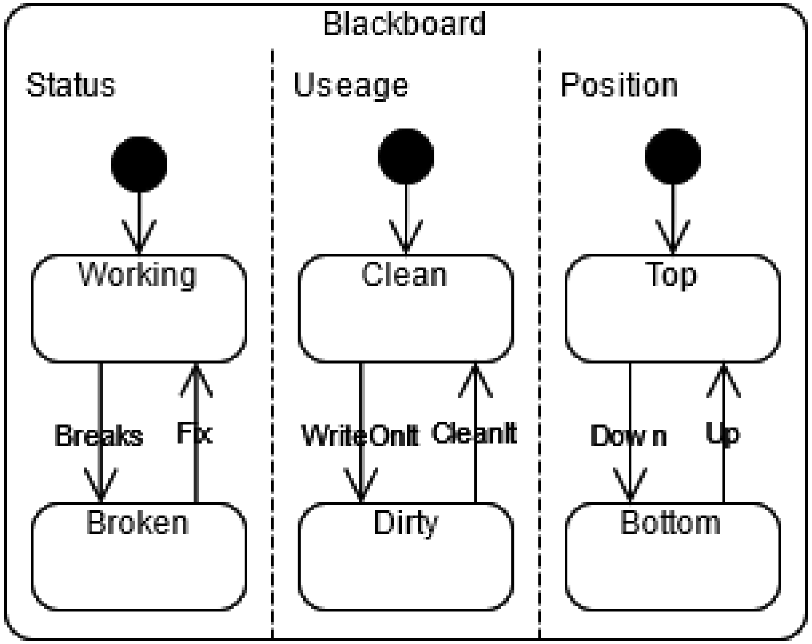


[Example-statechart for the blackboard]

Dynamic Storytelling Based on Complex Graphs

Dominik Huber

29.07.2021

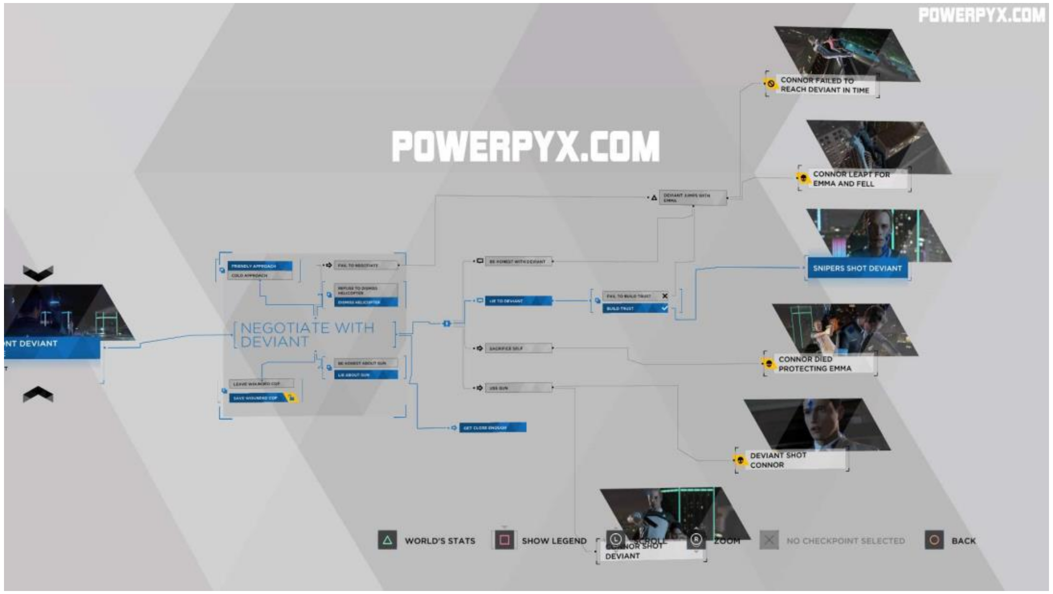
Final: Bachelor Informatics: Games Engineering

Supervisor: Prof. Gudrun Klinker, Ph.D.

Motivation

- Informal graphs to visualize the storyline on wiki and game-guide webpages
- Graphs only useful as visualization-format or also usable for implementation of storytelling in games?
- Dynamic stories as reactive systems

  → Statecharts



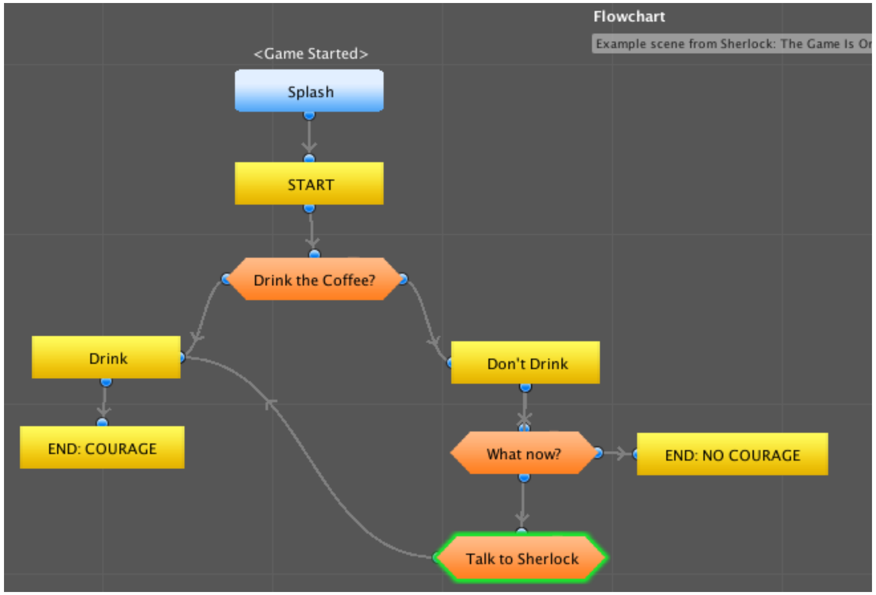
[Powerpyx.com – Detroit: Become Human]

Problem Description: Issue

To what extend can statecharts be used as an environment for representing dynamic stories in games?

Existing Solutions / Related Work

- Informal graphs on wiki and game-guide webpages
- Plugins for “Unity 3D”: “Inky”, “Twine”, “YarnEditor”, “Talkit”, “Fungus”
- Hierarchical task networks
- Planning algorithms
- SCXML



[Example game scenario from “Fungus” in “Unity 3D”]

Goals of this Thesis

- Create mapping of storytelling elements onto statechart features to obtain statecharts as an environment for representing dynamic stories in games.
- Testing of this mapping by applying an exemplary game scenario within a paper-prototype

Dynamic Storytelling

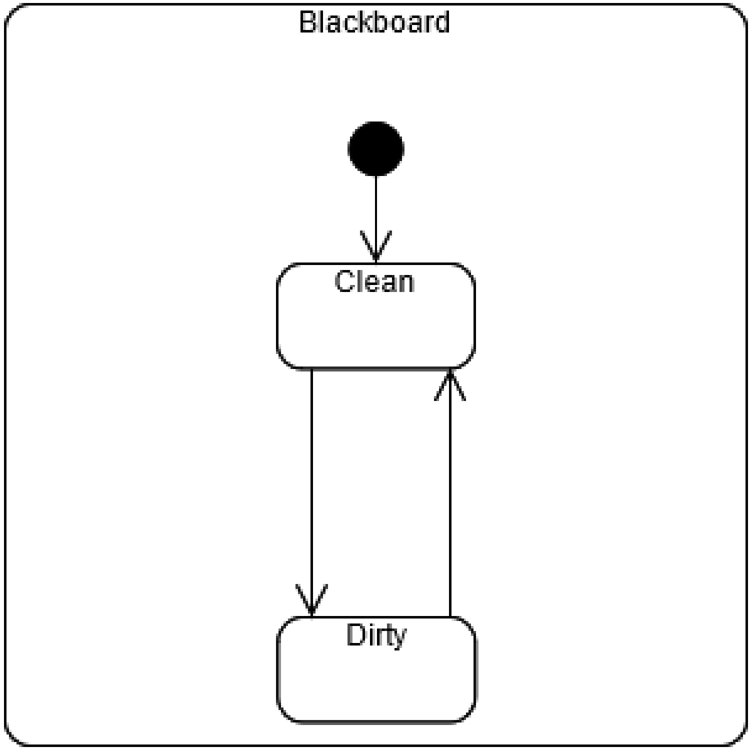
- Decisions and actions of the player shape the path of the storyline → story changes dynamically
- Story elements:
  - Setting
  - Protagonist and antagonist
  - Structure elements: main storyline and parallel side storylines, quests and interrupts
  - Storytelling methods: Dialogs, Foreshadowing of future events, flashbacks of past events

Statecharts – Definition

“Statecharts constitute a visual formalism for describing states and transitions in a modular fashion, enabling clustering, orthogonality (i.e., concurrency) and refinement, and encouraging ’zoom’ capabilities for moving easily back and forth between levels of abstraction.“ – David Harel, 1987, in “Statecharts: A Visual Formalism for Complex Systems“

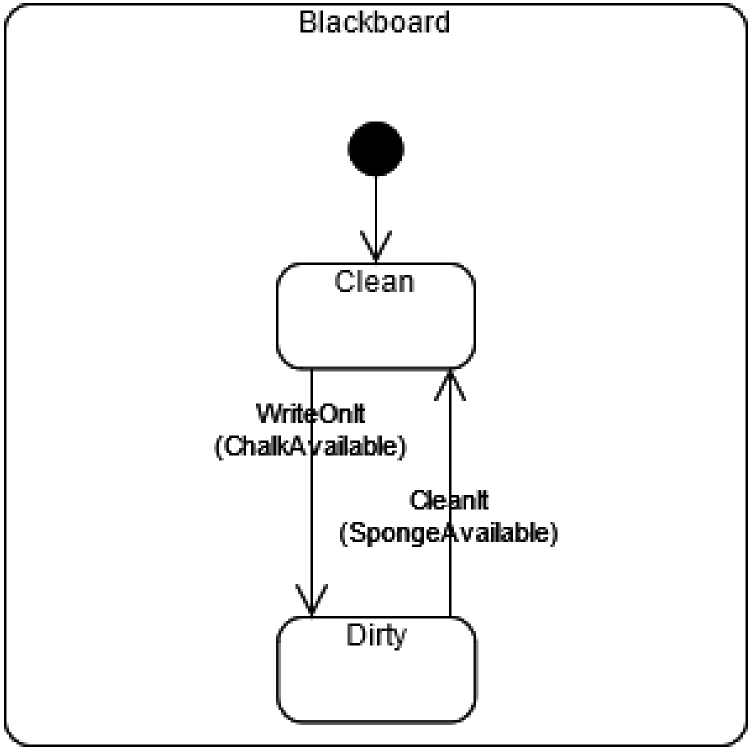
Statecharts – Basics

- **Nodes**: States of the system
- **Hierarchy**: Super-states contain substates
- **Default states**: Mark state that is activated when entering component
- **Transitions**: Changes between different states of the system



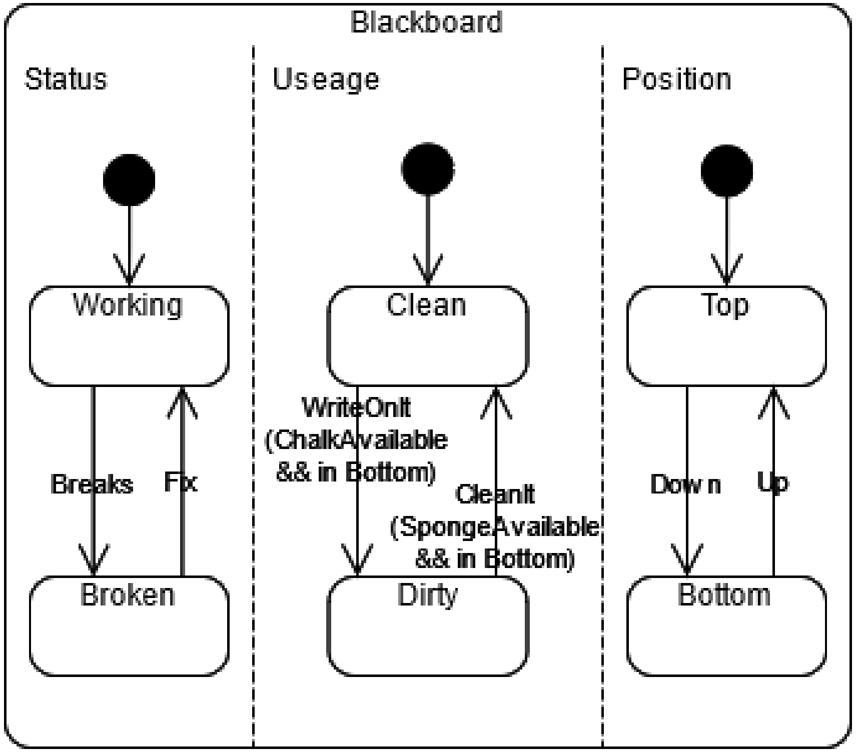
Statecharts – Basics

- **Trigger-Events**: Events that activate transitions
- **Conditions**: Conditions that must be true to traverse the transitions



