Achievements Plus

Documentation

Pre-processor define

Pre-processor defines are defined when the game is initiated. You can create a file with all the things related to Achievements Plus, and it will only be included when the base mod is active.

Read more about pre-processor in the Handbook.

```
#if ACHIEVEMENTS_PLUS_ACTIVE
    include achievements.lemon
#endif
```

Coins Functions

void AchievementsPlus.RewardsManager.loadPersistendData()

- Loads the Coins data stored in Persistent Data to the Shared Memory address.
- Notes:
    - The Shared Memory address is 0x851300.
    - To get the address value, use u32[0x851300].

bool AchievementsPlus.RewardsManager.addCoins(u16 value)

- Adds the specified value of Coins to the Shared Memory address and stores the new value in Persistent Data.
- Returns true when the function is successful.
- Parameters:
    - value is the number that will be used for adding the Coins value to the Shared Memory.
- Notes:
    - Error (200) notifies about the inability to add Coins. For example, if the specified value is negative or too large.

bool AchievementsPlus.RewardsManager.paidCoins(u16 value)

- Withdraw the specified value of Coins from the Shared Memory address and stores the new value in Persistent Data.
- Returns true when the function is successful.
- Parameters:
    - value is the number that will be used to withdraw the Coins value to the Shared Memory.
- Notes:
    - Error (201) notifies you that it is impossible to withdraw Coins. For example, if the value of Coins to withdraw exceeds the number of saved coins.
    - Other than that, an error may appear if you use too many withdrawals, or use a negative value.

Achievements Functions

void AchievementsPlus.AchivementsManager.loadPersistentData()

- Loads the Achievements data stored in Persistent Data to Shared Memory addresses.
- Notes:
    - Unlike Coins address, Achievement address are more dynamic.
    - The base Shared Memory address is 0x851304.
    - To get the address value, you will need to specify the ID of the achievement you are interested in, and multiply it by 4. Use u32[0x851304 + achievementID * 0x04].

string AchievementsPlus.AchivementsManager.getAchievementDetail(u16 achievementID, u8 parse)

- The function returns the specified data for the specified Achievement ID. This will return "" if there is no data for the specified parse.
- Parameters:
    - achievementID is the achievement ID for which you want to retrieve data.
    - parse is the type of data you want to receive from the specified achievement ID:
      0x00 = Return the title of the achievement.
      0x01 = Return the rating (stars) of the achievement.
      0x02 = Return the sub-title of the achievement.
      0x03 = Return a description of the achievement.
      0x04 = Return thumbnail (icon) of the achievement.

u32 AchievementsPlus.AchivementsManager.getAchievementAddress(u16 achievementID)

- The function will return the u32 address of the specified achievement ID.
- Parameters:
    - achievementID is the achievement ID for which you want to get an address.

void AchievementsPlus.AchivementsManager.addAchievementValue(u16 achievementID, u16 value)

- The function will add the specified value for the Shared Memory address of the specified achievement and store it in Persistent Data.
- Parameters:
  - achievementID is the achievement ID for which you want to add a new value.
  - value is the number you are going to add to the data about this achievement.
- Notes:
  - Error (205) notifies you that it is impossible to add your value to the data of the specified achievement. For example, if the value is too large or negative.

u32 AchievementsPlus.AchivementsManager.getAchievementValue(u16 achievementID)

- The u32 function returns the specified achievement ID.
- Parameters:
  - achievementID is the achievement ID for which you want to receive data.
- Notes:
  - Actually returns a value of u32[0x851304 + achievementID * 0x04].
  - Return 0 if there is no data to achievement.
  - Flags:
  - 0xa00000 establishes when conditions have developed in such a way that an achievement has been completed.
  - 0xf00000 can be set only after receiving the Completed flag.
    It is set when the toast of this achievement disappears behind the screen. Used to create a queue of toasts, so that several accomplished achievements show their toasts in turn, not us
  - 0x0f0000 sets the claim button in the achievement menu when you click. Means that for this achievement, the award was declared and received.

void AchievementsPlus.AchivementsManager.setAchievementComplete(u16 achievementID)

- Sets the specified achievement ID to Completed.
- Parameters:
  - achievementID is the achievement ID for which you want to set the status to Completed.
- Notes:

    - When the function is called, the 0xa00000 flag will be set to achievement.

bool AchievementsPlus.AchivementsManager.isAchievementComplete(u16 achievementID)

- The function will return true when the achievement has a status of Completed, or false if otherwise.
- Parameters:
    - achievementID is the achievement ID for which you want to check the completion status.

