

How to Edit and Test Snapse Source Code

1. Download Unity Editor (easiest way)
2. Open Unity Editor
3. Open the SNP-Editor/Assets/Scripts folder and select the file you need to modify
4. Press the play button in the upper portion of the Unity Editor to test the program
5. When done, go to File > Build Settings and set the Target Platform
6. Press Build
7. Open the build to test

1. Feature: Firing One Step

- a. *GUI* Forward button
- b. *EditorController.StartFire()*
- c. *EditorController.FireOneStep()*
 - i. *NeuronsController.Fire()*
 1. *NeuronController.StartFire()*
 - a. *NeuronController.CheckRules()*
 - i. *NeuronController.ProcessRule()*
 1. *NeuronController.StoreRule()* OR
NeuronController.CloseNeuron()
 - ii. *EditorController.CheckHalt()*
 - iii. *EditorController.StopContinuous* OR *EditorController.WaitForGuided()*
 1. *Case WaitForGuided()*: *EditorController.CreateGuidedMenus()*
 - a. *GuidedMenuController.SetUpMenu()*
 - i. *GuidedMenuController.SetNeuronNo()*
 - ii. *GuidedMenuController.AddRules()*
 1. *RulePrefabController.SetRule()*
- d. *EditorController.EndFire()*
 - i. *NeuronsController.EndFireNeurons()*
 1. *NeuronController.EndFire()*
 - a. *NeuronController.Fire()*
 - i. *NeuronController.Receive()*
 - ii. *NeuronController.AnimateFire()*
 - ii. *NeuronsController.UpdateOutputNeurons()*
 1. *NeuronController.UpdateOutput()*
 - iii. *EditorController.AddChoiceElement()*

2. Feature: Continuous Firing

- a. *GUI* Play button
- b. *EditorController.PlayButton()*
- c. *EditorController.StartContinuous()*
- d. *EditorController.ContinuousFire()*
- e. *EditorController.StartFire()*
- f. *EditorController.FireOneStep()*
 - i. *NeuronsController.Fire()*

- ii. *EditorController.CheckHalt()*
- iii. *EditorController.StopContinuous* OR *EditorController.WaitForGuided()*
 - 1. *Case WaitForGuided(): EditorController.CreateGuidedMenus()*
 - a. *GuidedMenuController.SetUpMenu()*
 - i. *GuidedMenuController.SetNeuronNo()*
 - ii. *GuidedMenuController.AddRules()*
 - 1. *RulePrefabController.SetRule()*
- g. *EditorController.EndFire()*
 - i. *NeuronsController.EndFireNeurons()*
 - 1. *NeuronController.EndFire()*
 - a. *NeuronController.Fire()*
 - i. *NeuronController.Receive()*
 - ii. *NeuronController.AnimateFire()*
 - ii. *NeuronsController.UpdateOutputNeurons()*
 - 1. *NeuronController.UpdateOutput()*
- iii. *EditorController.AddChoiceElement()*

3. Feature: Going Back One Step

- a. GUI Back button
- b. *EditorController.GoBackOne()*
 - i. *EditorController.SetAllSpikes()*
 - ii. *EditorController.SetAllConfig()*
 - iii. *NeuronsController.RetractOutput()*
 - 1. *NeuronController.Retract()*

4. Feature: Choosing a Rule

- a. GUI Rule button
- b. *RulePrefabController.SendRule()*
- c. *GuidedMenuController.ChoiceMade()*
 - i. *EditorController.SetGuidedChoice()*

5. Feature: New Neuron

- a. GUI 'New Neuron'
- b. *EditorController.NewNeuron()*

6. Feature: New Output Neuron

- a. GUI 'New Output'
- b. *EditorController.NewNeuron()*

7. Feature: Create a New Synapse

- a. Enter New Synapse Mode
 - i. GUI 'New Synapse'
 - ii. *EditorController.NewSynapseToggle()*
 - 1. *EditorController.NewSynapseOn()*
 - a. *EditorController.SetStatusText()*
 - b. *EditorController.SetFreeMode()*

