Visual Sentence – Code Documentation

Michel Pahud – 1/15/08

Table of Contents

Introduction

This document is intended to give a quick overview of Visual Sentence for the person responsible to modify or extend the code of Visual Sentence. The document assumes that the reader is already familiar with Visual Sentence intent.

Overview of Visual Sentence

Visual Sentence uses very simple 2D animations and was designed to be scalable in term of art.

The sentence structure supported contains: *primary actor*, *action*, *optional secondary actor (also known as patient)*, *optional background*. For example the sentence “The dragon ate the broccoli in the forest” would have the following primary actor: “dragon”, action: “eat”, secondary actor: “broccoli”, background: “forest”. If no background is specified, the last background that was specified in previous sentences will be used.

Figure 1 below shows the overall architecture of Visual Sentence. Visual Sentence takes a sentence in input (primary actor, action, secondary actor, and background), query a database that has all the images for the actors (actor instances), the action programs, and the images of the backgrounds (scene instances).

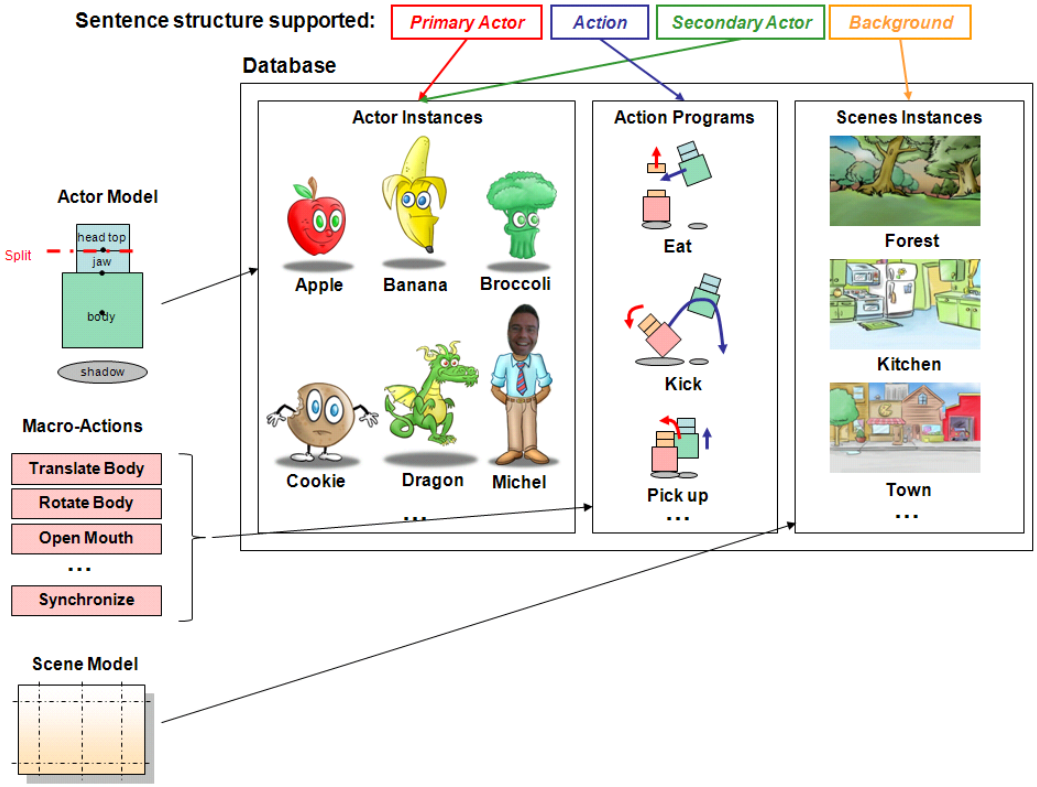


Figure 1: Overview of Visual Sentence

Note: The secondary actor is called patient in the source code.

The end result is an animation of the primary and secondary actors on the scene.

