enta 1231, [twine story] nothing out of the ordinary

Date Finished	@29 January 2026
Course	enta 1231 visual scripting in game development
Software	Twine
Type	Written
Status	Done

▼ description

Instructions

Assignment 1 - Twine Story

Create a complete interactive story in Twine that demonstrates your understanding of visual scripting concepts from Labs 1 & 2.

Your story must remember player choices, track state, and determine outcomes based on that state. The system—not a single final choice—should decide how the story ends.

Starting Point

You must continue from the Twine story developed in Labs 1 & 2.

Your story should already include:

- A clear introduction
- Multiple events
- Choices with visible outcomes
- A defined ending

Do not create a new story. This assignment completes your existing one.

Core Requirements

Your Twine story must include **all** of the following.

Variables and State

- Player choices must set variables
- Variables must represent narrative intent
- The story must remember what the player has done

Conditional Logic

- Story content must change based on earlier choices
- Conditions should alter dialogue, options, or outcomes

Branching and Endings

- The story must branch meaningfully
- The story must contain multiple distinct endings
- Endings must be determined by player state, not a single final choice

System-Driven Consequences

- Earlier choices must affect later events
- Tracked choices must matter somewhere later in the story

System Explanation

- Submit a short document or diagram explaining:
    - Your variables

    - What they represent
    - Where they are set and checked

Scope and Expectations

- Keep systems simple and readable
- Avoid complex mechanics (inventory, currency, reputation)
- Focus on clarity, correctness, and intent

Testing Requirement

Before submitting:

- Play through your story multiple times
- Confirm that:
    - Variables update correctly
    - Conditions trigger as expected
    - Different choices lead to different outcomes

Submission Requirements

Submit:

- Your completed Twine story file
- A screenshot of your final twine graph
- Your system explanation document or diagram

Evaluation (25% of Grade)

Your assignment will be evaluated using the same criteria as the requirements above.

- Player choices set and use variables
- The story remembers and responds to player actions
- Conditional logic meaningfully changes content
- Branching leads to multiple distinct endings
- Outcomes are system-driven

- System explanation clearly matches the implementation
- Story is complete and playable

---

1. my main goal

making choices feel meaningful without expanding the story endlessly

the core loop i'm using is:

choice → variable → state → consequence

this means the story can reconverge in the middle, but the outcome still changes depending on what the player did earlier.

what i was missing in lab 2

in lab 2, i was tracking choices, but i wasn't resolving them. i had variables changing, but no final state check that turned those values into clear outcomes.

for this assignment, the "arrival" passage fixes that by acting as a final gate that evaluates the player's accumulated state.

---

2. quick story summary (player experience)

from the player's perspective:

- i'm driving to meet my sister, mufudzi
- i go through a checkpoint / traffic stop
- i choose how i cope with stress afterward
- i have a blowout and decide how to handle being stranded
- the story reconverges at the same arrival location

even though the location is the same, the **ending changes** based on how i behaved.

---

3. how the system works

inputs (what the player does)

inputs are the moments where the player actively makes a decision. each input does **one specific task** in this system.

1. checkpoint behaviour

this is the player's response during the stop.

the options are:

- calm
- rude
- silent

each option:

- changes behaviour variables
- does *not* branch the story location
- creates a record of how the player handled the situation aka the authority.

this input exists to establish a baseline behaviour early in the story.

---

2. coping choice

this happens after the checkpoint and focuses on internal response, not action.

the options are:

- deny (light a cigarette)
- panic (spiral internally)
- reach out (call mama)

each option:

- sets the `$coping` variable
- may slightly affect `$heat`
- does not move the story forward geographically

this input is about *emotional regulation*, not progress.

---

3. blowout response

this happens when the player is stranded.

the options are:

- call for help
- try to handle it alone

this input:

- sets `$calledHelp`
- increases `$heat`
- acts as a stress test for earlier behaviour

this is the last meaningful player choice before the ending.

---

processing (what the system does)

processing is everything the game does **between** player choices and endings.

the player doesn't see this directly, but it's for us to see and how the system determines the outcome

---

step 1: update variables

after each choice:

- one or more variables are updated
- nothing resolves immediately

examples:

- rude response → `$defiance + 1`
- calling for help → `$calledHelp = true`

variables accumulate quietly in the background.

---

step 2: convert variables into states

instead of you checking many variables later, the system groups them into states at specific checkpoints. that's why we created them.

this happens:

- after the checkpoint encounter
- after the blowout

states simplify decision-making at the end.

---

step 3: evaluate states at the final gate

the final scene (`arrival`) checks:

- `$checkpointState`

- `$riskState`
- `$coping`

based on a priority order, the system selects **one** ending.

the player's final click does not decide the ending — the accumulated state does.

---

outputs (what the player gets)

outputs are the visible results of the system.

the player always arrives at the same location, but the tone and outcome change.

possible outputs include:

- confrontation or emotional blow-up
- support and reunion
- emotional shutdown or silence
- unresolved but changed ending

these outcomes are determined by behaviour patterns, not single choices.

i originally had a more extreme "caught" potential ending but removed it to keep the focus on emotional and relational consequences rather than escalation.

---

4. variables (raw values)

variables are the most basic units in the system. they change directly when the player makes a choice.

they are not shown to the player, but everything else depends on them.

---

behaviour flags (numbers)

these track how the player interacts with authority.

- `$authority`

  increases when the player responds calmly or tries to de-escalate

- `$defiance`

  increases when the player challenges, snaps back, or resists

- `$compliance`

increases when the player stays silent or follows instructions without comment

these values are not mutually exclusive, but usually one dominates.

---

coping style (string)

- `$coping`

this records **how** the player deals with stress.

possible values:

- `"deny"`
- `"panic"`
- `"reachout"`

this variable is set once and carries through to the ending.

---

risk tracker (number)

- `$heat`

this tracks escalation over time.

it increases when:

- the player is confrontational
- the player involves outside help
- the player draws attention to themselves

higher heat increases the chance of negative outcomes.

---

help choice (boolean)

- `$calledHelp`

this records whether the player relied on external support during the blowout.

true = called for help

false = handled it alone

this variable affects risk but also signals dependency vs isolation.

---

5. state variables (checkpoints)

states are simplified labels created from raw variables.

they exist to make the ending logic readable and manageable.

`$checkpointState`

this summarises how the player behaved at the checkpoint.

possible values:

- `"calm"`
- `"defiant"`
- `"compliant"`
- `"unknown"`

this state influences whether the arrival scene becomes confrontational.

`$riskState`

this summarises overall risk based on `$heat`.

possible values:

- `"low"`
- `"medium"`
- `"high"`

this state represents how visible or vulnerable the player is by the end.

`$ending`

this is a temporary label used to track which ending is triggered.

it helps with debugging and ensures only one ending displays.

---

6. checkpoint rules

checkpoint a — after the checkpoint encounter

this checkpoint converts behaviour variables into a single state.

this happens either:

- immediately in the branch, or

- later using conditional logic

example rule set:

- if `$defiance >= 1` → `"defiant"`
- else if `$authority >= 1` → `"calm"`
- else if `$compliance >= 1` → `"compliant"`

this locks in the player's approach to authority early.

---

checkpoint b — after the blowout

this checkpoint converts `$heat` into `$riskState`.

example rule set:

- heat ≥ threshold → `"high"`
- heat = 1 → `"medium"`
- heat = 0 → `"low"`

this allows earlier choices to influence the ending without adding new branches.

---

7. making choices impactful without expanding scope

instead of branching constantly, i let the middle of the story reconverge.

impact comes from:

- how variables accumulate
- how states are calculated
- how the final gate evaluates those states

this means i only add **one major new passage** (`arrival`) instead of rewriting the whole story and adding a whole bunch of passages

---

simple diagram

```
Intro
  -> Event 1 (Checkpoint encounter)
      sets authority/defiance/compliance + reputation/heat
      -> CHECKPOINT A: AttitudeState
```

```
   -> Event 2 (Cigarette / coping)
       sets coping (+ optional heat)
   -> Event 3 (Blowout)
       sets calledHelp (and heat)
       -> CHECKPOINT B: HeatState + HelpChoiceState
   -> Final Gate (Meet Sister)
       -> Ending 1 / 2 / 3 / 4 (based on states)
```

8. flow map (logic overview)

- intro → initialise variables
- event 1 → checkpoint encounter
  - sets behaviour variables
  - checkpoint a calculates `$checkpointState`
- event 2 → coping moment
  - sets `$coping`
- event 3 → blowout
  - sets `$calledHelp` and `$heat`
  - checkpoint b calculates `$riskState`
- arrival → system checks states and outputs ending

---

9. arrival as a final gate

the arrival passage:

1. acknowledges time passing so the transition feels natural
2. calculates `$ending` based on priority rules
3. displays exactly one ending

priority order:

1. high risk or defiant behaviour → confrontation
2. reach out → reunion
3. denial → silence

4. anything else → unresolved

this ensures the system, not the last click, decides the outcome.

---

10. use of imagery

images are used only at major beats:

- checkpoint
- coping moments
- arrival endings

the monochrome, fine-line style matches the quiet tension of the story and reinforces emotional states.

ai-generated images were used for this project and are acknowledged as such.