Sketch-Based Interaction for Planning-Based Interactive Storytelling

Edirlei Soares de Lima
Faculty of Design, Technology and Communication – IADE
Universidade Europeia
Lisbon, Portugal
edirlei.lima@universidadeeuropeia.pt

Felipe João Gheno
Faculty of Design, Technology and Communication – IADE
Universidade Europeia
Lisbon, Portugal
felipepletz@gmail.com

Ana Viseu
Faculty of Design, Technology and Communication – IADE
Universidade Europeia
Lisbon, Portugal
ana.viseu@universidadeeuropeia.pt

***Abstract* — Drawings have been used for thousands of years as a visual complement to oral and written storytelling. The evolution of technology and the advent of interactive narratives brings the possibility of exploring drawings and storytelling in new ways. This paper presents a new sketch-based interaction method for planning-based interactive storytelling systems, which uses a deep learning model based on a Convolutional Neural Network to recognize digital hand-drawn sketches. By combining real time sketch recognition with a planning-based plot generation algorithm, the proposed system allows users to interact with narratives by sketching objects on smartphones or tablet computers, which are then recognized by the system and converted into virtual objects in the story world, thereby affecting the plot of the narrative. Preliminary results show that the sketch recognition model has a remarkable accuracy for small sets of sketch classes (accuracy of 95.1% for 14 classes), which are sufficient to provide a good variety of interaction options. In addition, it can also be extended to more complex scenarios while maintaining a considerable accuracy (87.4% for 172 classes and 71.6% for 345 classes).**

Keywords — interactive storytelling, sketch-based interaction, automated planning, sketch recognition, interactive narratives

I. Introduction

Humans (and their ancestors) have told stories for over forty thousand years. Storytelling, in its visual, oral, and later written components, is an important means of cultural creation and transmission. For young(er) generations whose identity and behaviors are deeply interweaved with digital technologies, these have been added to the set of tools that can be utilized in, and for, storytelling. In this scenario, digital storytelling supports experimentation with new approaches and virtual experiences that create unique opportunities for entertainment and learning. The association between digital technologies and interactivity also looms large. While the concept of interaction is continuously changing – from desktops to mobile devices, passing through wearables, smart home assistants, and the internet of things – it has come to seem natural to us in our relations with technology.

The increasing interactive capabilities of new devices have created the potential for audiences to have an active role in the development of the narrative itself. In many settings, traditional storytelling may be considered old-fashioned, especially when we have games with a strong focus on storytelling that benefit from current technologies and give players the ability to actively participate in the narrative. Already in 1967, John Bart, in his essay “The Literature of Exhaustion” [1], questioned what would be the future steps of storytelling. Modern cinema, television, and streaming platforms, among others, can be considered an evolution of the ways in which traditional storytelling is performed. However, games and interactive storytelling applications, go one step further as they provide the audience with a greater role in their interactive experiences, moving from listeners or spectators to actors.

Interactive storytelling has two main pillars: (1) interactivity, which focuses on the end-user experience and defines how users engage with the story; and (2) authorship, which is based on the use of computational methods to give users the role of co-creators of an ongoing narrative. Over the last decade, several interaction methods have been explored by the research community, including traditional user interfaces [4], interactive books [18], smart toys [28], tangible objects [30], and hand-drawn sketches [19]. Among the main interaction methods for interactive storytelling, the use of hand-drawn sketches is showing promising results, especially for educational purposes. According to Cox [5], the act of drawing allows children to construct sign and meaning through a constructive process of thinking. In addition, sketch-based interaction can embrace the four main pillars of children’s basic learning (auditory, visual, tactile, and kinesthetic) [23]. Besides the educational characteristics, sketch-based interaction also increases the sense of authorship since the user’s creation is directly used in the development of the ongoing story [19].

In this paper, we propose a new sketch-based interaction method for planning-based interactive storytelling systems. The method uses a Convolutional Neural Network to recognize digital hand-drawn sketches and a story generation algorithm based on Automated Planning that supports real-time interaction. This method allows users to interact with narratives by drawing objects on a smartphone or tablet computer, which are then recognized by the system and converted into virtual objects in the story world, where they affect the decisions of virtual characters and change the plot of an ongoing narrative. The main objective of this paper is to present our method and to validate its precision and real-time performance on highly interactive storytelling environments.

The paper is organized as follows: Section II describes related work. Section III presents the proposed interaction method and describes the implementation details of the sketch recognition model and its integration with the story generation algorithm. Section IV describes a prototype application that uses the proposed method. Section V presents a technical evaluation of the method. Section VI offers concluding remarks.

II. Related Work

In 1955, Crockett Johnson published a children's book titled Harold and the Purple Crayon [13]. This book, still successfully sold today, depicts the story of a boy who uses his purple crayon to create images that come alive and take him through numerous adventures. Though not interactive, this story can be seen as an inspiration to the research on sketch-based interaction, an active research topic in the area of human-computer interaction. Although its applications are numerous and widespread, this related work section focuses primarily on sketch-based interaction methods for interactive storytelling and games. A more general review of the state of the art on sketch-based interaction is presented by Bonnici et al. [3].

One of the earliest interactive narratives to use hand-drawn sketches as a form of interaction is "The Lost Cosmonaut" [29], an art installation with a database of narrative units (imagens, movies and sounds) that are triggered and exhibited as result of pen strokes on a special paper. The system relies on a special pen, known as the Anoto Digital Pen, to record the pen strokes and send the drawing information to a computer. However, the system only triggers narrative units as response to pen strokes, without recognizing the meaning of the hand-drawn sketches. Another interactive narrative that also relies on the Anoto Digital Pen technology is "Papyrate's Island" [14], which explores the idea of transferring paper drawings to a 3D virtual world. Although the system is capable of converting the hand-drawn sketches into 3D objects, it does not recognize the content of the drawings. Instead, it relies on the narrative context to determine the functionality of the object. For example, when the story requests users to draw a fire-extinguisher, any object will be considered valid and the direction of the water jet is defined by the final stroke of the pen that moves from the drawing to the margin of the paper. A similar approach is explored by Feng et al. [6] in an augmented reality modeling system to create 3D cartoon scenes based on children drawings.

