

5 in 1 Puzzle Starter Kit

Version 1.3.2

Features

- 5 types of puzzle game
- Included character attribute system
- Included turn-base RPG battle
- Character skill will be implemented next version
- Old unity support will be implemented next version

About puzzle gameplay

There are 5 types of puzzle game included in this package

1. **Drag and Drop match 3 puzzle (Puzzle and Dragon)**, player can move a node everywhere in a board while a moving node collide with another node their position will be swap, player can solve a puzzle by match 3 or more nodes of same color to trigger character actions
2. **Classic swap match 3 puzzle**, player can swap 2 nodes by drag a node to target node which player want to swap to solve a puzzle by match 3 or more nodes of same color to trigger character actions
3. **Tap match 3 puzzle**, solve a puzzle by tap on a nodes that are beside each other 3 or more nodes of same color to trigger character actions
4. **Slide match 3 puzzle**, solve a puzzle by move an nodes in a row or column to match 3 or more nodes of same color to trigger character actions
5. **Line link match 3 puzzle**, player can drag from node to a node which is same color to link nodes that want to solve, link 3 or more nodes of same color to trigger character actions

Components explanation

CharacterAction

Is a component of character's attack, using skill actions. You can inherit this component class to create your own action for attack or skill special effects while character attacking or using skill.

Action Duration – Duration for this action

Can see /Script_Demo/ AttackAction_Demo.cs as example.

Character

Is a component of character which is part of RPG game rules, you can configuring following variables from inspector:

- **Controller Type** - Set this option follow this rule if this character is player's character set this option to "NORMAL", if this character is enemy(AI)'s character set this option to "AI" ["NETWORK" is not currently implemented]
- **Price_sell** - May be used in inventory management scene, developers can implemented as they wish
- **Price_upgrade** - May be used in inventory management scene, developers can implemented as they wish
- **Attr_level** - Minimum level of this character
- **Attr_max_level** - Maximum level of this character
- **Attr_exp** - Collected EXP(experience) points of this character on current level
- **Attr_next_exp** - EXP which is require for next level
- **Attr_total_exp** - Total collected EXP(experience) points of this character
- **Attr_star** - Star quantity, may be used in inventory management scene, developers can implemented as they wish
- **Attr_cost** - Cost used for squad setting, may be used in inventory management scene, developers can implemented as they wish
- **Attr_element** - Element of this character, if node with this element has been solved this character will attacking
- **Attr_atk** - Minimum attack damage which decreasing character's HP or player's HP
- **Attr_def** - Minimum defend which decreasing attack damage from player's character
- **Attr_rcv** - Minimum recover which recovering player's HP
- **Attr_hp** - Minimum HP (Health point)
- **Attr_max_atk** - Maximum attack damage
- **Attr_max_def** - Maximum defend
- **Attr_max_rcv** - Maximum recovery
- **Attr_max_hp** - Maximum HP
- **Fix_atk** - If it's true, current attack will be fix to minimum attack damage
- **Fix_def** - If it's true, current defend will be fix to minimum defend
- **Fix_rcv** - If it's true, current recover will be fix to minimum recover
- **Fix_hp** - if it's true, current HP will be fix to minimum HP
- **Growth_atk** - Growth rate which is used to calculate current attack
- **Growth_def** - Growth rate which is used to calculate current defend
- **Growth_rcv** - Growth rate which is used to calculate current recover
- **Growth_hp** - Growth rate which is used to calculate current HP
- **Name** - Character's name
- **Description** - Character's description
- **Icon** - Character icon which appear as player's characters
- **Portrait** - Actor which appear in battle scene as enemy
- **Skill Index** - (1-index) in "Skills" array (a GameDatabase variable) which skill will be set to

- **Current Skill Level** – Current skill level of this character
- **Skill** - CharacterSkill component
- **Turn_length** - Turn length before start attack (for AI)
- **Attack Action** – Character Action component for character attack