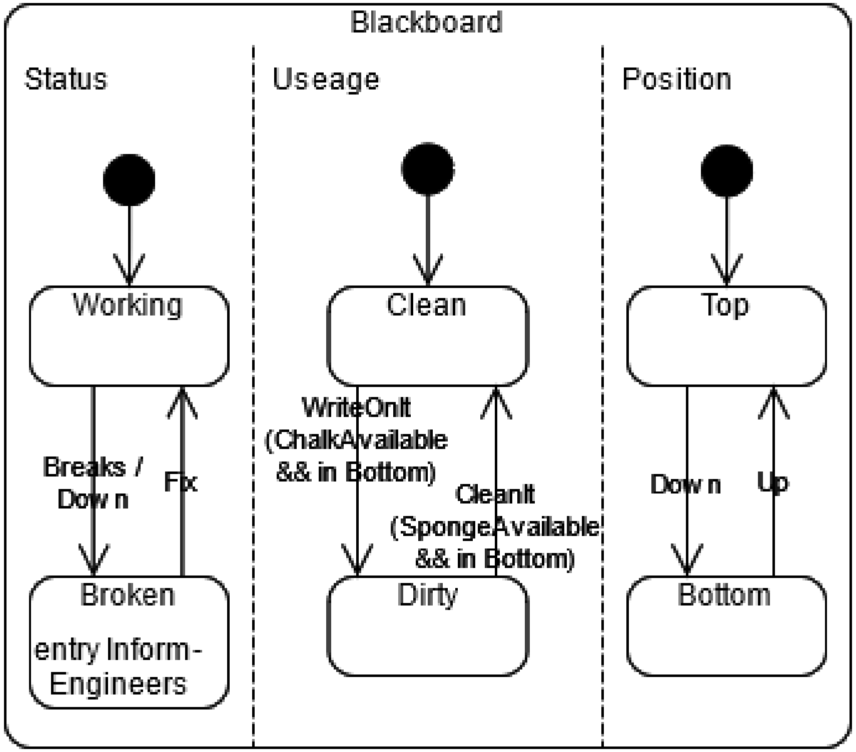
Statecharts – Basics

- **Orthogonality**: Multiple independent states are active
- **Dependencies** between parallel components: “in <StateName>”-condition


Blackboard
Status
Useage
Position
Working
Broken
Breaks
Fix
Clean
Dirty
WriteOnIt (ChalkAvailable && in Bottom)
CleanIt (SpongeAvailable && in Bottom)
Top
Bottom
Down
Up

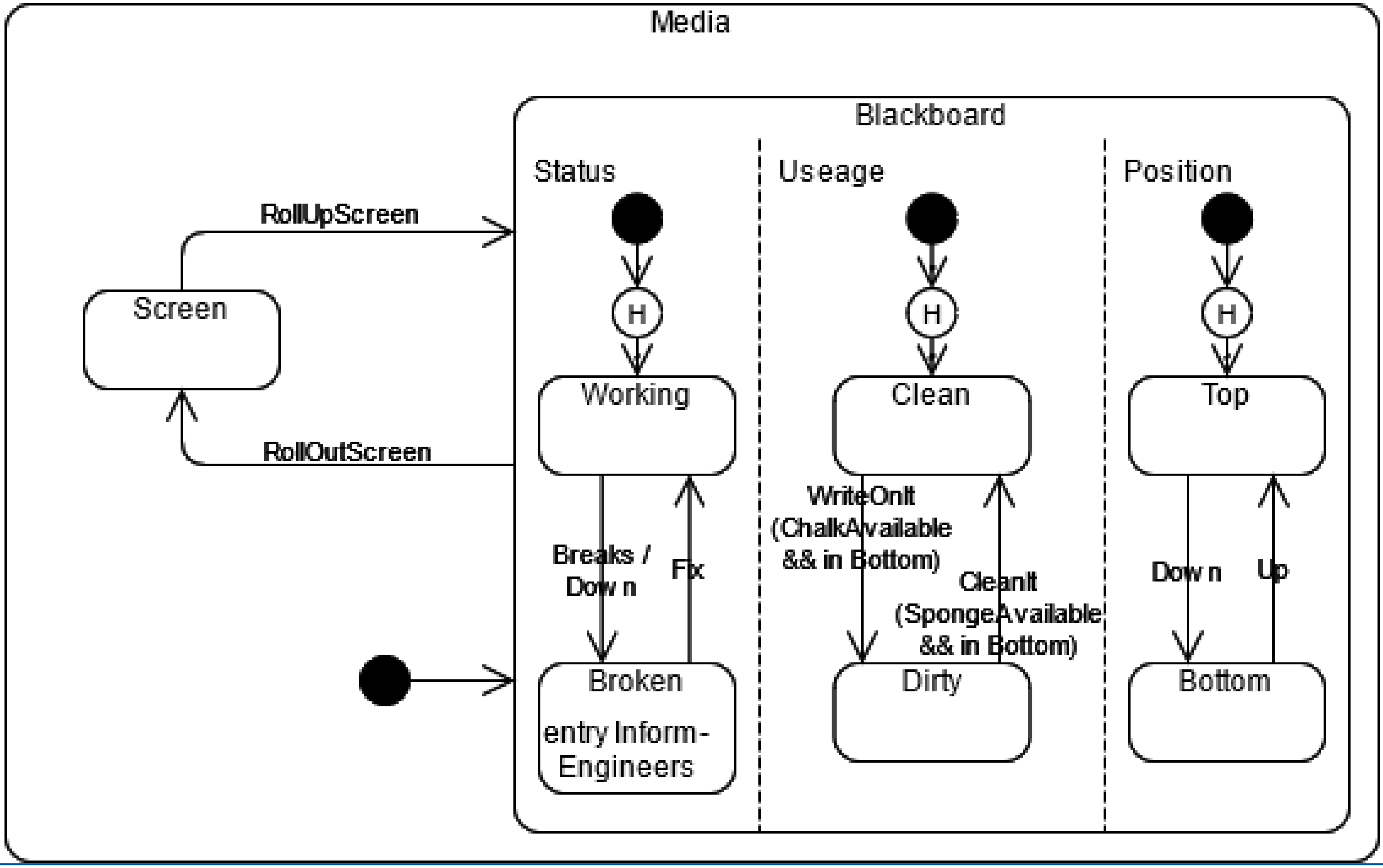
Statecharts – Basics

- **Actions**: Generating events and output



Statecharts – Basics

- **History state**: Entering the last visited substate when reentering super-state



Mapping Storytelling Elements onto Statechart Features

Statechart Feature	Storytelling Elements	Examples in the context of games
Node, atomic state	Concrete state of the plot	The Player tries to shoot the dragon
Arrow/Transition	Step to the next section of the plot/part of the storyline	Change in the game world, like when a dragon is sighted
Triggering event	Change in the game world	A dragon is sighted
Condition	Event or Action that must have occurred or variable that must have a certain value	When building a house, the roof can just be started when the walls are finished. The completion of the walls can be managed by a condition

Mapping Storytelling Elements onto Statechart Features

Statechart Feature	Storytelling Elements	Examples in the context of games
Action	Instantaneous action or occurrence triggered or produced (indirectly) through the player's action which can have an influence on the entire game world (1); Generation of an internal event or change of an internal variable, which represents something of the game world or of the game mechanics (2)	- (1): The player reaches the presidential suite, but he is dead. That action triggers a transition with the event "presidentWasKilled" in a parallel storyline. In this orthogonal storyline, the player cannot give the gun to the president anymore. - (2): The variable "VillageAttacked" is set during the attack of the dragon on the village to "true". After coming back to the building site, the condition "VillageAttacked == true" evaluates to true and results in an earlier state of the building process, since the building site was devastated by the dragon.

Mapping Storytelling Elements onto Statechart Features

Statechart Feature	Storytelling Elements	Examples in the context of games
And decomposition in super-state/orthogonality/ concurrency	One single event causing two independent happenings leading to two parallel storylines, like plot and sub-plot, like actions of the main character and actions of the non-playable characters or antagonist, or two parallel quests (1), or different subsystems (2)	- (1): Two parallel quests: In one the player helps the blacksmith forging a sword for the king and in the other he helps mercenaries to plan and execute a raid on a bandit camp in parallel - (2): Payer is in tutorial (storyspace) and the tutorial takes place in the castle (worldspace)

Mapping Storytelling Elements onto Statechart Features

Statechart Feature	Storytelling Elements	Examples in the context of games
Shallow history (H), deep history (H*)	Coming back to a point in the plot; digress and return in a dialog; return after interrupt in the plot; pause the story/game	The player is building a house when a dragon attacks. After defeating the dragon, the player returns to the construction site. Now there are two options: Return to unchanged point, where storyline was interrupted – construction site unchanged or return to changed point, where storyline was interrupted – construction site was devastated by the dragon

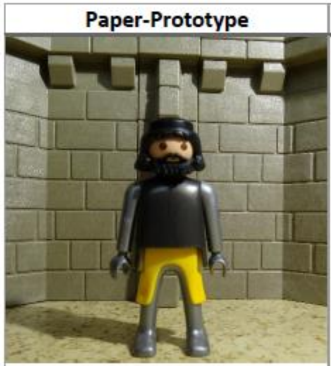
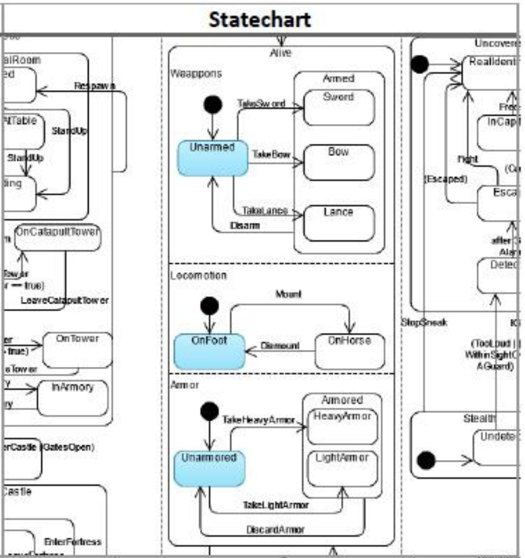
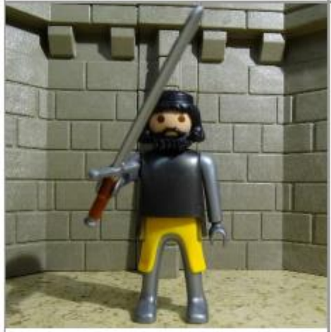
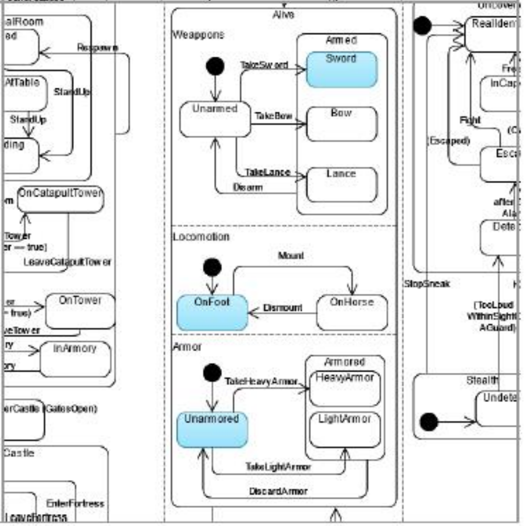
Paper-Prototype – Format

- Advantages: Simplicity, low cost and reduction to show the essentials
- Type of Paper-Prototype: Photographically captured gameplay-situations represented by “Playmobil” figures

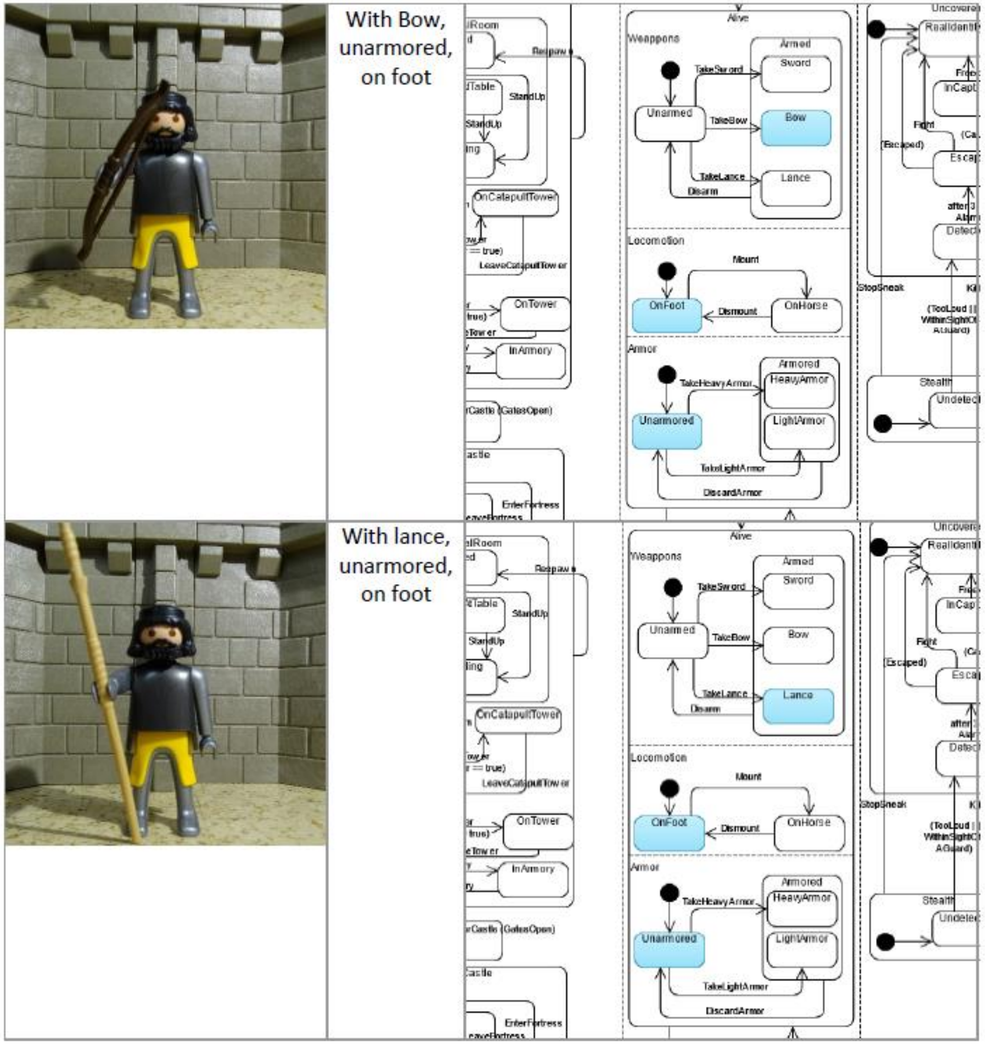
Paper-Prototype – Setting

- Medieval fantasy environment
- Protagonist: Knight who lives in a castle
- Open world game with quests and interrupts