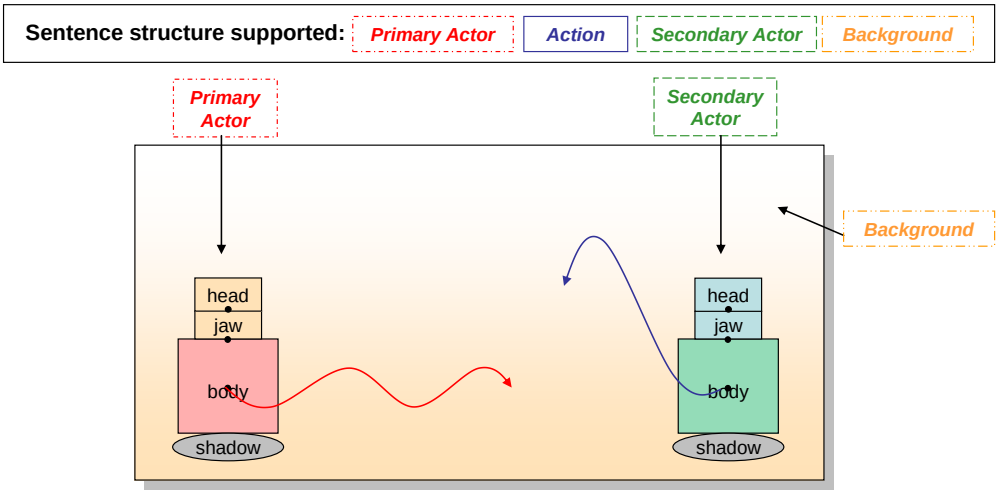


Figure 2: Scene in Visual Sentence

Note: On the scene the primary actor always starts on the left and the secondary actor on the right.

Visual Sentence API

The namespace for Visual Sentence is: KidTab.GameModes.StoryBaker.SBAnim

The class SBAnimation allows generating animations and has the following members:

- **Constructor: public SBAnimation()**

  Create a SBAnimation object

- **Method: public void AnimateSentence(Renderer renderer, string actor, string action, string patient, string background)**

  Animate a sentence with an actor, action, patient and background

  - Actor name without pronouns (i.e. dragon, house, etc.)
  - Action name in its basic form (i.e. kick, eat, etc.)
  - Patient name without pronouns (i.e. dragon, house, etc.)
  - Background name without pronouns (i.e. forest, beach, etc.)

- **Property: public Sprite Background (get only)**

  Get the background image

- **Property: public float BackgroundSFXVolume (get and set)**

  Get or set the volume of the background sfx

Example of Visual Sentence API Usage

```
using KidTab.GameModes.StoryBaker.SBAnim;

    public class VisualSentenceTest : Entity
    {
      private SBAnimation _sbAnim;
      private bool _animPlaying = false;
      ...
       public VisualSentenceTest(Game game)
       {
          KidTab.Drawing.Renderer renderer = game.Renderer;

          ...

          // Animate the sentence: “The man kicked the house in the forest.”
          string _actor1 = "man";
          string _action = "kick";
          string _actor2 = "house";
          string _background = "forest";

          _sbAnim = new SBAnimation();
          _sbAnim.AnimateSentence(
            Game.Instance.Renderer,
            _actor1,
            _action,
            _actor2,
            _background
          );
          _animPlaying = true;
       }

       protected override void OnTick(float gameTime, float dt)
       {
          base.OnTick(gameTime, dt);

          if (_animPlaying)
             _sbAnim.Tick(gameTime, dt);
       }
```

```
    protected override void OnRender(KidTab.Drawing.Renderer renderer)
    {
        base.OnRender(renderer);

        if (_animPlaying)
            _sbAnim.Draw(renderer);
    }

}
```

Storage of information

The database is using the file system. By default the root of the dictionary folder is at: C:\KidTab\Resources\KidWords.

The following folders are used within KidWords:

\Actions

This folder contains all the action programs for the primary actors and secondary actors. There are 10 variants of animations for each primary actor or secondary actor. The variant are picked based on the actor metadata.

Primary actor:

Filename: <<action>> + "_actor" + <<actionVariant 0..9>> + ".sba"

Example: eat_actor0.sba, for the variant 0 of the action eat

Secondary actor:

Filename: <<action>> + "_patient" + <<actionVariant 0..9>> + ".sba"

Example: eat_patient0.sba, for the variant 0 of the action eat

The action programs are written using macro-actions (see below “List of Macro-Actions”)

\Images

Contains all the images of actors

- Filename: <<actor>>_head.png => Image of head top (above the split)
- Filename: <<actor>>_jaw.png => Image of the head bottom (below the split)
- Filename: <<actor>>_body.png => Image of the body

\Metadata

Contains the metadata for the actors. The metadata are described below in SBMetadata class.

Filename: <<actor>>.meta

\SFX

Contains all the sfx

Filename: <<sfx>>.wav

Note: There are also some sfx file that has variant 0..9 (i.e. happy0.wav – happy9.wav). These files are used for emotions and/or for idle behaviors. The appropriate variant is picked based on the actor's metadata.

Description of Classes

SBActor class

Represents a primary or secondary actor for Visual Sentence: Split head, body, metadata and queue of macro-actions that is sequentially executed by the actor. A macro-action is done when the goal is reached (i.e. when the actor reaches the end point during a reposition)

SBActorAnim class

In place animation of the actor or patient to show an emotion using translation, rotation, and scaling of the body parts.

