Jelly Cube - Code Documentation

Main Prefabs

_Game (_Game.prefab)

- GameManager.cs
    - *NextLevelName (string)*
    - *Animator (AnimatorComponent)*
- InputManager.cs
- CubeManager.cs
    - *LevelRoot (Transform)*

_UI (_UI.prefab)

- Animator (Unity component)

Cube (CubePlayer.prefab/CubeBlue.prefab/CubeRed.prefab/CubeWall.prefab)

- CubeController.cs

CheckPoint (CheckPointGreen.prefab, CheckPointBlue.prefab, CheckPointRed.prefab)

- CheckPoint.cs
    - *CubeTag (string)*

Camera (optional controller)

- CameraController.cs
    - LookAtTransform (Transform)
    - LookAtDamping (float)
    - CameraMoveDamping (float)

Program Flow

CubeController.cs
When the level starts, this controller register itself into a List<CubeController> inside the CubeManager.cs

CubeManager.cs
When there is no cubes moving, this controller can receive a roll command from the InputManager.cs. It iterates with a List<CubeController> checking wich cubes can roll. The cubes

that are allowed to roll, are registered into a second list of cubes moving, and after the completion of all moves, this list is cleared. If there is no element in the list, means it´s ready to receive another roll input from the InputManager.cs. It will listen to all CheckPoints in scene to check if the puzzle is complete. When all checkpoints are solved and the puzzle is completed, the CubeManager will trigger a UnityEvent called OnPuzzleSolved(true). This UnityEvent will tell the GameManager the level is completed.

InputManager.cs
When a valid touch or keyboard event is detected, it sends the roll command to the CubeManager.cs

CheckPoint.cs
When the level starts, this controller is registered into an array inside the CubeManager.cs. This controller checks for any collision with a cube that has the Tag equals to its field "CubeTag". When the trigger is positive (the tags are the same), then this checkpoint is complete and the CubeManager will receive this information. There is an UnityEvent called OnCheckPointSolved, that allows the developer to implement features like visual feedbacks easily.

GameManager.cs
GameManager.cs controls the level loading, level completion, and ui. The CubeManager.cs is responsible to send this information to the GameManager.cs, using the UnityEvent OnPuzzleSolve(true).

Main Components

GameManager.cs

_Game.prefab

Fields	Type	Description
m_NextLevelName	public string	Name of the level to be loaded when this level is complete
m_Animator	public Animator	Animator component with the animated UI
m_AnimationClipsLength	Dictionary<string,float>	Dictionary of all animations found in the m_Animator, with its respective duration
m_CameraController	CameraController	CameraController reference
Instance	GameManager	Singleton instance reference

Method	Description
SetLevelComplete	if levelCompleted is true, then this method loads the LevelComplete method
ReloadLevel	Starts the ReloadLevelRoutine
ReloadLevelRoutine	coroutine that controls the UI animations and reload level
LevelComplete	coroutine that controls the UI animations and advance to next level

InputManager.cs

_Game.prefab

Fields	Type	Description
m_LockControls	bool	Ignore input events when true
m_MoveDirection	Vector2	Touch or Keyboard direction
m_Moved	bool	If moved and still touching is true
m_TouchHoldTimer	float	Hold time duration
Instance	InputManager	Singleton instance reference
TOUCH_HOLD_TIME_LIMIT	const float	Limit to start moving continuously
TOUCH_SENSIBILITY	const float	Touch sensibility to roll a cube

Method	Description
LockControls	Disable input controls
UnlockControls	Enable input controls
LateUpdate	Where all input events are calculated

CubeManager.cs

_Game.prefab

Fields	Type	Description
m_LevelRoot	Transform	GameObject that contains all level and cubes

m_OnPuzzleSolved	public UnityEvent<bool>	Dispatch tru/false when the level is complete (by default it sets the level complete in the GameManager)
m_CubeControllers	List<CubeController>	All CubeControllers in scene
m_Checkpoints	CheckPoint[ ]	Array of all scene checkpoints
m_CubeMoving	List<CubeController>	All moving CubeControllers in scene
Instance	CubeManager	Singleton instance reference