Paper-Prototype – Player‘s Basic States

Paper-Prototype	Description	Statechart
	Unarmed, unarmored, on foot	
	With sword, unarmord, on foot	

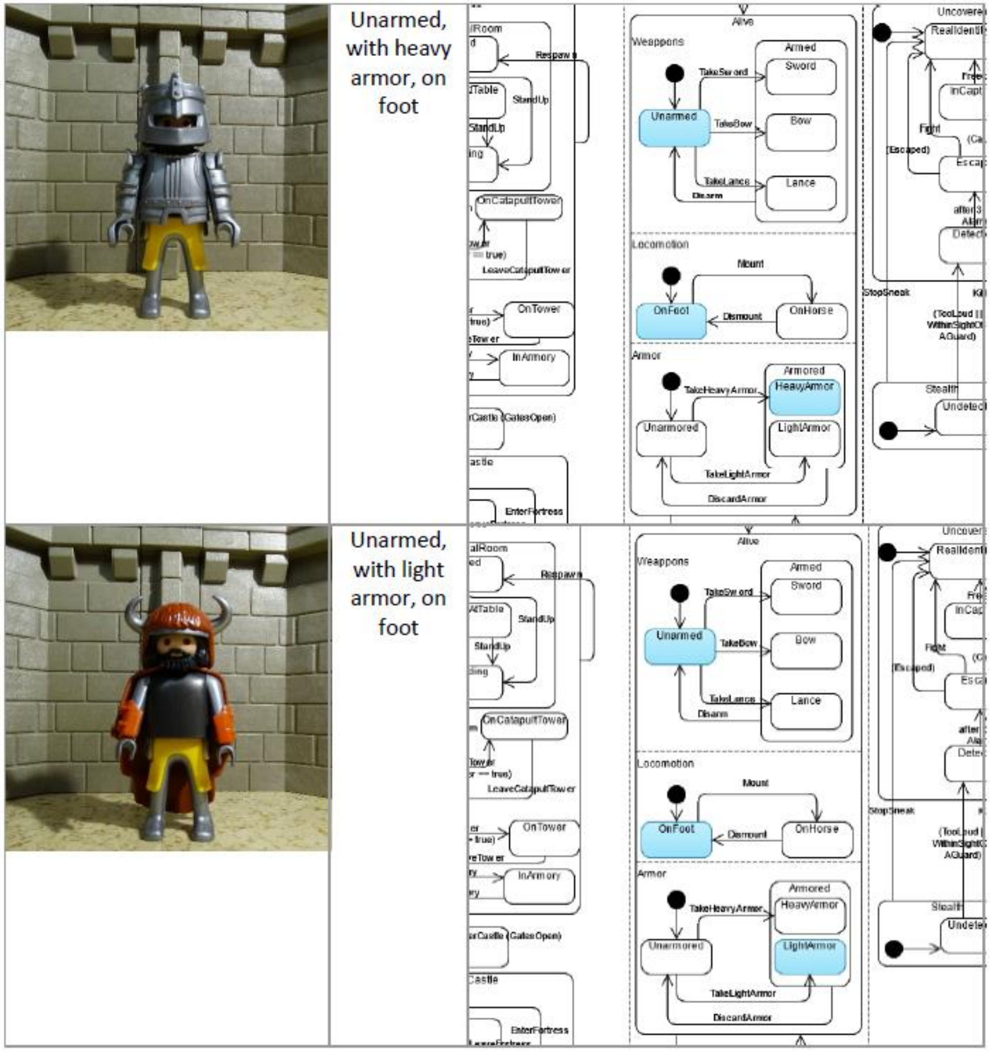
Paper-Prototype – Player‘s Basic States



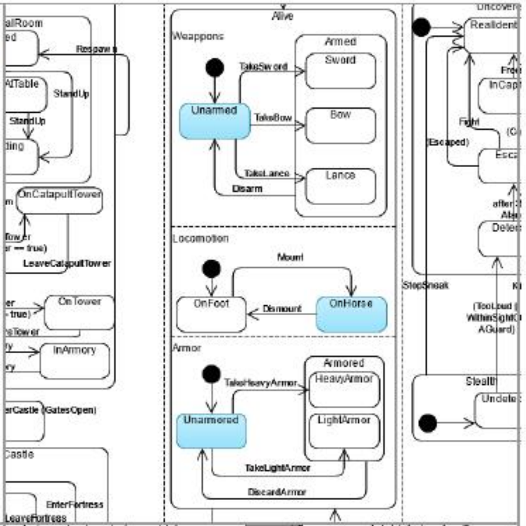
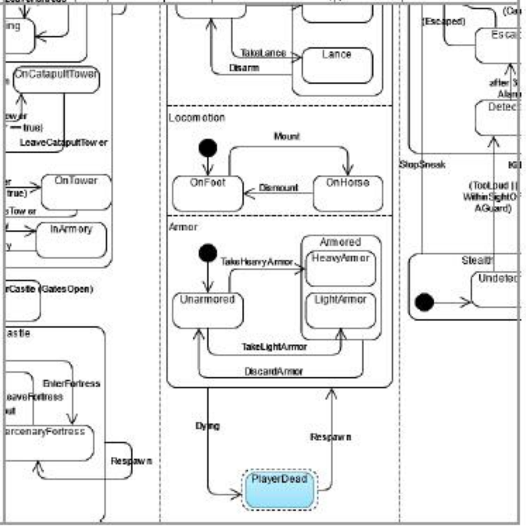
With Bow, unarmored, on foot

