Reinforcement based Coherent Training Framework to Improve Predictive Skills for Individuals with Autism

Thesis submitted by

SATYA KISHORE BUDUMURU
2012CS50284

under the guidance of

Prof. Tapan Kumar Gandhi and Prof. M. Balakrishnan

in partial fulfilment of the requirements
for the award of the degree of

Bachelor and Master of Technology

Department Of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY DELHI

September 2017

THESIS CERTIFICATE

This is to certify that the thesis titled **Reinforcement based Coherent Training Framework to Improve Predictive Skills for Individuals with Autism**, submitted by **Satya Kishore Budumuru**, to the Indian Institute of Technology, Delhi, for the award of the degree of **Master of Technology**, is a bonafide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Tapan K. Gandhi
Asst. Professor
Dept. of Electrical Engineering
IIT-Delhi, 110016

Place: New Delhi

Date: 14th September 2017

ACKNOWLEDGEMENTS

I express my sincere gratitude to my supervisor, mentor Dr. Tapan K. Gandhi for all his support and guidance over the course of this project, and for being such a relaxed but motivating supervisor. Without his guidance, the work would not see the light of the day. I would also like to thank Dr. M. Balakrishnan for his support and feedback throughout the project.

I would like to thank Dr. Nidhi Singhal, Director, Action For Autism for her valuable feedback. I would like to thank Dr. Shruthi along with Dr. Nidhi for helping us clear ethics and providing us children with autism for the project. I would also like to thank all the children with ASD who played the game and their parents for their valuable time and patience.

I would like to thank Tarun Kota for helping me in making games in Unity and all other friends who have supported me. Finally, I express my gratitude to my parents who have shown a great deal of faith in me.

Satya Kishore Budumuru

DEDICATION

Dedicated to all individuals in Autism Spectrum Disorder and their parents.

ABSTRACT

KEYWORDS: Autism; Magical World; Impaired Predictive Abilities; Knowledge space; Reinforcement Learning.

Individuals with autism, a heritable life-long neurodevelopmental disorder, often face social communication difficulties and have restricted repetitive behaviours. These individuals experience a "magical" world i.e., they can't figure out the causes of events due to their impaired predictive skills. A lot of work has been focused towards training individuals with autism using Virtual Reality, etc. Here, we make the training mechanism more effective by creating a mathematical framework which attempts to train individuals with autism by taking their impaired predictive skills into account. The framework first models reward in terms of predictive ability of individual in different dimensions and uses it to find the predictive ability of the individual from the actions he/she has taken. This will help the framework create a world which is not magical to autistic individual, thus making the reinforcement more effective.

The model framework uses to asses the impaired predictive skills uses a Gaussian distribution, with it's variance as a parameter that defines individual's predictive skills (IPS). Next, from the actions individual takes, his predictive skills are computed using Markov Decision Process. For each puzzle, the reinforcement this puzzle provides for a given IPS is computed and the best puzzle is provided to them so that the learning is maximized. Every decision making problem can be categorized into spatial or temporal or both. To evaluate the framework, we created two puzzles namely "K" in spatial domain and "Stick Hero" in temporal domain with coherent and random training mechanisms.

As pilot testing, we got 3 children for testing. From the results obtained we can state that the framework is helping individuals with autism learn 20% faster in Spatial Domain and 25% times faster in Temporal Domain. Rigours evaluation will be done after ethics get cleared in October, 2017. Since, the framework is domain independent, it can be further deployed into real life situations like road crossings, motor skills etc.,.

Contents

List of Figures

ABBREVIATIONS

IITD	Indian Institute of Technology, Delhi
IPS Map	Individual's Predictive Skills Map
HPC	High Performance Map
MDP	Markov Decision Process

NOTATION

σ	Variance in Gaussian Distribution

Chapter 1

MOTIVATION

"If they can't learn the way we teach, we teach the way they learn."

O. Ivar Lovaas

Autism is a heritable lifelong neuro-developmental disorder characterized by impaired social communication, verbal and non-verbal communication. Individuals with autism often have restricted, repetitive patterns of behaviors, interests, or activities [LRL+00, LRLC94]. The difficulties individuals with autism experience which include motor control [Hug96, SMBA03, BWS+12, FKR+10, MDB03, MSJF04], language [TFPL05], social interactions & planning [Fee10, Rob08] can be traced back to their inability to predict things [SKG+14]. Also, because of their inablity to predict things, they experience the world **magically** i.e., individuals with autism can't trace the cause of events or dependencies between events. Unlike, the belief that individuals with autism are bad at everything, there are areas which include mathematics [IRLS+14], block design test, rubiks cube solving, drawing abilites, physics where these individuals perform better or atleast at par with other individuals. All these areas require rule based, where predictive abilities are less required. A lot of literature suggests individuals with autism are good at 'if p then q' types of rules [BCAA+09]. Also, individuals with autism have impaired predictive skills because of their skewed priors [PB12], one of the two parameters in prediction, in bayesian decision theory [KW04].

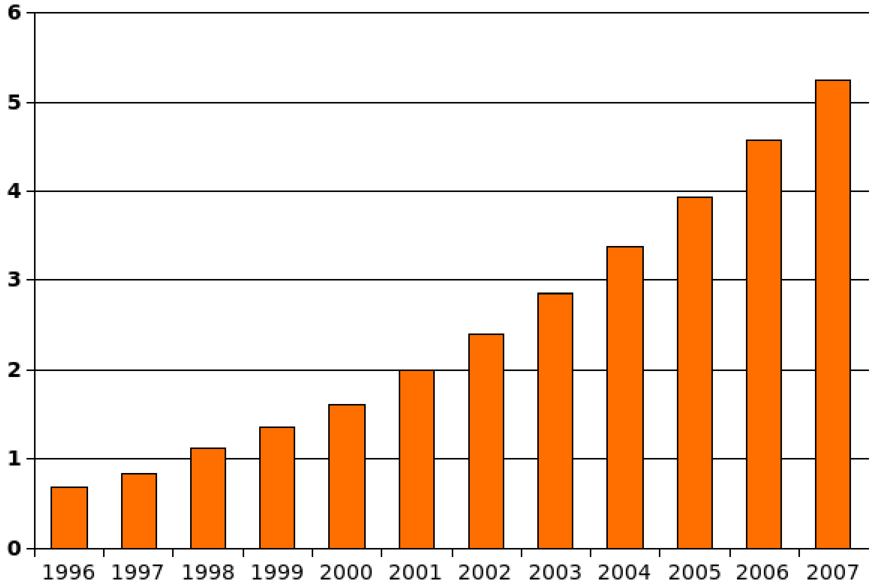
1.1 Demographics

As of 2015, autism is estimated to affect 24.8 million people across the globe. About one in 68 in US are diagnosed with ASD as of 2014. Due to improvements in diagnostic practices, the number of people diagnosed has been increasing dramatically since the 1980s. Figure 1.1 shows the increased in number of children with autism in US.

1.2 Framework

In this project, we created a framework to add this research in the design of useful tools in intervention and training. Earlier the tools of intervention for autism employed conventional

Figure 1.1: Reports of autism cases per 1,000 children in the US from 1996 to 2007



techniques to provide puzzles. In this project, we created an mathematical framework which can provide puzzles in a coherent way to maximize the learning by individuals with autism.

The framework proposed in this project can predict the magicalness of the world experienced by individuals with autism from the actions the individual has taken. It also has pre-computed reinforcement map for each puzzle for each magical world to predict the expected reinforcement user gets when he/she tries to solve it. These maps are useful for the framework to present the puzzles in a coherent way to maximize the user's learning.

Chapter 2

Technical Background

"If I have seen further than others, it is by standing upon the shoulders of giants."

Isaac Newton, 1676

2.1 Bayesian Principle

Bayes theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. For example, if cancer is related to age, then, using Bayes' theorem, a person's age can be used to more accurately assess the probability that they have cancer, compared to the assessment of the probability of cancer made without knowledge of the person's age.

2.2 Markov Decision Process

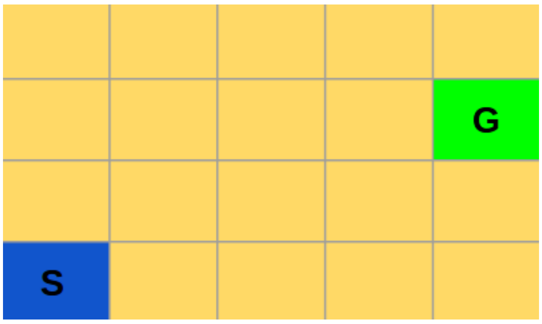
Markov decision process provides a mathematical framework for modelling decision making in scenarios where the outcome of action is not completely deterministic.

2.2.1 MDP Description

There are 4 important parts in decision making namely State Space, Action Space, Transition Probability and Reward.

- **State Space: S**
  The set of all possible states. For example, on, off describe the state of the fan.
- **Action Space: A**
  A fixed set of all possible actions. For example switching on or switching off the fan.
- **Transition Probabilities: T(<S, A, S'>)**
  The probability of going from state S to state S' given an action A. For example, T(on, switch off, off) is the probability of an fan switching off if the action is switch off. Ideally, it should be 1, but if the switch is faulty then it could be less than 1.

Figure 2.1: Game Board



- **Reward: R(S)**
  Defines the reward when state is S. Generally, this is the reinforcement which guides the planning.

2.2.2 MDP Example

In figure 2.1, there is a grid. The goal of the game is to move the S from blue area to green. At each point, a direction has to be chosen by user. After that a dice is rolled and the S has to move equal number of grids in the direction specified. If the S can't move by those many steps, the dice is rolled again. The objective function of the game is reduce the number of dice rolls. For ease of calculations, we took dice having 5 numbers. Now, to get the best actions, let's model it to MDP.

- **State Space: S**
  <x, y> where x, y is location of S on the grid.

- **Action Space: A**
  <Left>, <Right>, <Up>, <Down>

- **Transition Probabilities: T(<S, A, S'>)**
  T(<x, y>, right, <x+1, y>) = 1/5
  T(<x, y>, right, <x+2, y>) = 1/5
  ....
  T(<x, y>, up, <x+6, y>) = 1/5
  T(<x, y>, up, <x, y+1>) = 1/5
  T(<x, y>, up, <x, y+2>) = 1/5
  ....
  T(<x, y>, up, <x, y+6>) = 1/5
  T(<x, y>, up, <x, y+2>) = 1/5
  ....
  ....

- **Reward: R(A)**
  -1 for each action and 100 if the avatar reaches the goal.

Solving the MDP

While solving the MDP using value iteration, an algorithm to solve the MDP, each state is assigned a value which is the average amount of reward that is obtained from that state to reward state. **V** is an array that saves the. Also, at each state, the optimal action is also selected and saved in an array π.
The algorithm has two steps, which are repeated for all states until there is no further change. The steps are recursively defined as follows:

$$\pi(S) = \arg\max_{action}(\textstyle\sum_{S'} T(S, action, S') * [R(S, a) + V(S')])$$

$$V(S) = \textstyle\sum_{S'} T(S, \pi(S), S') * [R(S, \pi(S)) + V(S')])$$

2.3 Reinforcement Learning

Reinforcement learning is a technique to compute unknown transition probabilities or unknown reward in MDP models. In reinforcement learning, the agent or in our case the user, takes actions which in general are not optimal, and receives a reward which is not as expected. The agent/user then updates the transition probabilities to justify the observation. In the popular Q-learning algorithm for reinforcement learning, the reinforcement or the amount of learning is stated as the difference of actual reward and expected reward.
Reinforcement = Actual Reward - Expected Reward.

2.4 Gaussian Distribution

Gaussian Distribution is a continuous probability distribution often used to describe random variables natural sciences. The problems dealt in this thesis is similar to ball in a Galton Box which is briefly described in figure 2.2. The distribution achieved in figure 2.2 is binomial distribution which approximates to Gaussian when n is considerably large.
Galton box machine consists of a vertical board with interleaved rows of pins. Balls are dropped from the top, and bounce either left or right as they hit the pins. Eventually, they are collected into one-ball-wide bins at the bottom as shown in figure 2.3. The resulting distribution is binomial distribution. In Probability density function, μ is the mean or expectation of the distribution, σ is the standard deviation of the distribution and σ^2 is the variance of the distribution.

Figure 2.2: Reinforcement Learning

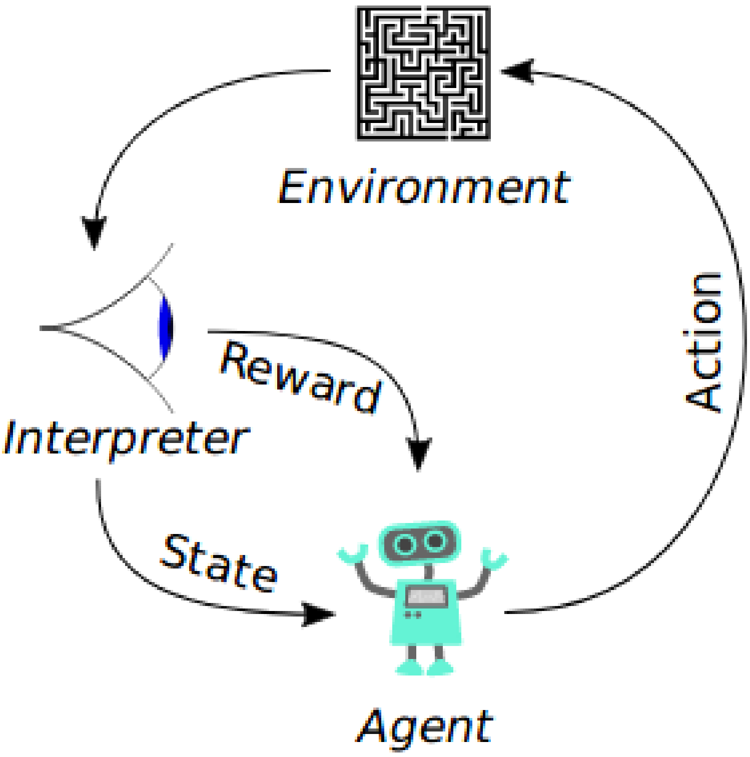


Figure 2.3: Probability Density Function of Gaussian Distribution

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Figure 2.4: Probability Density Function of Gaussian Distribution

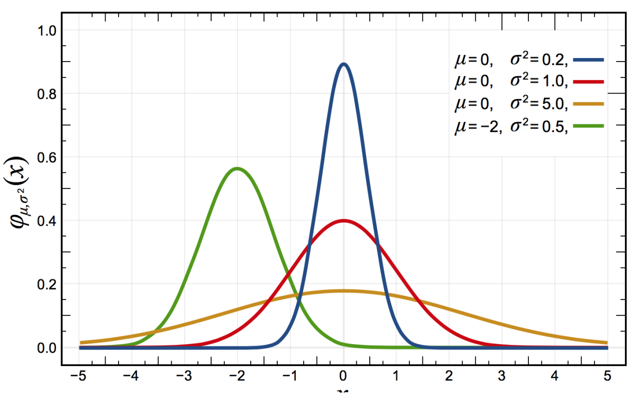


Figure 2.5: The probability of a ball in a Galton Box

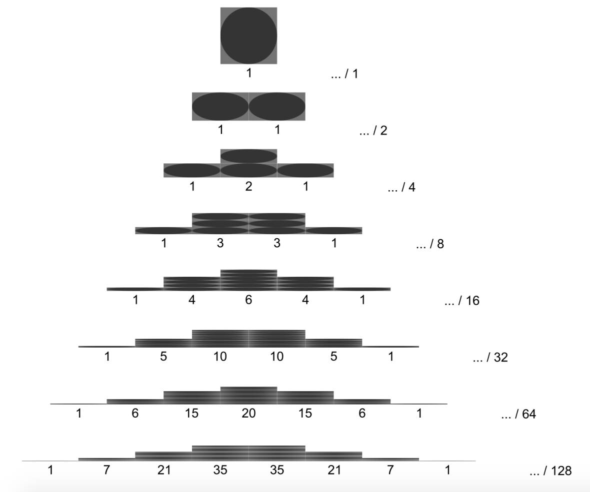
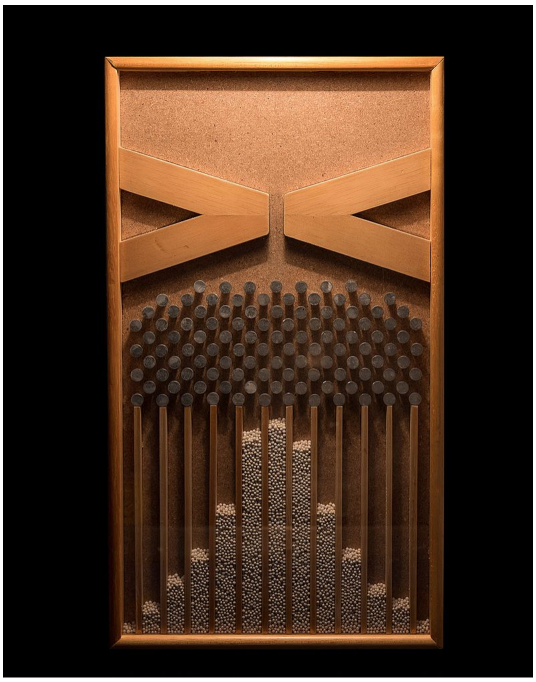


Figure 2.6: Description of Galton box



Chapter 3

Framework

Markov Decision Process (MDP), provides a mathematical framework for modelling decision making in various scenarios. As seen in section 2.2, there are 4 important parts in decision making namely State Space, Action Space, Transition Probability and Reward. A lot of research states that there is no problem for autistics in detecting 'if p, then q' type of rules. [cite talent]. Thus State Space, Action Space and Rewards for autistics is similar to that of high functioning individuals and can be modelled in a similar way. Also, to model autistics way of decision making, we need to modify the transition probabilities for they can't predict well. In this project, we model the transition probabilities using Gaussian Distribution whose variance defines the degree of impairedness of predictive skills (IPS) of the autistic individual. Since, decision making in humans uses bayesian theory [cite here] the usage of MDP to compute rewards will be an good approximation.