Node

Is a component of node which is part of puzzle game and RPG game rules, you can configuring following variables from inspector:

- **Element** - character which same element with this node will attacking if a node solved
- **Killed Effect** - effect which instantiating while destroy
- **Effect Offset** - effect's offset

BoardController, BoardControllerSwap, BoardControllerTap, BoardControllerSlide, BoardControllerLineLink

Is a component which handle all puzzle solving rules, you can configuring following variables from inspector:

- **Moving Node Offset** - Offset from finger which touched on a node
- **Grid Width** - Number of node per row
- **Grid Height** - Number of node per column
- **Cell Width** - Width of cell
- **Cell Height** - Height of cell
- **Match Count** - Minimum quantity of nodes to be solve
- **Levelling Count** - Minimum quantity of nodes to increase a node level after solve [not implemented]
- **Swap Speed** - Move speed of node while swapping position
- **Drop Speed** - Move speed of node while dropping to a board
- **Clear Delay** - Delay before solved nodes are clearing
- **Drop Delay** - Delay before dropping new nodes
- **Node Prototypes** - An nodes game object which instantiating later in a board
- **Input Camera** - A camera which use to receiving touch/click input, if leave it none "Main Camera" will be used, A orthographic camera which can see every nodes is recommend
- **Is Init On Awake** - Initializing nodes on awake
- **Is Drop Init Nodes** – Dropping nodes after board initialized, If not nodes will be appeared without dropping animation
- **Re Init If Cant Match** – Re initialize nodes if puzzle game can't playing
- **[Additional variables for BoardControllerSwap]**
 - **Reverse Delay** – Delay before invalid swapped nodes will reverse back
- **[Additional variables for BoardControllerSlide]**
 - **Is End On Touch End** – If set to true, after player left a touch from dragging nodes, player turn will end

[Additional variables for BoardControllerLineLink]

- **Can Link Difference Node**– If set to true, player can link with nodes which is not same color
- **Line Prototype**– Line renderer prototype, line which links between nodes
- **Highlight Prototype** – Highlight effect of linked nodes

- **Line Offset**– Line offset from nodes
- **Highlight Offset**– Highlight effect offset from nodes
- **Can Circuit Link On**– If linked equal or more than this value, player can link with first node as circuit link (Minimum: 3, lesser will not able for circuit link)
- **Show Linking Line**– If true, line from last node to touch position will be shown

GameManager

Is a component which handle all game rules, you can configuring following variables from inspector:

- **Board** - A board component, it's required (can be **BoardController**, **BoardControllerSwap**, **BoardControllerTap**, **BoardControllerSlide**, **BoardControllerLineLink**)
- **Characters** - An player's characters
- **Max Member Quantity** - which use for define limit team member
- **Is Init Character By Id** - if this is true character initialization by character ids.
- **Character_ids** - an player's characters ids
- **Character Container** - a transform which being contains player's characters
- **Enemy Character Container** - a transform which being contains enemy's characters
- **Character Positions** - an local positions inside "Character Container", an characters which is same index in array with this will set position following this value of same index
- **Enemy Character Positions** - an local positions inside "Enemy Character Container", an characters which is same index in array with this will set position following this value of same index
- **Area Data List** - An enemy(AI)'s characters in each area
- **Hp Bar** - HP (Health point) gauge bar
- **Time Bar** - Puzzle solving time gauge bar
- **Character Hp Bar** - HP (Health point) gauge bar (for enemy character)
- **Character Hp Bar Offset** – HP gauge bar offset
- **Character Turn Count Label** – Label for turn count before start attack (for enemy character)
- **Character Turn Count Label Offset** – Turn count label offset
- **Matching Count Down Time** - Puzzle solving duration
- **Change Attacking Team Delay** - Delay before enemy(AI)'s character start attacking
- **Change Area Delay** - Delay before moved to next area
- **Element Recovery** - Element value which declared in "Node" component that desire to use as recovery node (Recovery player's HP when matches)
- **Element Pairs** - An advantage and disadvantage elements
- **Advantage Element Damage Rate** - Character damage will be increase by this rate if its element is advantage element and target element is disadvantage

Some game events are not implemented developer can do this by inherit with "GameManager" component class and override following functions:

- **OnAreaChange (int area)** - Event trigger before area change.