The actual recognition of hand-drawn sketches of objects as a form of interaction for interactive narratives is explored by Lima et al. [19]. Their mixed reality interactive storytelling system allows users to interact with virtual characters by sketching objects on a conventional sheet of paper with a regular pen or pencil. By using computer vision techniques and a support vector machine classifier, their system can identify a predefined set of sketches and convert them into 3D objects in an augmented reality environment. This system was later extended by Franco and Lima [7] to support the recognition of generic hand-drawn sketches of environmental objects (such as clouds, rives, and walls), which are used to modify the environment of the virtual world. The main limitation of Franco and Lima's work is the limited set of drawings that can be recognized by the system (in their experiments, the authors' used only 6 classes of drawings). Although the repertoire of recognizable drawings can be extended, their procedure to acquire training samples is complex and time-consuming as it requires real photos of sketches taken from different view angles. In addition, the inclusion of more classes negatively affects the accuracy of their support vector machine classifier, which uses only Hu descriptors [11] to characterize the classes of drawings.

Sketch-based interaction has also been explored in games. Hagbi et al. [10] describe a content-authoring tool that uses hand-drawn sketches to create 3D environments for augmented reality games. Their system can identify predefined sketches, which are converted into 3D objects using predesigned models – as well as, generic geometric objects, which are transformed into 3D objects according to specific reconstruction rules. Their process to recognize sketches is based on the use of image processing and computer vision algorithms that break the sketch content into basic sketch elements, which have their meaning established according to a predefined set of visual language rules. While Hagbi et al. explore the use of sketches for authoring game content, other authors apply sketches for gameplay interaction, such as Williford et al. [31], who propose a game called ZenSketch, which utilizes gesture-based sketch recognition to translate features of line drawings into game mechanics, such as connecting floating islands to build bridges, where the quality of the line defines how good is the bridge. In the context of board games, Huynh et al. [12] use hand-drawn physical tokens to represent individual towers, and Monteiro et al. [20] propose a sketch-based version of the Sudoku game.

In addition to the aforementioned research works, there are some published games that also use sketches as a form of interaction, such as Line Rider[1], Crayon Physics Deluxe[2], and Draw Story![3]. Among these games, "Draw Story!" stands out for its interactive storytelling elements, which allow players to use sketches to interact with the story. In the game, the main character is constantly asking the player to draw certain objects in order to help him deal with several game situations. Although the game does recognize the sketches, the story is completely linear, so the sketches are used only to move the story forward.

Most of the previous works on sketch-based interaction methods for interactive storytelling were applied only on systems that relied on simplified computational models for narrative management, such as predefined branching narrative structures or linear stories, which reduces the user's sense of authorship as all interaction points are predefined by the author. The system presented by Lima et al. [19] is one the few that explores a more complex story generation method, which is based on a simulation of characters' emotions and scripted behaviors (character-based approach). However, all possible characters' reactions are still manually defined by the author. The most robust forms of interactive narratives usually rely on artificial intelligence techniques, such as planning [8], to dynamically generate the sequence of narrative events, creating real emergent narratives that allow users to effectively interact and change the story at any time. Although planning can improve the users' sense of agency and authorship, it also brings new challenges to the interaction process, such as how to balance interaction freedom while managing the effects of potential any-time user interactions in the plot. In this paper, we address some of these challenges and present a new sketch-based interaction method for planning-based interactive storytelling.

[1] https://www.linerider.com/
[2] http://www.crayonphysics.com/
[3] https://play.google.com/store/apps/details?id=com.gamejam.draw.story

III. Sketch-Based Interaction

The proposed sketch-based interaction method was designed for an educational and collaborative multi-user interactive storytelling system called “História Viva” (*Living History*). As illustrated in Fig. 1, “História Viva” is a multi-projector system that projects the story content onto the walls of a room, which create an immersive storytelling environment for children that favors learning. The dramatization system uses animated 2D graphics for the visual representation of characters, objects, and environments. The user interface of the sketch-based interaction system runs on mobile devices, such as smartphones and tablet computers, and it allows users to interact with the story by sketching objects. When a sketched object is recognized by the system, a virtual representation of that object is inserted into the virtual world where it can influence the decisions of characters and affect the story’s outcome. For example, if a character is lost in a dark forest during the night, users can help him by drawing the sun, which will turn night into day and the character will be able to find his path. However, if users decide to draw something different, like a candle or a campfire, the character can end up starting a wildfire by mistake, which leads to a completely different storyline.

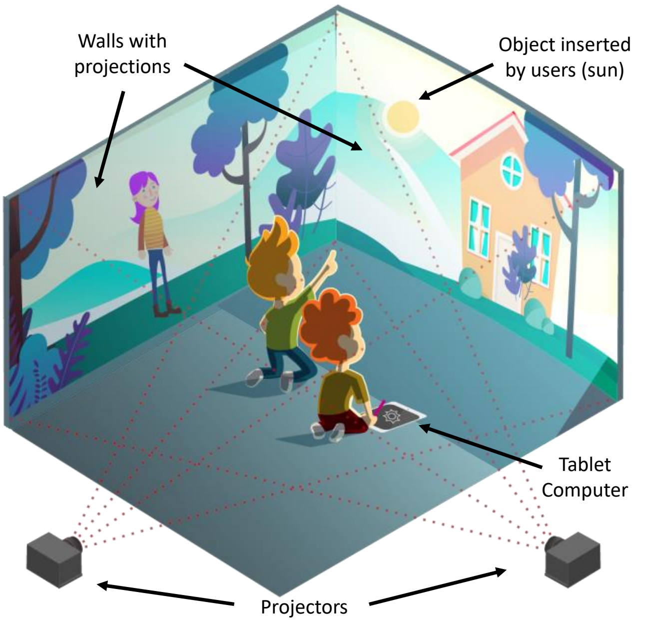

Fig. 1. Illustration of the components and environment of our interactive storytelling system.

The architecture of our interactive storytelling system is based on a client-server model (Fig. 2), where the modules responsible for the generation and dramatization of the story are part of the server and the sketch-based interaction interface is a client. On the client-side of this distributed model, the Sketch-Based Interaction Client is implemented as a mobile app, where the Sketch Input Interface is responsible for receiving the user’s sketches and sending images of them to a Convolutional Neural Network classifier. Once a sketch is recognized by the Convolutional Neural Network, an identifier of the sketch class is sent to the Planning-Based Storytelling Server through a TCP/IP network message. In addition, the Sketch Input Interface is also updated to inform the user about the identified object.