3.1 Parameterizing IPS

The intuition behind adding Gaussian Distribution is if we ask an individual to predict where a ball will hit a stick after reflection or when thrown directly, for multiple times and plot them on a distribution, we will get a gaussian distribution.

Now, the variance of this gaussian distribution implies the degree of impaired predictive skills (IPS) of the individual.

So, each problem can be modelled into this modified MDP with added parameters. So, IPS of an individual is an vector of size defined by the modelling, each dimension is a sigma of that modified parameter. Each IPS defines an magical world and thus magical world and IPS are used interchangeably.

Figure 3.1: Modeling Decision Making in individuals with autism

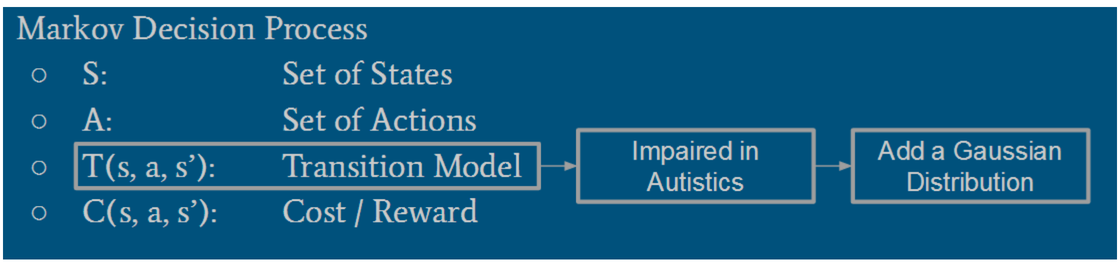


Figure 3.2: Intuition behind Gaussian Distribution - 1

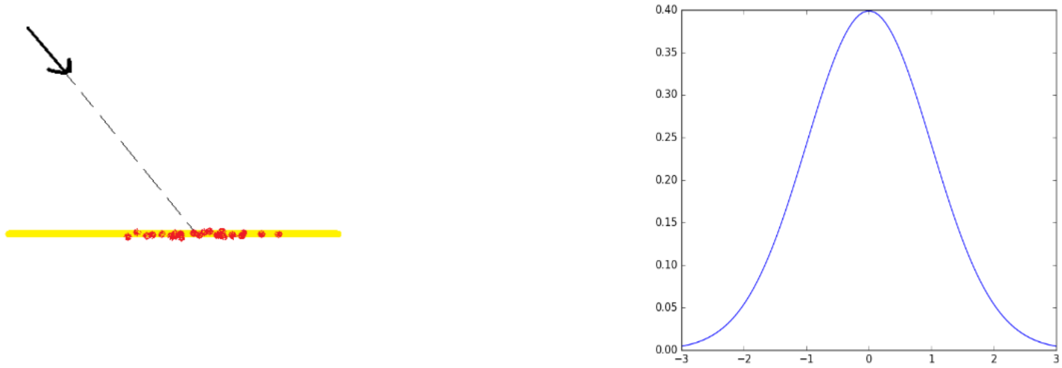


Figure 3.3: Intuition behind Gaussian Distribution - 2

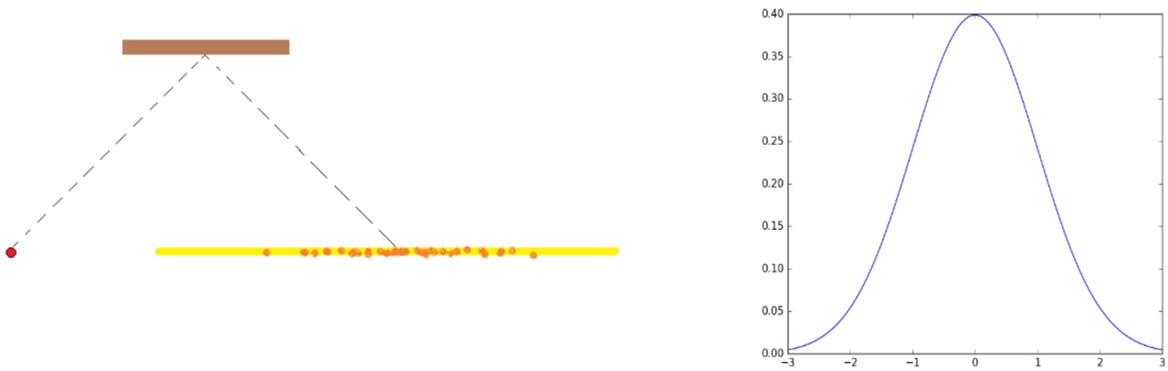


Figure 3.4: Variance as IPS - 1

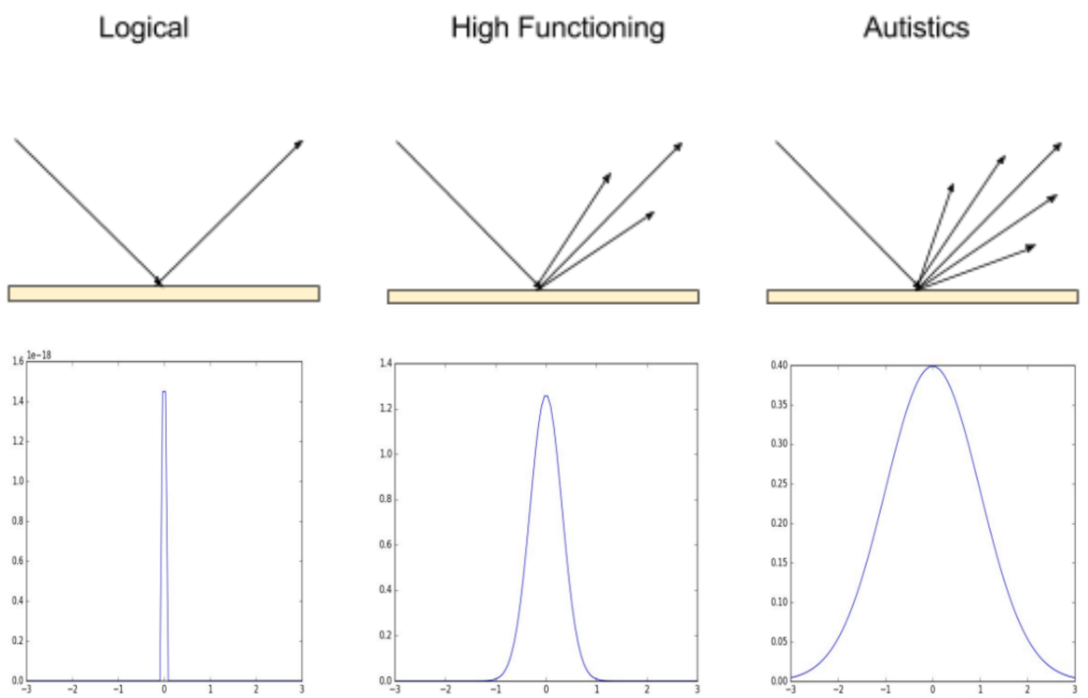
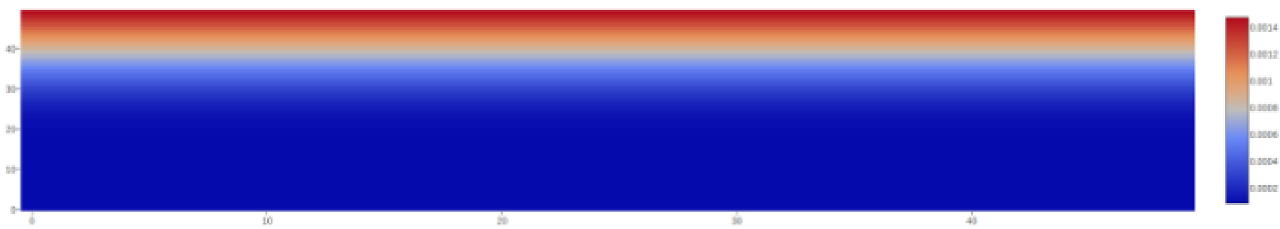


Figure 3.5: Sample IPS map

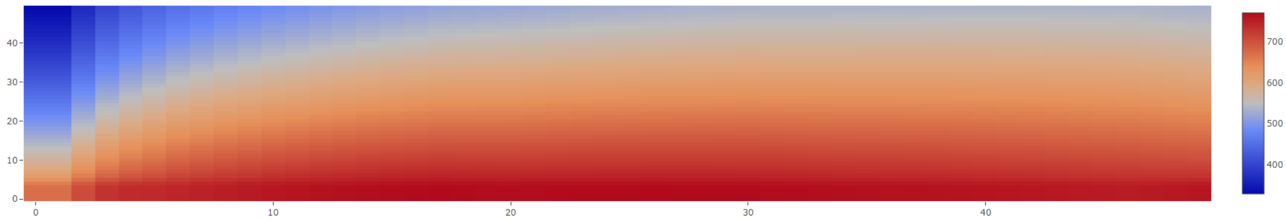


3.2 IPS Map

Since it is very difficult to find the exact IPS of an individual, we define IPS maps. Where each point in the map is a possible IPS of an individual i.e., defines a magical world with rules defined by gaussian with variance as the vector defined by it and the value in it is the probability of the individual having that IPS.

$$IPS = \underset{IPS}{\arg\max}\, Reward(State, Action, Transitions, Rewards, IPS)$$

Figure 3.6: Sample Reward map



$$Probability(IPS) = \frac{Reward(State, Action, Transitions, Rewards, IPS)}{\sum_{IPS} Reward(State, Action, Transitions, Rewards, IPS)} \tag{3.1}$$

Where Reward is computed using markov decision process. (explained in detail in the next chapter)

3.3 Reward Map

Reward map for a puzzle is expected reward for each IPS. Value at each point of a puzzle's reward map is the reward a person having this magical world expects when he tries to solve this puzzle.

$$Reward(IPS) = \sum_{Action} Reward(State, Action, Transitions, Rewards, IPS) * Probability(Action) \tag{3.2}$$

$$Probability(Action) = \frac{Reward(State, Action, Transitions, Rewards, IPS)}{\sum_{Action} Reward(State, Action, Transitions, Rewards, IPS)} \tag{3.3}$$

Thus,

$$Reward(IPS) = \frac{\sum_{Action} [Reward(State, Action, Transitions, Rewards, IPS)]^2}{\sum_{Action} Reward(State, Action, Transitions, Rewards, IPS)} \tag{3.4}$$

Where Reward is computed using markov decision process. (explained in detail in the next chapter)

Figure 3.7: Filter to convert reward map into reinforcement map

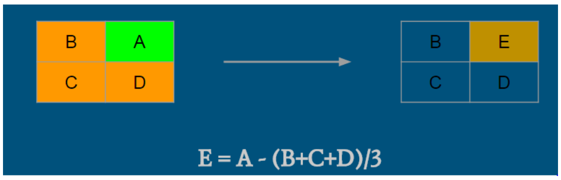
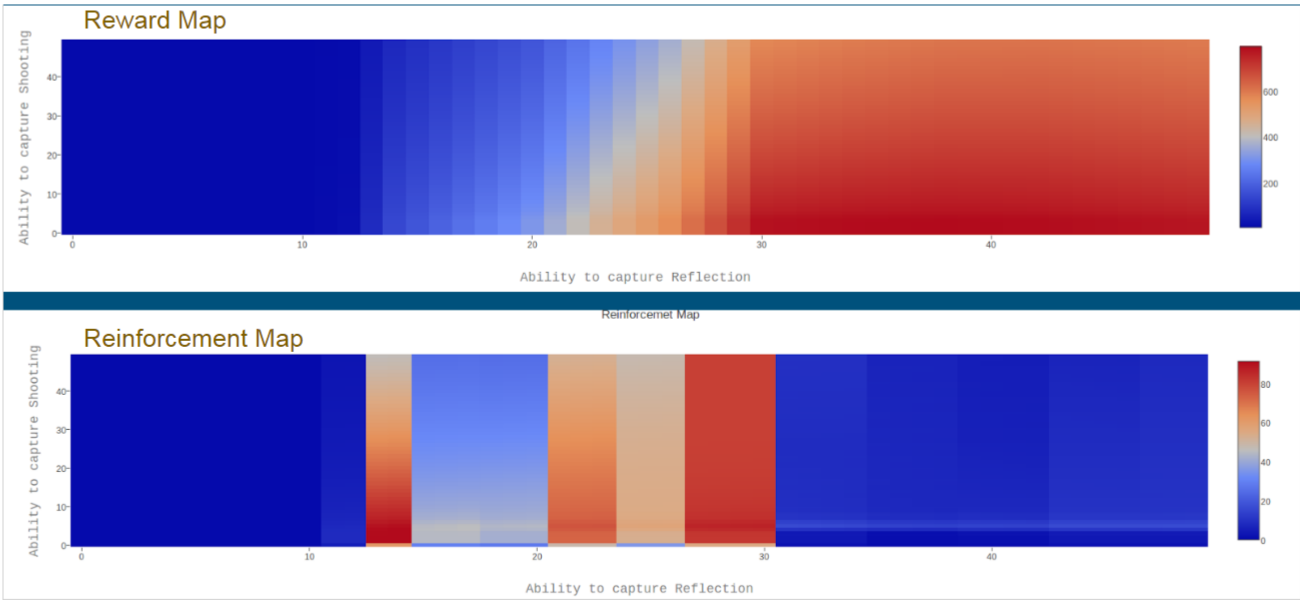


Figure 3.8: Sample Reinforcement map from Reward map



3.4 Reinforcement Map

For each puzzle, the framework computes the Expected Reinforcement in terms of IPS. Traditionally, Reinforcement is the difference between Expected Reward and Actual Reward. But, in case of Autistics, since they can't sense the actual world for it is too magical, the framework estimates reinforcement to be the difference between Expected Reward with current IPS and Reward with slightly better IPS.

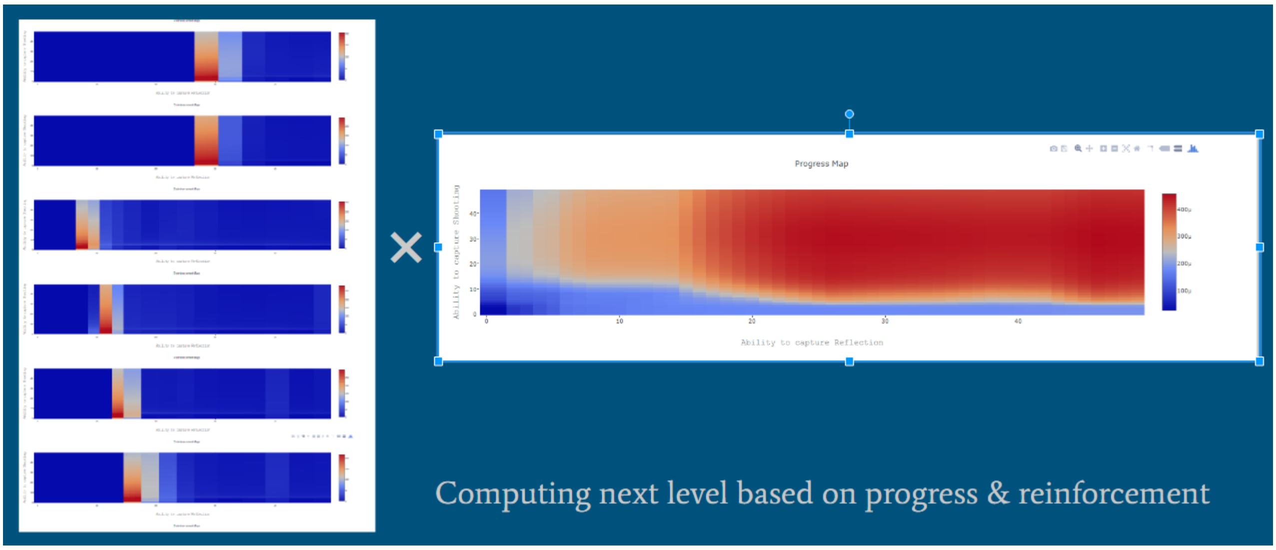
$$Reinforcement(IPS) = Reward(IPS) - Reward(IPS - \delta)$$

Thus reinforcement map can be computed by passing the filter in Figure 3.7 on the reward map. A sample output of reinforcement map obtained by passing this filter is in Figure 3.8.

3.5 Coherency Module

Adding, the probability distribution of IPS to the reinforcement maps, we get expected reinforcement an individual obtains when he/she solves that puzzle as the dot product of

Figure 3.9: Sample computation of next puzzle map using reinforcement maps. Left side are the reinforcement of different puzzles. Right side is the IPS map of an individual.



IPS map of that individual and reinforcement map for that puzzle.

$$Reinforcement = \sum_{IPS} Probability(IPS) * (Reinforcement(IPS)) \tag{3.5}$$

Then the framework will ask the individual to solve the puzzle that gives maximum reinforcement.

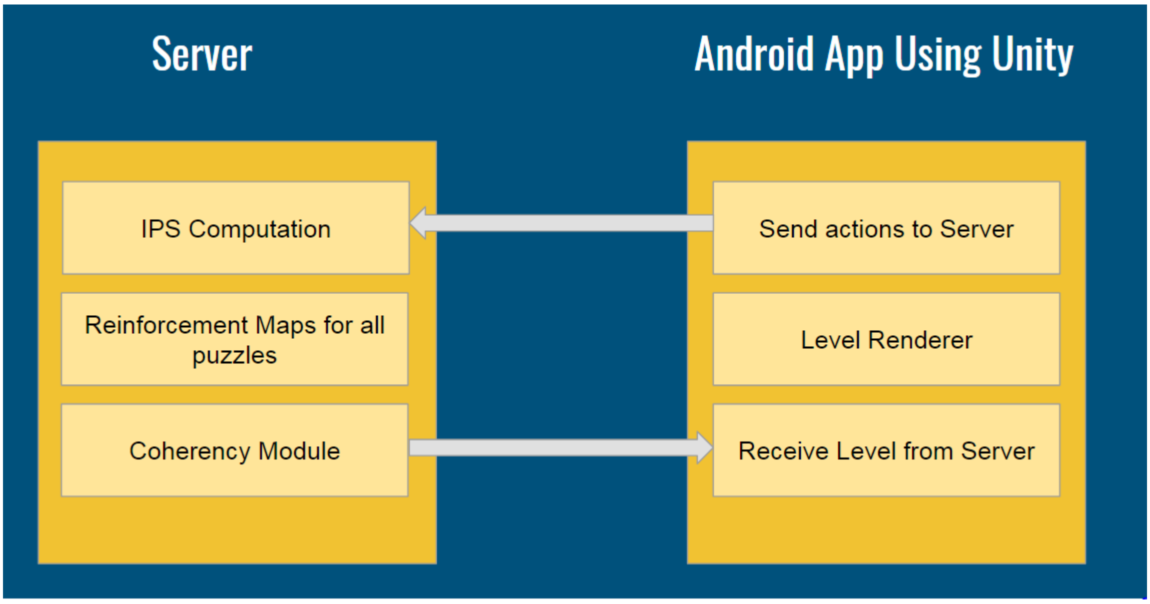
$$NextPuzzle = \arg\max_{Puzzle}[Reinforcement(Puzzle).IPSMap(Individual)]$$

3.6 Framework Design

IPS map computation is computationally intensive task. And so needs a server to be computed fast enough to make the game free of delays. The architecture of the framework is shown in Figure 3.10. There are 6 important tasks in the framework.

- IPS Computation, this task requires several MDP computations and is thus computationally intensive. So, reduce the time this IPS computation takes, several techniques like dynamic programming, caching are used.
- Level Renderer, this modules is designed in unity to render puzzles which it receives from the server and to let user solve these puzzles.
- Send Actions to Server, each action the individual takes are sent to server so that their IPS is updated and the next puzzle given based on reinforcements is more appropriate.

Figure 3.10: Framework Architecture



- Reinforcement Maps for all puzzles are pre-computed and saved in the server to save time. Computation of these maps is very very computationally intensive task and is done on high performance computation (hpc) facility of IITD.
- Coherency module, this module as described in the previous section computes the next puzzle to be given from IPS of the individual and the reinforcement maps.
- Receive Level from server, the puzzle to be given in the server needs to be sent to the device so that it can be rendered whenever needed.

Chapter 4

Spatial Domain

4.1 Game Description

To test the framework in Spatial Domain, we choose Okay Game available on Google Play Store and Apple App Store. Each level in the game has some rectangular & circular objects and the player is given a small ball which can be thrown once. The ball after it hits any object, the object gets vanished and the small ball gets perfectly reflected. The goal at each level is to make the small ball hit all the objects. To make the reinforcement more robust, we replay the recent action. Figure 4.1 is an example puzzle in the game.

4.2 Using MDPs to Solve

The State Space of this game is, ball at a position say x,y moving at an angle θ with respect to X-axis and some of the object vanished. The action space is leaving the ball from a position say x,y at an angle θ or ball getting reflected. There are two notable transitions, which ball is getting reflected after it hits an object and the ball going in the direction expected.

- **State Space** $= <$ **(x, y),** θ**, objects on screen**$>$ i.e., ball at position (x, y) moving at an angle of θ with respect to x-axis and number of objects on screen.

Figure 4.1: Game Description

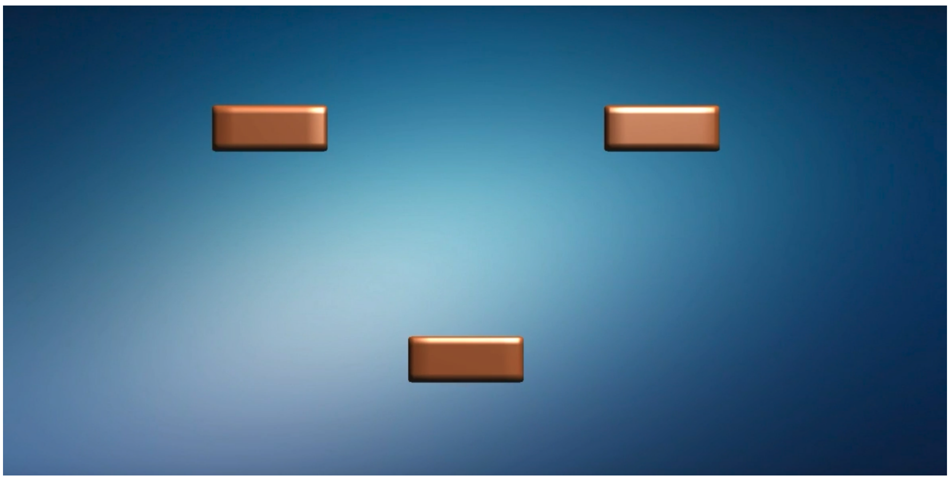
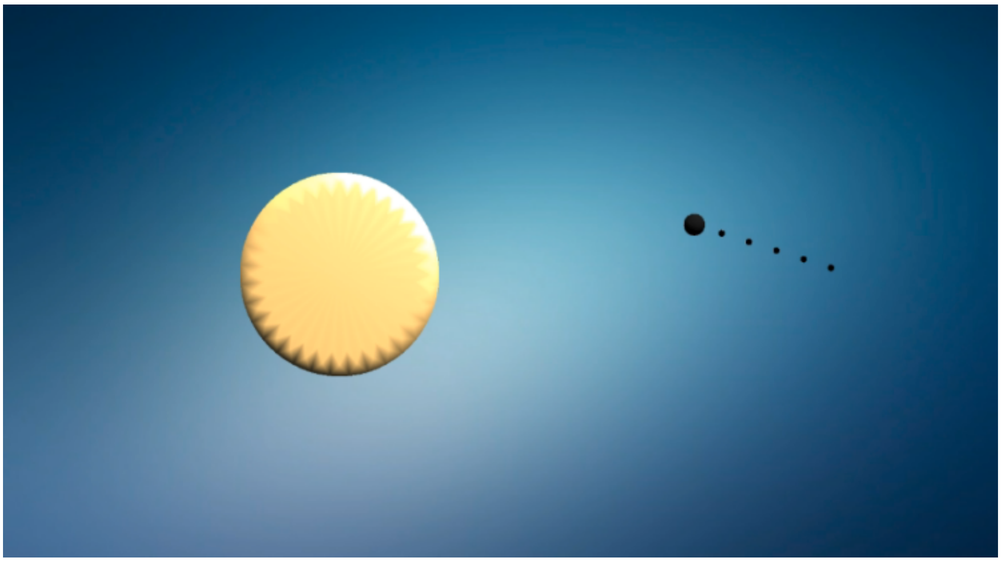


Figure 4.2: A puzzle where the user was unable to shoot the object properly.



- **Action Space** are
  $<$**(x, y)**, $\theta$$>$ i.e., ball thrown at an angle θ from position (x, y) &
  $<$**reflection**$>$ i.e., ball getting reflected after hitting an object.

- **Transition Probabilities** are
  $\mathbf{T(<(x, y), \theta, objects>, <(x, y), \theta>, <(x, y), \theta, objects>) = 1}$

  $\mathbf{T(<(x, y), \theta, objects>, <reflection>, <(x, y), \theta', objects - 1>) = 1}$
  where θ' is the angle of reflection when the ball hits at an angle θ and objects - 1 is the object with which the ball hits getting vanished.

  T(S, A, S') $=$ **0** otherwise

- **Rewards** are
  $<$ **(x, y)**, θ, **objects**$>$ $=$ **0** if (x, y) is outside the screen.
  $<$ **(x, y)**, θ, **{}**$>$ $=$ **1000** i.e., all the objects are vanished from the screen.

After solving the MDP which is the modelling of the game, we get reward for each possible action which can state the correct solutions and wrong solutions based on the reward value. But this is for the ideal case.

4.3 Understanding IPS Maps

Now to model it to individuals with autism, where their priors are skewed we need to update the transition probabilities. There are two predictions involved in this game.

- The prediction of direction of the ball when thrown in a specific direction.
- The prediction of direction of the ball when it is reflected after hitting an object.

Figure 4.3: IPS map for action in figure 4.2

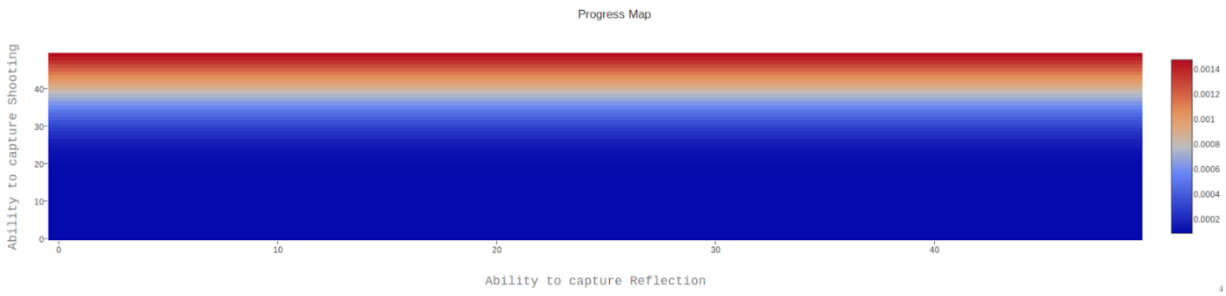


Figure 4.4: A puzzle where the user shot the object perfectly.

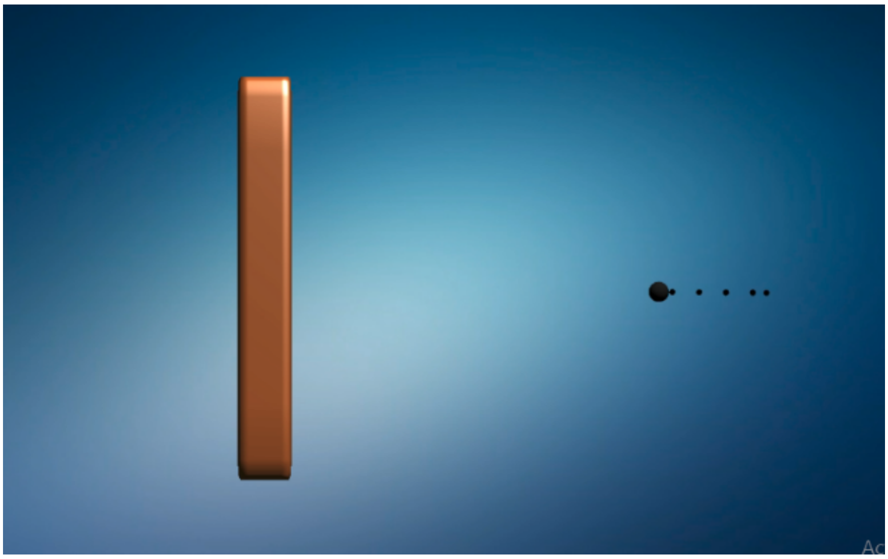


Figure 4.5: IPS map for action in figure 4.4

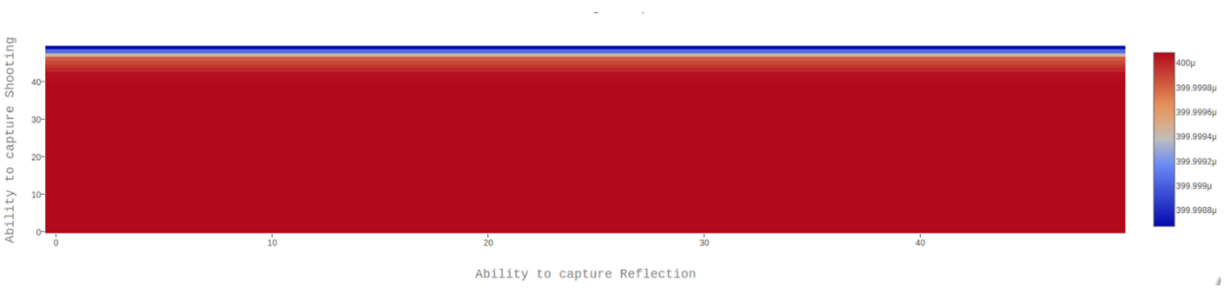


Figure 4.6: A puzzle where the user was unable to shoot the objects properly.

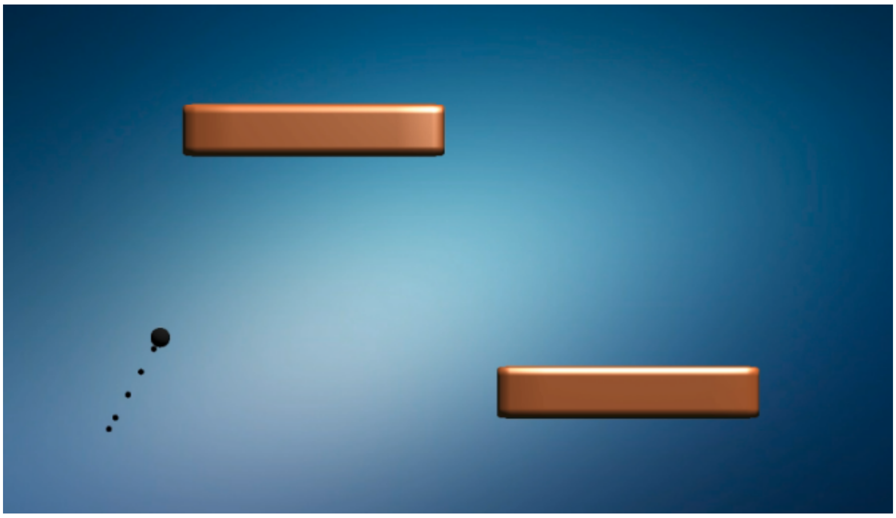
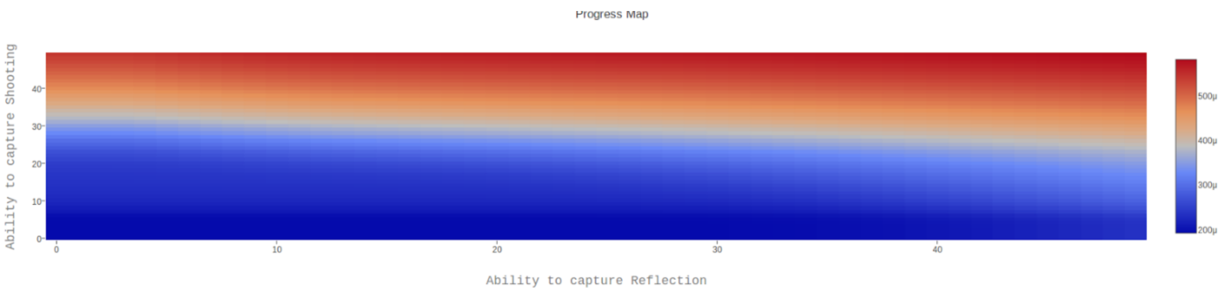


Figure 4.7: IPS map for action in figure 4.6



In figure 4.8, the bold black color represents the actual line of the movement of the ball. The need for two gaussians are included in the figure. While throwing the ball, the individual with autism hopes the ball goes between dotted black to dotted yellow making the wrong decision too have some reward, thus causing the individual to take that decision. Now, consider the action taken in Figure 4.6. The individual with autism might be thinking that the ball will move as show in Figure 4.8. Now, there are 2 ways to explain the action.

- Thin line besides the bold line which is hitting the bottom left object. Here, the individual's predictive skills for shooting can be perfect but predictive abilities for reflection should be very bad for the action to have any reward.
- The thin left most line. Here, the individual's predictive skills for shooting are bad but predictive skills for reflection can be better for the action to give reward.

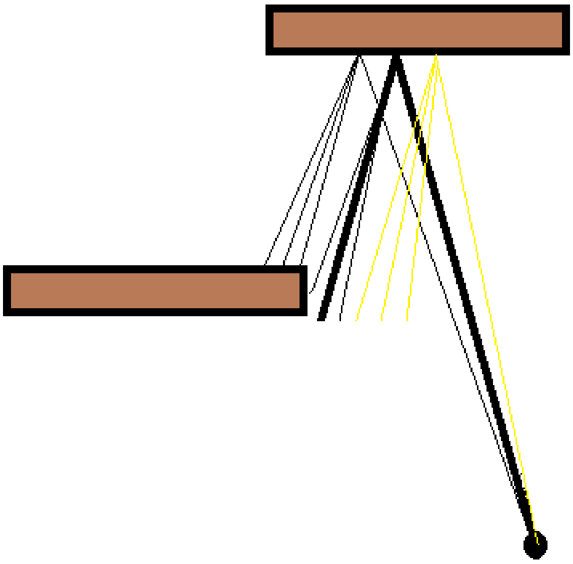
Now, the above two points can be seen in IPS map in Figure 4.7. In the left, the reflective abilities are good which implies shooting abilities should be bad. In the right, the reflective abilities are bad, so shooting abilities are better than in the previous case.

So adding skewed probabilities to the MDP modelling, we get,

- **State Space** $=$ $<$ **(x, y),** θ**, objects on screen**$>$ i.e., ball at position (x, y) moving at an angle of θ with respect to x-axis and number of objects on screen.
- **Action Space** are
  $<$**(x, y),** $\theta$$>$ i.e., ball thrown at an angle θ from position (x, y) &
  $<$**reflection**$>$ i.e., ball getting reflected after hitting an object.
- **Transition Probabilities** are
  T($<$**(x, y),** θ**, objects**$>$**,** $<$**(x, y),** $\theta$$>$**,** $<$**(x, y),** θ **+ i, objects**$>$**)** $=$
  Gaussian(i, σ**1)** where i is the deviation and $\sigma 1$ is a parameter which is specific to the individual.
  T($<$**(x, y),** θ**, objects**$>$**,** $<$**reflection**$>$**,** $<$**(x, y),** θ**' + i, objects - 1**$>$**)** $=$
  Gaussian(i, σ**2)**
  where θ' is the angle of reflection when the ball hits at an angle θ and $\sigma 2$ is again a parameter specific to the individual.
  T(S, A, S') $=$ **0** otherwise
- **Rewards** are
  $<$ **(x, y),** θ**, objects**$>$ $=$ **5** if (x, y) is outside the screen. A reward of 5 is given to regularize the model.
  $<$ **(x, y),** θ**, {}**$>$ $=$ **1000** i.e., all the objects are vanished from the screen.

For each prediction, an Gaussian term is added to the T(S, a, S') term is added to compute the expected reward. Now, there are two hidden individual specific parameters which we need to find. Instead of trying to find the exact $<\sigma 1, \sigma 2>$ we assign a probability to each $<\sigma 1,\sigma 2>$ based on the actions individual has taken. For visualization, we plotted it using heat maps in which x-axis defines the σ for prediction of direction of the ball and

Figure 4.8: Understanding Gaussian Parameters in Spatial Game



y-axis defines the σ for prediction of reflection of the ball. Less is sigma, better is the ability. Each set of sigmas i.e., $<\sigma 1, \sigma 2>$, state the magicalness of the magical world individual with autism experiences. Thus The following will illustrate the same.

- Figure 4.3 is the IPS map for the action in figure 4.2. In this case, the individual is unable to shoot the object properly. Thus the redness which indicates the probability distribution is in the bottom. Since, there is no reflective predictions involved in this problem, the weights from left to right are all equal.
- Figure 4.5 is the IPS map for the action in figure 4.4. In this case, the individual is unable to shoot the object properly. Thus the probability distribution is in the top. Since, there is no reflective predictions involved in this problem, the weights from left to right are all equal.
- Figure 4.7 is the IPS map for the action in figure 4.6. In this case, the individual is unable to hit the two objects. This can be attributed to both inability to predict the ball direction or inability to predict the reflection. Thus the probability weight is in the top.

After each action the individual takes, the IPS map is recomputed by multiplying with discounted previous IPS map. i.e.,
New IPS Map = (IPS map based on action) * (Previous IPS map)$^{0.8}$

Figure 4.9: A sample puzzle whose reward map is in Figure 4.10.

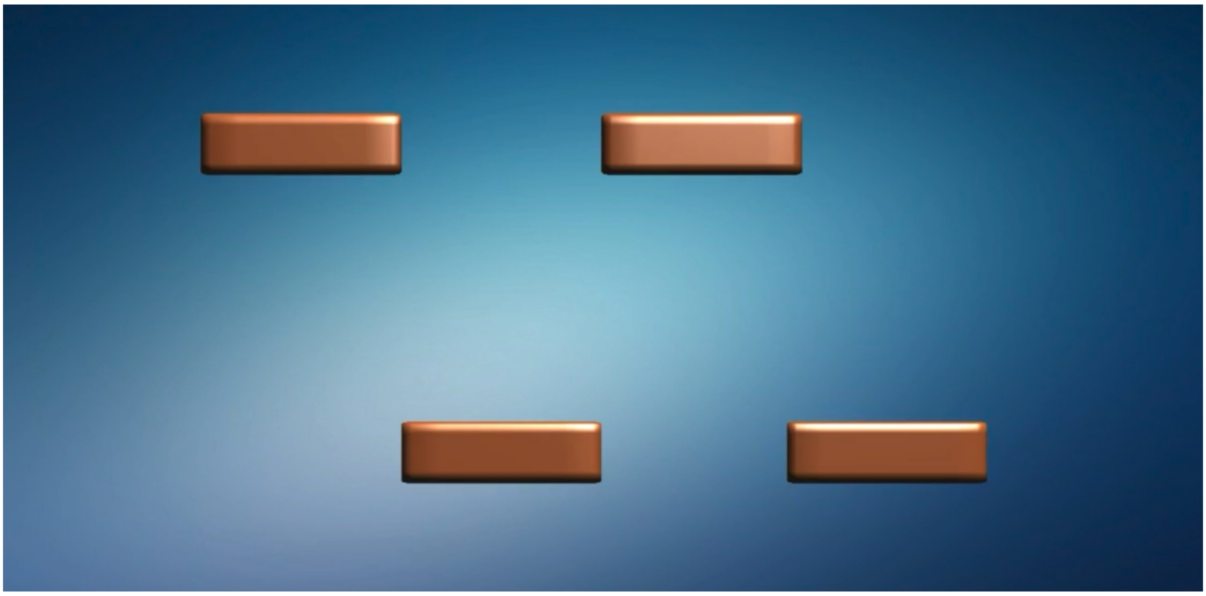


Figure 4.10: Reward map for puzzle in figure 4.9

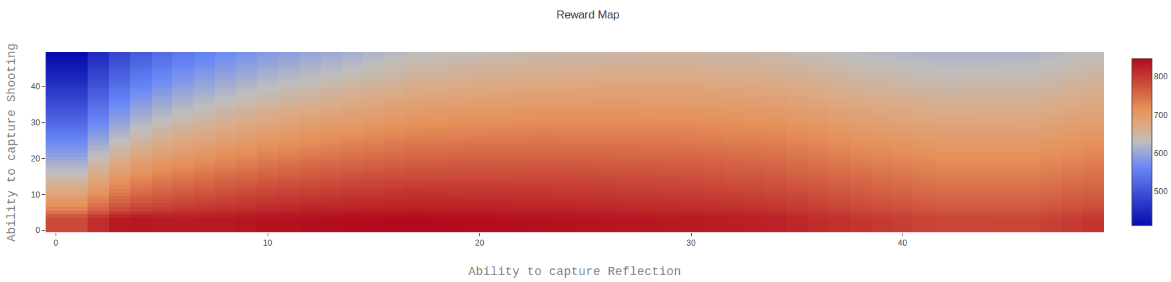


Figure 4.11: Filter to convert reward map to reinforcement map.

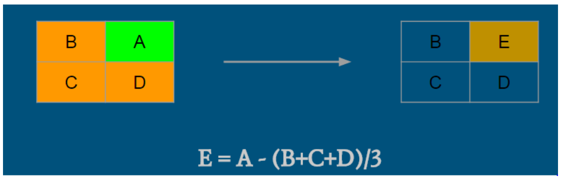


Figure 4.12: Intuition behind the filter in figure 4.11

4.4 Understanding Reward Maps & Reinforcement Maps

As stated in Chapter 3, each point of reward map states the reward an individual in the magical world defines by sigmas i.e., co-ordinates in the reward map. The following will illustrate some examples of the same.

- Figure 4.10 is the reward map for the puzzle in figure 4.9.

Reward maps as such doesn't have any significance in the framework. But using the filter described in Figure 4.12 and technique described in Section 3.4, we can compute the reinforcement maps using reward maps which gives an estimate of the reinforcement an individual with autism experiences after he/she tries to solve the puzzle.

The intuition behind the filter can be explained using Figure 4.12. The area in which an individual with autism expects is colored with green. A slightly better expectation is colored with yellow. Now, we would like to push the predictive abilities of the individual to yellow from green. The value under A in the filter stated in figure 4.11 states the reward if the ball moves in green and yellow region. The average value under B, C & D in the filter states the reward if the ball moves in yellow region. Thus the value computed by the filter is proportional to the probability with ball misses the goal and moves in the yellow colored region. Thus it causes a **oh no!** movement thus reinforcing the understanding of the laws of the world.

4.5 Adding Framework to Game

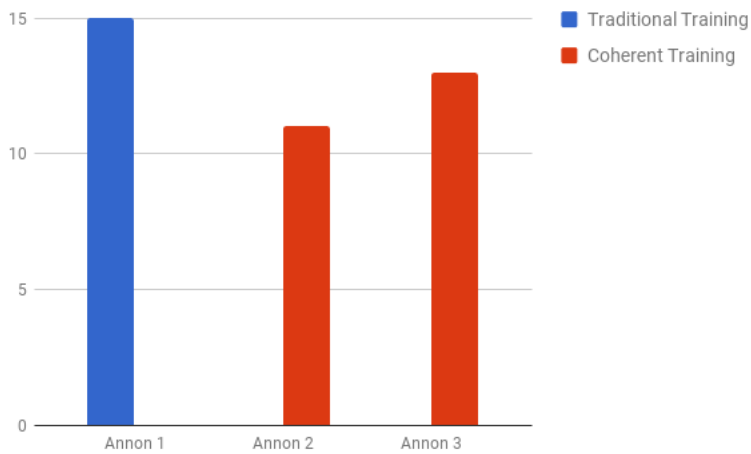
As stated in section 3.6, the android game app communicates with the server using API calls. The following steps are used for creating the game with this coherent framework.

- 10 puzzles, all with different arrangement of objects, are created for the game.
- Small changes are made to each of these 10 puzzles to make them cover all ranges of IPSs and 50 puzzles are made.
- Reinforcement map for each of these 50 puzzles are computed in hpc facility at IITD.
- A server with login credentials and user-defined content is developed using web2py framework and is hosted at Amazon AWS server. This server communicates with the app using API calls.
- As stated in section 3.6, each time the app asks the server for puzzles and server return the puzzle. There are 2 ways of training namely
  - **Coherent**
    In this training, puzzle that gives the maximum reinforcement and that the user has not solved yet out of those 50 puzzles is given to the user.
  - **Traditional**
    In this training, an ordering between the puzzles is created before hand manually based on intuition. A puzzle is repeated until the user solves the puzzle. After the user solves, the next puzzle in the ordering is given.
- The action the user takes gets back to the server using API call for IPS map computation. The IPS map computation is computationally intensive. So, a lot of computation required is stored in database i.e., memorized for faster computation. Time required for computation is reduced from 64s to 2s using this technique. In these 2s, a replay of the action the user has taken is played.

4.6 Results

To evaluate the framework using this game, we gave 25 puzzles in coherent or traditional way. After that to test the individuals learning, we asked him to solve 3 puzzles. The number of retries user has taken to solve these puzzles is counted. The better the learning, the lesser the retries will be. As a pilot testing, we got 3 patients. The results are shown in graph 4.13. From the pilot testing though we can't comment for the test subjects are too less, we can say there is around 20% improvement in the training. Rigirous testing will be done from October 1st, 2017 after the ethic clearance from Action for Autism Centre, New Delhi.

Figure 4.13: Bar Graph showing the results



Chapter 5

Temporal Domain

5.1 Game Description

To test the framework in temporal domain, we choose Stick Hero game available on Google Play store and Apple app store. The main goal of the game is make the avatar move between blocks using stick. User needs to estimate the length of the stick and keep on taping the screen until the stick's length is as estimated. In figure 5.1, the screen shot is while the user is tapping the screen and the stick length is increasing. After the user feels that the stick length is equal to the distance between the black blocks he can stop tapping the screen after which, the stick falls. If the other end of the stick is not on the next block, the avatar falls and game over screen appears. Otherwise, a score of 2 if the stick lands on the red mark or 1 otherwise will be awarded and a new block appears.

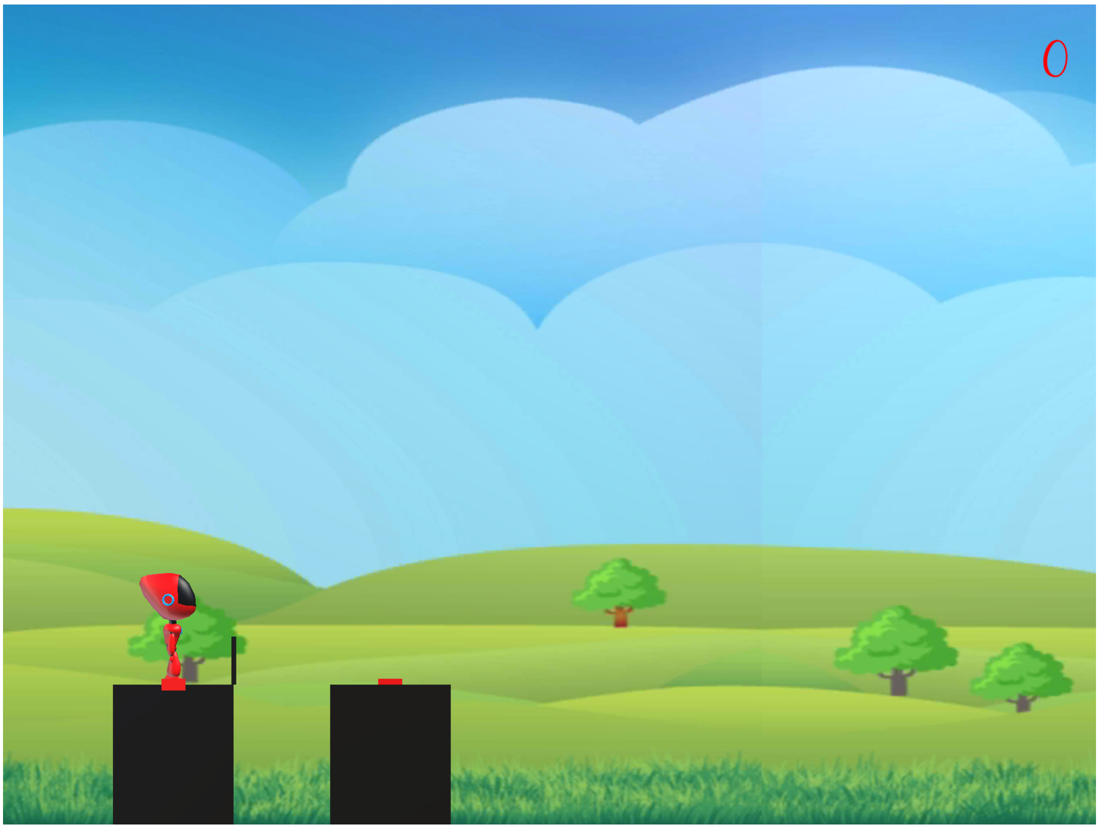
5.2 Using MDPs to Solve

The State Space of this game is, the stick having a length of x, the distance between the blocks and the width of the block. The action space is tap for t units of time. Ideally, there is only one transition i.e., stick length increasing.

- **State Space** = **<x, d, w>** i.e., stick having a length of x, distance between blocks is d and the width of the target block is w.
- **Action Space** = **<t>** i.e., the user tapping the screen for t seconds.
- **Transition Probabilities** are
  - **T(<0, d, w>, <t>, <t*r, d, w>) = 1** where r is the rate of increase of the stick.
  - **T(S, A, S')** = **0** otherwise
- **Rewards** are
  - **R(<x, d, w>) = 1000** if d-w <= x <= d+w
  - **R(S) = 0** Otherwise

After solving this MDP which is the modelling of the game, we get reward for each possible tap duration which can state if the tap duration is correct or wrong based on the value of the reward. But this is for ideal case.

Figure 5.1: A sample puzzle



5.3 Understanding IPS Maps

Now to model it to individuals with autism, where the priors are skewed we need to update the transition probabilities. We have considered two predictions which might be involved in the game.

- Estimating the rate of increase of the stick.
- Estimating the tap time which also comes as distance.

Adding these skewed probabilities to the MDP modelling, we get,

- **State Space** = **<x, d, w>** i.e., stick having a length of x, distance between blocks is d and the width of the target block is w.
- **Action Space** = **<t>** i.e., the user tapping the screen for t seconds.
- **Transition Probabilities** are
  - **T(<0, d, w>, <t>, <(t+j)*(r+i), d, w>) = Gaussian(σ1,i)*Gaussian(σ2,j)** where r is the rate of increase of the stick, i is the deviation from estimation of the rate of increase of the stick and j is the deviation from tapping the screen on time.
  - **T(S, A, S') = 0** otherwise

Figure 5.2: A sample action showing wrong estimation of speed

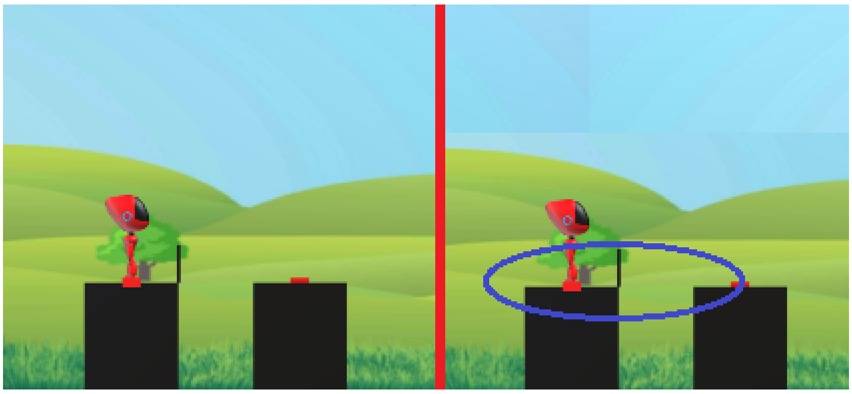


Figure 5.3: A sample action showing wrong estimation of distance

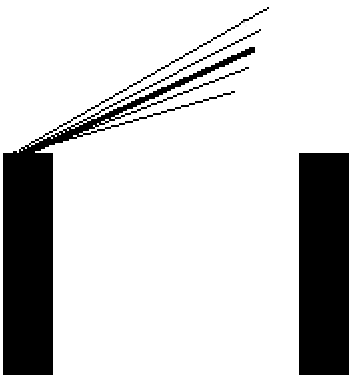
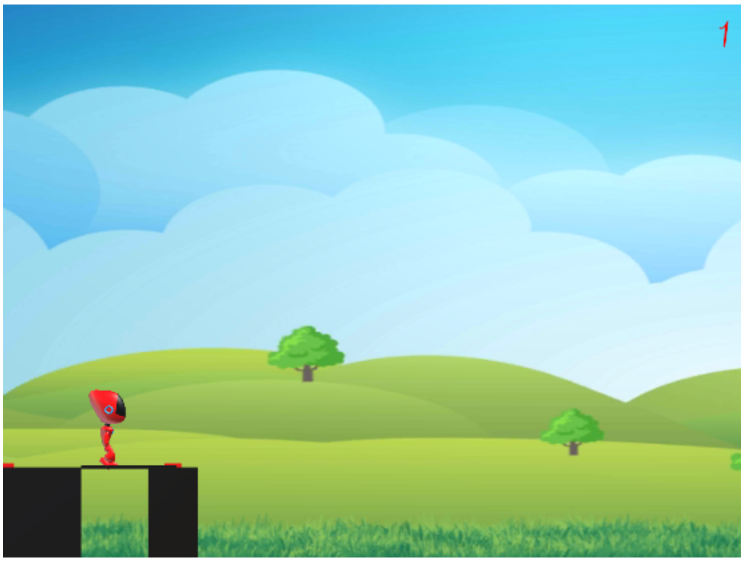


Figure 5.4: A sample perfect action



- **Rewards** are
  - $\mathbf{R(<x, d, w>) = 1000}$ if $d\text{-}w <= x <= d+w$
  - $\mathbf{R(S) = 0}$ Otherwise

For each prediction, an Gaussian term is added to the T(S, a, S') term is added to compute the expected reward. Now, there are two hidden individual specific parameters which we need to find. Instead of trying to find the exact $<\sigma 1, \sigma 2>$ we assign a probability to each $<\sigma 1,\sigma 2>$ based on the actions the individual has taken. For visualization, we plotted these maps using heat maps in which x-axis defines the σ for prediction of speed and y-axis defines the σ for prediction of time to stop tapping the screen. Less is sigma, better is the ability. Each set of sigmas i.e., $<\sigma 1, \sigma 2>$, state the magicalness of the magical world individual with autism experiences. Thus The following will illustrate the same.

- Figure 5.5 is the IPS map for the action in figure 5.4. In this case, the individual took an perfect action. As we can see in Figure 5.5, the maxima for red area is in left bottom corner which should be the case.
- Figure 5.7 is the IPS map for the action in figure 5.6. In this case, the individual took a wrong action. But this can be explained in terms of bad estimation of distance or speed.

Using the same techniques developed in Chapter 3 and section 4.4, reward maps and reinforcement maps are computed for each puzzle.

5.4 Adding Framework to Game

As stated in section 3.6 and 4.5, the android game app communicates with the server using API calls. The following steps are used for creating the game with this coherent framework.

Figure 5.5: IPS Map of action in Figure 5.4

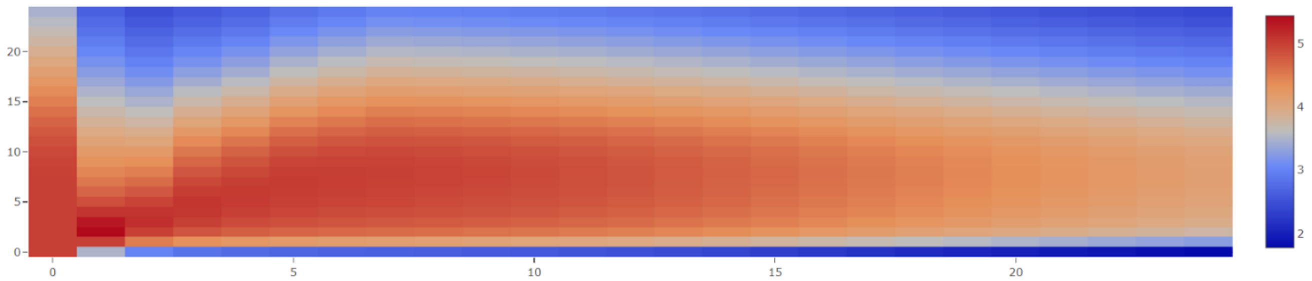


Figure 5.6: A sample wrong action

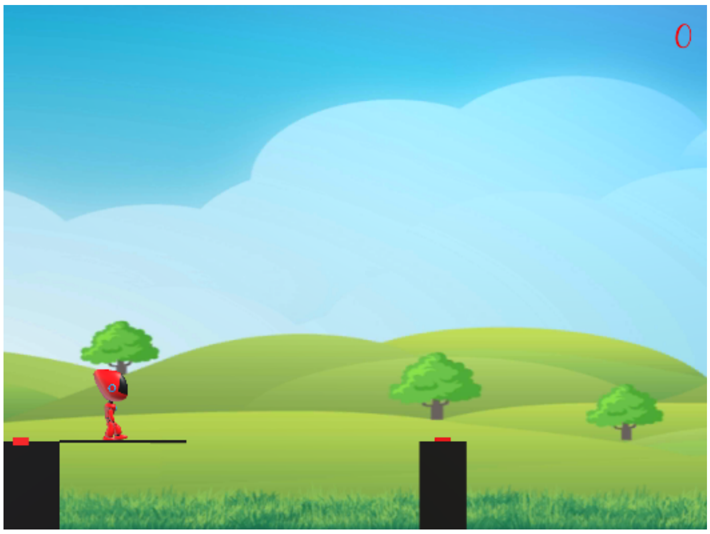
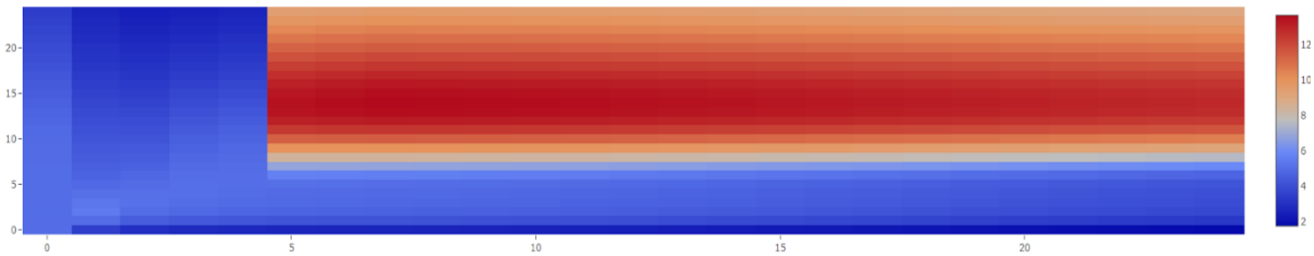


Figure 5.7: IPS Map of action in Figure 5.6



- The span of all possible puzzles, i.e., set of all possible <distance, width> of blocks where distance is an integer in range of 1 to 25, width is an odd integer less than 23 and sum of distance and width is less than 25. A total of 144 puzzles are created based on these constraints.
- Reinforcement map for each of the 144 puzzles are computed in hpc facility at IITD.
- A server with login credentials and user-defined content is developed using web2py framework and is hosted at Amazon AWS server. This server communicates with the app using API calls.
- A stated in section 3.6, each time the app asks the server for puzzles and server returns the puzzle. There are 2 training ways namely,
  - **Coherent**
    In this training, puzzle that gives the maximum reinforcement out of the 144 puzzles is given to the user. To evaluate the user's learning, for every 10 games, 2 games are given which use traditional training mechanism.
  - **Traditional**
    In this training, a randomly puzzle is selected from these 144 puzzles and is given to the user.
- The action the user takes gets back to the server using API call for IPS computation. In case of this game, IPS map computation is not that computationally intensive.

5.5 Results

To evaluate the framework in temporal domain using stick hero game, individuals are divided into 2 parts, one for playing the puzzles generated by the framework and the other who plays the puzzles generated in traditional way. A game over screen appears when the user takes wrong action, and a score of +1 is awarded for correct action and a score of +2 for perfect action. The user is asked to play 25 games on a daily basis for a week. The score is counted and is graphically plotted on server as shown in Figure 5.8. Average of the score is also counted (Figure 5.9) and is available on the server. As a pilot testing, we got 3 patients. The results are shown in Figure 5.10. From the pilot testing though we can't comment for the test subjects are too less, we can say there is around 25% improvement in the training. Rigorous testing will be done from October 1st, 2017 after the ethics clearance from Action for Autism Centre, New Delhi.

Figure 5.8: Graph plotting the score of an individual playing the game

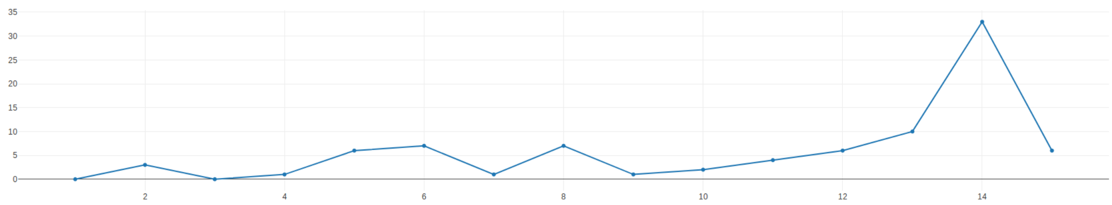


Figure 5.9: Graph plotting the average daywise score of an individual playing the game

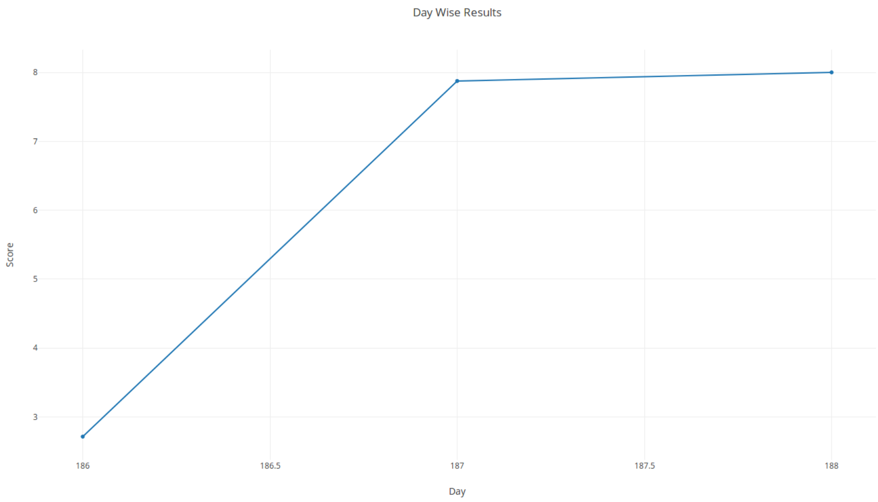


Figure 5.10: Graph comparing the scores of individuals in Coherent Framework and Traditional

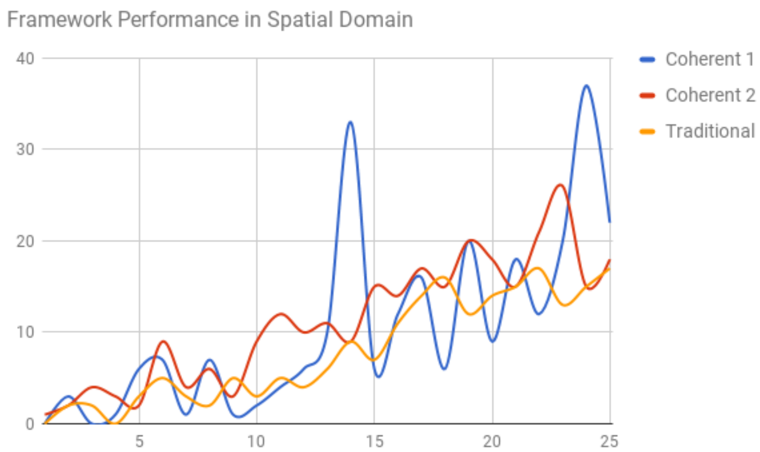
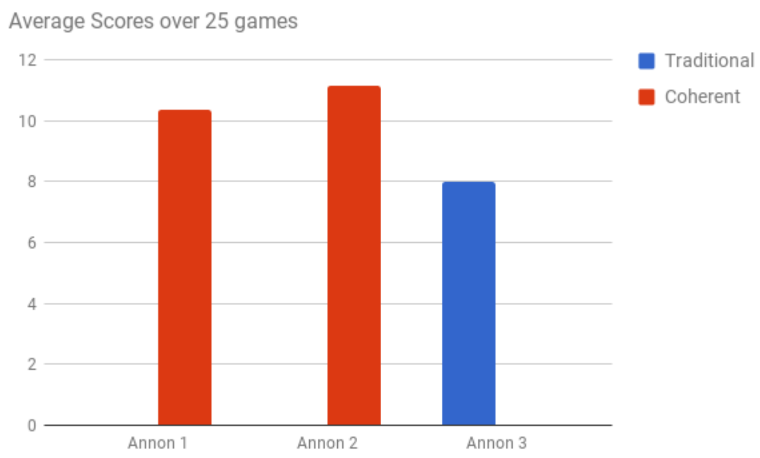


Figure 5.11: Graph comparing average scores of individuals in Coherent Framework and Traditional



Chapter 6

Game Play

As seen earlier, there are two games in two different domains.

- Spatial Domain - Okay Game
- Temporal Domain - Stick Hero Game

For the purpose of evaluation, the test subjects are divided equally in two parts namely

- One who plays puzzles generated in coherent way by framework.
- Other who plays puzzles generated in traditional way.

6.1 Spatial Domain

To evaluate in spatial domain, we designed a game called "Okay".

- The subject needs to be assigned an a training mechanism i.e., traditional or coherent.
- Next, an account needs to be created on the name of the subject with the assigned training mechanism on the address Sign Up Page for Spatial Domain. (http://www.wikiwithwisdom.org/AssistingAutistic/CreateUser/index). A screenshot is also attached in Figure 6.1.
- Next, using the credentials, the user can login on "Okay" game available on the android tablet. A screenshot is also attached in Figure 6.2.
- After the user logged in, puzzle appears. A sample puzzles is attached in Figure 6.3
- The major goal of the game is to hit all objects on the screen using the ball given. The ball can be thrown only once. The subject can be stated how the ball can be thrown. A screenshot is attached in Figure 6.4.
- Another screenshot of the puzzle is attached in Figure 6.5 for reference.
- After the user throws the ball, a replay of the action is played to make the reinforcement more robust. A screenshot is attached in Figure 6.6
- Subject will trained on 25 puzzles. Then, the abilities of the subject are tested using 3 puzzles. The number of retries the subject takes to complete these levels is counted and is a score to measure the subject's abilities.
- The 3 levels in which subject's abilities are tested are in Figure 6.7, Figure 6.8 and Figure 6.9.
- All results are recorded on the server.