SBMetadata class

Contains the metadata for actor or patient in Story Baker

Physical:

- Ref to head and body image files (.ktai)
- Weight [0..9] (0 – Light, 9 – Heavy)
- Strength [0..9] (0 – Weak, 9 – Strong)
- Soft/Hard [0..9] (0 – Soft, 9 – Hard)

Personality:

- Type {Human, Animal, Vegetable, Mineral}
- Serious/Silly [0..9] (0 – Serious, 9 – Silly)
- Shy/Ongoing [0..9] (0 – Shy, 9 – Outgoing)
- Lazy/Hard worker [0..9] (0 – Lazy, 9 – Hard worker)
- Grumpy/Happy [0..9] (0 – Grumpy, 9 – Happy)
- Dumb/Smart [0..9] (0 – Dumb, 9 - Smart)
- Sleepy/Awake [0..9] (0 – Sleepy, 9 – Awake)

Special Info:

- Head image filename (Head/Jaw where head is the image of the top part above the split and jaw the image of the bottom part below the split)
- Body image filename

SBMetaFormulas class

Contains the formulas metadata for actor or patient.

The result allows to control perturbation such as:

- Angle max of rotation for shaking/rotating
- Size of opening for mouth opening
- Height of jump
- Variant of action
- Variant of sfx
- etc…

Examples of formulas:

Macro-action	Metadata	Metadata automatic effect	Formula
Talk	Shy/Ongoing	• Size of the mouth when open depends on ongoing/shy (ongoing actors open the mouth more than shy)	• Jaw opens to: value(Shy/Ongoing) • Heads opens to: value(Shy/Ongoing) * 2 + 10
Idle	Shy/Ongoing Weight Serious/Silly	• Duration between farts depends on shy/ongoing (ongoing fart more often) • Sfx fart depends on Weight • Sfx burp depends on Serious/Silly	• Time between farts [seconds] = 9 – value(Shy/Ongoing) + random(0-2) • Time between burps [seconds] = 10 – value(Shy/Ongoing) /2 + random(0-1)
Emotion	Grumpy/Happy Shy/Ongoing Weight	• Emotion is picked in function of the metadata	• Grumpy/Happy > 7 -> Happy • Grumpy/Happy < 4 -> Grumpy • Shy/Ongoing > 7 -> Confident • Shy/Ongoing < 4 -> Shy • Others: Surprised

SBCommands file

Contains all the classes to hold the macro-actions

Macro-actions classes

- Reposition class
- Talk class
- Shake class
- Rotate class
- Idle class
- IdleSync class
- Emotion class (*)
- etc.

(*) Emotions type

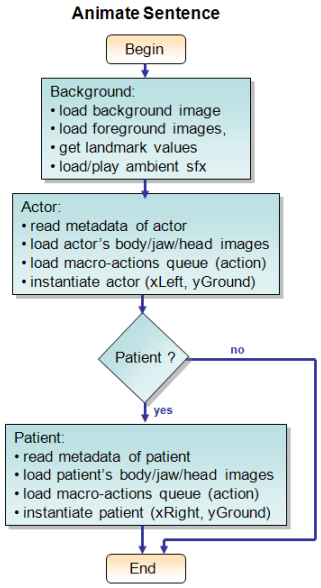
- confident
- happy
- angry
- sad
- surprise
- disgust
- shy

SBEntity class

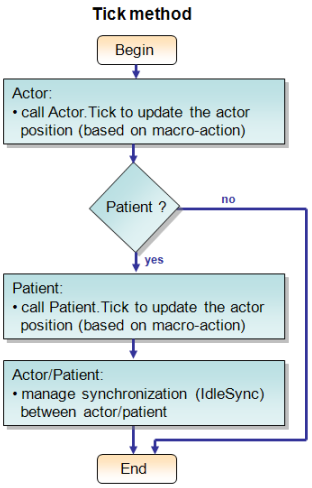
Contains an extension of Entity with image support