On the server-side, the Interaction Server module is responsible for receiving and interpreting the sketch classes sent by clients. Two interaction modes are supported: (1) free interaction mode, in which all objects identified in users’ sketches are directly inserted into the story world as soon as they are received; and (2) voting interaction mode, in which the Interaction Server collects all identified objects during a certain time-window and then selects the one to be inserted into the story through a voting process. Free interaction is the default mode, but the Plot Manager can request the Interaction Server to switch to the voting mode in response to certain story events where virtual characters explicitly ask the audience for specific objects. Besides selecting the interaction mode, the Plot Manager is also responsible for: (1) controlling the execution of the story events by sending action requests to virtual Characters; (2) applying the effects of objects inserted by users in the World State; and (3) requesting new story plans from the Story Planner when user interactions affect the consistency of the current plot. The World State aggregates all information about the current situation of story world, which is affected by characters’ actions and objects inserted by users. The Story Planner uses observations extracted from the World State to find a valid plan of actions to achieve authorial goals. All modules of the Planning-Based Storytelling Server work according to the story context defined in the Story Domain Database, which establishes all characters, locations, visual assets, planning operators, authorial goals, initial state, and the effects of each recognizable object.

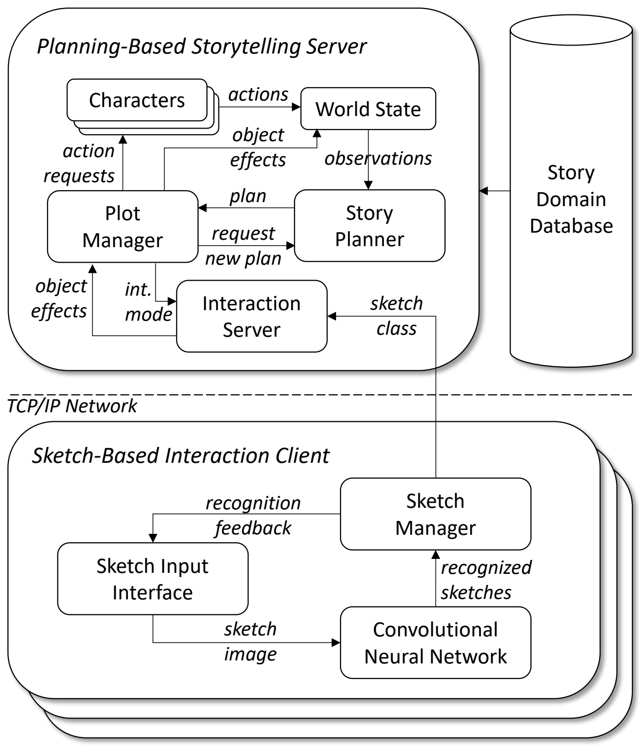

Fig. 2. Architecture of our interactive storytelling system.

A. *Sketch Recognition*

Sketch recognition has been an active research topic since Ivan Sutherland proposed the now famous SketchPad system in 1963 [27]. Since then, a variety of computational methods to recognize sketches have been explored, including hand-crafted models and deep learning models (see [32] for a survey on sketch recognition methods). In general, the process of identifying the class of a sketch can be modeled as a classification problem, where the system knows a set of drawing's classes (a vocabulary) and must recognize a new sketch based on its similarity to some member of the known set, which is a typical machine learning problem.

Deep learning models are known for their effectiveness in numerous computer vision tasks, including sketch recognition [32]. However, deep learning typically requires massive training data, which can be difficult to obtain. However, the recent release of the Quick Draw Dataset [25][9] has opened up new possibilities for research on sketch recognition. Quick Draw is the largest available dataset of sketches, containing over 50 million sketches of 345 different classes, such as clouds, cats, swords, cars, etc. The dataset was created with samples collected via an online web game called "Quick, Draw!",[4] where players were asked to sketch a specific object from 345 possible categories, in 20 seconds. Over 15 million players from around the world played the game and contributed to the dataset. Therefore, the dataset not only contains a huge number of samples from diverse object types, but also involves a diversity of drawing styles from different cultures, ages, and backgrounds.

The sketch recognition module of the proposed interaction method uses a deep learning model based on a Convolutional Neural Network trained with samples extracted from the Quick Draw dataset. More specifically, we use a preprocessed version of the dataset, where all sketches are rendered into 28x28 grayscale bitmaps in the NumPy file format (.npy), which matches the expected input data that the model will receive for the recognition process in real-time. The model's output comprises the classes of sketches that can be identified by the system, which is a subset of the 345 classes included in the Quick Draw dataset. Although the architecture of our Convolutional Neural Network can support the recognition of all 345 classes, using only the subset of sketches that are relevant for the story domain will improve the accuracy of the model (an evaluation of the model's accuracy for different subsets of sketches is presented in section V) and is therefore recommended.

Computer vision methods usually comprise three stages: feature extraction, feature reduction, and classification. In contrast to traditional methods where feature extraction and feature reduction are performed in a preprocessing phase, Convolutional Neural Networks combine these stages into the network itself. Therefore, input features do not need to be manually selected and extracted. Instead, Convolutional Neural Networks use their initial layers (convolutional layers and pooling layers) to automatically extract feature from the input data. While convolutional layers employ a set of filters (convolutional kernels) that are convolved with the input to generate feature maps, pooling layers operate on blocks of feature maps to reduce the dimensions of the data by combining the blocks into a single feature in the next layer. In the final layers of a Convolutional Neural Network, the extracted features are provided as input to fully connected layers that perform the classification process.

The architecture of our Convolutional Neural Network is illustrated in Fig. 3. The network consists of two convolutional layers, each followed by a max-pooling layer, which is then complemented by a flatten layer and two fully connected layers. The flatten layer converts input of any dimensionality to a dimensionality of $1 \times n$ (e.g., $5 \times 5 \times 64$ becomes 1×1600). In our network, convolutional layers and fully connected layers employ a Rectifier Linear Unit (ReLU) activation function, whereas the final output layer uses a Softmax function. There is a total of N neurons in the output layer, where N is defined by number of classes of sketches that can be identified by the Convolutional Neural Network, which depends on the story context where the sketch-based interaction method is being used. The structure of the network and its parameters were carefully chosen through preliminary experiments to maximize accuracy.