- c. EditorController.NewSynapseStart()
 - d. NeuronsController.NewSynapseMode()
 - i. NeuronController.NewSynapseModeReceiver()
 - ii. NeuronController.SynapseV1ModeReceiver()
 - iii. EditorController.SetStatusText()
 - b. Select source neuron for synapse
 - i. GUI Neuron
 - ii. NeuronController.OnMouseDown()
 - iii. NeuronsController.SynapseCoordinate1()
 - iv. NeuronController.SynapseV1ModeReceiver()
 - v. NeuronController.SynapseV2ModeReceiver()
 - vi. EditorController.SetStatusText()
 - c. Select target neuron for synapse
 - i. GUI Neuron
 - ii. NeuronController.OnMouseDown()
 - iii. NeuronsController.SynapseCoordinate2()
 - iv. NeuronController.SynapseV1ModeReceiver()
 - v. NeuronController.NewSynapseModeReceiver()
 - vi. EditorController.NewSynapse()
 - vii. Check if neuron is valid target
 - 1. If invalid
 - a. EditorController.SetStatusText()
 - b. EditorController.NewSynapseError()
 - 2. if valid
 - a. EditorController.InstantiateSynapse()
 - b. if synapse is not loaded from a file
 - i. EditorController.NewSynapseEnd()
 - 1. EditorController.SetStatusText()
 - 2. EditorController.NewSynapseStart()
 - a. //Starts a new synapse creation cycle
 - d. Feature: Exit New Synapse Mode
 - i. GUI 'New Synapse'
 - ii. EditorController.NewSynapseToggle()
 - 1. EditorController.NewSynapseOff()
 - a. EditorController.SetStatusText()
 - b. NeuronsController.NewSynapseMode()
 - i. NeuronController.NewSynapseModeReceiver()
 - ii. NeuronController.SynapseV1ModeReceiver()
 - c. EditorController.SetFreeMode()
- 8. Feature: Edit Neuron**
- a. Enter Edit Neuron Mode
 - i. GUI 'Edit Neuron'
 - ii. EditorController.EditNeuronToggle()

1. EditorController.EditNeuronOn()
 - a. EditorController.SetFreeMode()
 - b. EditorController.EditNeuronStart()
 - i. NeuronsController.EditNeuronMode()
 1. NeuronController.EditNeuronModeReceiver()
- b. Edit a neuron's rules
 - i. GUI Neuron Rule Container
 - ii. NeuronController.EditRules()
 1. EditorController.EditNeuron()
 - a. EditorController.EditRulesStart()
 - i. EditorController.EditRules()
 1. NeuronController.GetRules()
 - iii. Confirm edit
 1. GUI 'Confirm' in Rules Window
 2. EditorController.EditRulesConfirm()
 - a. NeuronController.SetRules()
 - b. SetStatusText()
 - iv. Cancel edit
 1. GUI 'Cancel' in Rules Window
 2. EditorController.EditRulesCancel()
 - a. SetStatusText()
- c. Edit a neuron's spikes
 - i. GUI Neuron Spike Container
 - ii. NeuronController.EditSpikes()
 1. EditorController.EditNeuron()
 - a. EditorController.EditSpikesStart()
 - i. EditorController.EditSpikes()
 1. NeuronController.GetSpikesNum()
 - iii. Confirm edit
 1. GUI 'Confirm' in Spikes Window
 2. EditorController.EditRulesConfirm()
 - a. NeuronController.SetSpikes()
 - b. SetStatusText()
 - iv. Cancel edit
 1. GUI 'Cancel' in Spikes Window
 2. EditorController.EditRulesCancel()
- d. Exit Edit Neuron Mode
 - i. GUI 'Edit Neuron'
 - ii. EditorController.EditNeuronToggle()
 1. EditorController.EditNeuronOff()
 - a. EditorController.SetFreeMode()
 - b. EditorController.EditNeuronEnd()
 - i. NeuronsController.EditNeuronMode()