- **OnWin ()** - Event trigger after all enemies died
- **OnLose ()** - Event trigger after out of player's HP
- **OnCharacterAttack (Character attacker, CharacterAction attack, int damage, Character target)** - Event trigger when character attack
- **OnCharacterSkill (Character skillUser, CharacterSkill skill, int damage, Character target)** - Event trigger when character use skill
- **OnCharacterSpawn (Character owner)** – Event trigger on character spawn to the scene
- **OnCharacterDead (Character owner)** - Event trigger after out of character's HP
- **OnCharacterTurnCountDown (Character owner)** – Event trigger on character turn counted down
- **OnCharacterSkillTurnCountDown (Character owner)** - Event trigger on character's skill turn counted down
- **OnEnemyCharacterAttack (Character attacker, int damage, Character target)** - Event trigger when character attack (for enemy character)
- **OnEnemyCharacterSkill (Character skillUser, CharacterSkill skill, Character target)** - Event trigger when character use skill (for enemy character)
- **OnEnemyCharacterSpawn (Character owner)** – Event trigger on character spawn to the scene (for enemy character)
- **OnEnemyCharacterDead (Character owner)** - Event trigger after out of character's HP (for enemy character)
- **OnEnemyCharacterTurnCountDown (Character owner)** – Event trigger on character turn counted down (for enemy character)
- **OnEnemyCharacterSkillTurnCountDown (Character owner)** - Event trigger on character's skill turn counted down (for enemy character)

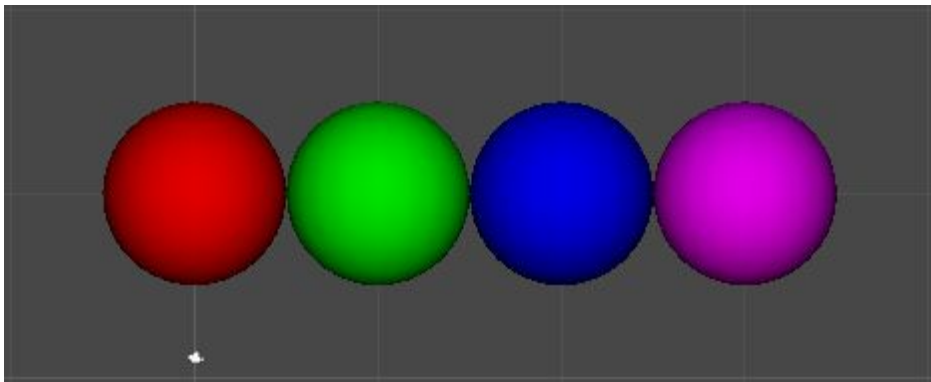
Can see /Script_Demo/GameManager_Demo.cs as example.