The Convolutional Neural Network was implemented in Python using the libraries Keras and TensorFlow. For the training and validation processes, we conducted several experiments with varying dataset sizes and classes of sketches. The results of these experiments and more details about the training procedure are presented in section V.

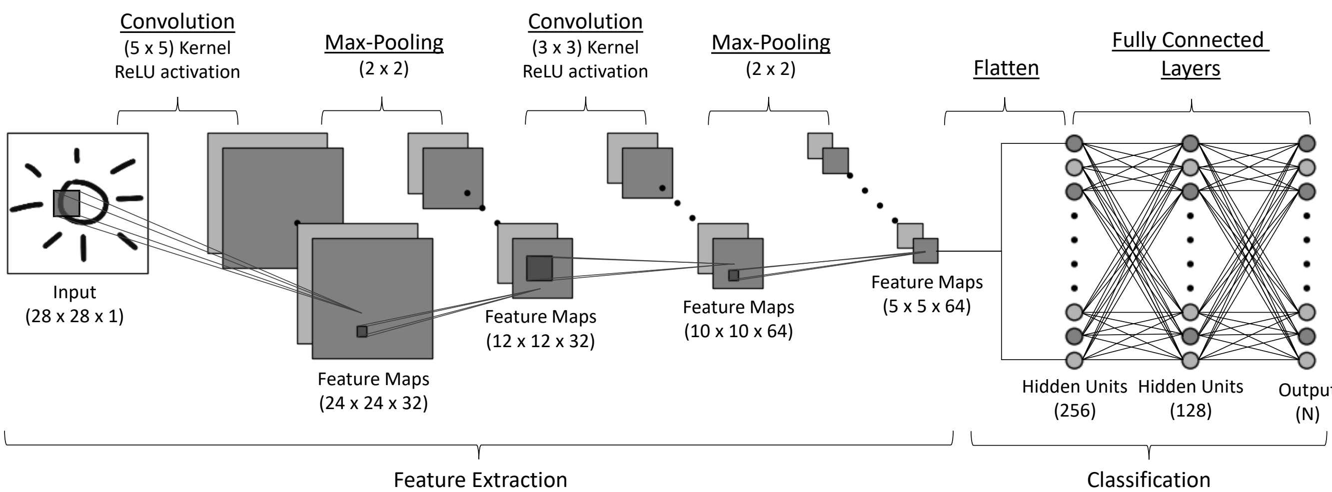

Fig. 3. Architecture of our Convolutional Neural Network.

[4] https://quickdraw.withgoogle.com/

In terms of application, the Convolutional Neural Network is integrated into an Android app, which uses the trained model to recognize the sketches drawn by users. The application makes use of the TensorFlow Lite library to load a compressed version of the model (.tflite file format), which is specially designed for mobile devices and includes several optimizations that do not affect accuracy. When a new sketch is drawn by users, the image is scaled down to 28x28 (matching the size of the training samples) and provided as input to the Convolutional Neural Network, which performs the classification of the sketch.

B. Story Planner

The story generation module of our system is based on a plot-based approach for interactive storytelling, where the story plot is automatically built and updated in real time by a planning algorithm, which was implemented as part of our previous work on automated planning for games [15][16][17]. The Story Planner uses a standard Heuristic Search Planning algorithm [2] that performs a forward search in the space of world states using a weighted A* algorithm [24]. The planner solves STRIPS-like planning problems that are dynamically formulated by the system according to a set of alternative goals and observations about the current state of the story that are extracted from the World State.

In our system, a planning problem is expressed by the tuple:

$$\Gamma = (P, O, S_0, G),$$

where P is a set of *atoms* (atomic formulas), O is a set of planning operators, S_0 is the initial state of the story (or the current state of the story when the problem is formulated for an ongoing story), and G is the goal state. An atom is an expression of the form $p(v_1, \ldots, v_k)$, where p is a predicate symbol and $v_1, \ldots, v_k$ are variable terms (e.g. `CH` and `PL`) or ground terms (e.g. `ana` and `forest`). Both $S_0 \subseteq P$ and $G \subseteq P$ are sets of ground literals, where each literal is an atom p or the negation of an atom $\neg p$.

An operator $o \in O$ represents a possible type of event that can occur during a narrative, which is denoted by:

$$o = (name(o), precond(o), effect(o)),$$

where:

- $name(o)$ is an atom $op(x_1, x_2, \ldots, x_k)$, where op is a unique symbol that identifies the operator, and x_i is a variable symbol that represents a parameter of o;
- $precond(o)$ is a set of literals that defines the preconditions of o (i.e. literals that must be true in the current state to allow the operator o to be executed); and
- $effect(o)$ is a set of literals that defines the effects of o (i.e. the positive and/or negative literals that will hold after the execution of the operator o).

Both P and O are defined as part of the conceptual schema of the story domain and are used to compose all planning problems. On the other hand, S_0 and G (initial state and goal state) can vary from one problem to another, expressing different situations that can occur in the course of a narrative as result of user interactions. The planning system handles the effects of user interaction with the story plot by adopting an online re-planning strategy [26], in which the Plot Manager is continuously monitoring the execution of the story to verify the consistency of the plot. If it detects that the current world state is different from the expected state described in the current plan in a way that violates the preconditions of future events, it requests a new plan to the Story Planner, where the current state of the world will be used as the initial state S_0 of the new planning problem.

To handle different goal states, the Story Planner uses a totally ordered set of alternative authorial goals $\Upsilon = \{g_1, g_2, \ldots, g_n\}$ under the relation $g_i \prec g_j$ (meaning that the attempt to achieve the goal g_i must occur before the attempt of reaching the alternative goal g_j). Each goal is a pair $g_i = \langle C_i, T_i \rangle$, where C_i is a set of literals that defines the preconditions of g_i (i.e. positive and/or negative literals that must hold in the current state to allow the planner to select g_i), and T_i is a set of ground literals that describe the goal state of g_i and can be used to establish G when formulating a new planning problem to be solved by the planner.

