

CANDICE AI FOR GAMES

CHANGELOG



CHANGELOG

Document Version: 2.0

Released: 2023-01-16

Comments:

This is a major release of Candice AI; with almost every aspect of its makeup, functioning and implementation-time significantly improved.

This is not an API document, and thus only mentions the namespace(s), types, methods and functions implemented in v2 that are new, and those from earlier versions that have been modified. See API Document for a detailed description of how you can inherit from v2.



Table of Contents

V2	4
New Animations Management System:	4
Libs > Candice Animations Management >.....	4
Modules > CandiceModuleAnimations.cs (:CandiceCreateAnimationActions) >	4
Additions made to:.....	5
public class CandiceAIController : CandiceAnimationManager	5
Fields made public:	5
New fields:	5
New proprieties exposed as:	6
Modifications made to Die and OnCollisionEnter methods.	6
CandiceAIPlayerController.cs.....	7
CandiceProjectile.cs	7
New stand-alone only scripts added:	7
CandiceAnimationManager	7
CandiceSceneBuilder	7
EnemySpawners.....	7
genericDrop (empty).....	7
TimedDestroy.....	7
TrigerNextScene.....	7
CandiceDrop.....	7
Base improvements over Candice AI V1.2.1:.....	8
PRE V2:	9
V1.2.1	9
V1.2.0	9
V1.1.0	9
V1.0.4	10
v1.0.3.....	10



v1.0.2..... 11

v1.0.1..... 11

v1.0.0..... 11



V2

New Animations Management System:

Libs > Candice Animations Management >

1. Actions (Defines animation actions into standard and rigged categories such CandiceStandardActions, CandiceHumanoidMelee, CandiceHumanoidRanged, CandicePlayerOverrides) etc.
2. Camera (Uses 3rd party Smooth Camera Shaker free asset and defines various Camera Techniques, such as KillCam)
3. Handicaps (used to inflict Bleed, Daze and other detrimental Player effects)
4. Inventory (base prototype for inventory system, currently only supports CandiceDrop)
5. Middleware (type checking middleware, under-construction)
6. Possessor (Fast PlayerController switching built for CandiceAIPlayerController, allows the use of a static CandiceAIPlayerController instance to be assigned to any CandiceAIController agent.)
7. UI (Uses TextMeshPro Base). Health indicator, music player, save file manager (autosave & load), SceneBuilder Reset, UI indicators and more.

Modules > CandiceModuleAnimations.cs (:CandiceCreateAnimationActions) >

1. AnimationController(ModellImporter)
2. GetTriggers()
3. CheckTrigger(AnimatorControllerParameter)
4. ModifyParameter(int, string)
5. AddParameter(AnimatorController, string, AnimatorControllerParameterType)



Additions made to:

`public class CandiceAIController : CandiceAnimationManager`

Fields made public:

```
public float moveSpeed;
public float attackDamage;
public GameObject projectile;
public bool isAttacking;
```

New fields:

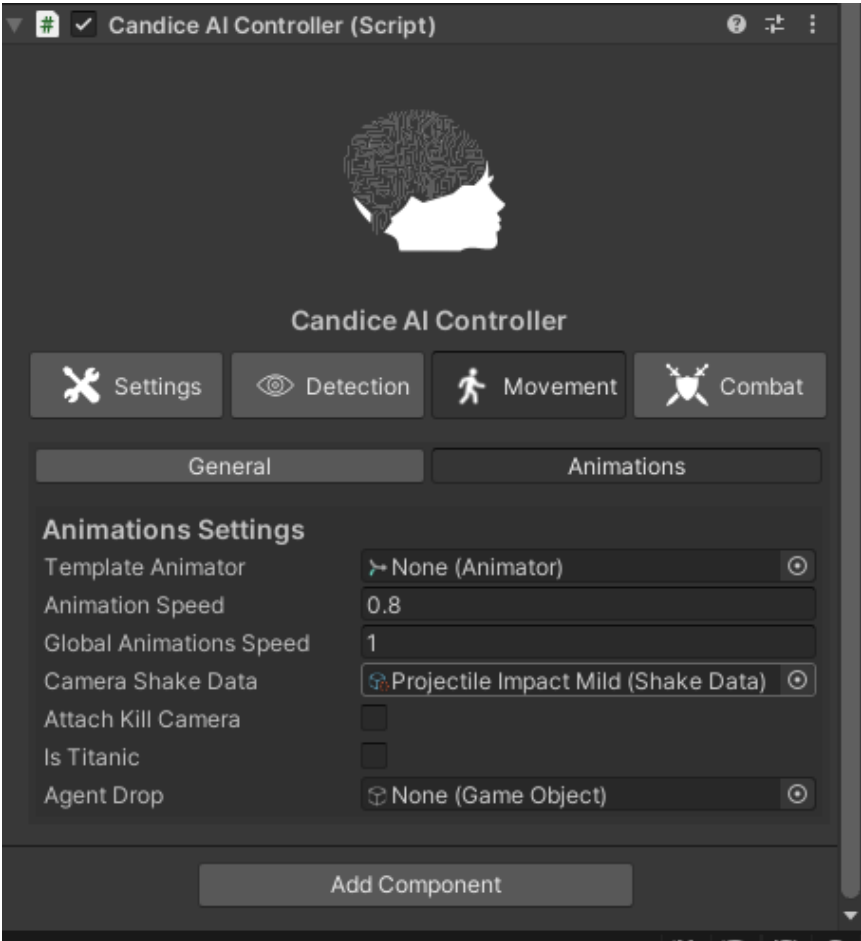
```
/*
 * New Animation Variables
 */
[SerializeField]
private Camera mainCamera;
[SerializeField]
private GameObject rig;

//New Animation System fields
[SerializeField]
public Animator templateAnimator;

/*
 * End Of New Animation Variables
 */
```



New proprieties exposed as:



Modifications made to Die and OnCollisionEnter methods.