Method	Description
Register	Add a CubeController into m_CubeControllers list
RegisterMove	Add a moving CubeController into m_CubeMoving list
UnregisterMove	Remove a CubeController from m_CubeMoving list
Move	Try to move all CubeControllers registered in m_CubeControllers

CubeController.cs

CubePlayer.prefab, CubeBlue.prefab, CubeRed.prefab, CubeWall.prefab

Fields	Type	Description
m_Cube	public Collider	main cube object collider
m_CanControl	public bool	Cube can move after Input event
m_PushCubeType	public enum PushType - DontPushCubes - PushCubesWhenMove - PushCubesAfterMove	Cube can push another cube when start a move or finish a move.
m_CanShake	public bool	Cube shakes after collision.
m_CanBePushed	public bool	Cube will move after a moving cube collision
m_MoveType	public enum MoveType - Roll - Slide	Choose the move animation between roll or slide

m_MoveSpeed	public float	Move speed
m_OnMoveStart	public UnityEvent	Dispatched when cube starts to move
m_OnMoveStart	public UnityEvent	Dispatched when cube movement ends
m_OnMoveByImpactSuccess	public UnityEvent	Dispatched when cube receive a push force and can move
m_OnMoveByImpactFail	public UnityEvent	Dispatched when cube receive a push force but can't move
m_LastMove	Vector3	Last move position
m_LastInputDirection	Vector3	Last direction
m_Pivot	Transform	Rotation pivot created at runtime
m_Contents	Transform	Safe container to the children objects inside the cube
m_CubeSize	float	Stores the cube size on level start
SHAKE_SCALE	const float	Shake scale intensity

Method	Description
DoMove	Try to move the cube. Will call DoRoll or DoSlide according with m_MoveType.
DoRoll	if m_MoveType = Roll, the cube will to tween a roll animation
DoSlide	if m_MoveType = Slide, the cube will to tween a slide animation
DoPush	Try to move another cube.
GetCubeDirection	Get Cube direction using the root transform as LevelRoot
CheckCollisionRecursive	Check if a neighbor cube also has a collision, recursively.
DoShake	if m_CanShake is true, when the cube is hitted by a second cube, it will shake
OnMoveComplete	Called after any movement completion
GetPositionOnFloor	Get center of the cube, but at the pivot level.
ResetPosition	Make sure the cube is on a flat grid rotation and position
RoundVector	Round a Vector and span it to an angle value or grid position

CheckPoint.cs

CheckPointGreen.prefab, CheckPointBlue.prefab, CheckPointRed.prefab

Fields	Type	Description
m_CubeTag	public string	cube tag necessary to make the checkpoint complete
m_OnCheckPointSolved	public UnityEvent<bool>	Dispatch an event when the completion state of this check point is changed

Method	Description
SetCheckpointSolved	Update the checkpoint state when a valid cube interacts through OnTriggerEnter and OnTriggerExit

Helper Components

CameraController.cs

CameraController.cs is a MonoBehaviour that when attached to a camera, make it look smoothly to selected transform target.

Tweener.cs

Twenner.cs is a small and very simple twenner system developed to this game. It can be replaced for any twenner you like.

CubeParticleEmitter.cs

CubeParticleEmitter.cs is a simple script to play the ParticleSystem containing trails and splashs.

Sound.cs

Sound.cs is a basic MonoBehaviour component with an AudioSource control and a command “DontDestroyOnLoad” to make this object always active until the end of the game.

RubberEffect.cs

RubberEffect.cs is a script that changes the vertex of an object in runtime and apply a rubber behaviour on it. This is a modified version of the previous script that worked using vertex colors (http://rodrigopegorari.com/blog/?p=58). This version was modified to be rigid on bottom of the object, and soft on upper side, like it should do in real life.