The set of alternative goals Υ is defined by the author of the story and it is used by the system to guide the development of the narrative towards admissible outcomes. When a new plan is requested by the Plot Manager, the Story Planner performs a search in Υ to find a goal g_i such that $C_i \in g_i$ holds in the current state of the story. Then, respecting the order in which the alternative goals are defined in Υ, the planner formulates a planning problem for g_i and tries to solve it. If the planner succeeds, the resulting plan is sent to the Plot Manager to be executed. Otherwise, if the goal state $T_i \in g_i$ cannot be achieved, the planner tries the next authorial goal (g_{i+1}). The planner can fail to achieve a desired goal either if there is no valid sequence of events that leads from the current state of the story to the goal state; or if the prescribed time limit for the searching process is exceeded. In both cases, the planner tries to achieve the next goal from the set of alternative goals. In this way, the author of the story domain can define the priority of the possible story outcomes by establishing the order of the of alternative goals in Υ.

The following examples show four alternative goals from the story domain of our prototype application:

```
g1:
  C1: currenttime(night), hungry(ana),
      ¬cansee(ana, village), ¬cansee(ana,
      camp).
  T1: met(ana, traveler), currenttime(night),
      cooked(ana, stonesoup, forest),
      ate(ana, stonesoup).
g2:
  C2: currenttime(day), hungry(ana),
      cansee(ana, village), ¬met(ana,
      traveler).
  T2: met(ana, friar), cooked(ana, stonesoup,
      friarhouse), ate(ana, stonesoup).
g3:
  C3: currenttime(night), hungry(ana),
      cansee(ana, camp).
  T3: met(ana, werewolf), cooked(ana,
      stonesoup, camp), ate(ana, stonesoup).
g4:
  C4: currenttime(day), hungry(ana), met(ana,
      traveler)
  T4: cooked(ana, stonesoup, forest),
      ate(ana, stonesoup).
```

The above examples define four different alternative goals: g_1 leads `ana` to meet a `traveler` at the `forest` during the night, where they cook and eat a stonesoup; g_2 leads ana to meet a friar at his house during the day, where they also cook and eat a stonesoup; g_3 also leads ana to cook and eat a stonesoup, but at this time with a werewolf that she meets at a camp; and g_4 is similar to g_1, but occurs during the day. The preconditions of the goals (C_1, C_2, C_3, and C_4) establish that: both g_1 and g_3 can only occur during the night, while g_2 and g_4 can only occur during the day; g_2 and g_3 require ana to be able to see the village or the camp, respectively; g_1 requires ana not to be able to see the village nor the camp; g_2 can only occur before ana meets the traveler, which is the opposite of g_4 that requires ana to meet the traveler as a precondition; and all goals require ana to be hungry.

When a goal is selected, a new planning problem can be formulated. The following example illustrates the representation of a planning problem:[5]

```
P: character(C), place(P), ingredients(I),
   food(F), time(T), object(O), effect(E),
   at(C, P), path(P1, P2), currenttime(T),
   met(C1, C2), has(C, O), hungry(C),
   knowneed(C2, C1, I), cooked(C, F, P),
   ate(C, F), cansee(C, P), cancause(C, E).

o1:
  name: go(CH, PL1, PL2)
  precond: character(CH), place(PL1),
           place(PL2), at(CH, PL1),
           path(PL1, PL2), cansee(CH, PL2)
  effect: at(CH, PL2), ¬at(CH, PL1)

o2:
  name: meet(CH1, CH2, PL)
  precond: character(CH1), character(CH2)
           place(PL), at(CH1, PL),
           at(CH2, PL)
  effect: met(CH1, CH2)

o3:
  name: ask(CH1, CH2, IN, PL)
  precond: character(CH1), character(CH2),
           ingredients(IN), place(PL),
           at(CH1, PL), at(CH2, PL),
           has(CH2, IN), hungry(CH)
  effect: knowneed(CH2, CH1, IN)

o4:
  name: give(CH1, CH2, IN, PL)
  precond: character(CH1), character(CH2),
           ingredients(IN), place(PL),
           at(CH1, PL), at(CH2, PL),
           has(CH2, IN),knowneed(CH2, CH1, IN)
  effect: has(CH2, IN), ¬has(CH1, IN)

o5:
  name: cook(CH, FO, IN, PL)
  precond: character(CH), food(FO),
           ingredients(IN), place(PL),
           at(CH, PL), has(CH, IN), hungry(CH)
  effect: cooked(CH, FO, PL), has(CH, FO),
          ¬has(CH, IN)

o6:
  name: eat(CH, FO, PL)
  precond: character(CH), food(FO),
           ingredients(IN), place(PL),
           at(CH, PL), has(CH, FO), hungry(CH)
  effect: ate(CH, FO), ¬has(CH, FO),
          ¬hungry(CH)

S0: character(ana), character(traveler),
    character(friar), character(werewolf),
    food(stonesoup), place(forest),
    place(lake), place(village), place(camp),
    place(friarhouse), time(day), time(night),
    object(candle), effect(wildfire),
    ingredient(soapingredient), at(ana,
    forest), at(friar, village), at(traveler,
    lake), at(werewolf, camp), hungry(ana),
    cansee(traveler, forest), has(traveler,
    soapingredient), currenttime(night),
    path(lake, forest), path(forest, lake),
    path(forest, camp), path(camp, forest),
    path(forest, village), path(village,
    forest), path(village, friarhouse),
    path(friarhouse, village).

G: met(ana, traveler), currenttime(night),
   cooked(ana, stonesoup, forest), ate(ana,
   stonesoup).
```

In the above example, P defines the vocabulary of atoms used to describe the problem, and o_1 to o_6 describes six operators based on the well-known STRIPS formalism that represent possible events for the story (go, meet, ask, give, cook, eat). The initial state S_0 defines that ana, traveler, friar, and werewolf are characters; stonesoup is a food; forest, lake, village, camp, and friarhouse are places; day and night represent time; candle is an object; soapingredient is an ingredient; wildfire is an effect; ana is at forest; friar is at the village; traveler is at the lake; werewolf is at the camp; ana is hungry; traveler can see the forest and has the soapingredient; the current time is night; and there is a path connecting the lake with the forest (amongst other paths connecting places). Lastly, G is the goal state defined in g_1.