Figure 6.1: Sign Up page for Spatial Domain

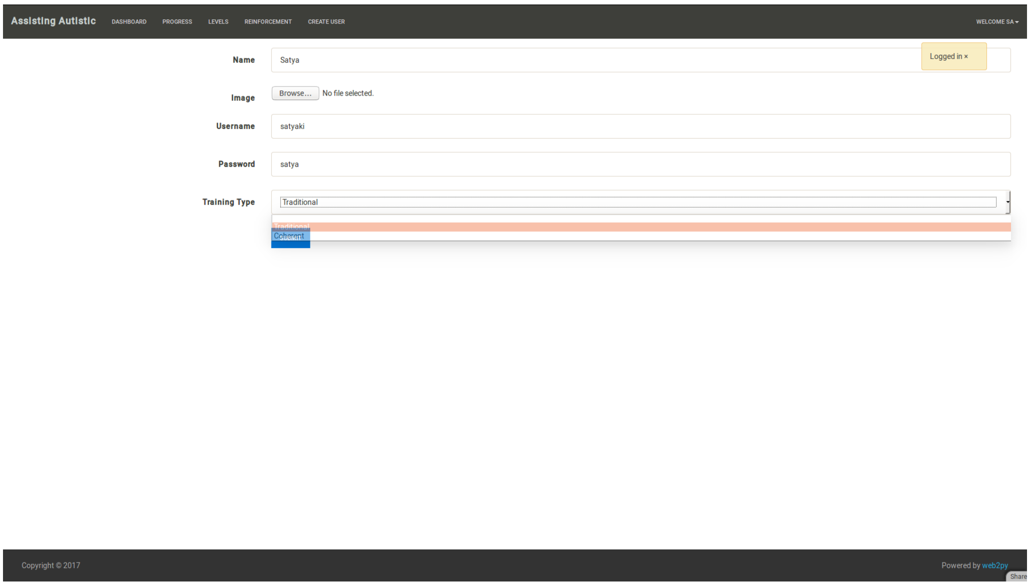


Figure 6.2: Login Page on Android Tab

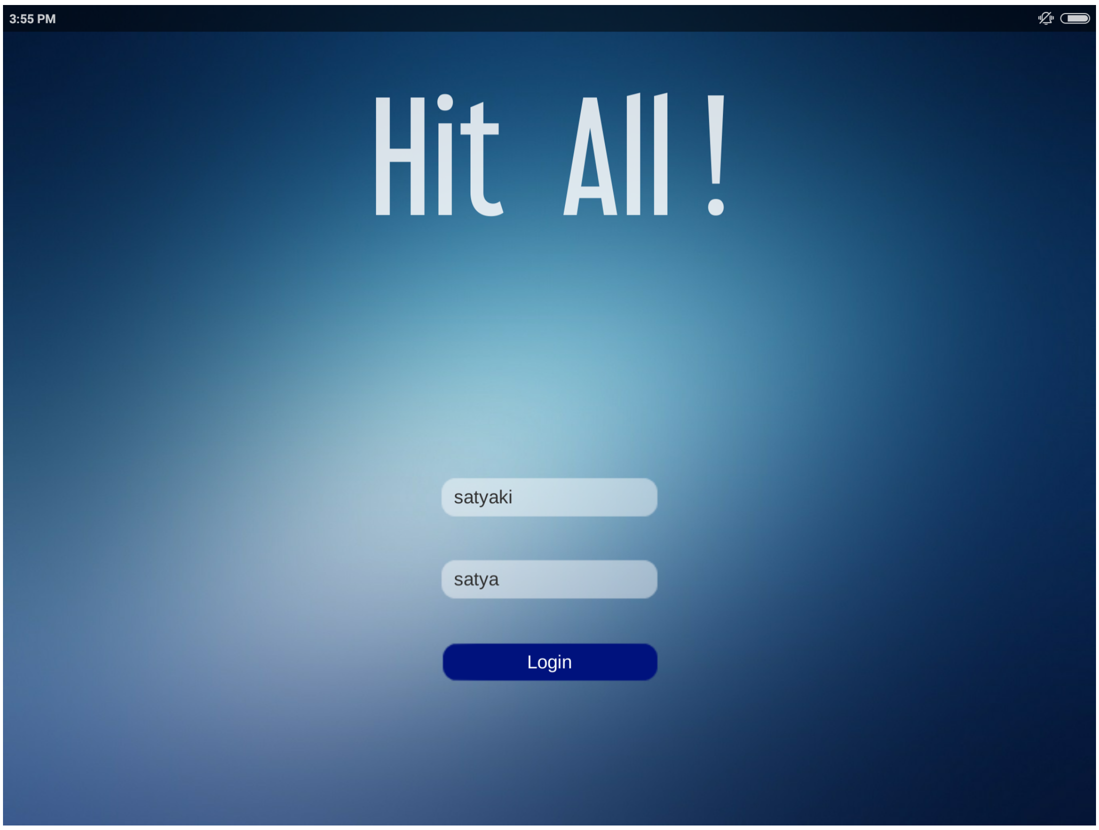


Figure 6.3: A sample puzzle

Figure 6.4: Sample Action

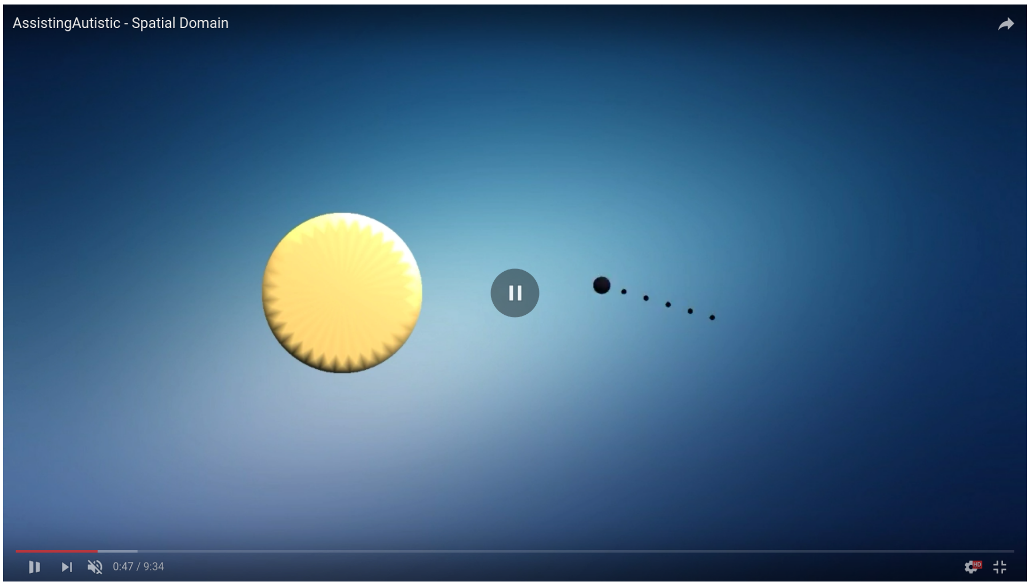


Figure 6.5: Another sample puzzle

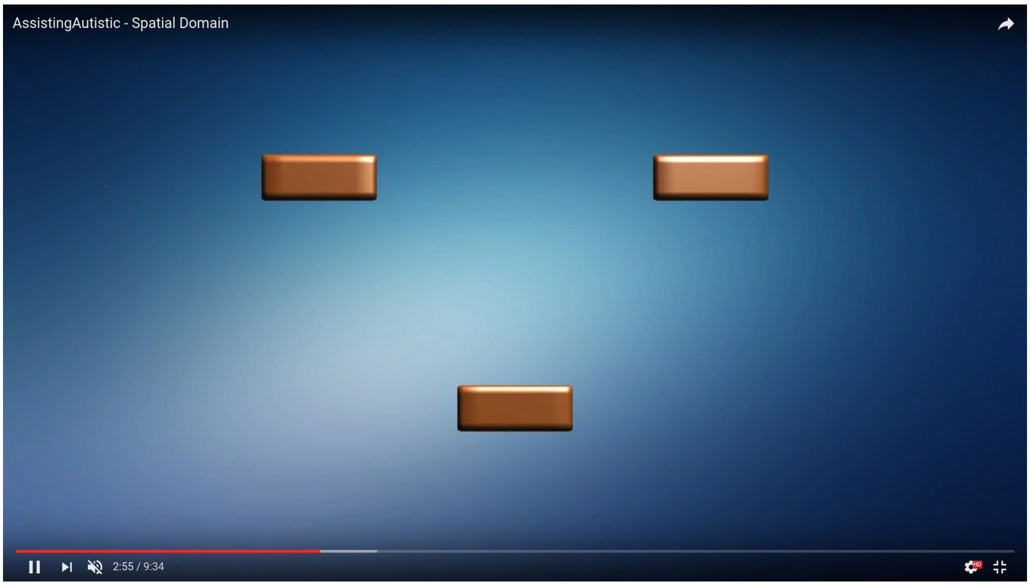


Figure 6.6: Replay of an action

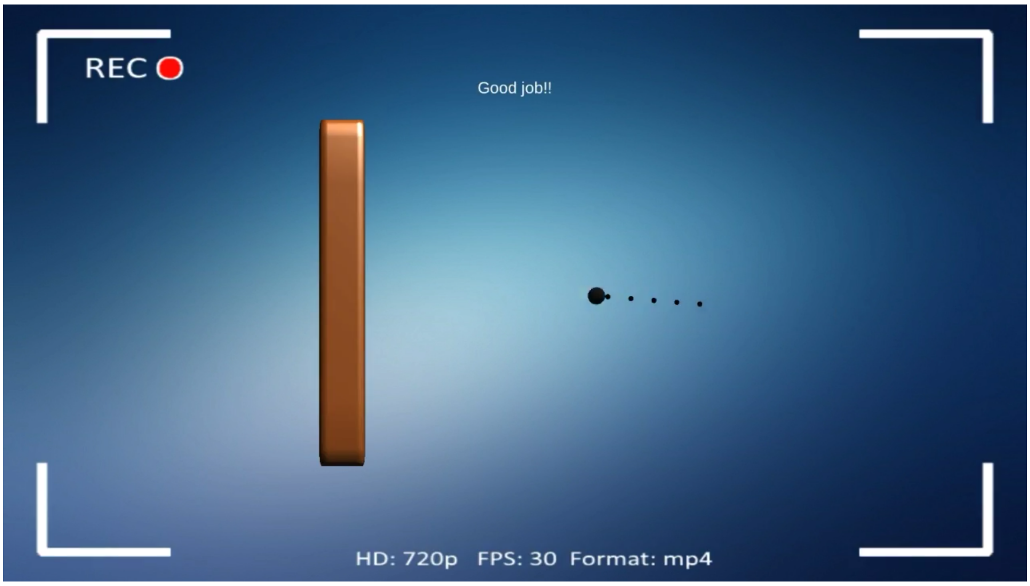


Figure 6.7: Test Puzzle 1

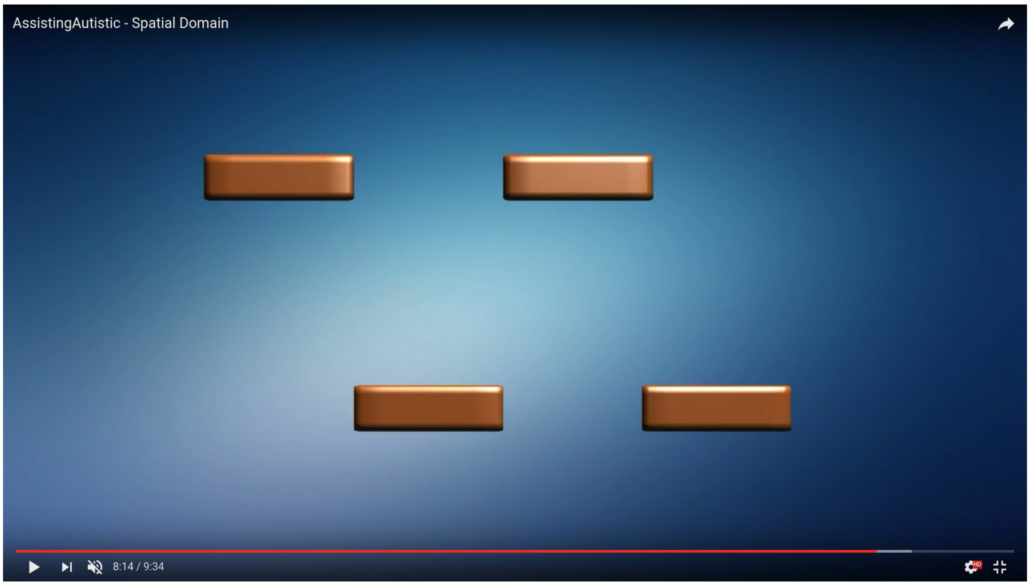


Figure 6.8: Test Puzzle 2

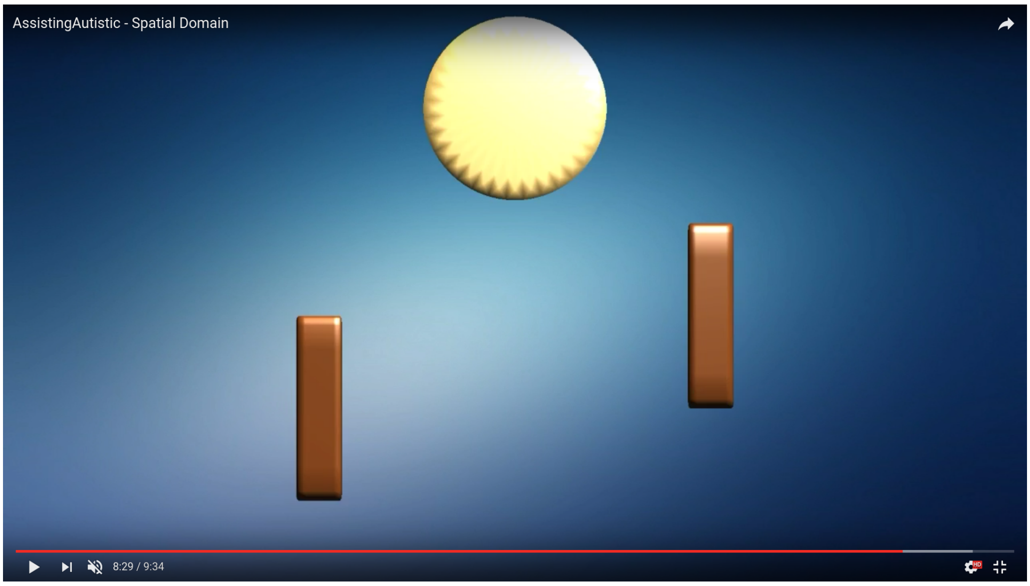
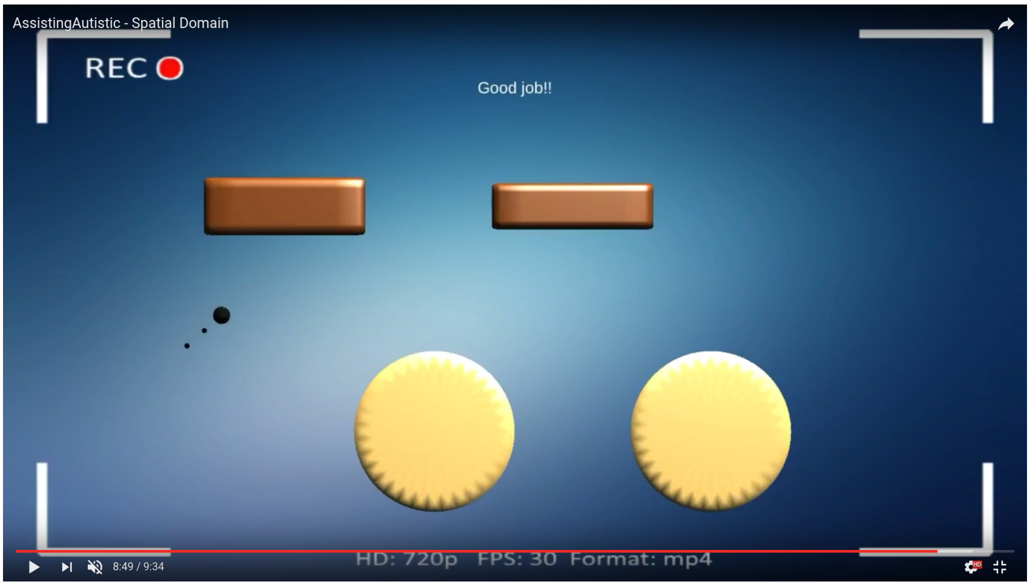


Figure 6.9: Test Puzzle 3



6.2 Temporal Domain

To evaluate in temporal domain, we designed a game called “Stick Hero”.

- The subject needs to be assigned an a training mechanism i.e., traditional or coherent.
- Next, an account needs to be created on the name of the subject with the assigned training mechanism on the address Sign Up Page for Spatial Domain. (http://www.wikiwithwisdom.org/StickHero/CreateUser/index). A screenshot is also attached in Figure 6.10.
- Next, using the credentials, the user can login on “Stick Hero” game available on the android tablet. A screenshot is also attached in Figure 6.11.
- After the user logged in, puzzle appears. A sample puzzles is attached in Figure 6.3
- The major goal of the game is to take the avatar with letting it fall. As the user taps on the screen, a stick appears and its size keeps on increasing. After the user stops the tap on the screen, the stick stops increasing and falls. To make sure that the avatar doesn’t fall, the user has to make sure that the stick fall on the black wall. If the stick falls on the red area i.e., perfect prediction, there is an extra reward. And if the size if lesser or greater i.e., the stick’s end doesn’t fall on the black wall, then the avatar falls, leading to a game over screen.

Figure 6.10: Create User Portal for Stick Hero Game (Temporal Domain)

