”. For example: “Target player or creature”. This method is called when this player is selected as the target. **If you’ve added a custom target type that can have current player as a target**, modify this method. |
| public static void RemoveEOTBuffsAndDebuffs () | Invoked at the beginning at each turn, removes buffs and debuffs that should last until the end of turn (EOT). Currently only removes attack and defense EOT buffs. **If you’ve added a custom buff or debuff that lasts until EOT that does not modify attack/defense stats**, add lines here. |
| public void GameWon () | Makes everything on the screen blue and initiates display of “Victory” text in VictoryDefeat.cs **Change if you need a different visual effect for victory.** |
| public void GameLost () | Makes everything on the screen gray and initiates display of “Defeat” text in VictoryDefeat.cs **Change if you need a different visual effect for defeat.** |
| public void IsAttacked (card Attacker) | Similar to the method with the same name in **card.cs**, calculates the combat damage dealt by the attacking card and lowers the player’s life accordingly. **Change if you’ve added a new combat mechanic** such as “triple strike” |
| public void IsHitBySpell (Vector3 param) | param.x = damage amount  param.y = damage type  param.z = the spell card's ID  Currently there are two damage types: fire and combat. The damage type is specified to indicate what kind of special effect and how to treat the damage.  **If you want some spells to have different visual/sound effects** (for ex. lightning spells), add a damage type here. Additionally, a player might have for ex. a resistance effect that causes it to only take half damage from fire sources. |
| IEnumerator MusicPlaylist () | If MainMenu.OptionGameMusic is set to ‘true’, plays the music during the game.  **Modify if you want to change how it is played: the order of tracks, duration of playback etc.** |
| void PlayerTurnPopup (int windowID) | Displays the “Your Turn” popup if MainMenu. OptionPlayerTurnPopup is set to ‘true’. Modify if you want to customize it. |
| void OnGUI () | Displays player name and player stats such as life and mana. Displays “Choose a target” text when needed. **Modify if you need to customize the appearance of player’s stats and “choose a target” text.** |
| public static void RearrangeCardsInHand () | Called when a player draws a card or a card is removed from the hand, positions the cards in hand according to their number – the more cards there is, the denser they are positioned. **Modify to change the position of cards in hand.** |
| public static void RearrangeCreatures (  bool AI=false) | Same as the method before, but for creatures. **Modify to change the position of cards in hand (for both players).** |
| public static void RearrangeLands (bool AI=false) | Same as the method before, but for lands. **Modify to change the position of lands (for both players).** |

The following methods can be useful in your custom effects and mechanics. I don't recommend modifying these unless absolutely needed, as it can break existing functionality.

|  |  |
| --- | --- |
| **Method** | **Description** |
| public static List<GameObject> CreaturesAsGameObjects () | Returns the list of this player’s creatures as GameObjects |
| public static GameObject RandomCreature () | Returns a random creature that belongs to the player. Useful for effects that target a random creature this player has in play. |
| public static List<GameObject> RandomCreatures (int number\_of\_creatures) | Returns a list of random creatures that belong to the player. The number of creatures returned is determined by **number\_of\_creatures** passed to the method. |
| public void RestoreColor () | Restores the player’s avatar color. Use it in your visual effects if they involve changing the player avatar color. |
| public static bool HasACreature () | Returns ‘true’ if the player has a creature in play. |
| public static bool HasAHero () | Returns ‘true’ if the player has a creature with an attribute “hero=1” in the xml. Useful if your game has 'Hero' cards, which are specified in .xml or the deck editor. |

**These methods are central to the kit's functionality. I don't recommend modifying the following methods or using them on their own**:

|  |  |
| --- | --- |
| **Method** | **Description** |
| public static void StartGame () | Resets the player stats at the beginning of the game. Add a line here if you’ve added a new game zone or a new player stat. |
| void AssignTarget () | Sets the target to be the current player. |
| public static void NewTurn () | Removes buffs that last until end of turn (EOT), resets mana (if MainMenu. OptionManaDoesntReset is set to ‘false’) and takes creatures off of “summoning sickness” at the beginning of each turn.  Also gives the player the amount of mana corresponding to the current turn if MainMenu.OptionManaAutoIncrementsEachTurn is set to ‘true’. |
| void Update () | Initiates creature stats updating if needed, cancels the current spell/ability if the player has right-clicked (by default, right-click is used to cancel an action). Also checks for win/lose conditions to end the game – modify it if your lose condition is different from player life being 0 or less. |
| public void IsHealed (int param) | Plays the healing sounds and particle effects, and increases the player's health by **param**. |
| public void TakesCardSFX () | Plays the “TakesCard” audio clip. |
| public void GainsMana (int amount) | Increases the player’s mana by **amount** and plays the particle effect for gaining mana. |
| public static void SendTargets () | Converts Player.targets to a comma-separated string and calls the RPC method in **Logic.cs** to send it to multiplayer opponent |
| public static void OpenListToChooseCard (List<card> cardlist, int cardtype = -1) | Displays a list of cards to choose from. Used in target types such as “a spell from graveyard”. |
| public static void OpenIntListToChooseCard (List<int> cardlist, int cardtype = -1) | The same as the previous method, but used for <int> lists such as Enemy.Deck |