Additional Functions

bool AchievementsPlus.AchivementsManager.isAchievementsEnabled()

- The function will return true when achievements can be obtained, false if otherwise.
- Achievements cannot be obtained in Debug mode and Game via Level / Act select mode, so the function will return false.
- Regardless of the activity of debug mode, achievements can be obtained in Competition mode and Blue Sphere game in any case.

u16 AchievementsPlus.AchivementsManager.getTotalAchievementsByRange(u16 startID, u16 lastID, bool returnCompleted)

- The function will return the maximum number of available achievements, from the specified range of ID achievements and parameters.
- Parameters:
    - startID is the starting achievement ID from which the calculations will emanate, up to and including the lastID.
    - lastID is the final achievement ID on which the calculations will end.
    - returnCompleted, if set, the function will return the number of completed achievements from the range.

u16 AchievementsPlus.AchivementsManager.getTotalNonClaimedByRange(u16 startID, u16 lastID)

- The function will return the number of achievements for which a claim reward is available from the specified achievement ID range.
- Parameters:

    - startID is the starting achievement ID from which the calculations will emanate, up to and including the lastID.
    - lastID is the final achievement ID on which the calculations will end.

u8 AchievementsPlus.AchivementsManager.getDecimalStars(u16 achievementID)

- The function will return the number of stars (rating) for the specified achievement ID as a number, instead of a string.
- Parameters:
    - achievementID is the achievement ID for which you want to get stars.

Walls Functions

void AchievementsPlus.WallManager.loadPersistentData()

- Loads the Walls data stored in Persistent Data to Shared Memory addresses.
- Notes:
    - The base Shared Memory address is 0x8510fb.
    - To get the address value, you will need to specify the ID of the wall you are interested in. Use u8[0x8510fb + wallID].
    - 0x8511fb sets the required Coins value to purchase the wall that the player touched. Use u8[0x8511fb].
    - 0x8511fc sets the ID of the wall that the player touched. Use u8[0x8511fc].

void AchievementsPlus.WallManager.addSessionWall(u16 px, u16 py, u8 width, u8 height, u8 wallID, u8 paidValue)

- The function will spawn the wall object at the level with the specified position and parameters.
- Session wall does not save its purchase status when restarting a level or exiting the game.
- Parameters:
    - px is the position of the object on the X axis. The first point is always on the top-left.
    - py is the position of the object on the Y axis. The second point is always on the bottom-right.
    - width is the width of the object.
    - height is the width of the object
    - wallID is the wall ID, used to identify the wall when buying it. Do not use single IDs for walls, because when you buy one, you will unlock others.
    - paidValue is the value of Coins needed to purchase this wall.

void AchievementsPlus.WallManager.addPurchaseWall(u16 px, u16 py, u8 width, u8 height, u8 wallID, u8 paidValue)

- The function will spawn the wall object at the level with the specified position and parameters.
- Purchase wall differs from Session wall in that it retains its unblocked status when buying a wall, unlike Session wall, which you need to buy every time you exit the game.
- Parameters:

- px is the position of the object on the X axis. The first point is always on the top-left.
- py is the position of the object on the Y axis. The second point is always on the bottom-right.
- width is the width of the object.
- height is the width of the object
- wallID is the wall ID, used to identify the wall when buying it. Do not use single IDs for walls, because when you buy one, you will unlock others.
- paidValue is the value of Coins needed to purchase this wall.