When the planning problem is solved, the plot is established as a linear sequence of events or actions to be performed by the virtual characters. In the above example, the plot comprises:

```
go(traveler, lake, forest), meet(ana,
traveler, forest), ask(ana, traveler,
soapingredient, forest), give(traveler, ana,
soapingredient, forest), cook(ana, stonesoup,
soapingredient, forest), eat(ana, stonesoup,
forest).
```

The classes of objects that can be sketched by users and inserted into the story world are defined in a set $\Lambda = \{ob_1, ob_2, \dots, ob_n\}$. Each object ob_i is a pair $ob_i = \langle N_i, E_i \rangle$, where N_i is the class name of ob_i, and E_i is a set of literals that defines the effects of ob_i (i.e. the positive and/or negative literals that will hold in the world state after the insertion of ob_i). The following examples illustrate the logic behind two objects (sun and candle):

```
ob1:
   N1: sun
   E1: ¬currenttime(night), currenttime(day),
       cansee(ana, village), cansee(ana,
```

[5] For the sake of clarity, we omitted some operators, atoms, preconditions, and effects that were not relevant for the example.

```
        lake), cansee(ana, camp), cansee(ana,
        friarhouse).
ob2:
    N2: candle
    E2: has(ana, candle), cansee(ana, camp),
        cancause(ana, wildfire).
```

As can be observed in the effects of ob_1, the insertion of a `sun` will change the current time from `night` to `day` and will allow `ana` to see all nearby places. In the case of ob_2, `ana` will receive the `candle` and will be able to see the `camp`, but also will be subject to the danger of causing a `wildfire`. The final result in the story also depends on the point of the plot where the object is inserted. For example, if the user inserts the `sun` while `ana` is lost in the `forest` during the `night`, the effects of the `sun` will create an inconsistency in the plan generated for g_1, which will trigger g_2 and lead the story towards the outcome where she goes to the `village`. However, if the user inserts the `sun` after `ana` has met the `traveler`, the only goal available will be g_4, which will lead `ana` and the `traveler` to continue their interaction in the `forest` during the `day`.

IV. Prototype Application

In order to test and validate the proposed interaction method, we implemented an educational interactive narrative for our interactive storytelling system. The narrative is designed for children (7-10 years old) and is aimed at increasing their interest and engagement with folktales. The narrative explores the intersection of elements extracted from four traditional Portuguese folktales: *A Sopa de Pedra* (The Stone Soup) [22], *O Lobisomem de Fareja* (The Werewolf from Fareja) [21], *A Velha e os Lobos* (The Old Woman and the Wolves) [21], and *Nossa Senhora da Alegria* (*Our Lady of Joy*) [21]. The narrative is structured around the storyline of "The Stone Soup" with additional branches that result from events, characters, and objects extracted from the other folktales. "The Stone Soup" is a traditional European folktale where a hungry foreigner persuades the residents of a village to share their ingredients for the preparation a soup, which has a stone as the main ingredient (in some variants, the stone is replaced by other objects, such as an axe, button, nail, and wood). The moral of the story teaches about the importance of sharing, which can then be used by educators as a starting point for further discussions on sharing.

In terms of user interaction, there are 14 objects that can be sketched by users and inserted into story world: "sun", which turns night into day; "campfire", "candle", "lighter", and "matches", which are used as sources of light during the night, but are also dangerous and can cause a wildfire; "axe", "sword", "knife", and "scissors", which can be used to free tied characters; "stone",[6] "carrot", and "broccoli", which are ingredients that can be used to cook a soup; "rain", which can extinguish a wildfire; and "cat", which can distract the characters of the story. Fig. 4 shows an example of each class of sketch that can be identified by the sketch recognition system in our prototype application.

Considering the opportunities for user interaction, the prototype application can generate a considerable number of diversified stories. In more conventional stories, Ana (the hungry stranger) finds her way out of the forest and meets a friar, who helps her cooking the stone soup in his house; in other variants, the friar refuses to help Ana, who decides to cook the stone soup outside of the friar's house while the friar curiously observes. In other storylines, Ana does not find her way to the village, but ends up finding a camp in the forest, where she meets a werewolf, who helps her cook the stone soup. In more unconventional stories, Ana causes a wildfire in the forest, which is extinguished by rain or by the miraculous apparition of Our Lady of Joy, who also provides Ana with some ingredients for the stone soup.

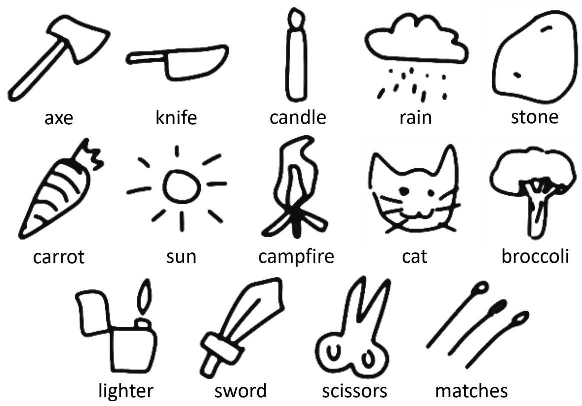

Fig. 4. Classes of sketches that can be identified by the sketch recognition system in our prototype application.

Fig. 5 shows a scene from the prototype application and the user interface of the mobile app, where the user is sketching an object to be inserted into the story world (a carrot).

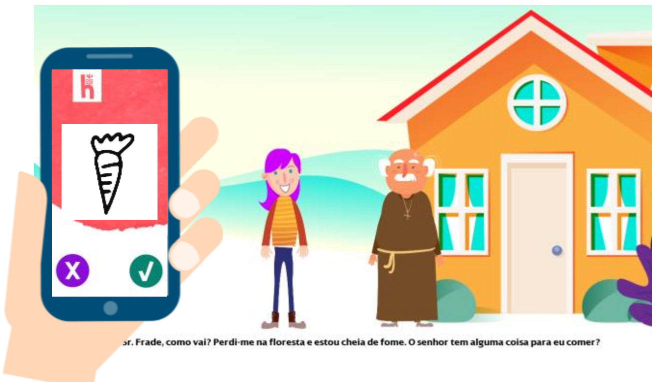

Fig. 5. A scene from the prototype application and the user interface of the mobile app.