With lance, unarmored, on foot

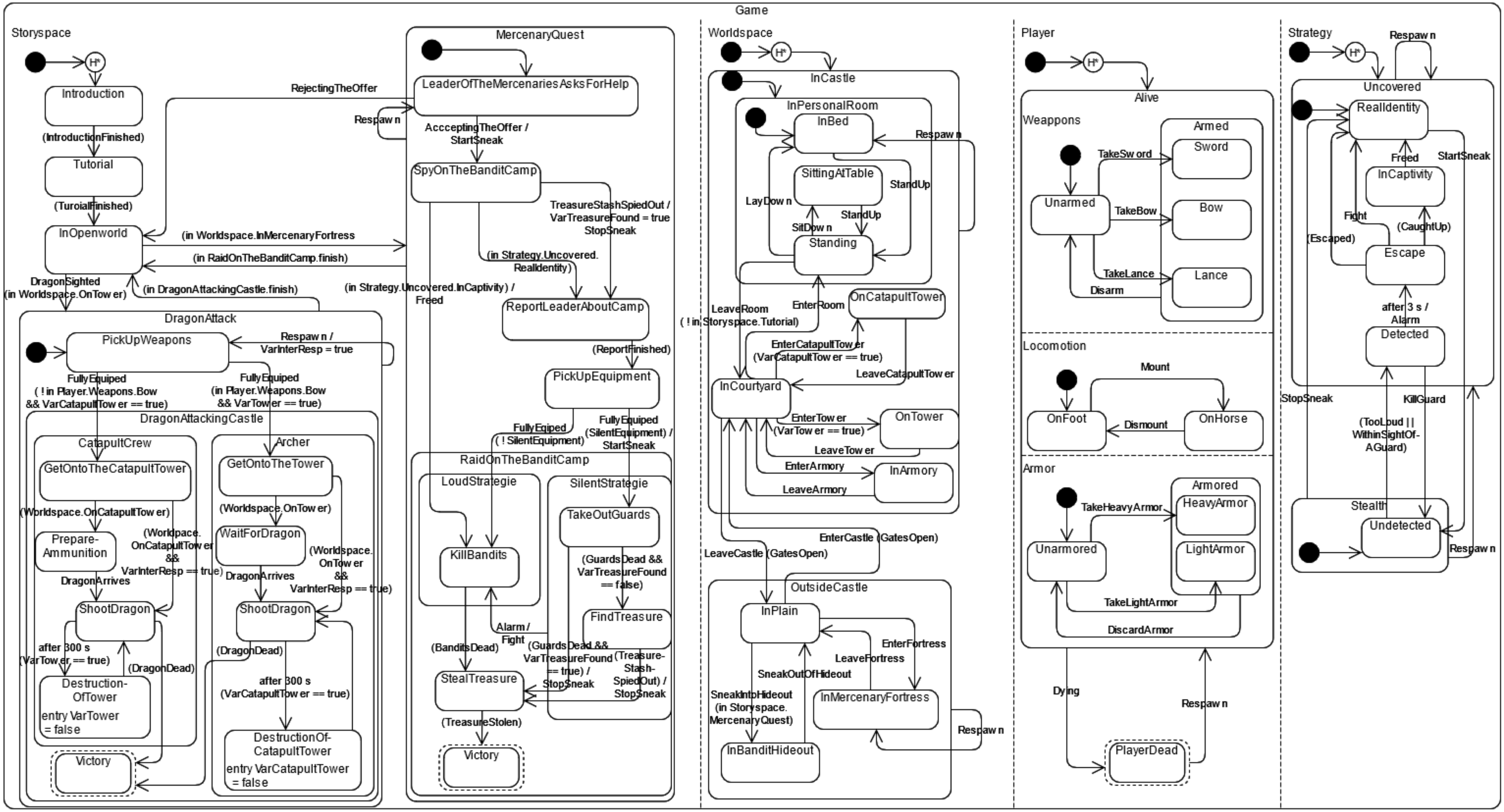
Paper-Prototype – Player‘s Basic States



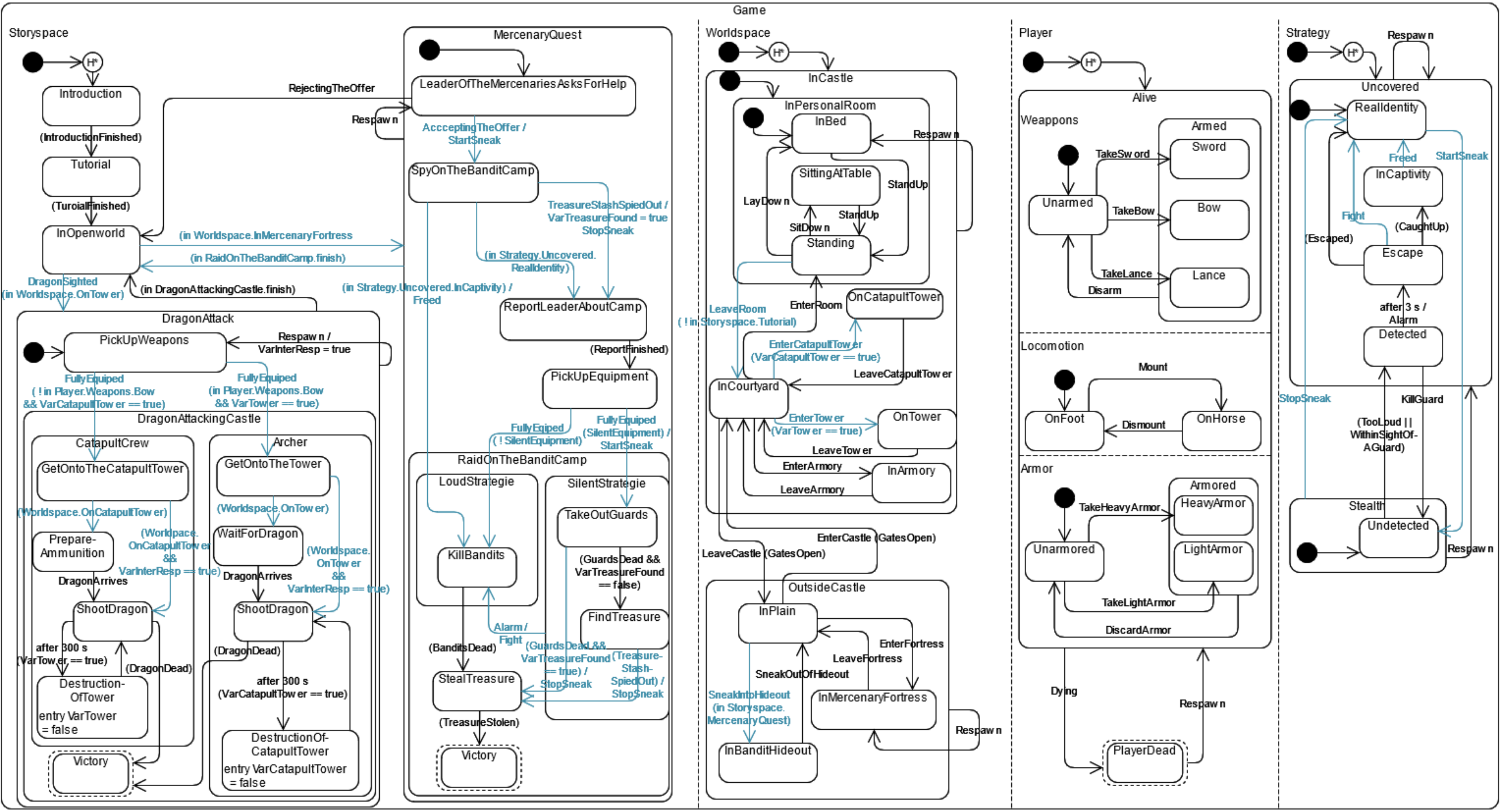
Paper-Prototype – Player‘s Basic States

	Unarmed, unarmored, on horse	
	Dead	

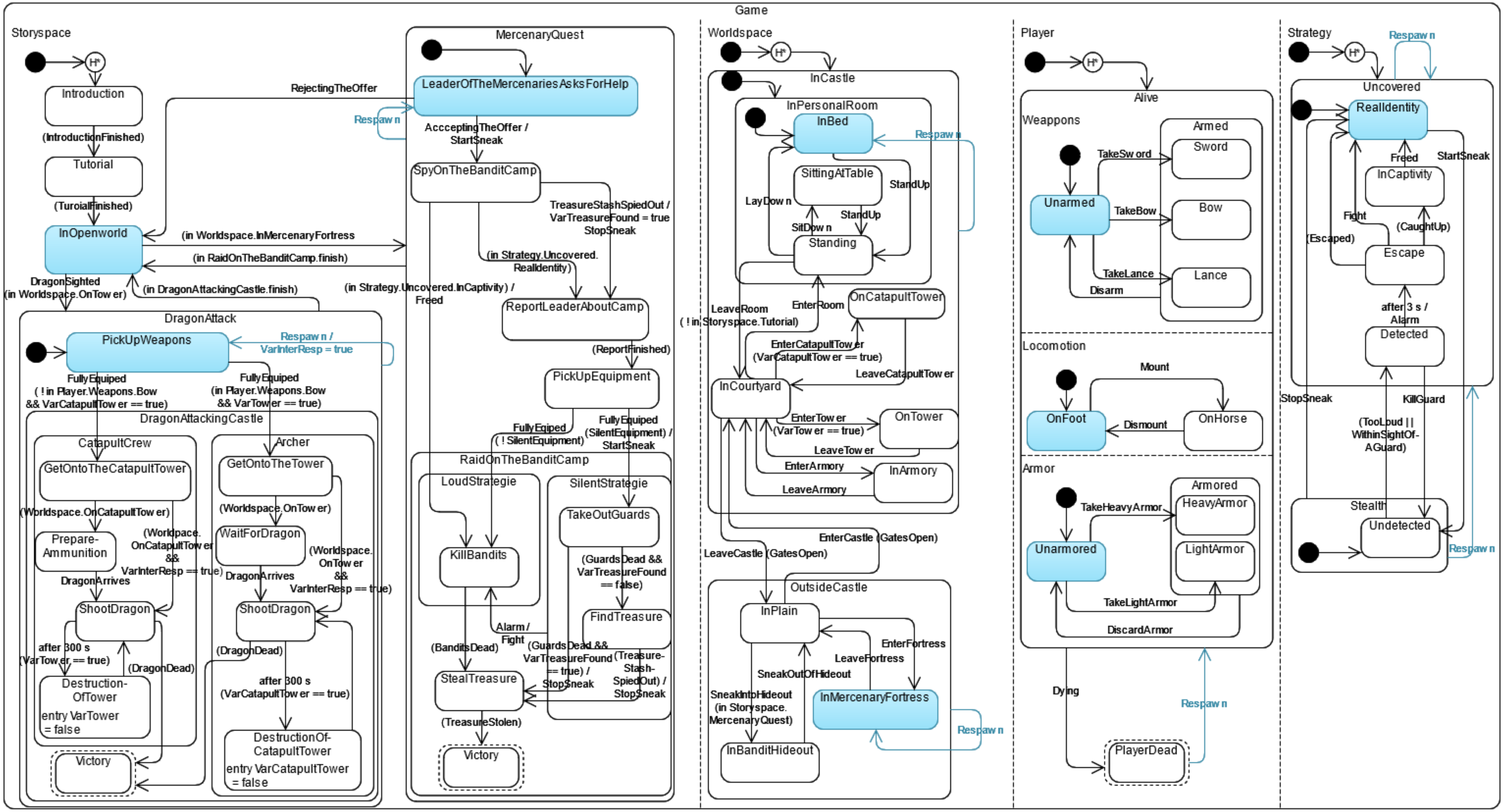
Paper-Prototype – Statechart



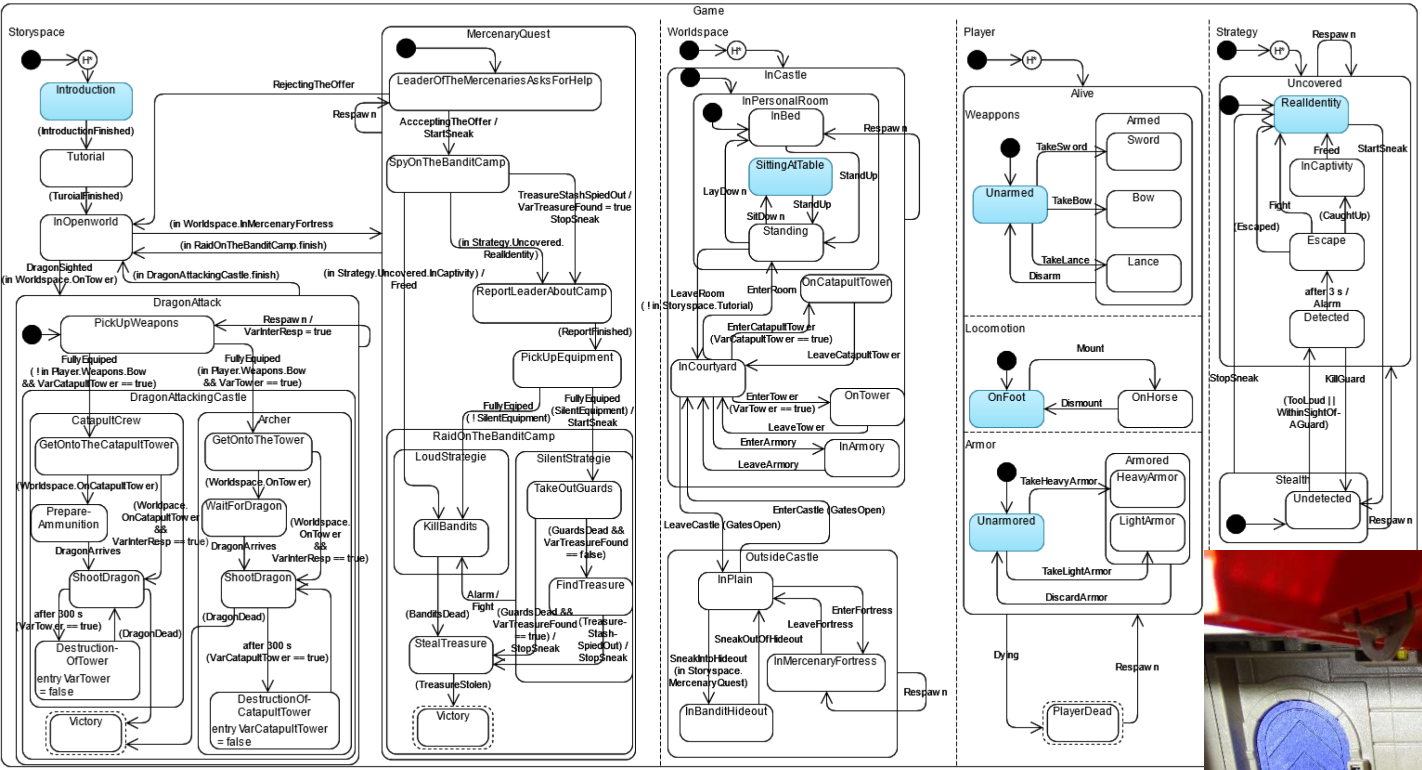
Paper-Prototype – Dependencies



