PlayerPrefs is used to preserve information from one scene to another. All values are set to the default when selecting a new game.

int: useKinect
This is used to store whether the user is playing the game via the kinect, or using a keyboard.
0 = user is using keyboard
1 = user is using kinect
Default: By default, when starting a new game, this value is set to 0, ie using keyboard. This is so any using without a kinect connected to the PC can still play the game. If kinect was default, they would not be able to change the value to allow them to use their keyboard. Any user who has a Kinect plugged into their PC, we can assume that they also have a keyboard attached to their PC. They can then use this keyboard to select the "kinect" option, setting useKinect to 1.
However, if the user has been previously playing the game, (and this means that there is a previous value useKinect), then this value is carried through when starting a new game.
Modified: optionsMenu.toggleKinect() accessed in the Option Submenu from the Pause menu. useKinect is set to (useKinect+1)%2, to toggle the value.
Read: read in levelProperties.Start(), to enable or disable the Kinect components. This value is always set in the Update() function, to take into account the user changing the value while in the middle of a level.

int: PlayerLives
Default: 3
Modified: Any changes occurred while playing through a scene are saved in completeLevel.save().
Read: The value is read in PlayerProperties.start(), to set the value of PlayerProperties.lives.

int: PlayerCoins
Default: 0
Modified: Any changes occurred while playing through a scene are saved in completeLevel.save().
Read: The value is read in PlayerProperties.start(), to set the value of PlayerProperties.coins.

Int: highestLevelCompleted
This is the index of the highest level (starting from 1) which the user has completed.
Default: 0
Modified: It is updated in completeLevel.save(). If the user is replaying a level (ie, the current level index < highestLevelCompeleted), this value is not updated in completeLevel.save().
Read: This value is used in the scene LevelSelect to determine if a level is accessible to the user. Sets value SelectLevel.isOpen, when SelectLevel.linkToWorldIndex is compared to worldOfHighestLevelCompleted and SelectLevel.linkToLevelIndex is compared to highestLevelCompleted

int: worldOfHighestLevelCompleted
This is the index of the world (starting from 1) which contains the level highestLevelCompeleted.
Default: 0
Modified: It is updated in completeLevel.save(). If the user is replaying a level (ie, the current level index < highestLevelCompeleted), this value is not updated in completeLevel.save().
Read: This value is used in the scene LevelSelect to determine if a level is accessible to the user. Sets value SelectLevel.isOpen, when SelectLevel.linkToWorldIndex is compared to worldOfHighestLevelCompleted and SelectLevel.linkToLevelIndex is compared to highestLevelCompleted

string: loadThis

This value is changed every time a new scene is to be loaded. Eg

```
PlayerPrefs.SetString("loadThis", level);
PlayerPrefs.Save();
Application.LoadLevel("loading");
```

The “loading” scene is loaded, then loads the scene that is contained within the variable “loadThis”. This is because it takes a few seconds when using the Kinect to load up the next scene. There is next to no wait time when using the keyboard. The loading level is currently not working as expected.

Default: “”

Modified: MapMenuScript_c.startLevel(), when starting/continuing a game from the main menu.
CompleteLevel.Upate(), when completing a level, and now going back to the “levelSelect” level.
selectLevel.loadLevel(), when selecting a level from the “levelSelect” level.

Read: LoadingScreen.OnLevelWasLoaded(), loads the level in “loadThis”