# TDD

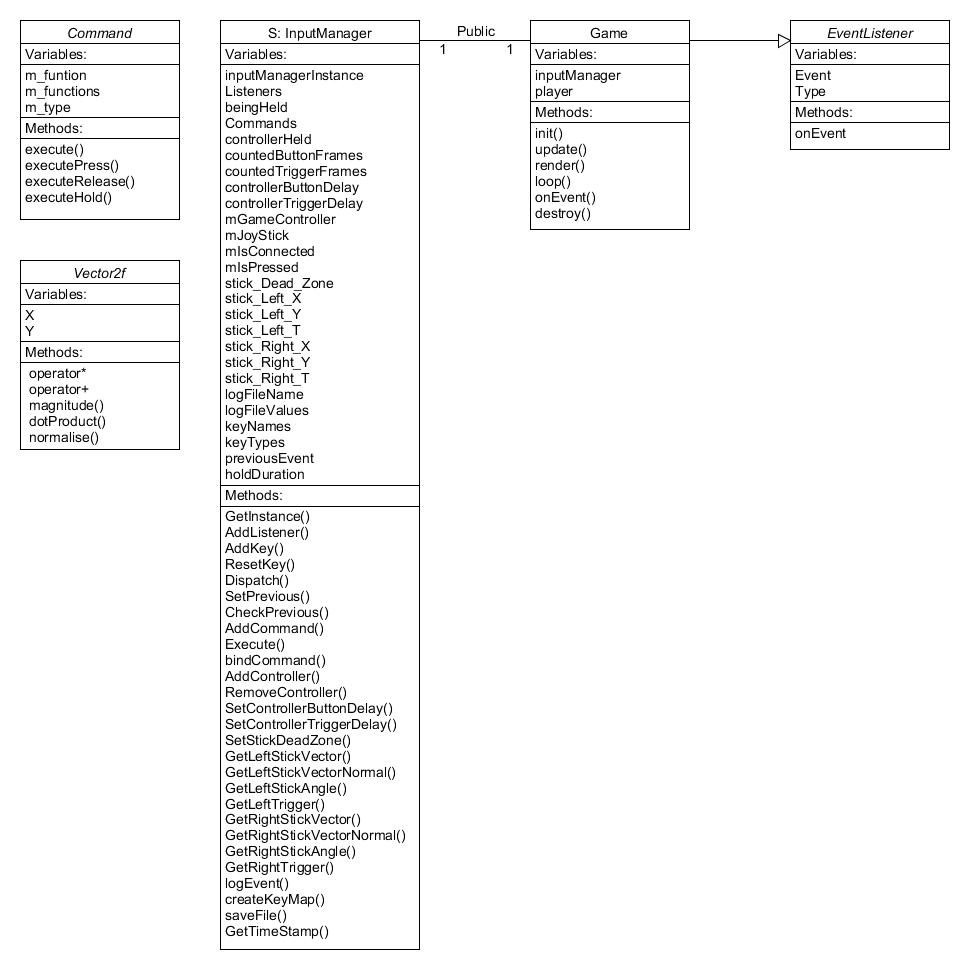
## CRC Cards:

|  |  |
| --- | --- |
| Event Listener | |
| onEvent() | Event  Type |

|  |  |
| --- | --- |
| Command | |
| execute()  executePress()  executeRelease()  executeHold() | m\_funtion  m\_functions  m\_type |

|  |  |
| --- | --- |
| Input Manager | |
| GetInstance()  AddListener()  AddKey()  ResetKey()  Dispatch()  SetPrevious()  CheckPrevious()  AddCommand()  Execute()  bindCommand()  AddController()  RemoveController()  SetControllerButtonDelay()  SetControllerTriggerDelay()  SetStickDeadZone()  GetLeftStickVector()  GetLeftStickVectorNormal()  GetLeftStickAngle()  GetLeftTrigger()  GetRightStickVector()  GetRightStickVectorNormal()  GetRightStickAngle()  GetRightTrigger()  logEvent()  createKeyMap()  saveFile()  GetTimeStamp() | inputManagerInstance  Listeners  beingHeld  Commands  controllerHeld  countedButtonFrames  countedTriggerFrames  controllerButtonDelay  controllerTriggerDelay  mGameController  mJoyStick  mIsConnected  mIsPressed  stick\_Dead\_Zone  stick\_Left\_X  stick\_Left\_Y  stick\_Left\_T  stick\_Right\_X  stick\_Right\_Y  stick\_Right\_T  logFileName  logFileValues  keyNames  keyTypes  previousEvent  holdDuration |

## Class Diagram:



## Features:

### AddKey / BindKey / (AddListener / Dispatch) / (AddCommand/ Execute)

* AddKey(Event, EventListener)
* Bind Event with Command\*
* Event sent in to AddListener() and AddCommand()
* When Dispatch() is called, a check is done to detect previous key, the the Execute() is called with the event and type.

### OnEvent

* AddListener(Event, EventListener) (Done automatically with AddKey())
* Override in Inherited class
* Switch(event) and designate event to detect dispatch.

### ResetKey

* Send in an Event
* Map “commands” second value which is a vector, is cleared.

### Add / Remove Controller

* SDL Event detects controller connect / disconnect.
* mIsConnected = true then controller input is updated.
* mIsConnected = false then controller input is ignored.

### Sticks (Left shown, Right also available)

* void SetStickDeadZone(int deadZone);
* Vector2f GetLeftStickVector()
* Vector2f GetLeftStickVectorNormal()
* float GetLeftStickAngle()
* float GetLeftTrigger()

### Trigger (Left shown, Right also available)

* GetLeftTrigger()
* Press, release, hold either a half or full trigger.

### Log

* LogEvent() adds the current event to a vector
* saveFile() converts each entry of the vector to a string and prints it in a .csv

### Command Class

* Takes in an function pointer and type.
* Mapped with Events to create a Key Command.
* Combination of type and event is used to call one of the 4 execute functions.
* A loop through all functions is used to activate all functions mapped to a key, based on type.

### Vector2f Class

* Takes in an x and a y variable.
* Can return the magnitude, dot product and normal.
* Custom \* and + operator to multiply and add vectors.