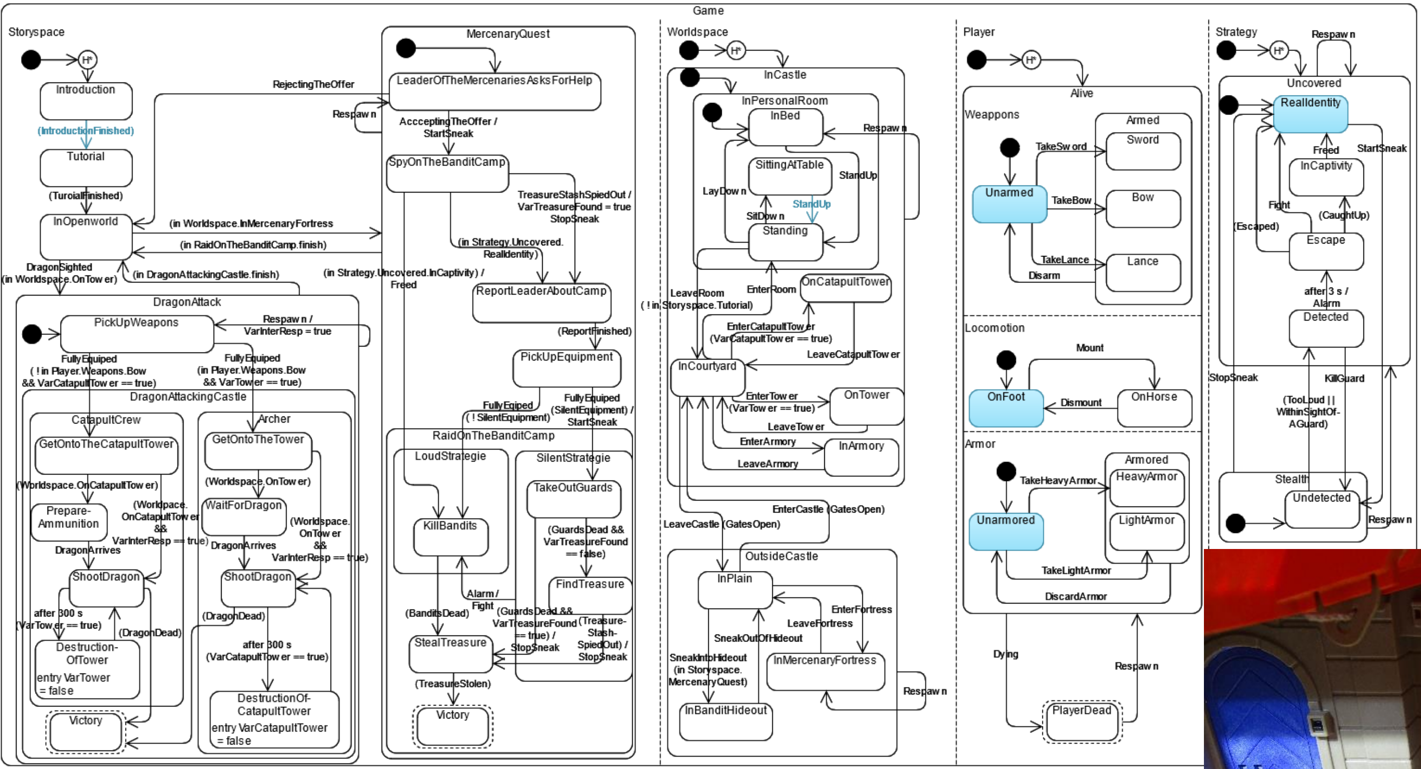
Paper-Prototype – Distributed Respawn



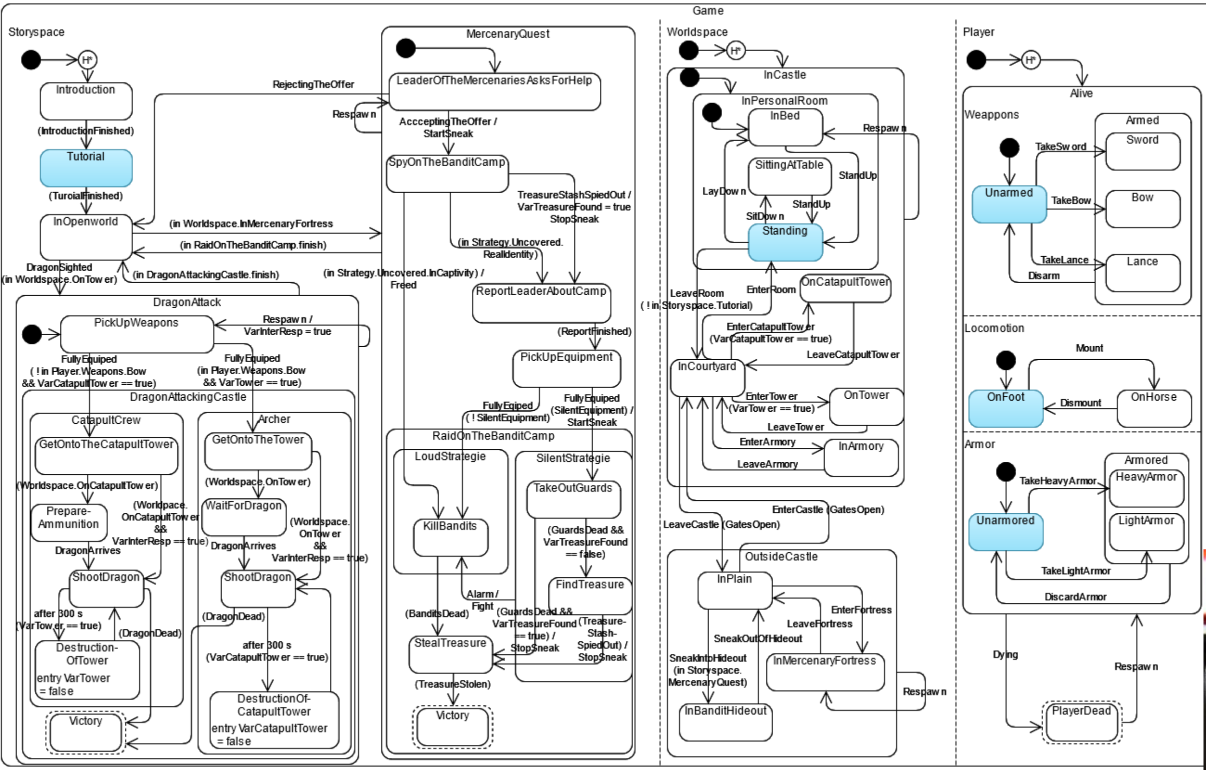
P.P. – Storyline: Introduction



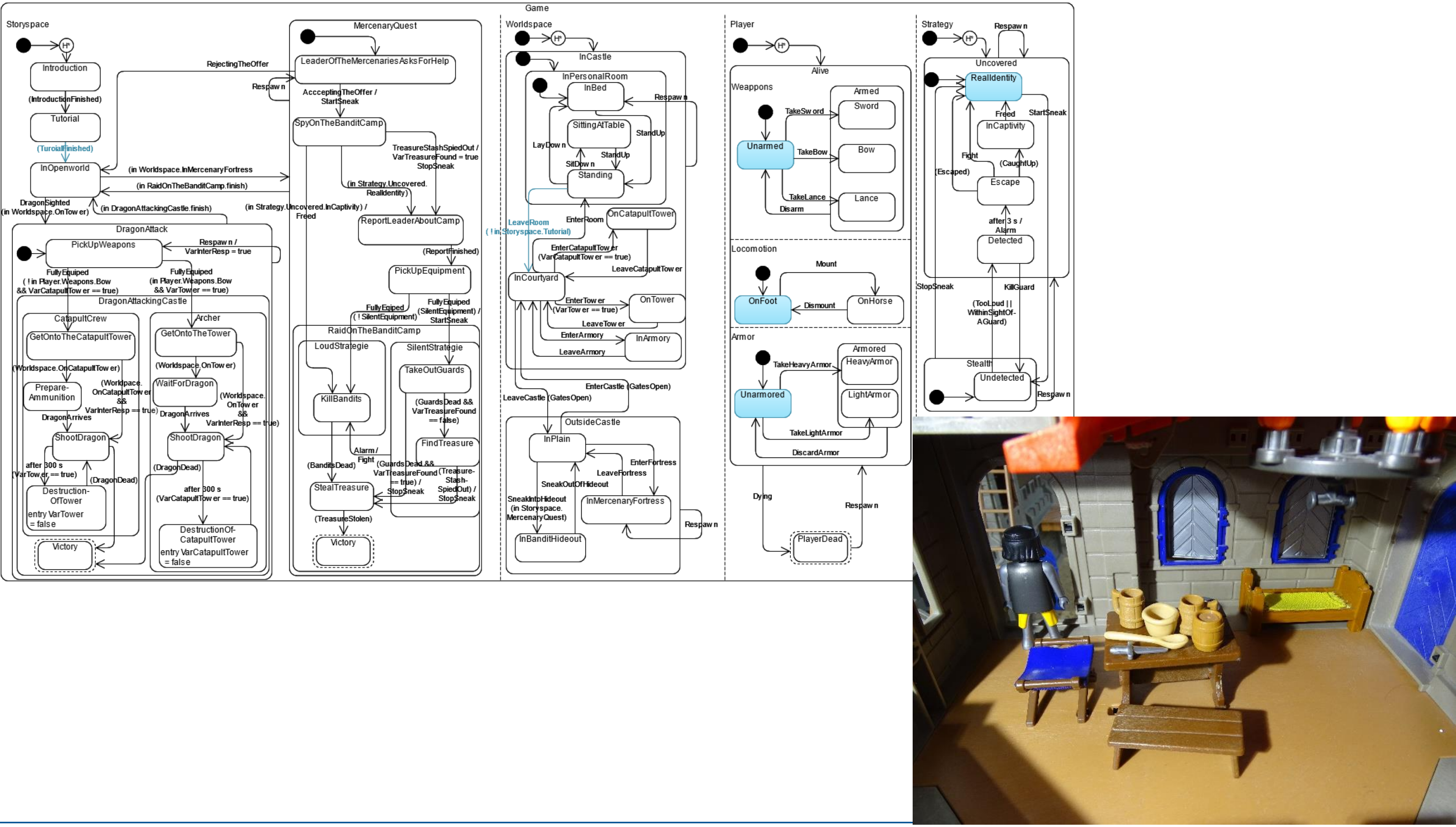
P.P. – Storyline: Introduction



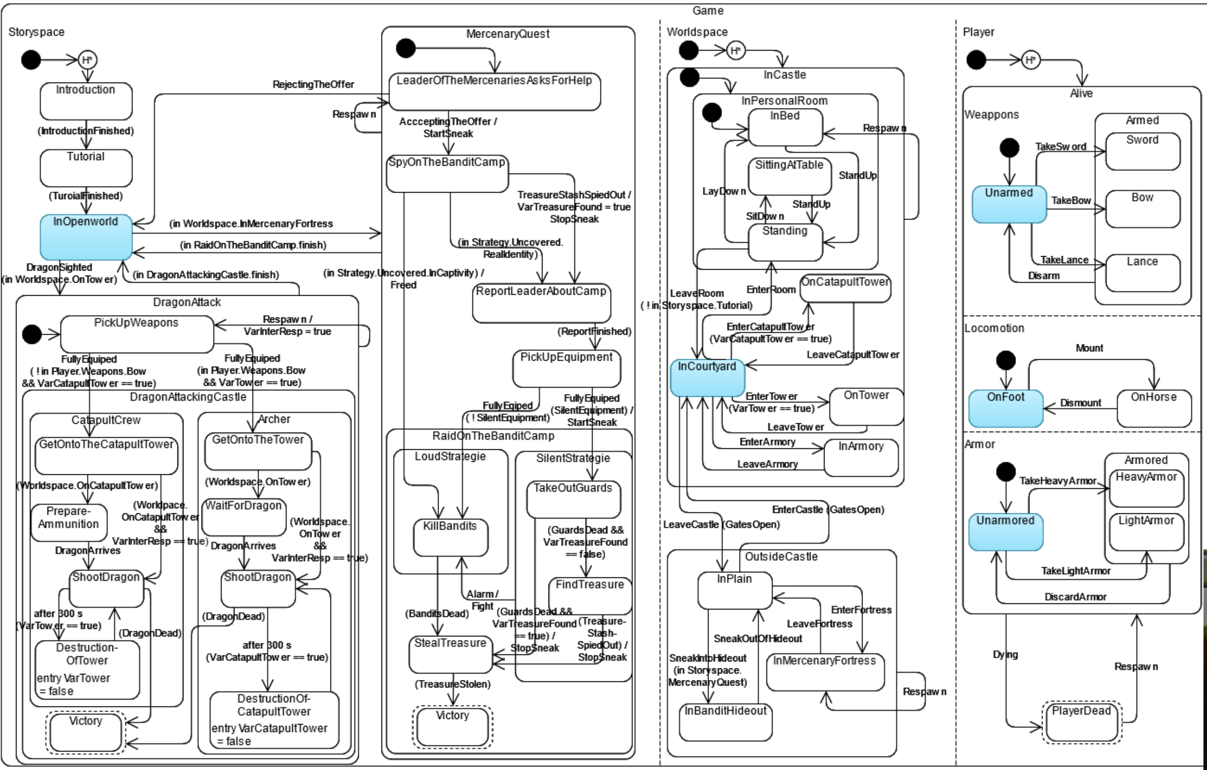
P.P. – Storyline: Tutorial



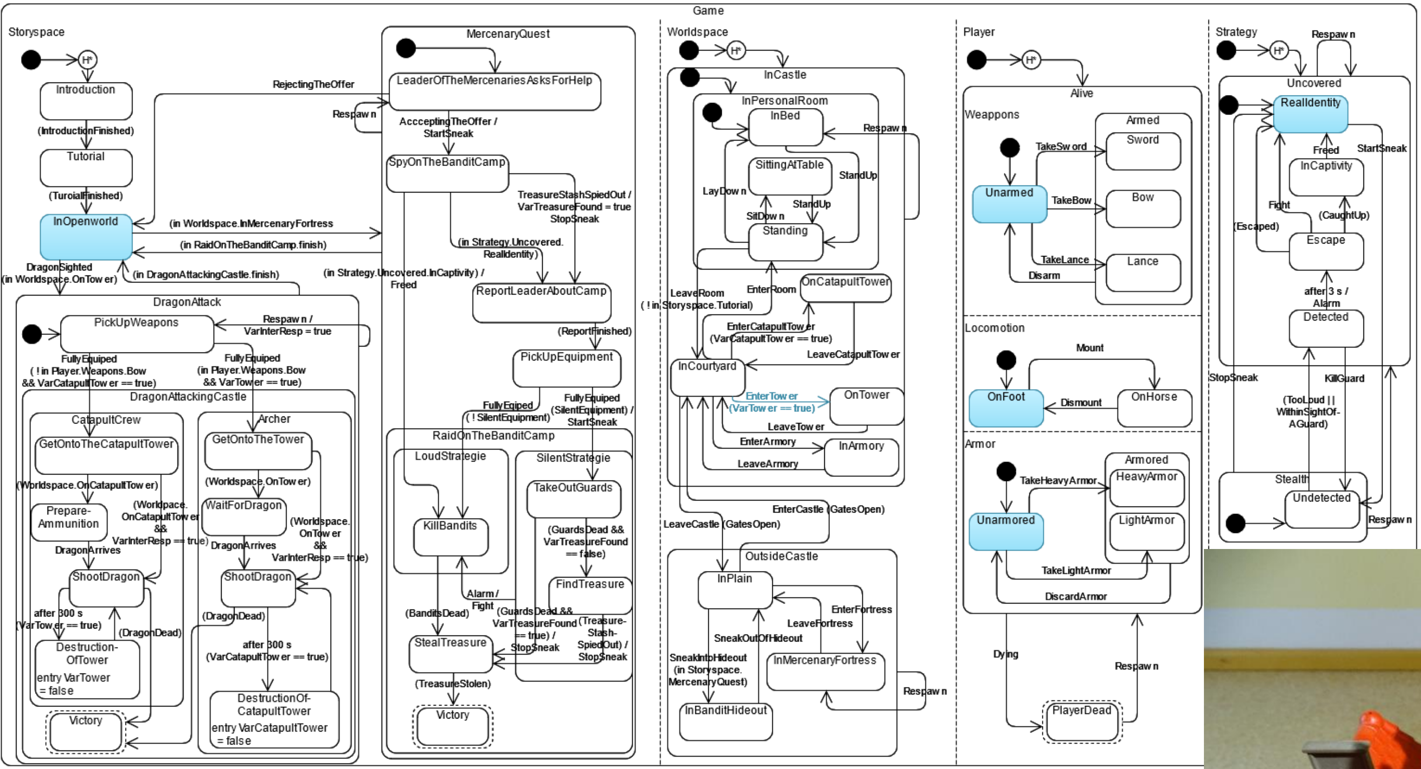
P.P. – Storyline: Tutorial