Setup guide

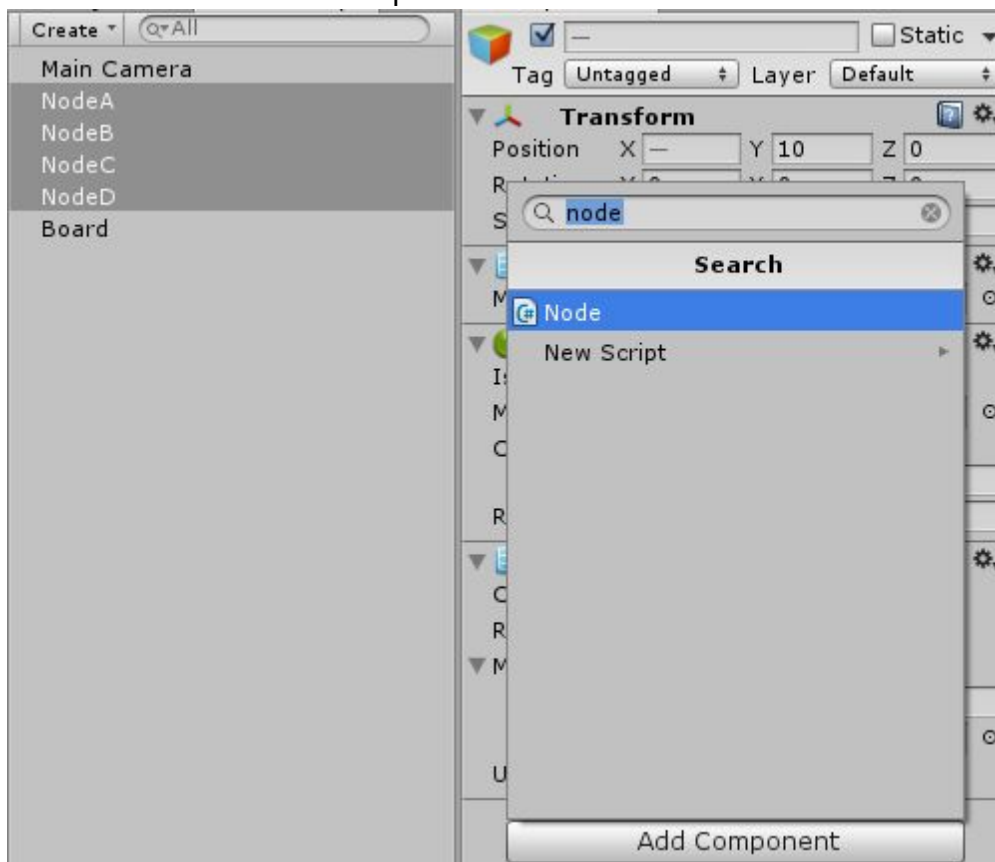
1. **Node** – You can add “Node” component to any game object which is included collider it's required for touch events and then set “Element” as you wish, an nodes you created will be used by “BoardController” component as prototype to create another nodes

Example

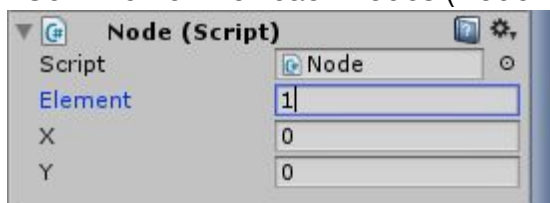
- I've create four spheres with difference color and *10 scales



- Then include “Node” component to each nodes



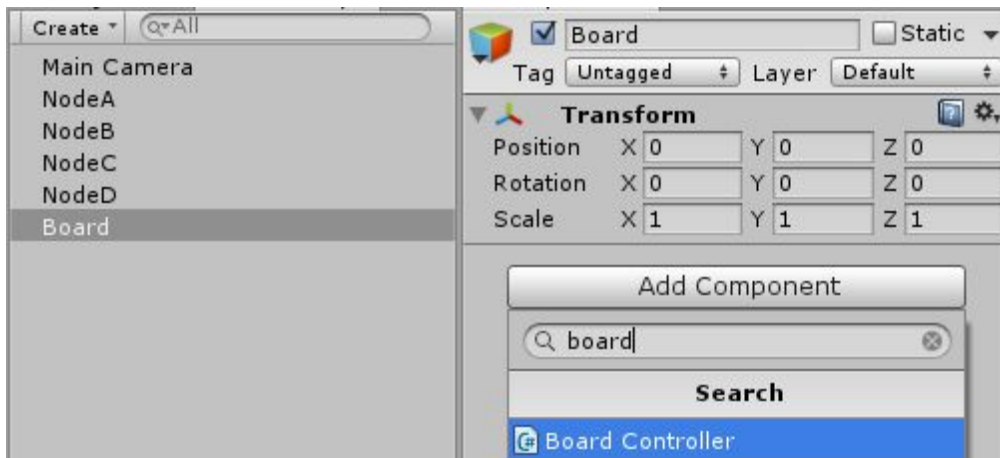
- Set “Element to” each nodes (NodeA = 1, NodeB = 2, NodeC = 3, NodeD = 4)