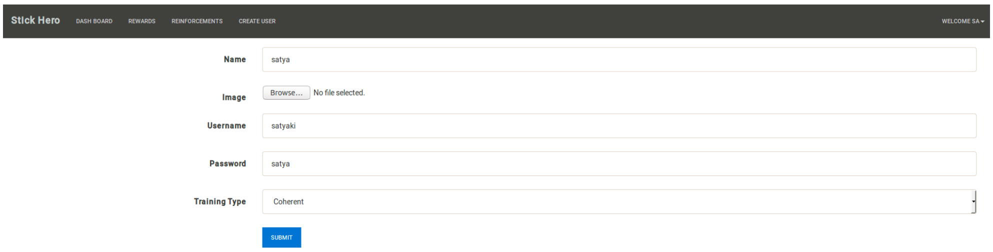


Figure 6.11: Login Page on App

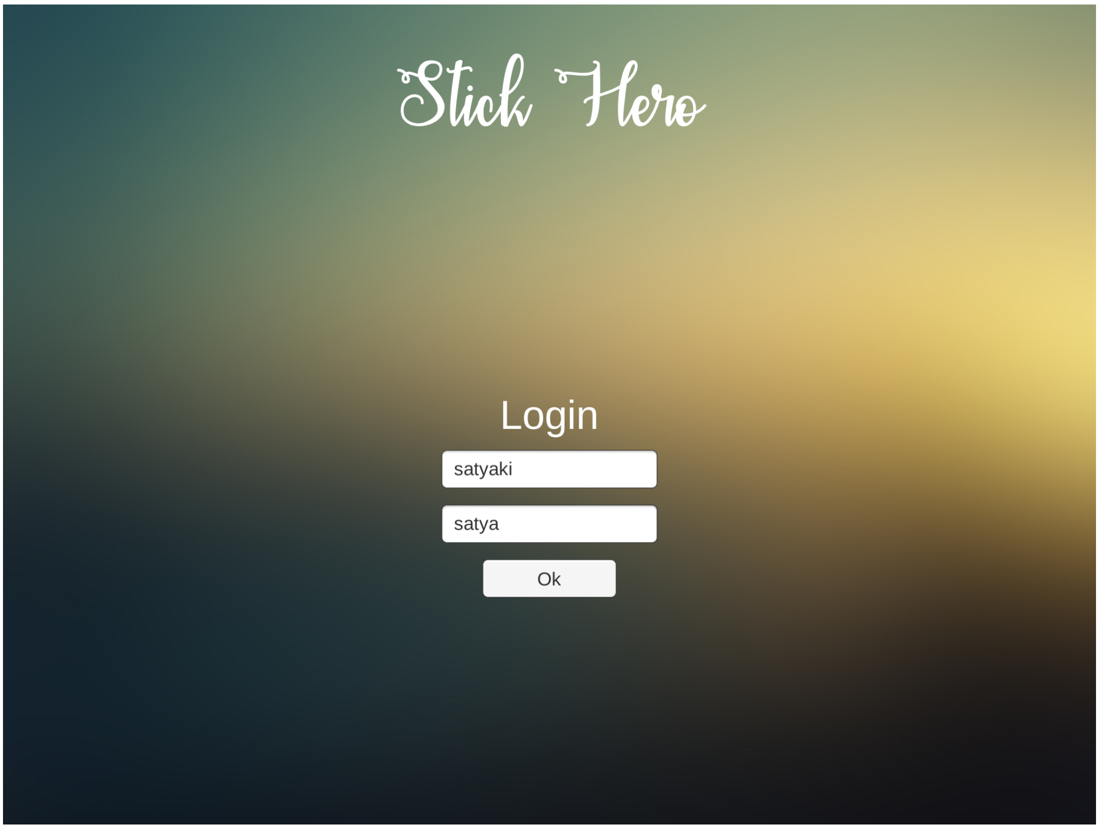


Figure 6.12: Sample Puzzle

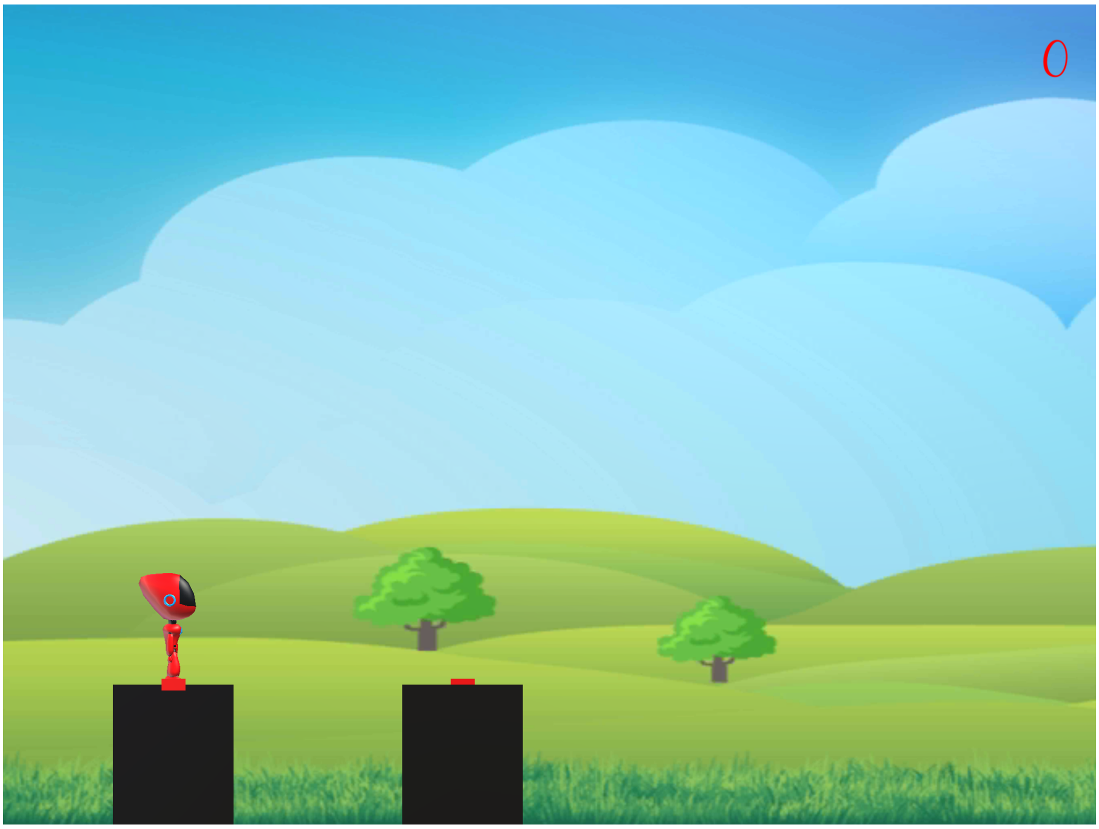


Figure 6.13: Screen Shot of Stick Increasing Phase

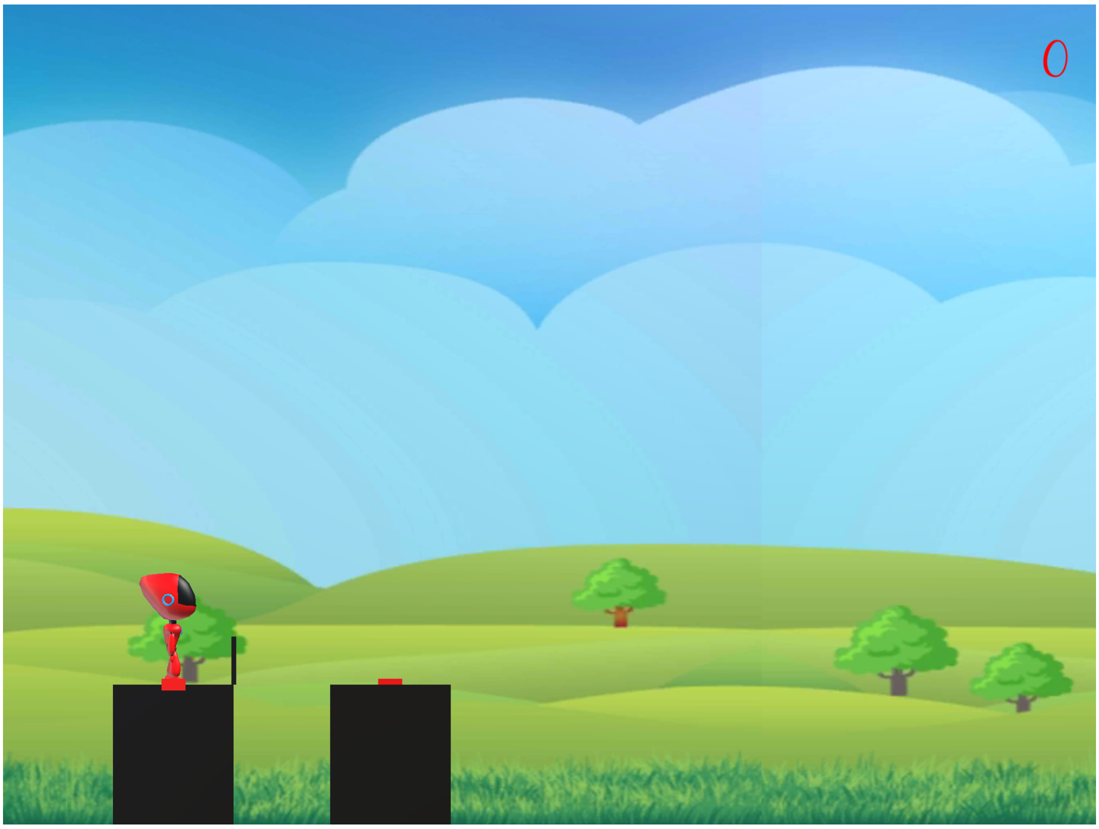


Figure 6.14: Screen Shot of Stick Falling Phase

Figure 6.15: Screen Shot of Robot Walking after perfect action

Figure 6.16: Screen shot of robot about to fall after wrong action - 1

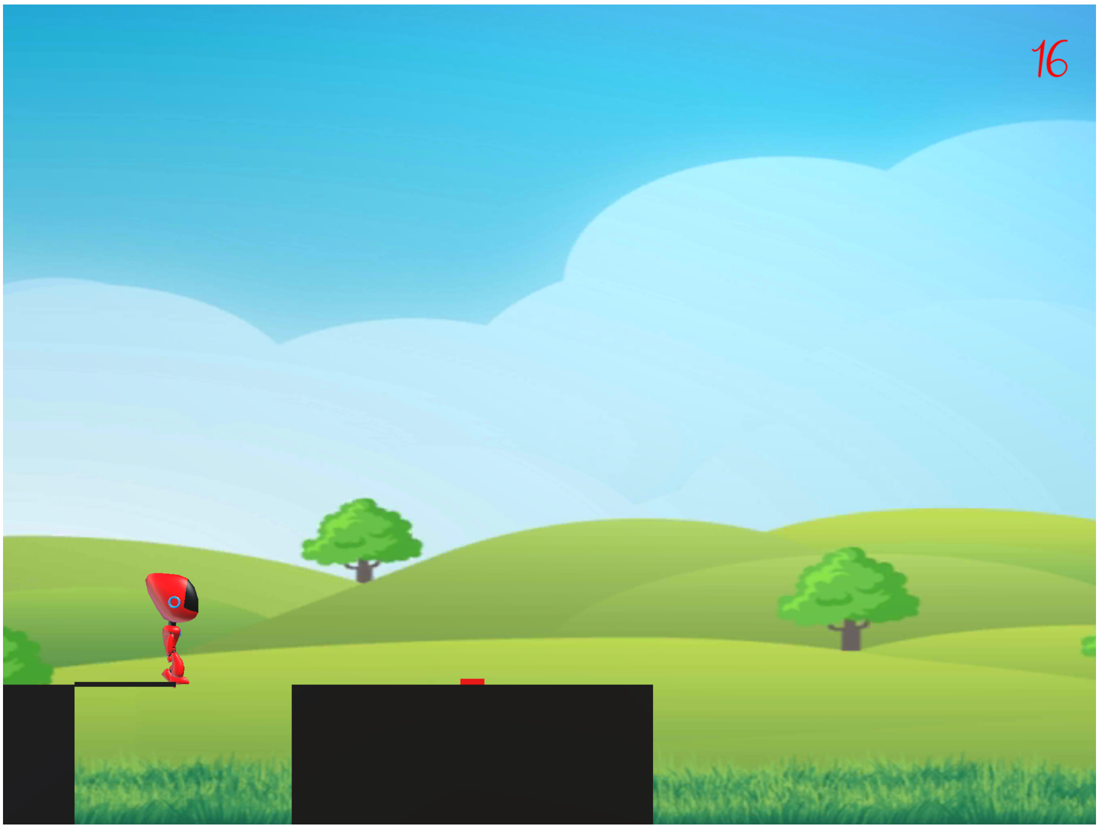


Figure 6.17: Screen shot of robot about to fall after wrong action - 2

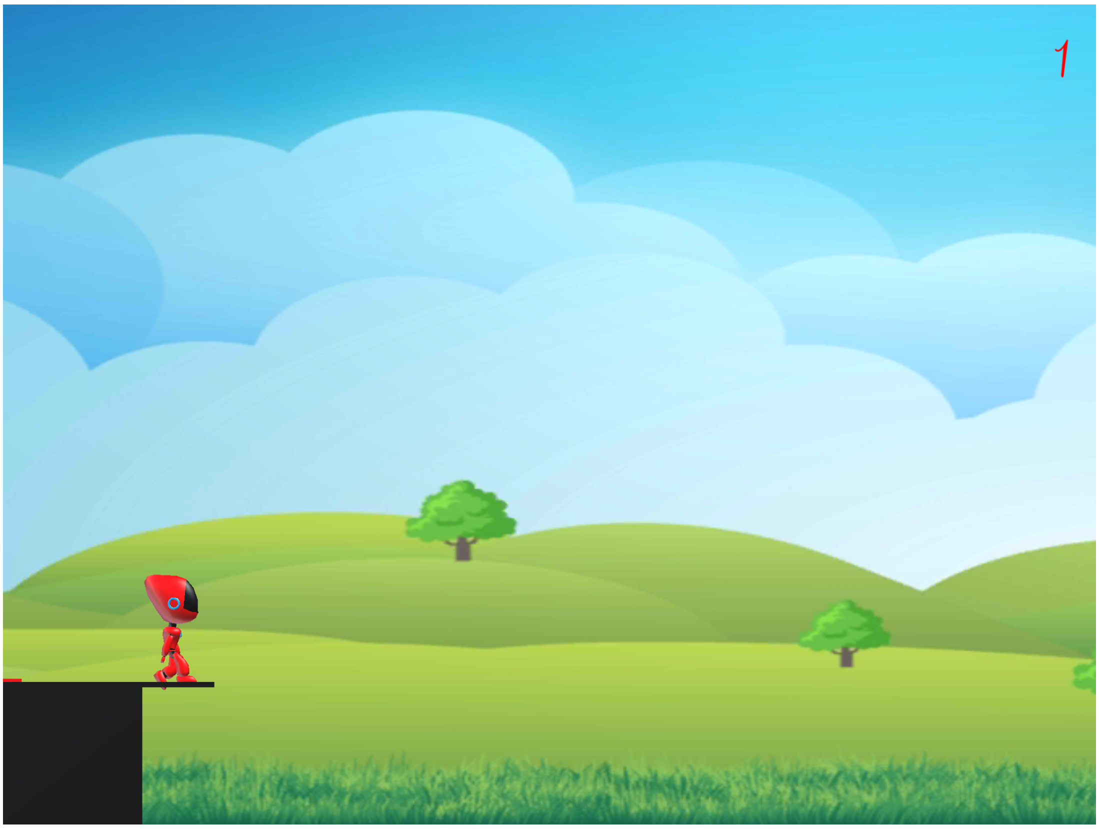
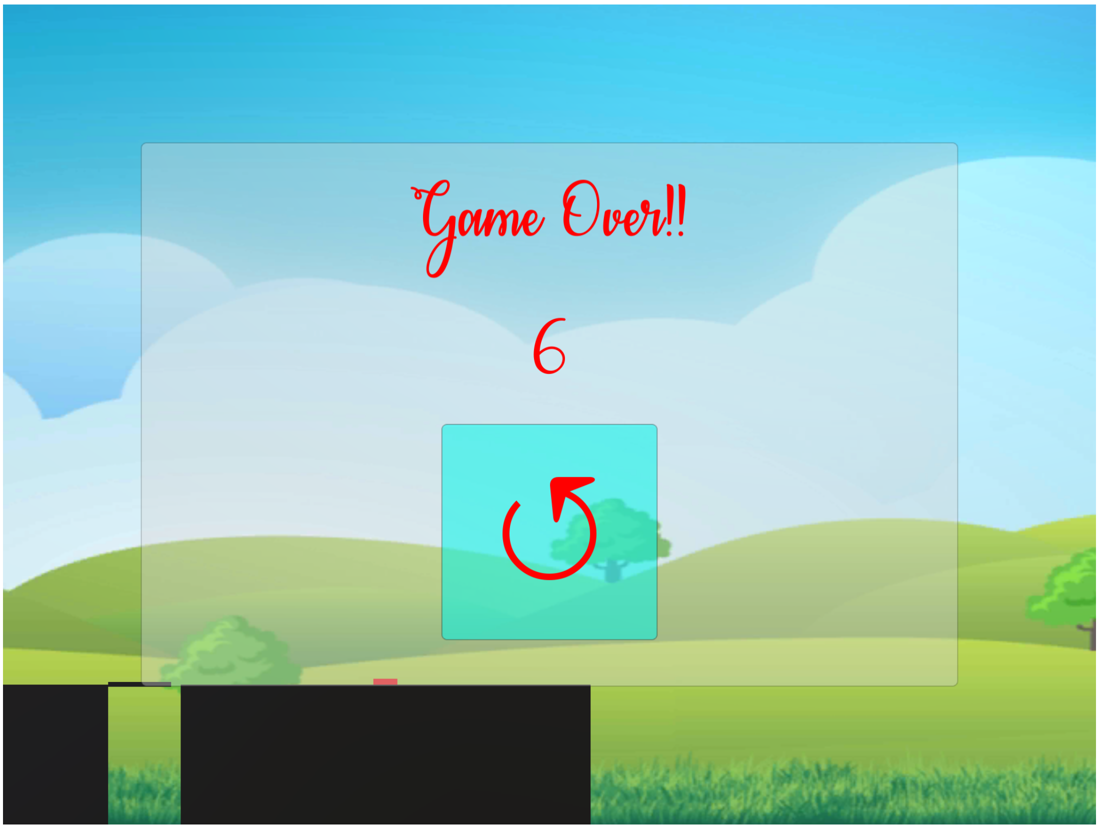


Figure 6.18: Game Over Scene



Chapter 7

Conclusion

In this project, we considered theoretical considerations of autism disorder such as disorder of prediction due to skewed priors, and parameterized the skewed priors on which a coherent training framework is developed. To evaluate this domain independent coherent training framework for individuals with autism on spatial and temporal domains we developed two games **Okay** and **Stick Hero**. The evaluation done until now is on pilot basis and is showing excellent results. After the clearance of ethics at Action for Autism center, a better evaluation will done.

This framework has potential to be included in various training therapies, and make them more effective. If not for their impaired predictive skills, individuals with autism are talented. So, we hope that this training framework can make their predictive skills better.

Bibliography

[BCAA+09] Simon Baron-Cohen, Emma Ashwin, Chris Ashwin, Teresa Tavassoli, and Bhismadev Chakrabarti. Talent in autism: hyper-systemizing, hyper-attention to detail and sensory hypersensitivity. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 364(1522):1377–1383, 2009.

[BWS+12] Julie Brisson, Petra Warreyn, Josette Serres, Stephane Foussier, and Jean Adrien-Louis. Motor anticipation failure in infants with autism: a retrospective analysis of feeding situations. *Autism*, 16(4):420–429, 2012.

[Fee10] Cecilia Feeley. Evaluating the transportation needs and accessibility issues for adults on the autism spectrum in new jersey. In *89th annual meeting of the transportation research board*, volume 25, 2010.

[FKR+10] Kimberly A Fournier, Cara I Kimberg, Krestin J Radonovich, Mark D Tillman, John W Chow, Mark H Lewis, James W Bodfish, and Chris J Hass. Decreased static and dynamic postural control in children with autism spectrum disorders. *Gait & posture*, 32(1):6–9, 2010.

[Hug96] Claire Hughes. Brief report: Planning problems in autism at the level of motor control. *Journal of autism and developmental disorders*, 26(1):99–107, 1996.

[IRLS+14] Teresa Iuculano, Miriam Rosenberg-Lee, Kaustubh Supekar, Charles J Lynch, Amirah Khouzam, Jennifer Phillips, Lucina Q Uddin, and Vinod Menon. Brain organization underlying superior mathematical abilities in children with autism. *Biological Psychiatry*, 75(3):223–230, 2014.

[KW04] Konrad P Kording and Daniel M Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427(6971):244, 2004.

[LRL+00] Catherine Lord, Susan Risi, Linda Lambrecht, Edwin H Cook, Bennett L Leventhal, Pamela C DiLavore, Andrew Pickles, and Michael Rutter. The autism diagnostic observation schedule—generic: A standard measure of social and communication deficits associated with the spectrum of autism. *Journal of autism and developmental disorders*, 30(3):205–223, 2000.

[LRLC94] Catherine Lord, Michael Rutter, and Ann Le Couteur. Autism diagnostic interview-revised: a revised version of a diagnostic interview for caregivers of individuals with possible pervasive developmental disorders. *Journal of autism and developmental disorders*, 24(5):659–685, 1994.

[MDB03] Cynthia A Molloy, Kim N Dietrich, and Amit Bhattacharya. Postural stability in children with autism spectrum disorder. *Journal of autism and developmental disorders*, 33(6):643–652, 2003.

[MSJF04] Nancy J Minshew, KiBum Sung, Bobby L Jones, and Joseph M Furman. Underdevelopment of the postural control system in autism. *Neurology*, 63(11):2056–2061, 2004.

[PB12] Elizabeth Pellicano and David Burr. When the world becomes 'too real': a bayesian explanation of autistic perception. *Trends in cognitive sciences*, 16(10):504–510, 2012.

[Rob08] John Elder Robison. *Look me in the eye: My life with Asperger's*. Random House, 2008.

[SKG+14] Pawan Sinha, Margaret M Kjelgaard, Tapan K Gandhi, Kleovoulos Tsourides, Annie L Cardinaux, Dimitrios Pantazis, Sidney P Diamond, and Richard M Held. Autism as a disorder of prediction. *Proceedings of the National Academy of Sciences*, 111(42):15220–15225, 2014.

[SMBA03] Christina Schmitz, Joëlle Martineau, Catherine Barthélémy, and Christine Assaiante. Motor control and children with autism: deficit of anticipatory function? *Neuroscience letters*, 348(1):17–20, 2003.

[TFPL05] Helen Tager-Flusberg, Rhea Paul, and Catherine Lord. Language and communication in autism. *Handbook of Autism and Pervasive Developmental Disorders, Volume 1, Third Edition*, pages 335–364, 2005.