CandiceAIPlayerController.cs

Built-in targeting buffer for dynamic camera, dynamic camera support, possessor support and various other features that do not break previous releases of the script, simply add functionality.

CandiceProjectile.cs

All propeties made public.

New stand-alone only scripts added:

CandiceAnimationManager

CandiceSceneBuilder

EnemySpawners

genericDrop (empty)

TimedDestroy

TrigerNextScene

CandiceDrop



Base improvements over Candice AI V1.2.1:

Modules > CandiceModuleMovement (:CandiceBaseModule) > LookAt(Transform, CandiceAIController)

```
public void LookAt(Transform transform, CandiceAIController aiController)
{
    //to be deprecated
    //transform.LookAt(aiController.MainTarget.transform);

    //need to grab attached collider to determine object height
    BoxCollider box = transform.gameObject.GetComponent<BoxCollider>();
    CapsuleCollider capsule = transform.gameObject.GetComponent<CapsuleCollider>();

    //buffer the y position for the transform.LookAt with the agent height.
    //This determines (inner-eye) level, which in turn helps determine realistic look direction (the agent will always try to look you in the eyes)
    if (box != null)
    {
        transform.LookAt(new Vector3(aiController.MainTarget.transform.position.x,
            aiController.MainTarget.transform.position.y + box.size.y,
            aiController.MainTarget.transform.position.z));
    }
    else if (capsule != null)
    {
        transform.LookAt(new Vector3(aiController.MainTarget.transform.position.x,
            aiController.MainTarget.transform.position.y + capsule.height,
            aiController.MainTarget.transform.position.z));
    }
}
```

LookAt now determines inner-eye level. Support for box and capsule colliders currently, with advanced collider support coming in next release.

Note that: Inner-eye level technology depends on the physics engine of Unity. It allows to determine what height you are at, as a player, a target, on various elevation terrain and have the AI agent look in that exact direction. The dependency on colliders means that your colliders have to be properly defined and constrained around your agent. If undesired behaviour occurs, tweak the Y size of your box or height of your capsule colliders.



PRE V2:

V1.2.1

1. Updated CandiceBehaviorTreeMono
 - a. Added method InitializeBehaviorTree()
 - b. Added variable prevSelectedBT.
2. Added CandiceBasicSaveSystem.cs
3. Added prototype

V1.2.0

1. Updated CandiceAIController
 - a. Added NavMeshPathfind() function.
 - b. Added navMeshAgent variable.
 - c. Changed targetPosOld to initialise with Vector3.zero.
2. Updated CandiceBehaviorNodeS
 - a. Changed variable 'function' from type integer to string.
3. Updated CandiceBehaviorTree
 - a. Added function GetNameFromIndex().
4. Updated CandiceBehaviorTreeMono
 - a. Added SelectedBehaviorTree enum.
 - b. Added public property selectedBehaviorTree.

V1.1.0

1. Updated CandiceAIController
 - a. Fixed 2D attack bug.
2. Updated CandiceGrid
 - a. Added 2D support.
3. Updated CandiceDefaultBehaviors
 - a. Added MoveTo() function.
 - b. Added Patrol() function
 - c. Added Orbit() function
 - d. Added Seek() function
 - e. Added Rotate() function
4. Updated CandiceModuleMovement
 - a. Added MoveTowards2D() function



- b. Added Orbit() function
 - c. Added Seek() function
 - d. Added Rotate() function
- 5. Updated CandiceBehaviorTreeMono
 - a. Added SelectedBehaviorTree enum.
- 6. Added CandiceBehaviorTreeMono_Editor.cs
- 7. Added BasicPlayerController.cs

V1.0.4

- 1. Updated CandiceAIController
 - a. Fixed Attack/Detection bug. Edited onObjectFound callback method. Clear detection lists at the start of the function.
 - b. Added Die() function. Called by the ReceiveDamage() function when the agents health drops to or below 0.
 - i. Destroys the gameobject.
 - c. Renamed "ReceiveDamage" function to "CandiceReceiveDamage", including all references.
- 2. Updated CandiceBehaviorTree
 - a. Changed "btName" variable name to "_name".

v1.0.3

- 1. Updated CandiceAIManager
 - a. Fixed CandiceTile bug
- 2. Updated CandiceAIController
 - a. Removed Callback parameter (_onAttackComplete) from DealTimedDamage and FireProjectile methods/functions.
 - i. The callback that was registered in the constructor of the CandiceModuleCombat class is now the only one used.
 - b. Removed onAttackComplete() function in AIController.
 - i. onAttackComplete(bool success) is still available in the CandiceAIController.



v1.0.2

1. Updated Combat Module.
 - a. Fixed damage calculation bug.
2. Updated Candice Projectile
 - a. Fixed Damage Bug when SendMessage() is called.
3. Updated Detection Module
 - a. Moved “perceptionMask” in the Detection GUI tab from Detection Settings to Obstacle Avoidance Settings.
 - b. Changed “perceptionMask” name to “obstacleAvoidancePerceptionMask”.
 - c. Removed “Detection Tags” label from the Detection Tab GUI.

v1.0.1

N/A

v1.0.0

N/A