V. Evaluation and Results

The sketch recognition process involves algorithms that are not completely accurate but provide the fundamental basis of the proposed interaction method. Therefore, an evaluation of their accuracy and performance is mandatory. For this evaluation, we conducted two experiments: (1) a precision test to check the accuracy of the Convolutional Neural Network for varying dataset sizes and number of sketch classes; and (2) a performance test to evaluate the real-time performance of the sketch recognition system on mobile devices.

For the precision test, we created 12 datasets with varying sizes (1000, 10000, and 100000 samples) and different

[6] Although the Quick Draw dataset does not have a category for stones, we used the category of potatoes to represent stones, which are visually similar when depicted as a sketch.

numbers of sketch classes (14, 56, 172, and 345 classes), which allowed us to evaluate the accuracy of the model for different usage scenarios. All datasets were created with samples randomly extracted from the Quick Draw dataset, but we maintained the datasets balanced (i.e. each class has the same number of samples in each dataset). The sketch classes were also arbitrarily selected, except for the datasets of 345 classes, which included all classes of the Quick Draw dataset; and the datasets of 14 classes, which included only the classes used by our prototype application (a real usage scenario). For the validation process, each dataset was divided into training and testing sets (75% of the samples of each dataset were used for training the Convolutional Neural Network and the remaining samples were used for testing). In all the experiments, we used a 10-fold cross-validation strategy, where the division of the datasets into training and testing sets was repeated 10 times (varying the samples used for training and testing), and then the average accuracy was calculated. Each network was trained for a maximum of 20 epochs.

The results of the precision test are shown in Table I. The accuracy of the Convolutional Neural Network clearly increases with the number of training samples and decreases when more classes are added. However, the accuracies obtained with the datasets of 100000 samples are promising and show that the system can correctly identify the sketches in most cases (even when all 345 sketch classes are considered). This is even more remarkable when considering that not all interactive narratives require large vocabularies of recognizable sketches. For example, in our prototype application, only 14 classes of sketches were enough to provide users with a good variety of interaction options; and for that set of sketches, the Convolutional Neural Network achieved an accuracy of 95.1%.

TABLE I. ACCURACY OF THE CONVOLUTIONAL NEURAL NETWORK FOR VARYING DATASET SIZES AND NUMBER OF SKETCH CLASSES.

Number of Classes	Number of Training Samples	Accuracy (%)
14	1000	84.6%
	10000	91.5%
	100000	95.1%
56	1000	73.9%
	10000	86.0%
	100000	90.6%
172	1000	66.0%
	10000	75.1%
	100000	87.4%
345	1000	56.5%
	10000	65.1%
	100000	71.6%

To evaluate the performance of the sketch recognition system on mobile devices, we performed the classification of 100 sketches drawn by a user in our mobile application. For each classification, we computed the time necessary to scale down the image and to recognize the sketch using our Convolutional Neural Network, which was trained with 100000 samples of 14 classes of sketches. The smartphone used to run the experiment was a Motorola Moto X4 XT1900, Octa-core 2.2 GHZ CPU, 4 GB of RAM. As a result, we got an average time of 5.4 milliseconds (standard deviation of 1.3 milliseconds), which indicates the applicability of the proposed method in highly interactive storytelling systems without noticeable delays in the interaction process.

VI. CONCLUSION REMARKS

In this article, we present a novel interaction method for interactive storytelling, which uses sketch recognition and planning-based plot generation to allow users to interact with narratives by drawing objects on smartphones or tablet computers. The proposed method provides designers, developers, and story writers with new ways of creating interactive narrative experiences. In addition, we believe that this means of play-learning, one that is based on first-hand, experimentation, is particularly suited for encouraging creativity while also stimulating children's thinking and imagination.

In our experiments, the proposed sketch recognition system reveals encouraging results. The Convolutional Neural Network shows good overall accuracy, especially considering that small sets of sketch classes are sufficient to provide a good variety of interaction options for most interactive storytelling systems. For example, the system presented by Lima et al. [19] uses 6 classes of sketches, which can be recognized with 93.8% of accuracy. In our prototype application, we expand the set to 14 classes and achieve a 95.1% of accuracy. Furthermore, the use of the Quick Draw dataset simplifies the process of creating new vocabularies of sketches for new narratives as the dataset already contains 345 classes of sketches ready to be used.

As further research, we intend to conduct user studies to evaluate our interaction method from the user's perspective. Since, so far, our primary focus has been the technical aspects, we have not yet conducted a rigorous user study, which surely is a mandatory task that will serve as orientation for the next stages of our project.

REFERENCES

[1] J. Barth, "The Literature of Exhaustion," in J. Barth (ed.), The Friday Book: Essays and Other Non-Fiction, London: The John Hopkins University Press, 1984.

[2] B. Bonet, and H. Geffner, "Planning as Heuristic Search," Artificial Intelligence, vol. 129 (1), pp. 5-33, 2001, doi: 10.1016/S0004-3702(01)00108-4.

[3] A. Bonnici et al., "Sketch-based interaction and modeling: where do we stand?," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol. 33 (4), pp. 370-388, doi: 10.1017/S0890060419000349.

[4] A. E. M. Ciarlini, C. T. Pozzer, A.L. Furtado, & B. Feijo, "A logic-based tool for interactive generation and dramatization of stories," Proceedings of the International Conference on Advances in Computer Entertainment Technology, Valencia, 2005, pp. 133-140, doi: 10.1145/1178477.1178495.

[5] S. Cox, "Intention and meaning in young children's drawing," International Journal of Art and Design Education, vol. 24 (2), pp. 115-125, 2005, doi: 10.1111/j.1476-8070.2005.00432.x.

[6] L. Feng, X. Yang, S. Xiao, "MagicToon: A 2D-to-3D creative cartoon modeling system with mobile AR," 2017 IEEE Virtual Reality (VR), Los Angeles, CA, 2017, pp. 195-204, doi: 10.1109/VR.2017.7892247.

[7] T. C. Franco, and E. S. Lima, "Paper and Pencil Interactive Storytelling Based on Generic Hand-drawn Sketches," Proceedings of the XVI Brazilian Symposium on Computer Games and Digital Entertainment, Curitiba, Brazil, 2017, pp. 594-597.