1. NeuronController.EditNeuronModeReceiver()

9. Feature: Delete Neuron

- a. Enter Delete Neuron Mode
 - i. GUI Delete Neuron
 - ii. EditorController.DeleteNeuronToggle()
 1. EditorController.DeleteNeuronOn()
 - a. EditorController.SetStatusText()
 - b. EditorController.SetFreeMode()
 - c. DeleteNeuronStart()
 - i. NeuronsController.DeleteNeuronMode()
 1. NeuronController.DeleteNeuronModeReceiver()
- b. Select Neuron Target
 - i. GUI Neuron
 - ii. NeuronController.OnMouseDown()
 1. NeuronsController.DeleteNeuronTarget()
 - a. EditorController.DeleteNeuron()
 - i. EditorController.removeConnectedSynapses()
 1. EditorController.RemoveOutSynapseFromSource()
 - a. NeuronController.DeleteOutSynapse()
 - ii. Destroy()
 - iii. EditorController.DeleteNeuronEnd()
 1. SetStatusText()
- c. Exit Delete Neuron Mode
 - i. GUI 'Delete Neuron'
 - ii. EditorController.DeleteNeuronToggle()
 1. EditorController.DeleteNeuronOff()
 - a. EditorController.SetFreeMode()
 - b. NeuronsController.DeleteNeuronMode()
 - i. NeuronController.DeleteNeuronModeReceiver()

10. Feature: Delete Synapse

- a. Enter Delete Synapse Mode
 - i. GUI 'Delete Synapse'
 - ii. EditorController.DeleteSynapseToggle()
 1. EditorController.DeleteSynapseOn()
 - a. EditorController.SetFreeMode()
 - b. EditorController.DeleteSynapseStart()
 - i. SynapsesController.DeleteSynapseMode()
 1. SynapseController.DeleteSynapseModeReceiver()

- a. DeleteButtonController.SetSynapse()
)
 - b. Select Synapse to delete
 - i. GUI 'X' Button on synapse
 - ii. DeleteButtonController.DeleteButtonPress()
 - 1. EditorController.DeleteSynapse()
 - a. SynapseController.GetSourceNeuron()
 - b. SynapseController.GetDestNeuron()
 - c. NeuronController.DeleteOutSynapse()
 - d. Destroy()
 - e. EditorController.DeleteSynapseEnd()
 - i. EditorController.SetStatusText()
 - ii. EditorController.DeleteSynapseStart()
 - 1. //Starts a new deletion cycle
 - c. Exit Delete Synapse Mode
 - i. GUI 'Delete Synapse'
 - ii. EditorController.DeleteSynapseToggle()
 - 1. EditorController.DeleteSynapseOff()
 - a. EditorController.SetFreeMode()
 - b. SynapsesController.DeleteSynapseMode()
 - i. SynapseController.DeleteSynapseModeReceiver()

11. Feature: Save Menu

- a. Open Save Menu
 - i. GUI 'Save' Button
 - ii. SaveMenuController.OpenSaveMenu()
 - 1. EditorController.DisableButtonsAll();
 - a. EditorController.DisableControlButtons();
 - b. EditorController.DisableMiscButtons();
 - c. EditorController.DisableSimulationButtons();
 - d. EditorController.DisableViewButtons();
 - 2. EditorController.DisableNonInteractable();
 - 3. EditorController.SetDragMode();
- b. Changing saving file path
 - i. GUI filename inputfield
 - ii. GUI 'Save' button
 - iii. SaveMenuController.Save()
 - 1. SaveMenuController.ValidPath()
 - a. if invalid
 - i. SaveMenuController.ErrorNotification()
 - b. if valid
 - i. SaveMenuController.SaveNotification()
 - 1. SaveMenuController.DisableNotif()
 - ii. SaveMenuController.UpdateAutoSavePath()
 - 1. EditorController.ChangeAutoSavePath()