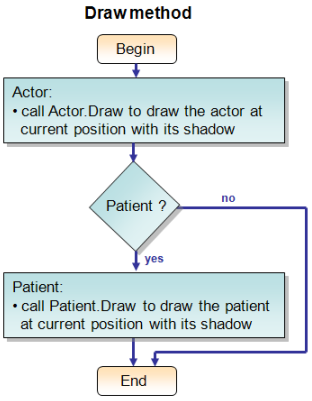
Main Process Diagrams

SBAnimation class process – AnimateSentence method

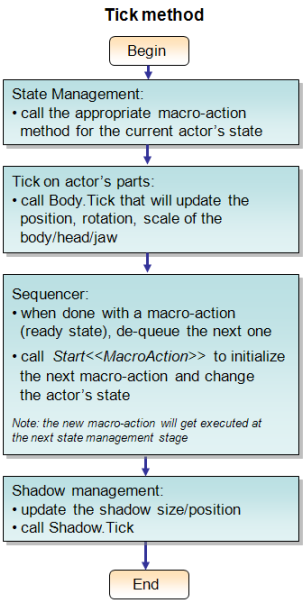


SBAnimation class process – Tick/Draw methods





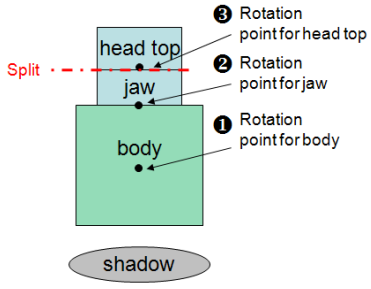
SBActor class process – Tick method



Rotation Points on the Actor Model

The actor model has 3 rotation points:

- Rotation point for the body
- Rotation point for the jaw (head bottom)
- Rotation point for the head top



List of Macro-Actions

Macro-Action	Description	Parameters	Example(s)
Reposition	Reposition an actor or patient horizontally from its current position to a locationX at the hight locationY Note: The actor is automatically looking in the direction of movement	• locationX: can be a value (in pixels) or xLeft, xMiddle, xRight, • locationY: can be a value (in pixels) or yGround, ySky (when ySky the actor is automatically rotated to show a fly behavior) • reposType: Linear or Parabolic • Speed: speed of repositioning in pixels/seconds • Sfx: ref to an optional sfx played during repositioning. By default a linear repositioning will play (hop0.wav – hop9.wav) every time the actor bounces	• Reposition: 500, yGround, Linear, 200, yup.wav • Reposition: xRight, yGround, Parabolic, 300,
Talk	Lips animation	• Speed: Speed of lips • Size: Amplitude of mouth open • Repeat: number of repetition • Sfx: ref to sfx. • Note: if the sfx is laugh.wav, an array of laugh (laugh0.wav – laugh9.wav depending of the actor's weight which is a value between 0-9) is automatically queried	• Talk: 400, 20, 11, laugh.wav • Talk: 400, 20, 11, gulps.wav • Talk: 400, 20, 11, xxx_intro.wav Note: If the house is executing this macro-action, it will look for house_intro.wav and play the sfx with lips animation if it exists.
Emotion	Show an emotion	• Type of emotion: confident || happy || angry || sad || surprise || disgust || shy Note: It automatically query the sfx (emotion0.wav – emotion9.wav depending of the actor's weight which is a value between 0-9)	• Emotion: happy • Emotion: Note: if the emotion is not specified, an emotion will be picked from the actor's metadata (see metadata formulas)
BodyRotation	Rotate the body of the actor	• Speed: degree / seconds of rotation • Angle: angle to reach • Counterclockwise: direction of the rotation • Sfx: ref to sfx to play while rotating	• BodyRotation: 1200, 360, true, Wzzzz.wav • BodyRotation: 1200, 0, false, zzzzW.wav
Look	Look left or right Note: The art has to be	• Direction: Direction toward the actor is looking	• Look: right

	authored to lok toward the right direction by default		
Depth	Change the depth of the actor	• Depth: [0..255] 0 - front, 255 – back Note: By default the actor/patient are place at depth of 128 with the patient at the front of the actor	• Depth: 129
Burst	Display a burst at the front of the actor	• grow speed: %of scale grow / seconds • scaleMin: starting scale • scaleMax: ending scale • gfxPath: optional gfx • sfxPath: optional sfx Note: If no gfx/sfx is specified, the burst will be chosen from the actor metadata between: Bang, Burp, Crash, Glass, Klonk, Ouch, Urkk, Zap (querying burst_xxx.ktai and bust_xxx.wav files)	• Burst: .7, 0.01, 1.4, burst_glass.ktai, burst_glass.wav • Burst: .7, 0.01, 1.4,,
Idle	Stay idle while playing idle behavior. Note: The actor/patient plays idle animation burp/fart using (burp0.wav – burp9.wav / fart0.wav – fart9.wav) depending on metadata during the idle state	• Duration: idle duration in seconds • SfxPath: optional sfx	• Idle: 2,
IdleSync	Idle with global synchronization. Stay idle until actor and patient calls this instruction Note: The actor/patient plays idle animation burp/fart using (burp0.wav – burp9.wav / fart0.wav – fart9.wav) depending on metadata during the idle state		• IdleSync: