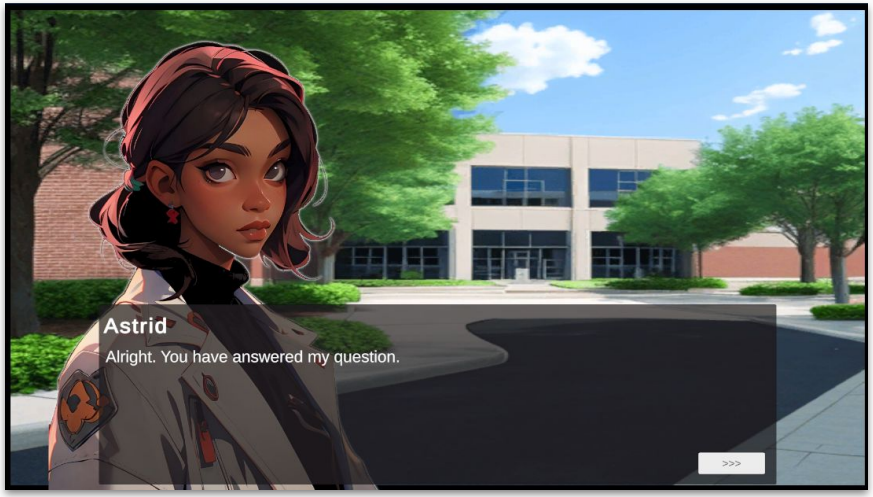
Anansi

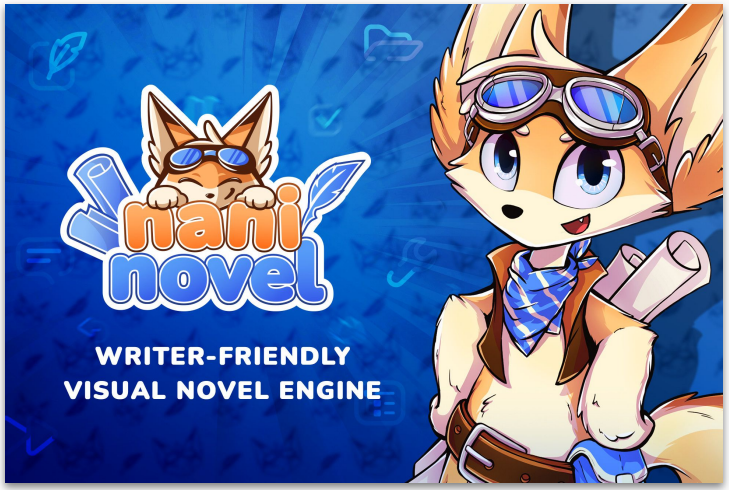
A visual novel framework for Unity

Overview

- Unity package that facilitates creating dynamic and branching visual novels
- Started working on it Summer 2023
- Write content using Inkle's Ink scripting language
- Integrates branching/storylet narratives with a social simulation
- Inspired by *Persona* 5's use of time and character relationships to drive narrative progression



Alternative Systems

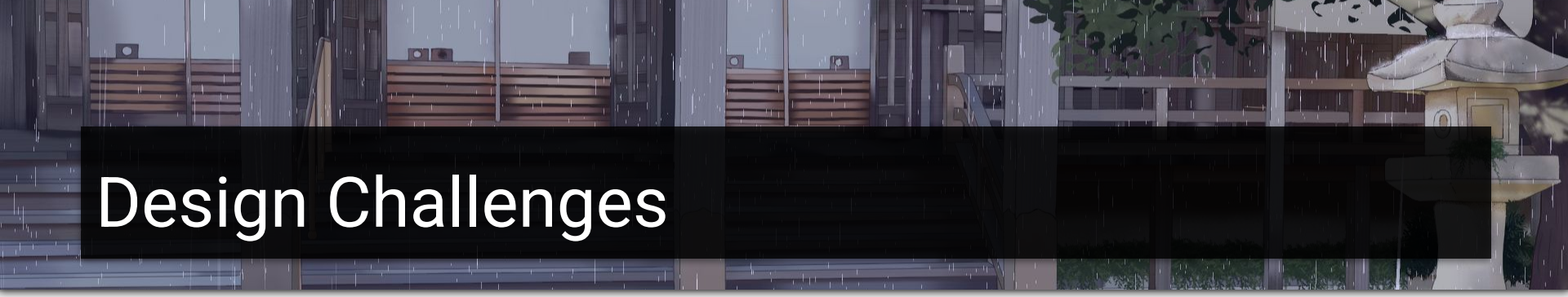


Fungus

Twine

Design & Personal Goals

1. Support dynamic casting of characters into storylines
2. Easy to add new characters and have them participate in existing content
3. Modular system that can adjust to end-user UI implementations
4. A writing interface that was easy for writers to work with
5. (Personal) Learn more about structuring storylet content



Design Challenges

- The problem here is that we want to treat the location like a storylet. So we can’t have direct diverts within the location because we need to mix mix these choices with dynamic choices made available by storylets. We CANNOT mix knot-level choices with storylet-level choices. navigating around the world operates at the realm of storylets. Conversations and intermediate scripts operate at the knot-level.
- Casting only happens when switching storylets. It does not happen when diverting to a new Ink knot

How do you write dialogue?

- All dialogue content is written using Inkle's Ink scripting language
- Ink is battle tested and has been used for multiple commercial games
  - Heaven's Vault, 80 Days, Sorcery 1 - 3
- Story structure can range from linear to branching to dynamic (storylets)

Start Storylet

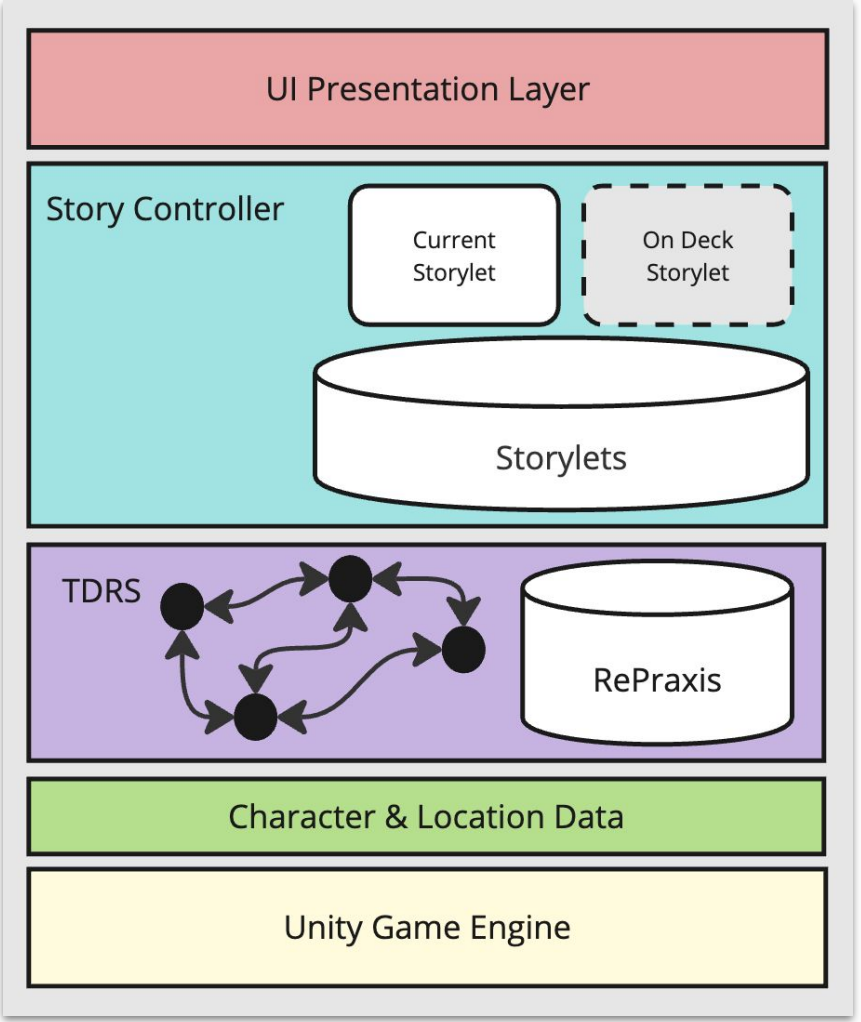
```
24 === start ===
25
26 // The system will always look for a knot with the name "start"
       to execute first. Inside of this knot, designers should set
       the current time of day and location. This will move all
       characters to where they need to be. Optionally, you can
       include prologue content under this knot to help introduce
       the player to the game.
27
28 -> location_outside_library
29
30 -> DONE
```

Basic Storylets

- Baseline storylet
- Might have some tags
- Users can use the “@query” command to specify preconditions
- Moves game control flow into Ink space

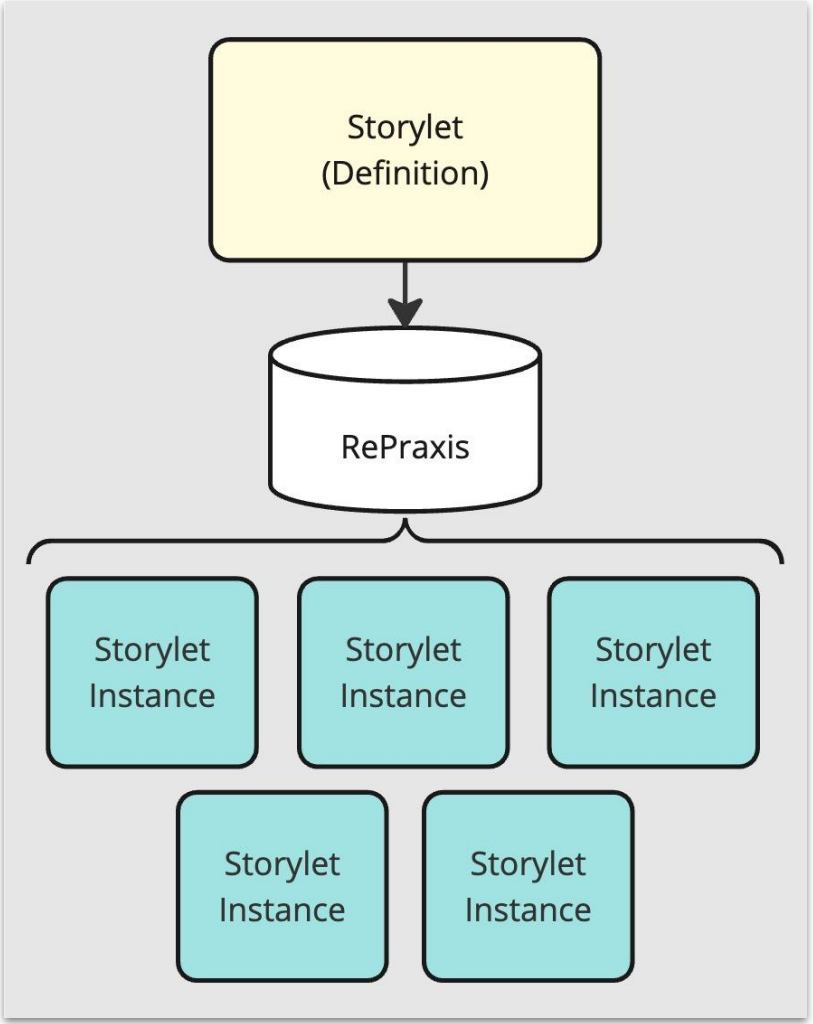
```
=== storylet_sample ===
# ---
# tags: evelyn, convo
# @using speaker as ?speaker
# @query
# ?speaker.relationships.player.stat.Friendship!?val
# eq ?val 0
# @end
# ===

{speaker}.surprised: Hi, nice to meet you.

-> DONE
```

Anansi Layers



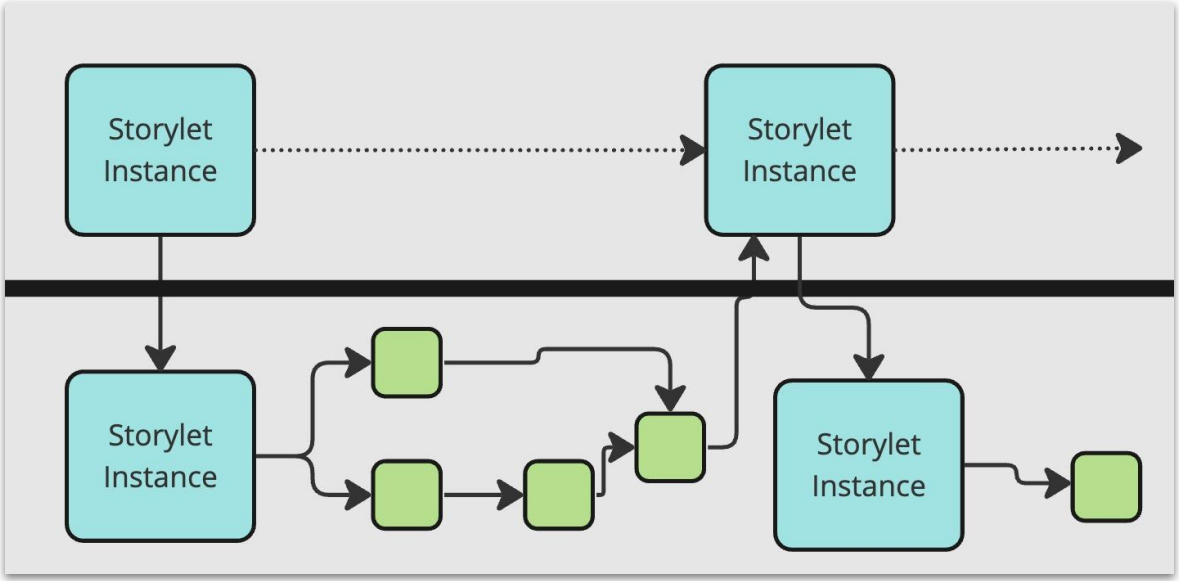
Storylet Instancing

- Storylets at their base are “definitions”
- They become instances when we run the precondition query
- Based on the results of the query, we create various instances of storylets
- Each instance has a different set of bindings from the precondition



Storylet Control Flow

- Choices and control flow in the story passes between two level
- Storylet-to-storylet level
- The Ink-level



Action Storylets

- Storylets corresponding to actions players can take
- Actions have a choice label (adapted from Story Assembler)
- Can also have precondition queries
- Can do all the Ink things
- Actions are presented outside the Ink control flow. They are not to be mixed with Ink-level stories

```
=== action_nap_in_room ===
# ---
# choiceLabel: Take a nap
# @query
# player.location!player_dormroom
# @end
# ===

You decided to take a nap...

{AdvanceTime()}

Time has advanced.

-> DONE
```

Location Storylets

- Corresponds to locations that exist in the game
- Also have choice labels
- Presented to the player outside the Ink control flow
- Locations can divert and do all the Ink things

```
112 === location_evelyn_dormroom ===
113 #---
114 # choiceLabel: Go to Evelyn's room.
115 #===
116
117 {SetLocation("evelyn_dormroom", "")}
118
119 {
120     - DbAssert("evelyn.location!astrid_dormroom"):
121         You are in Evelyn's room. She is looking looking out the
              window, listening to music
122     - else:
123         Evelyn is not here.
124 }
125
126 -> DONE
```

Writing in Ink

```
INCLUDE ./EvelynDialogue.ink

EXTERNAL SetLocation(location, tags)
EXTERNAL SetBackgroundOnly(location, tags)
EXTERNAL DbInsert(statement)
EXTERNAL DbDelete(statement)
EXTERNAL DbAssert(statement)
EXTERNAL AdvanceTime()
EXTERNAL QueueStorylet(storyletId)
EXTERNAL QueueStoryletWithTags(tags, fallback)
EXTERNAL GetInput(prompt, variableName, callback)

VAR speaker = "not-specified"
VAR player = "player"
VAR location = "not-specified"
VAR timesKnocked = 0
```

Dynamic Character Sprites

- Support for animated and static character sprites
- Designers assign one or more descriptive tags to sprite images
- Writers can specify zero or more tags for selecting a sprite to show when presenting spoken dialogue

Speaker's UID

```
{speaker}.surprised: Hi, nice to meet you.
```

Annoyed

Surprised

Default

Time and Schedule System

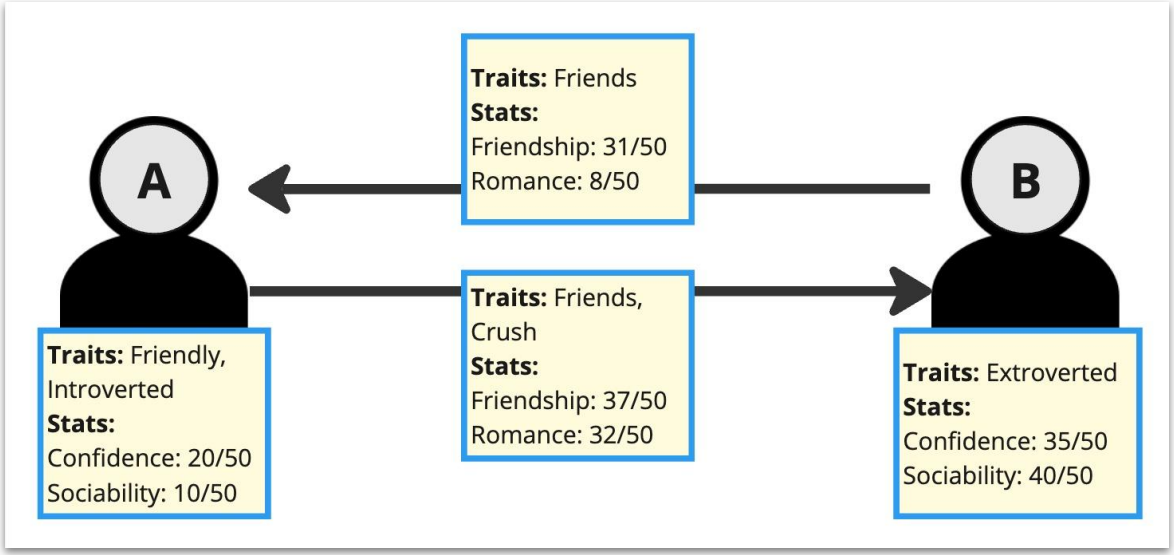
- Calypso tracks the current day and week
- Each days is subdivided into parts: morning, afternoon, evening, night
- Some player actions advance time
- Time can also advance within character-dialogue
- Characters move locations based on their daily-schedules
- Character can have multiple schedules at once

Trait-Driven Relationship System (TDRS)

- Tracks relationships between social agents
- Designers attach traits to characters and their relationships
- Traits can apply stat buffs
- Traits can define social rules that apply buffs to the owner’s relationships
- Social Events apply immediate and second order social effects
- Query the relationship system using Versu Praxis syntax (RePraxis)

TDRS: Agents and Relationships

- **Agent**: an entity that can form relationships (single character, faction, an idea/concept)
- **Relationship**: a social connection between two entities
- Agents and relationships have associated **traits** and **stats**
- Relationships are **unidirectional** from one agent to another



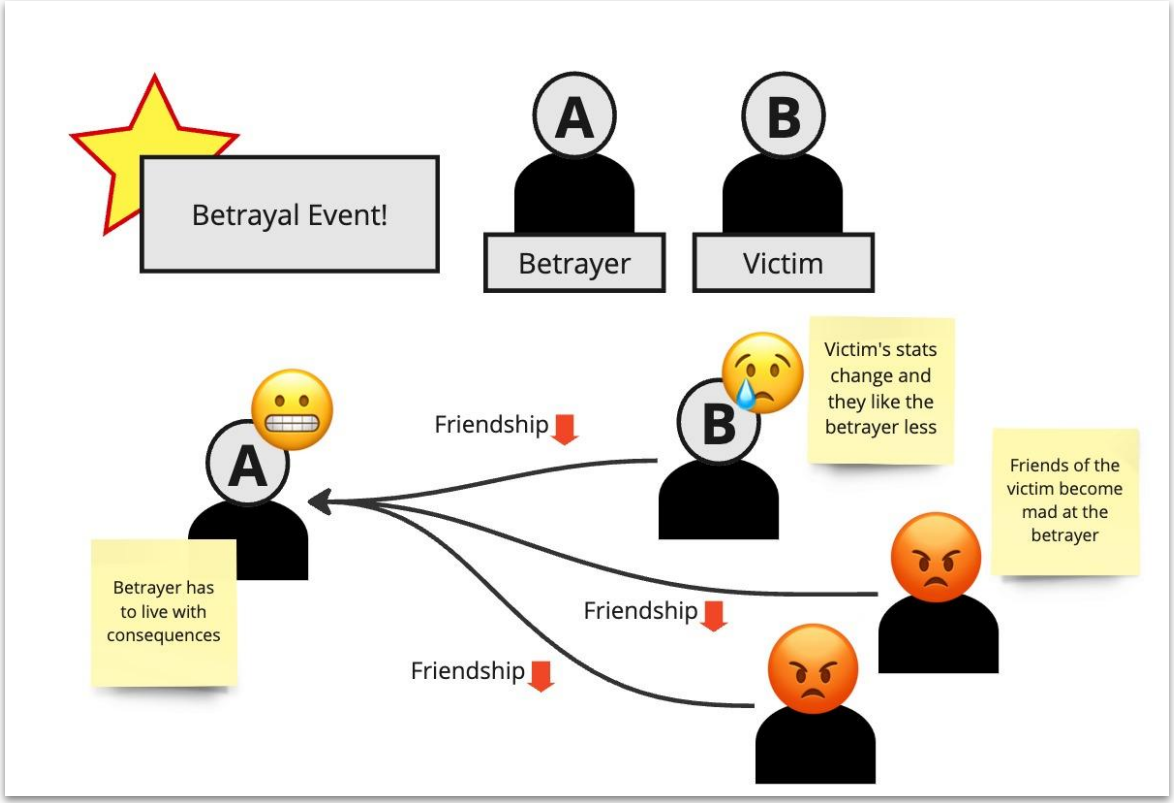
TDRS: Traits

- The main driver for TDRS logic
- Uses RePraxis syntax and string templates for effects and social rules
- Can be attached to agents or relationships

```yaml
- traitID: jerk
  traitType: agent
  displayName: Jerk
  description: "[owner] is a jerk"
  socialRules:
    - precondition: |
        not ?other.traits.jerk
      effects:
        - DecreaseRelationshipStat ?other ?owner Friendship 10
      description: "[owner] is a jerk"
    - precondition: |
        ?other.traits.jerk
      effects:
        - IncreaseRelationshipStat ?owner ?other Friendship 10
      description: "Jerks like other jerks"

- traitID: happy_go_lucky
  traitType: agent
  displayName: Happy-Go-Lucky
  description: "[owner] is happy-go-lucky."
  effects:
    - IncreaseAgentStat ?owner Sociability 10
  socialRules:
    - effects:
        - IncreaseRelationshipStat ?owner ?other Friendship 3
```

TDRS:
Social Events

- **Social Event**: Event that triggers a state change in the involved parties and their associates
- Bind agent roles and define effects based on preconditions
- Social events can add temporary or permanent traits or stat buffs



TDRS: RePraxis Query Language

- Based on the Praxis exclusion logic language used in the Versu Engine
- RePraxis is used to store all queryable data in social engine
- Supports variable unification
- Supported data types: Int, float, and string

```
- event: betrayal ?betrayer ?victim
  description: "[betrayer] betrayed [victim]."
  responses:
    - effects:
        - DecreaseRelationshipStat ?victim ?betrayer Friendship 10 5
        - RemoveRelationshipTrait ?victim ?betrayer friend
    - precondition: |
        ?victim.relationships.?victim_friend.traits.friend
        neq ?victim_friend ?betrayer
      effects:
        - DecreaseRelationshipStat ?victim_friend ?betrayer Friendship 5 6
    - precondition: |
        ?victim.relationships.?victim_family.traits.family
        neq ?victim_family ?betrayer
      effects:
        - AddRelationshipTrait ?victim_family ?betrayer angry_at 10
```

RePraxis syntax is used to specify social event roles and preconditions. The '?' character is used to denote variables

Thank you!

Contributors:

- Shi Johnson-Bey
- Kira Liao

Collaborators:

- Academical RCR Team

Links:

- Anansi: https://github.com/ShiJbey/Anansi
- TDRS: https://github.com/ShiJbey/Unity-TDRS
- RePraxis: https://github.com/ShiJbey/RePraxis
- BG Art by Noraneko Games: https://noranekogames.itch.io/