- a. EditorController.AutoSave()
 - i. EditorController.EncodeToFormat()
 - ii. EditorController.SetStatusText()
 - iii. EditorController.AutoSaveNotify()
 - c. Close Save Menu
 - i. GUI Save Menu 'x' button or 'Cancel' button
 - ii. SaveMenuController.CloseSaveMenu()
 - 1. EditorController.EnableButtonsAll();
 - a. EditorController.EnableControlButtons();
 - b. EditorController.EnableMiscButtons();
 - c. EditorController.EnableSimulationButtons();
 - d. EditorController.EnableViewButtons();
 - 2. EditorController.EnableNonInteractable();
 - 3. EditorController.SetDragMode();

12. Feature: Load

- a. Open Load Menu
 - i. GUI 'Load' button
 - ii. LoadMenuController.OpenLoadMenu
 - 1. EditorController.DisableButtonsAll();
 - a. EditorController.DisableControlButtons();
 - b. EditorController.DisableMiscButtons();
 - c. EditorController.DisableSimulationButtons();
 - d. EditorController.DisableViewButtons();
 - 2. EditorController.DisableNonInteractable();
 - 3. EditorController.SetDragMode();
- b. Load
 - i. File.Exists()
 - 1. if file exists
 - a. EditorController.LoadFromPath()
 - i. EditorController.DecodeFromFormat()
 - ii. LoadMenuController.CloseLoadMenu()
 - 2. if file does not exist
 - a. LoadMenuController.ErrorNotifcation()
 - i. LoadMenuController.DisableErrorNotif()
- c. Close Load Menu
 - i. GUI Load Menu 'x' button or 'Cancel' button
 - ii. LoadMenuController.CloseSaveMenu()
 - 1. EditorController.EnableButtonsAll();
 - a. EditorController.EnableControlButtons();
 - b. EditorController.EnableMiscButtons();
 - c. EditorController.EnableSimulationButtons();

- d. `EditorController.EnableViewButtons();`
- 2. `EditorController.EnableNonInteractable();`
- 3. `EditorController.SetDragMode();`

13. Feature: Choice History

*Note: Choice History is updated only through firing, specifically at `EndFire()`

- a. `EditorController.EndFire()`
 - i. `EditorController.AddChoiceElement()`
 - 1. `ChoiceNode()`

14. Feature: Zoom in

- a. GUI 'Zoom In'
- b. `ZoomController.ZoomIn()`

15. Feature: Zoom out

- a. GUI 'Zoom Out'
- b. `ZoomController.ZoomOut()`

16. Feature: Pan

17. Feature: Hide buttons

- a. Hide buttons
 - i. `EditorController.HideButtonsToggle()`
- b. Show buttons
 - i. `EditorController.HideButtonsToggle()`

18. Feature: Help

- a. Open Help Menu
 - i. GUI (?) button
 - ii. `EditorController.HelpMenuOpen()`
 - 1. `EditorController.SetFreeMode();`
 - 2. `EditorController.DisableButtonsAll();`
 - a. `EditorController.DisableControlButtons();`
 - b. `EditorController.DisableMiscButtons();`
 - c. `EditorController.DisableSimulationButtons();`
 - d. `EditorController.DisableViewButtons();`
 - 3. `EditorController.DisableNonInteractable()`
- b. Open Tutorial
 - i. GUI 'View Tutorial' button
 - ii. `TutorialController.OpenTutorial()`
- c. Close Help Menu
 - i. Help Menu 'x' button
 - ii. `EditorController.HelpMenuClose()`
 - 1. `EditorController.EnableButtonsAll();`
 - a. `EditorController.EnableControlButtons();`
 - b. `EditorController.EnableMiscButtons();`
 - c. `EditorController.EnableSimulationButtons();`
 - d. `EditorController.EnableViewButtons();`