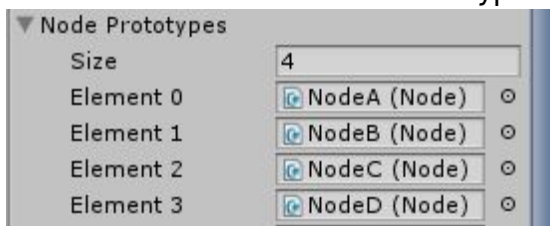
2. **BoardController** – You can add “BoardController” component to any game object then add an nodes to “Node Prototypes” and setting as you wish

Example

- I’ve create empty gameobject and then add “BoardController” component



- Place an nodes to “Node Prototypes”



- And then I’ve setting board following

----- Grid Width = 6, Grid Height = 6; Because I want 6*6 board

----- Cell Width = 12, Cell Height = 12; Because my sphere distance in this board would fit to this setting

----- Swap Speed = 100;

----- Drop Speed = 100;

----- Transform’s position = -30, -30, 0

***** These settings I have to repeat testing and setting but not so hard.

Now your puzzle game is ready to play but if you want to make Puzzle-RPG game you should learn following guide

3. **GameDatabase** - You can add “GameDatabase” component to any game object this component is collect presets of Buffs, Skills and Characters data.
4. **CharacterBuff** – You can add “CharacterBuff” component to any game object, I recommend it to be child of “GameDatabase” component then setting data as you wish and add to “GameDatabase” component, You should set size of “Turn Length” and “Values” equals to “Max Level” variable
5. **CharacterSkill** – You can add “CharacterSkill” component to any game object, I recommend it to be child of “GameDatabase” component.

Then setting data as you wish and add to “GameDatabase” component, You should set array size of “Buffs”, “Team Buffs”, “Enemy Buffs”, “Enemy Team Buffs”, “Turn Cost” and “Values” equals to “Max Level” variable and set variable for each level (from level 1 to max level).

An buffs can be included to skill by add buff index which is located in “GameDatabase” component as 1-based index (start from 1 not 0) in

- “Buffs” - Buff for the character whom use this skill
- “Team Buffs” - Buff for all characters that same team
- “Enemy Buffs” - Buff for the character whom has been a target
- “Enemy Team Buffs” - Buff for all characters that same team with target character

6. **Character** - You can add "Character" component to any game object, I recommend it to be child of "GameDatabase" component then setting data as you wish and add to "GameDatabase" component, defining character skill as skill index which is in "GameDatabase" component as 1-based index (start from 1 not 0)
7. **GameManager** – You can add "GameManager" component to any game object, then define "Board" as a board controller
 - you can set an enemy in a stage in "Area Data List" its array length is battle round, characters per element is an characters that will appearing per round defining characters as character index which is in "GameDatabase" component as 1-based index (start from 1 not 0)
 - an "Character_ids" is an player's characters define it as character index which is in "GameDatabase" component as 1-based index (start from 1 not 0), an "Character Positions" is an spawn positions inside "Character Container" that player's characters will be spawned as icon
 - an "Enemy Character Positions" is an spawn positions inside "Enemy Character Container" that enemy's characters will be spawned as portrait, index of "Enemy Character Positions" + 1 is quantity of characters that spawned and value per element is spawn position for that.

Thank you for purchase

Contact: ittipon.bay@gmail.com <Ittipon>

Demo: <http://bit.ly/1plmlqb>