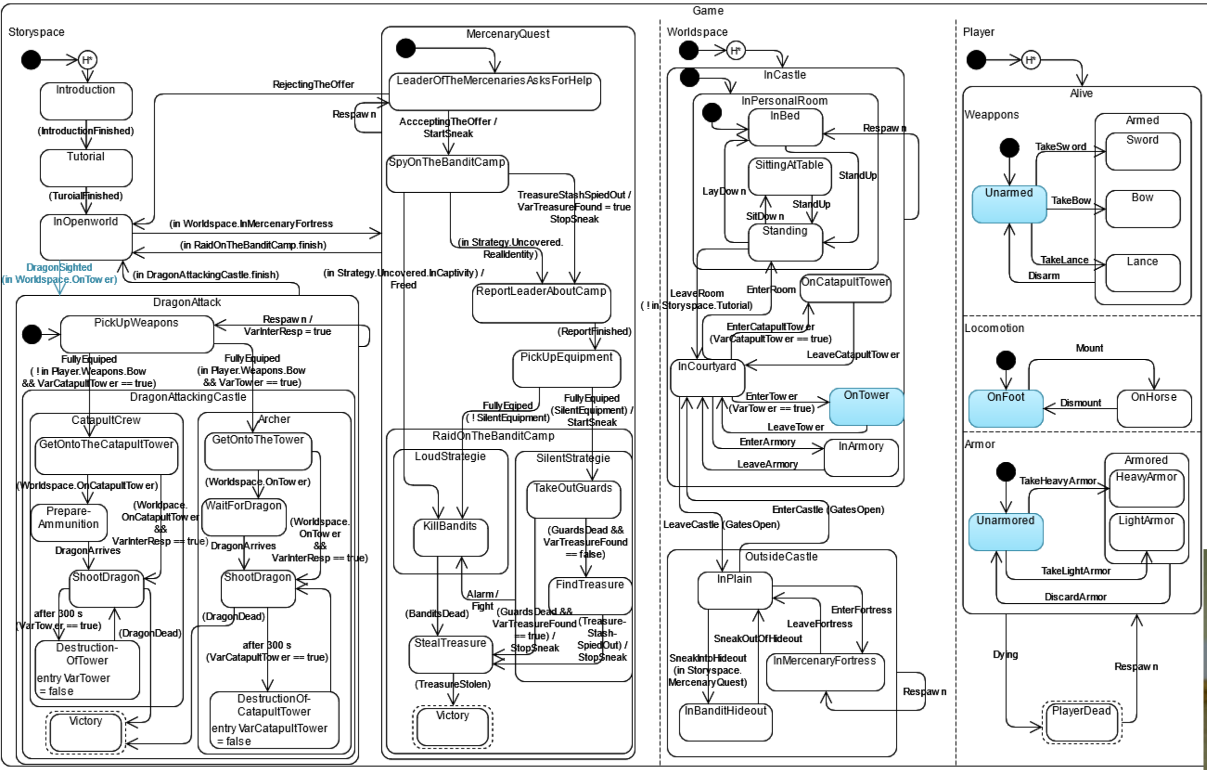
P.P. – Storyline: In Open World

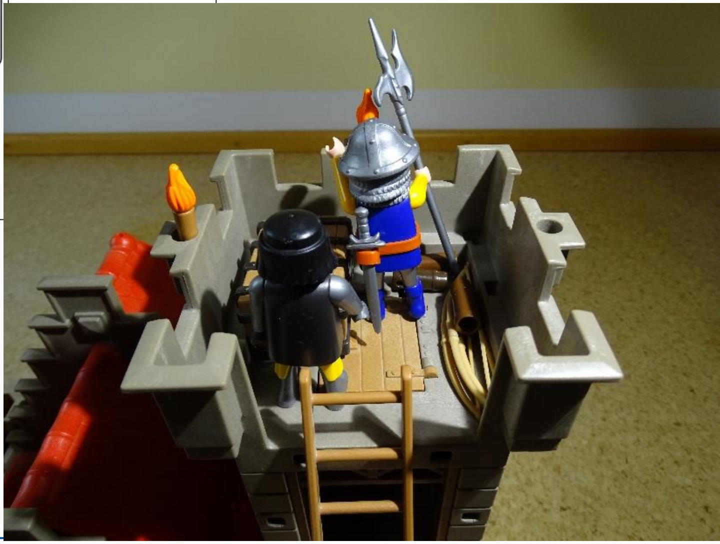


P.P. – Storyline: In Open World



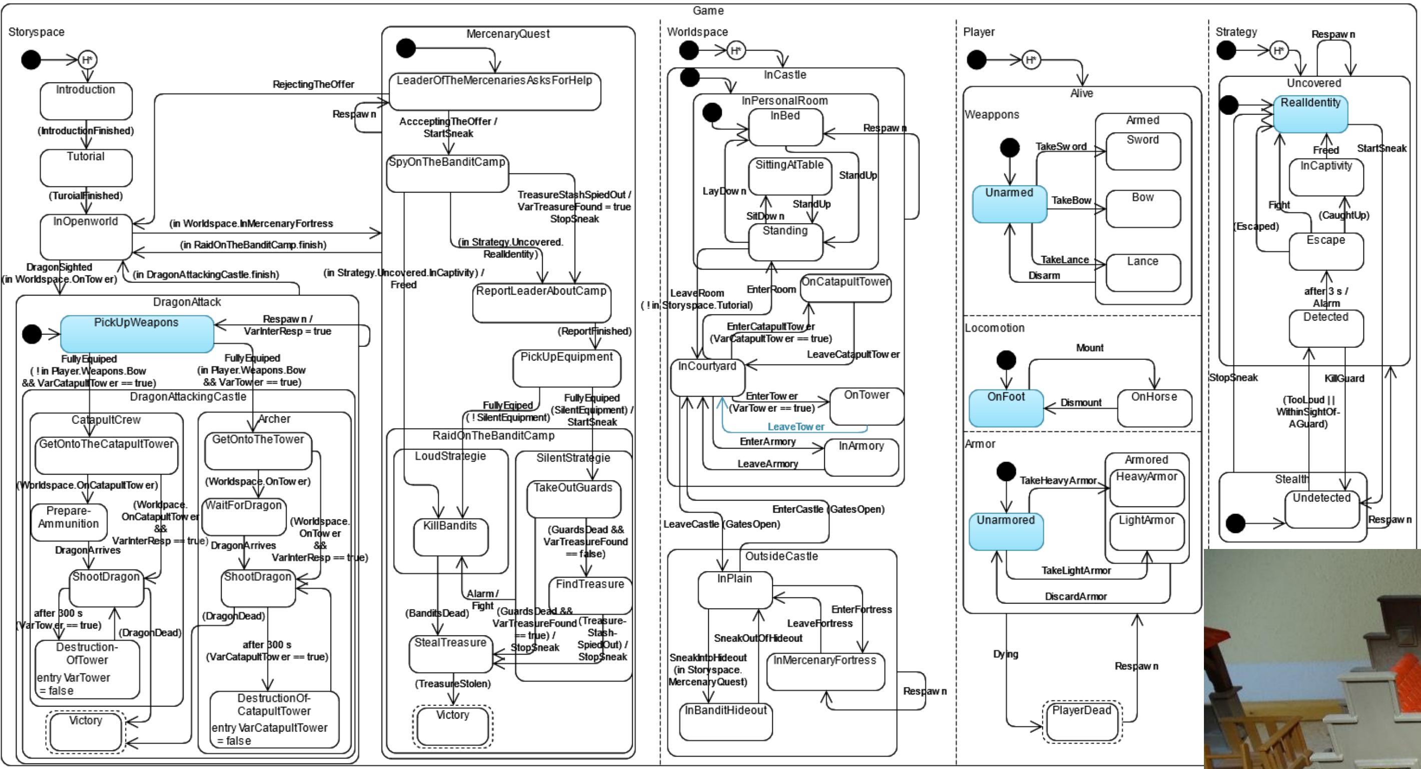
P.P. – Storyline: Dragon Sighted



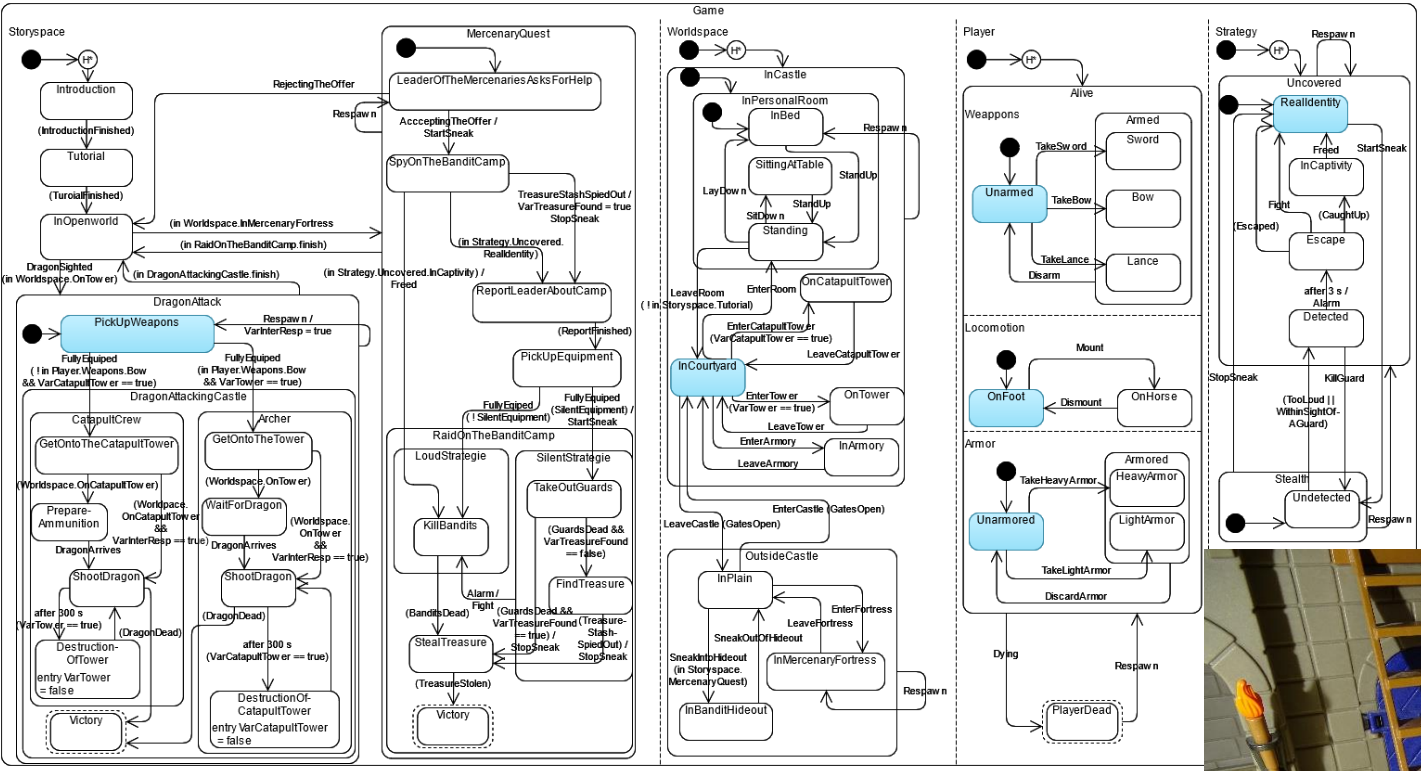


P.P. – Storyline: Dragon-Attack Interrupt

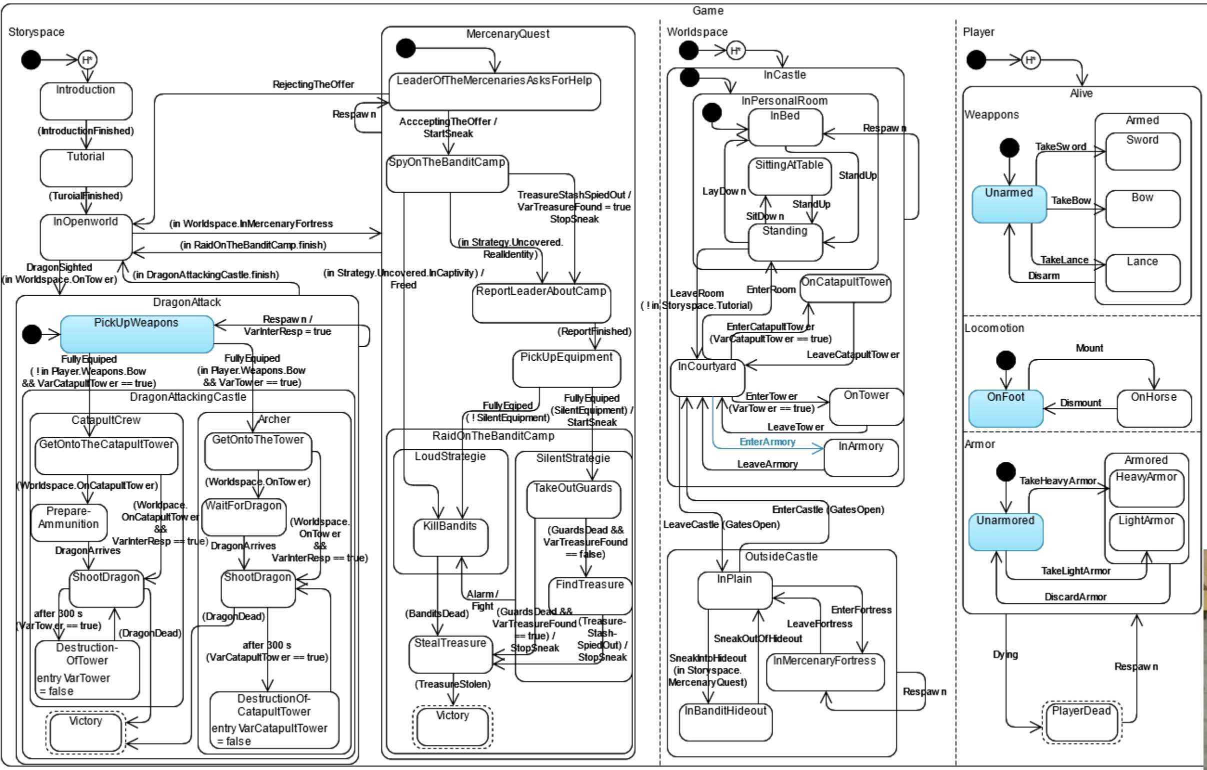
P.P. – Storyline: Pick Up Weapons