2. EditorController.EnableNonInteractable();

19. Feature: Settings

- a. Open Settings Menu
 - i. EditorController.SettingsMenuOpen()
 - 1. EditorController.SetFreeMode();
 - 2. EditorController.DisableButtonsAll();
 - a. EditorController.DisableControlButtons();
 - b. EditorController.DisableMiscButtons();
 - c. EditorController.DisableSimulationButtons();
 - d. EditorController.DisableViewButtons();
 - 3. EditorController.DisableNonInteractable()
- b. Pseudorandom Nondeterminism
 - i. GUI Pseudorandom Nondeterminism Radio Button
 - ii. EditorController.SetToPseudorandomMode()
- c. Guided Nondeterminism
 - i. GUI Guided Nondeterminism Radio Button
 - ii. EditorController.SetToGuidedMode()
- d. Show Rules
 - i. GUI Show Rules checkbox
 - ii. EditorController.ChangeShowRulesMode()
- e. Show Labels
 - i. GUI Show Labels checkbox
 - ii. EditorController.ChangeShowLabelsMode()
- f. Enable Animations
 - i. GUI Enable Animations checkbox
 - ii. ChangeEnableAnimationMode()
- g. Close Settings Menu
 - i. GUI Settings 'x' button
 - ii. EditorController.SettingsMenuClose()
 - 1. EditorController.EnableButtonsAll();
 - a. EditorController.EnableControlButtons();
 - b. EditorController.EnableMiscButtons();
 - c. EditorController.EnableSimulationButtons();
 - d. EditorController.EnableViewButtons();
 - 2. EditorController.EnableNonInteractable();

20. Feature: Tutorial

- a. Close Tutorial
 - i. GUI Tutorial Menu 'x' button
 - ii. TutorialController.CloseTutorial()
- b. Next Page
 - i. GUI > button
 - ii. TutorialController.NextPage()
- c. Previous Page
 - i. GUI < button

- ii. TutorialController.PrevPage()
- d. Show at startup
 - i. GUI 'Show at startup' checkbox
 - ii. TutorialController.ToggleShowAgain()

21. Function: EditorController.DecodeFromFormat()

- a. EditorController.BlankSlate()
 - i. EditorController.DeleteAllNeurons()
 - 1. EditorController.DeleteNeuron()
 - a. EditorController.RemoveConnectedSynapses()
 - i. EditorController.RemoveOutSynapseFromSource()
 - b. EditorController.DeleteNeuronEnd()
 - i. SetStatusText()
- b. No neurons error
 - i. EditorController.BlankSlate()
 - ii. EditorController.InvalidFileNotify()
- c. EditorController.NewNeuron()
- d. EditorController.NewOutputNeuron()
- e. EditorController.NewSynapse()
- f. NeuronController.SetSpikes()
- g. NeuronController.SetRules()
 - i. EditorController.ValidateRules()
 - 1. SetStatusText()
- h. NeuronController.SetOutSynapses()
- i. NeuronController.SetDelay()
- j. NeuronController.SetStoredGive()
- k. NeuronController.SetStoredConsume()
- l. NeuronController.SetToOutputNeuron()
- m. NeuronController.SetStoredReceived()
- n. NeuronController.SetBitString()

22. Function: EditorController.EncodeToFormat()

- a. NeuronController.isOutputNeuron()
- b. NeuronController.GetSpikesNum()
- c. NeuronController.GetRules()
- d. NeuronController.GetOutSynapses()
- e. NeuronController.isOutputNeuron()
- f. NeuronController.GetDelay()
- g. NeuronController.GetStoredGive()
- h. NeuronController.GetStoredConsume()
- i. NeuronController.isOutputNeuron()
- j. NeuronController.GetStoredReceived()
- k. NeuronController.GetBitString()
- l.