[8] M. Ghallab, D. Nau, and P. Traverso, Automated Planning: Theory and Practice, United States: Morgan Kaufmann Publishers, 2004.

[9] D. Ha, and D. Eck, "A neural representation of sketch drawings," arXiv preprint arXiv:1704.03477, 2017.

[10] N. Hagbi, R. Grasset, O. Bergig, M. Billinghurst, and J. El-Sana, "In-Place Sketching for Augmented Reality Games," Computers in Entertainment, vol. 12 (3), Article No. 3, December 2014, doi: 10.1145/2702109.2633419.

[11] M. K. Hu, "Visual pattern recognition by moment invariants," IRE Transactions on Information Theory, vol. 8 (2), pp. 179-187, 1962, doi: 10.1109/TIT.1962.1057692.

[12] D. T. Huynh, K. Raveendran, Y. Xu, K. Spreen, and B. MacIntyre, "Art of defense: a collaborative handheld augmented reality board game," Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games, New Orleans, United States, 2009, pp. 135-142, doi: 10.1145/1581073.1581095.

[13] C. Johnson, Harold and the Purple Crayon, New York: Harper, 1955.

[14] D. Kuka, O. Elias, R. Martins, C. Lindinger, A. Pramböck, A. Jalsovec, P. Maresch, H. Hörtner, and P. Brandl, "Deep space: high resolution VR platform for multi-user interactive narratives," Proceedings of the 2nd Joint International Conference on Interactive Digital Storytelling, Guimarães, Portugal, 2009, pp. 185-196, doi: 10.1007/978-3-642-10643-9_24.

[15] E. S. Lima, B. Feijó, and A. L. Furtado, "Player Behavior Modeling for Interactive Storytelling in Games," Proceedings of the XV Brazilian Symposium on Computer Games and Digital Entertainment (SBGames 2016), São Paulo, Brazil, 2016, p. 1-10.

[16] E. S. Lima, B. Feijó, and A. L. Furtado, "Player Behavior and Personality Modeling for Interactive Storytelling in Games," Entertainment Computing, vol. 28, p. 32-48, 2018, doi: 10.1016/j.entcom.2018.08.003.

[17] E. S. Lima, B. Feijó, and A. L. Furtado, "Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning," Proceedings of the XVIII Brazilian Symposium on Computer Games and Digital Entertainment (SBGames 2019), Rio de Janeiro, Brazil, 2019 p. 495-504, doi: 10.1109/SBGames.2019.00028.

[18] E. S. Lima, B. Feijó, A. L. Furtado, S. D. J. Barbosa C. T. Pozzer, and A. E. M. Ciarlini, "Non-branching Interactive Comics," Proceedings of the 10th International Conference on Advances in Computer Entertainment Technology, Netherlands, 2013, pp. 230-245, doi: 10.1007/978-3-319-03161-3_16.

[19] E. S. Lima, B. Feijó, S. D. J. Barbosa, A. L. Furtado, A. E. M. Ciarlini, and C. T. Pozzer, "Draw your own story: Paper and pencil interactive storytelling," Entertainment Computing, vol. 5 (1), pp. 33-41, 2014, doi: 10.1016/j.entcom.2013.06.004.

[20] C. D. D. Monteiro, M. Narayanan, S. Polsley, and T. Hammond, "A Multilingual Sketch-Based Sudoku Game with Real-Time Recognition," In: T. Hammond, A. Adler, and M. Prasad (eds) Frontiers in Pen and Touch. Human-Computer Interaction Series, Springer, pp. 187-196, 2017, doi: 10.1007/978-3-319-64239-0_13.

[21] J. V. Moutinho, Lendas de Portugal, Lisboa: Diario de Notícias, 2013.

[22] J. V. Moutinho, Seis Histórias Tradicionais Portuguesas, Lisbon: Fábula, 2017.

[23] B. Paulson, B. Eoff, A. Wolin, J. Johnston, and T. Hammond, "Sketch-based educational games: drawing kids away from traditional interfaces," Proceedings of the 7th International Conference on Interaction Design and Children, Chicago, Illinois, 2008, pp. 133-136, doi: 10.1145/1463689.1463739.

[24] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Boston: Addison-Wesley, 1985.

[25] Quick Draw Dataset, Google, 2017. [Online]. Available at: https://github.com/googlecreativelab/quickdraw-dataset.

[26] S. Russell, and P. Norvig, Artificial Intelligence: A Modern Approach. Third edition, New Jersey: Prentice Hall, 2010.

[27] I. E. Sutherland, "Sketchpad, a man-machine graphical communication system," Ph. D. Thesis, Dept. of Electrical Engineering, Massachusetts Institute of Technology, United States, 1963.

[28] S. ter Stal, "Designing an Interactive Storytelling System for Children using a Smart Toy Design of a prototype to investigate the effect of emotional behaviour of a toy on children's storytelling," M.S. Thesis on Human Media Interaction, Department of Electrical Engineering, Mathematics and Computer Science, University of Twente, Netherlands, 2017.

[29] A. Vogelsang, and B. Signer, "The lost cosmonaut: an interactive narrative environment on the basis of digitally enhanced paper," Proceedings of 3rd International Conference on Virtual Storytelling, Strasbourg, France, 2005, pp. 270-279, doi: 10.1007/11590361_31.

[30] D. Wang, L. He, and K. Dou, "StoryCube: Supporting children's storytelling with a tangible tool," The Journal of Supercomputing, vol. 70 (1), pp. 269-283, 2014, doi: 10.1007/s11227-012-0855-x.

[31] B. Williford, M. Runyon, J. Cherian, W. Li, J. Linsey, and T. Hammond, "A Framework for Motivating Sketching Practice with Sketch-based Gameplay," Proceedings of the Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '19), Barcelona, Spain, 2019, pp. 533-544, doi: 10.1145/3311350.3347175.

[32] X. Zhanga, X. Lia, Y. Liua, and F. Fengc, "A survey on freehand sketch recognition and retrieval," Image and Vision Computing, vol. 89 pp. 67-87, 2019, doi: 10.1016/j.imavis.2019.06.010.