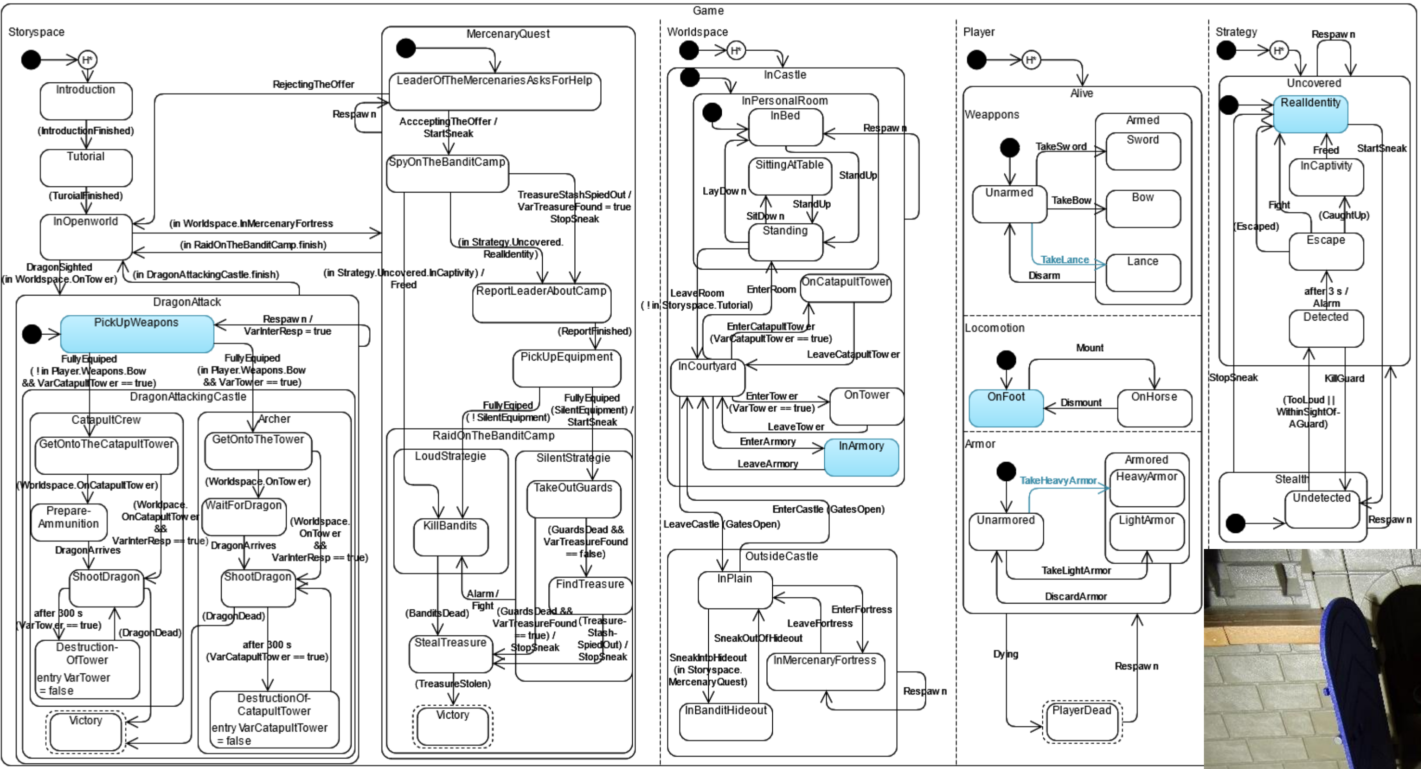
P.P. – Storyline: Pick Up Weapons



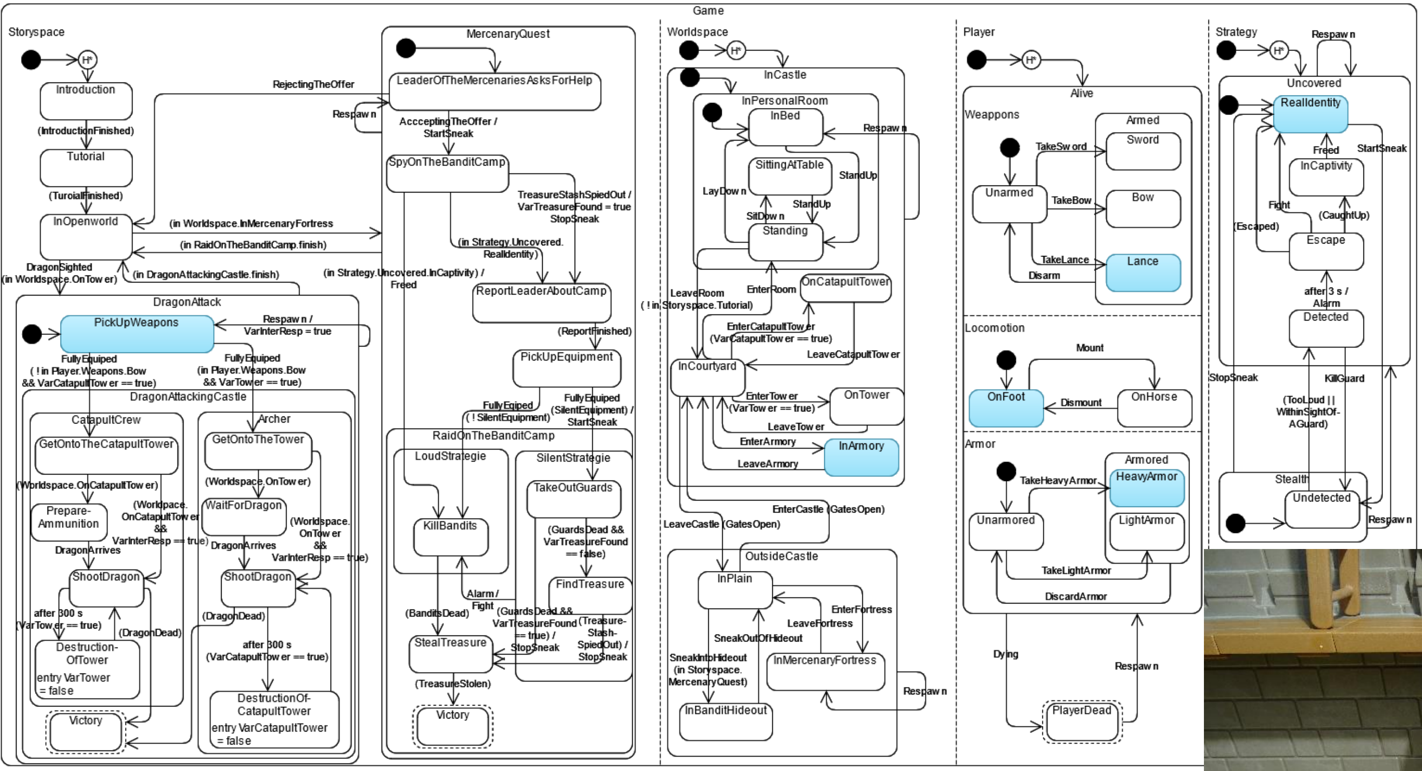
P.P. – Storyline: Pick Up Weapons

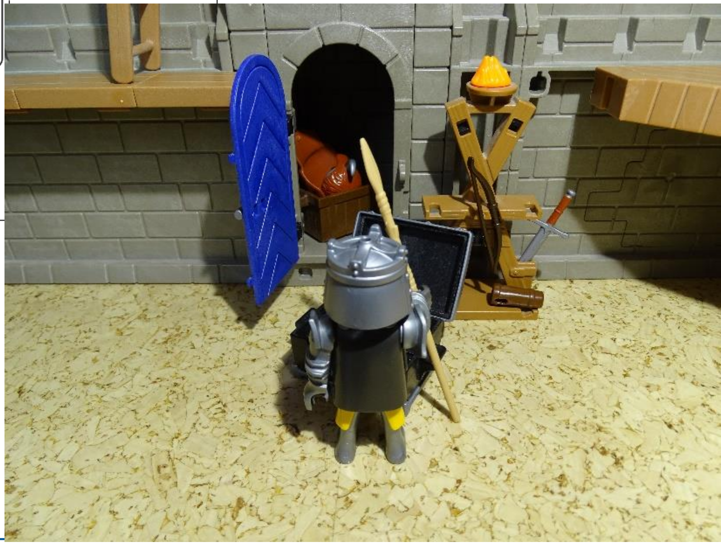


P.P. – Storyline: Pick Up Weapons

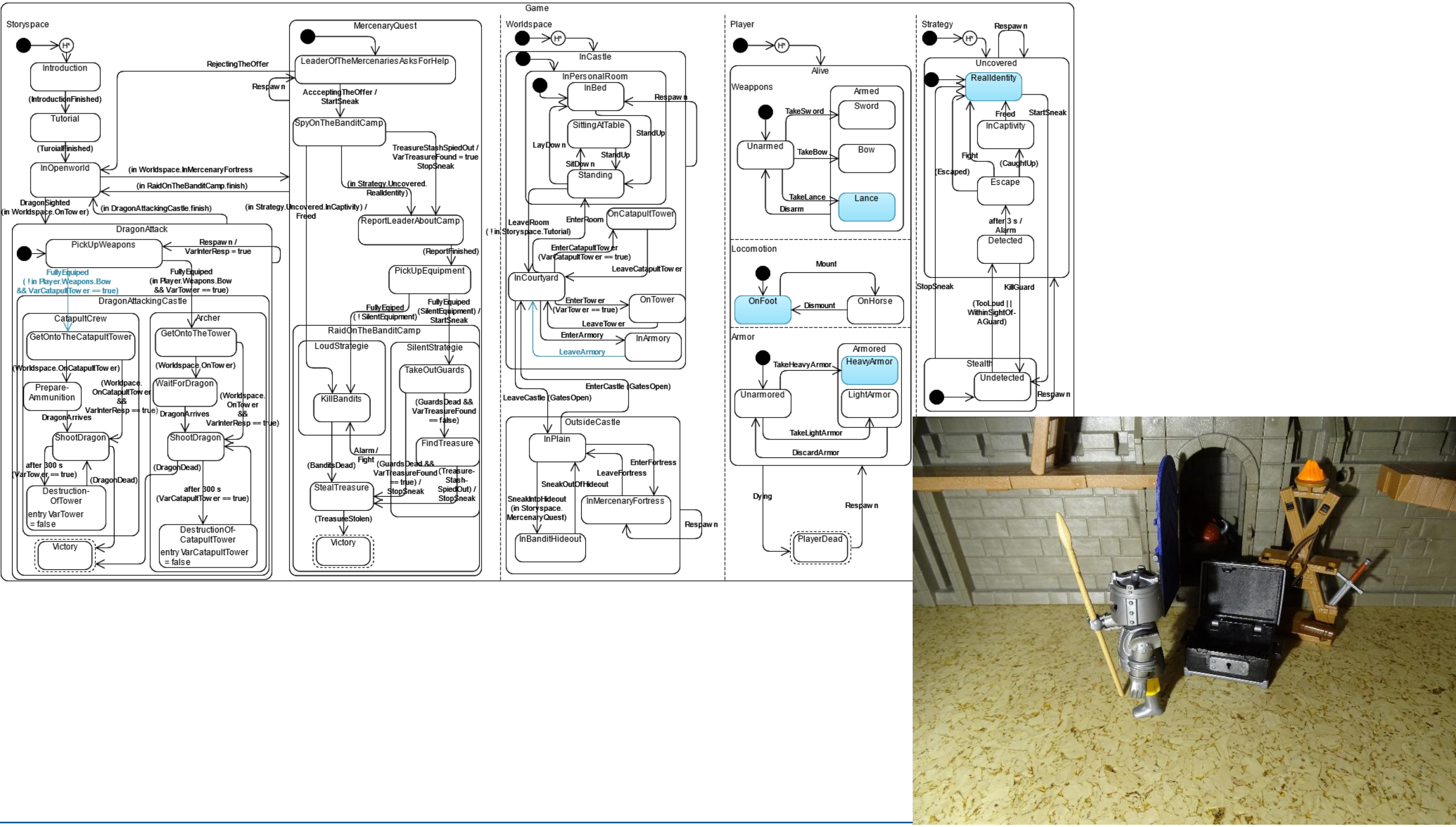


P.P. – Storyline: Pick Up Weapons

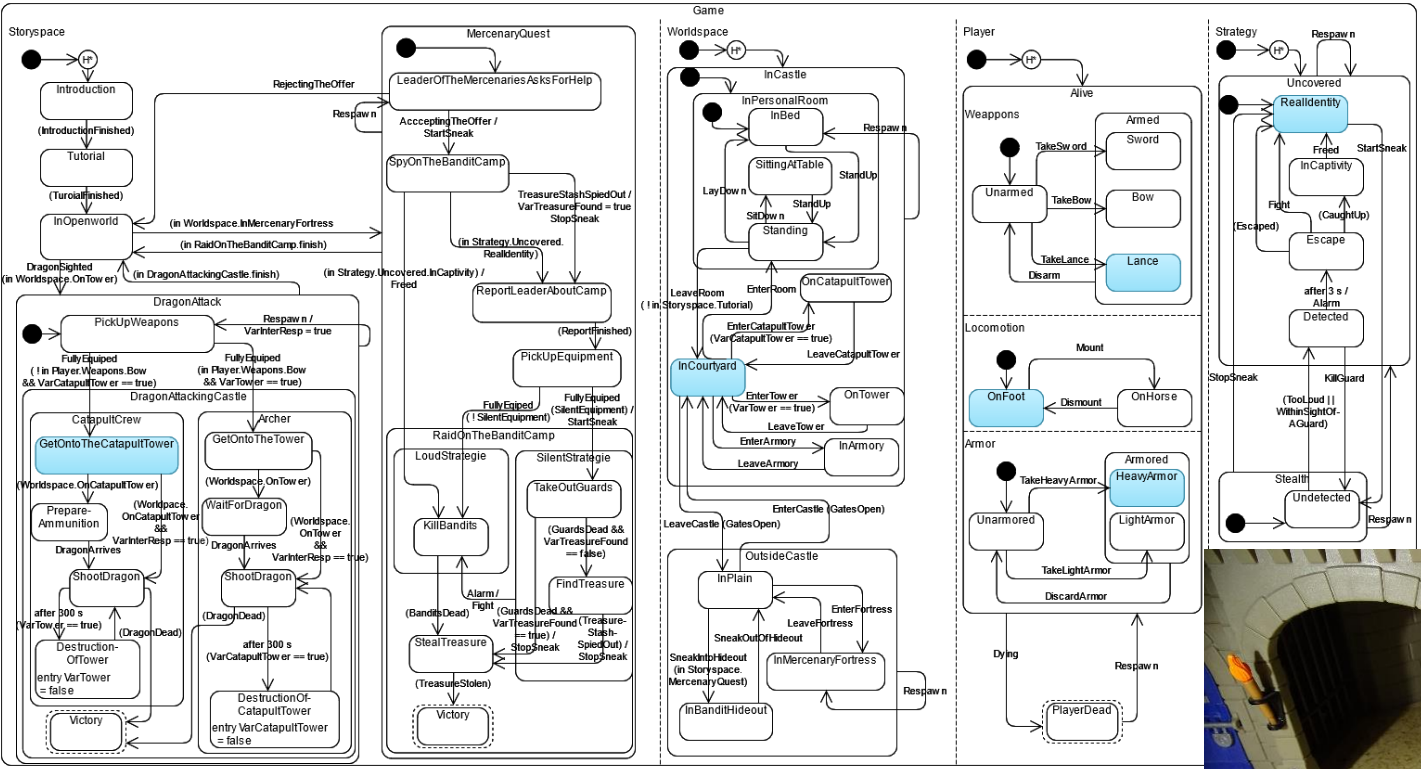




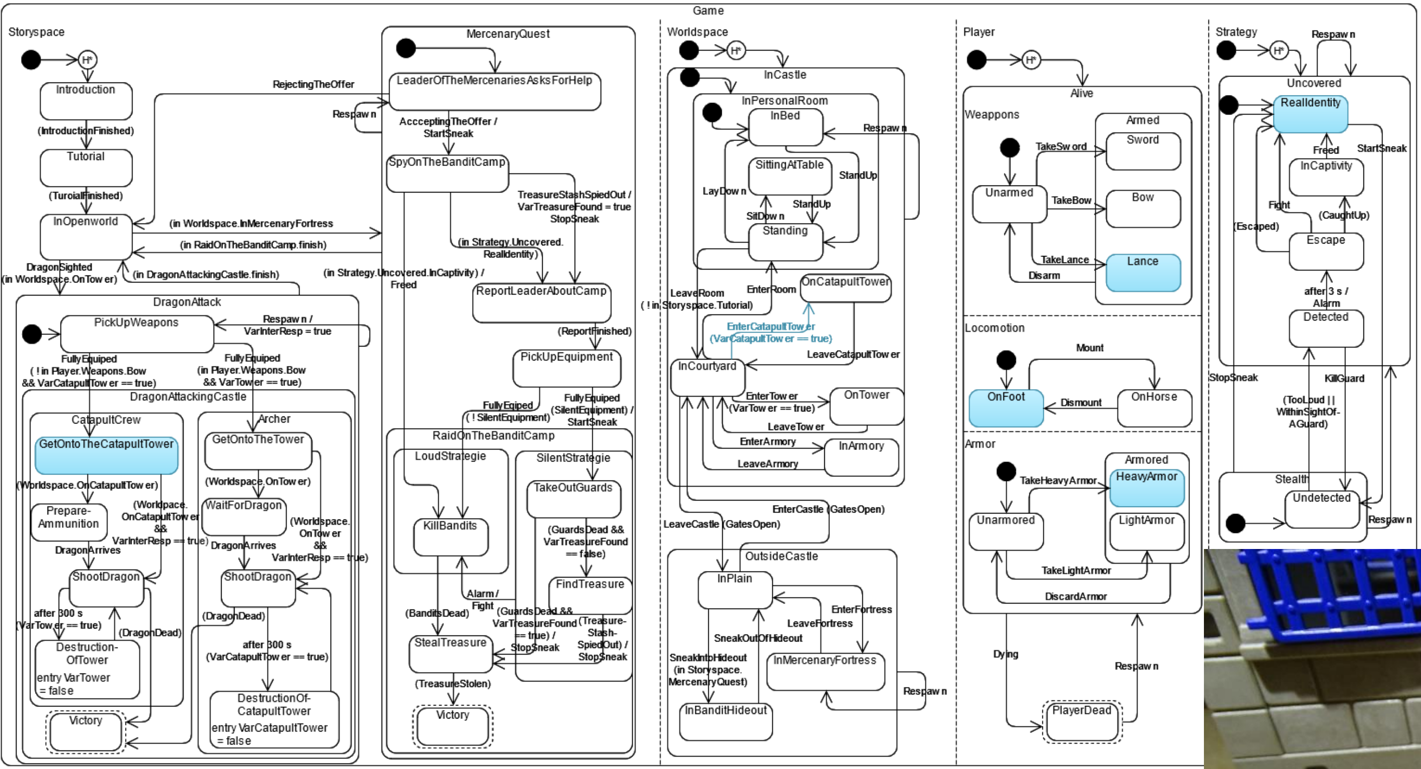
P.P. – Storyline: Pick Up Weapons



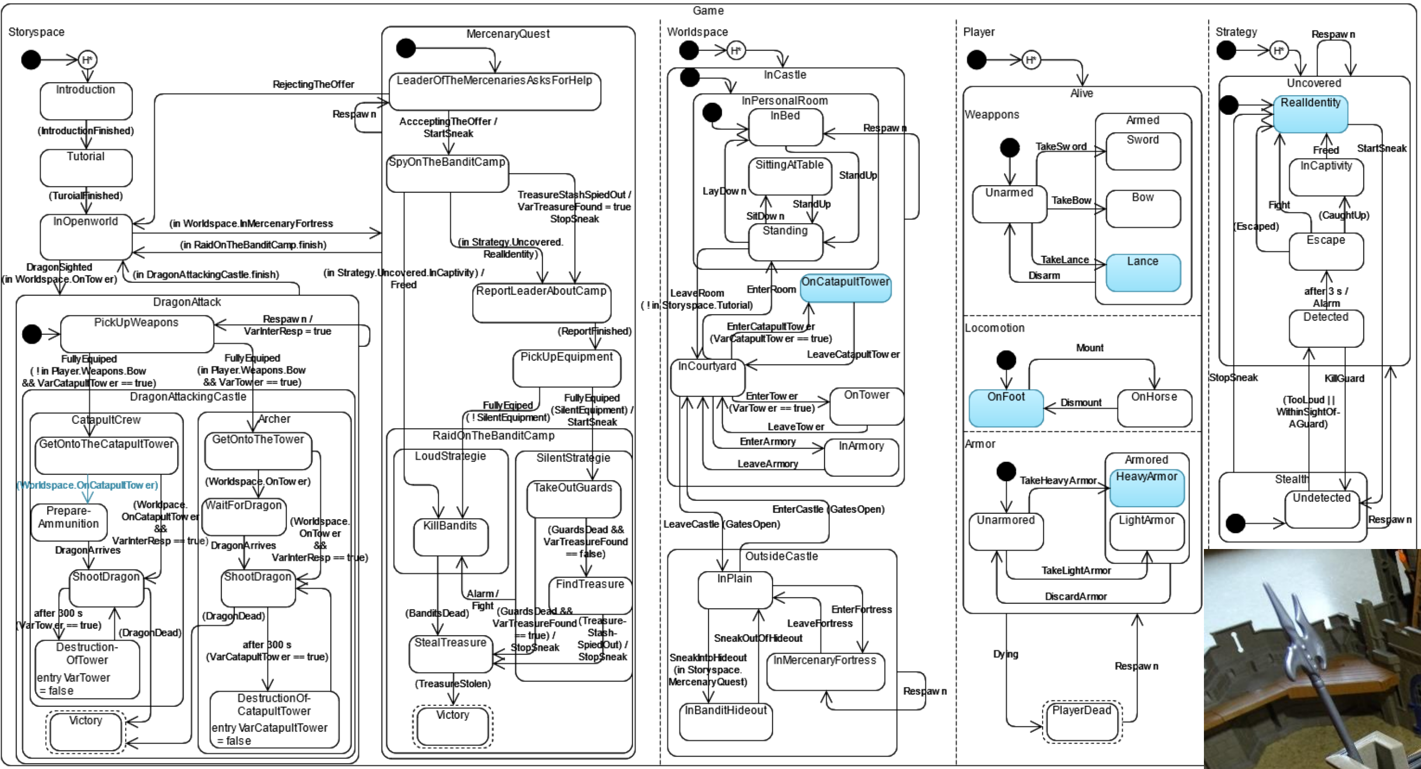
P.P. – Storyline: Get onto Tower

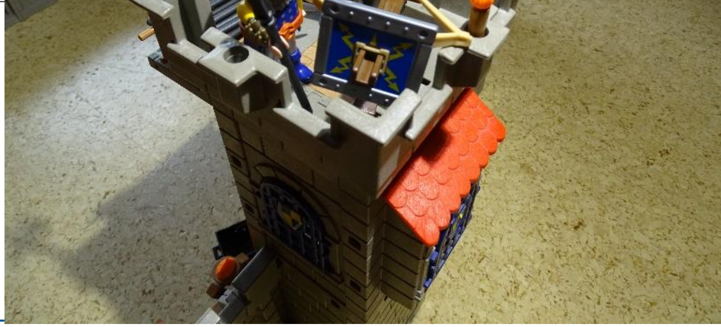


P.P. – Storyline: Get onto Tower

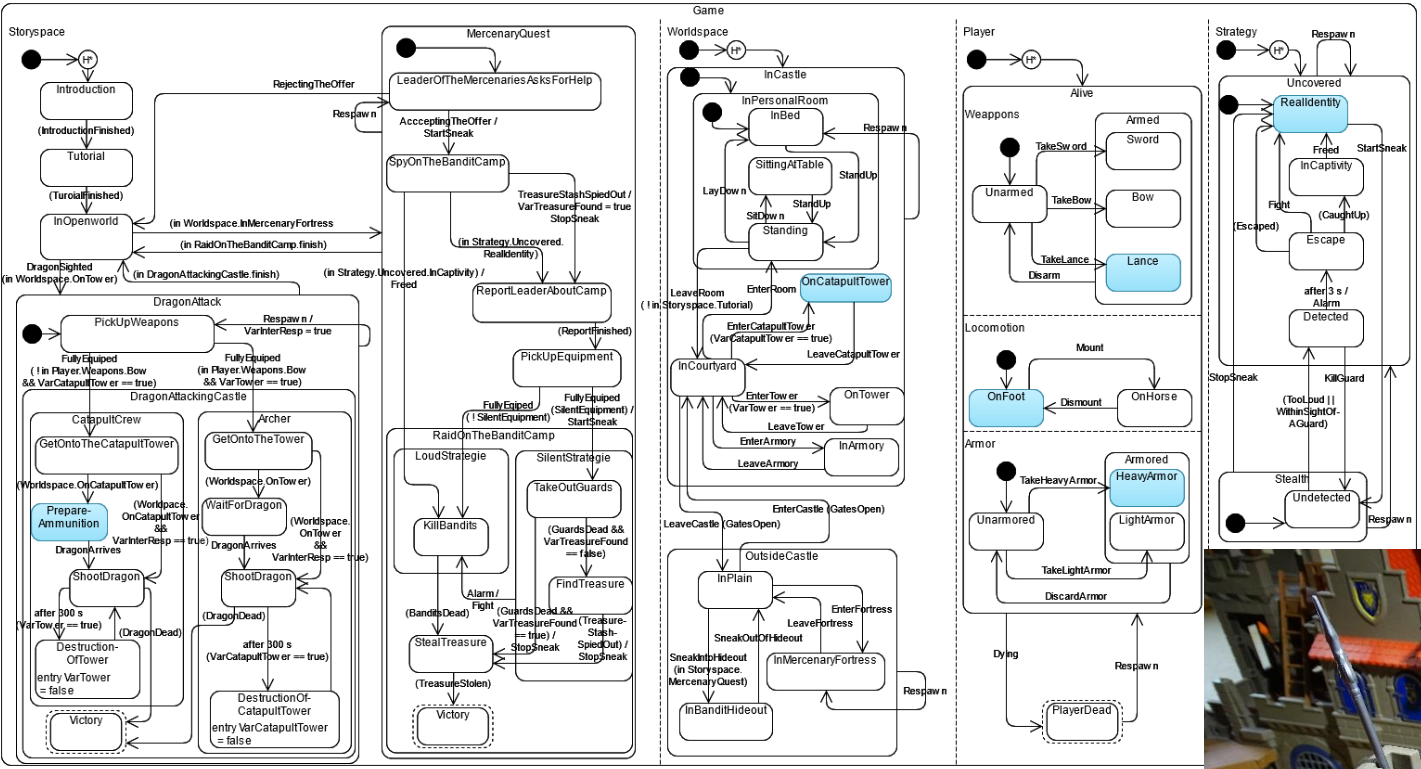


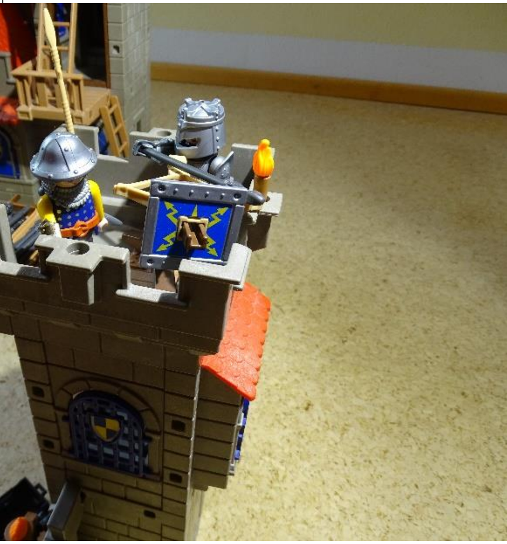
P.P. – Storyline: Get onto Tower



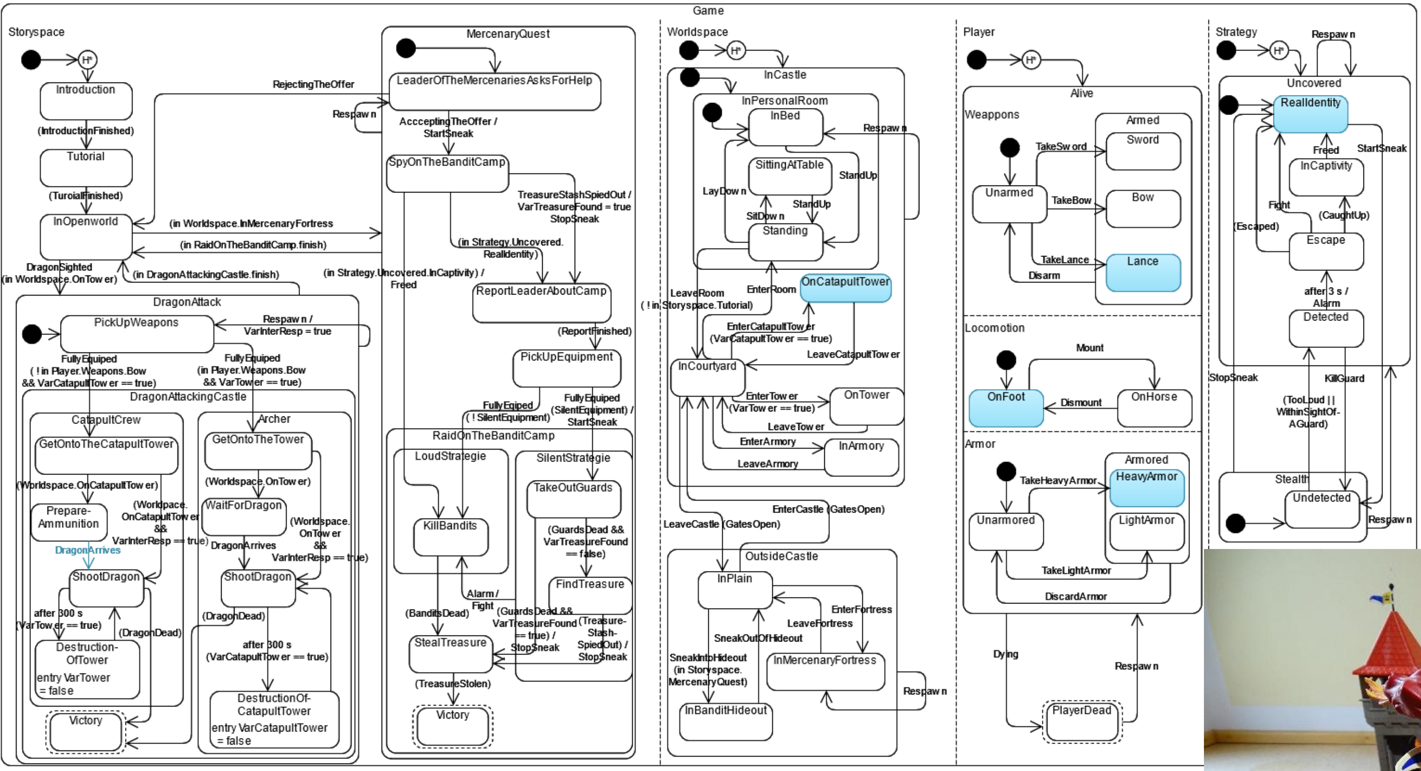


P.P. – Storyline: Prepare Ammunition

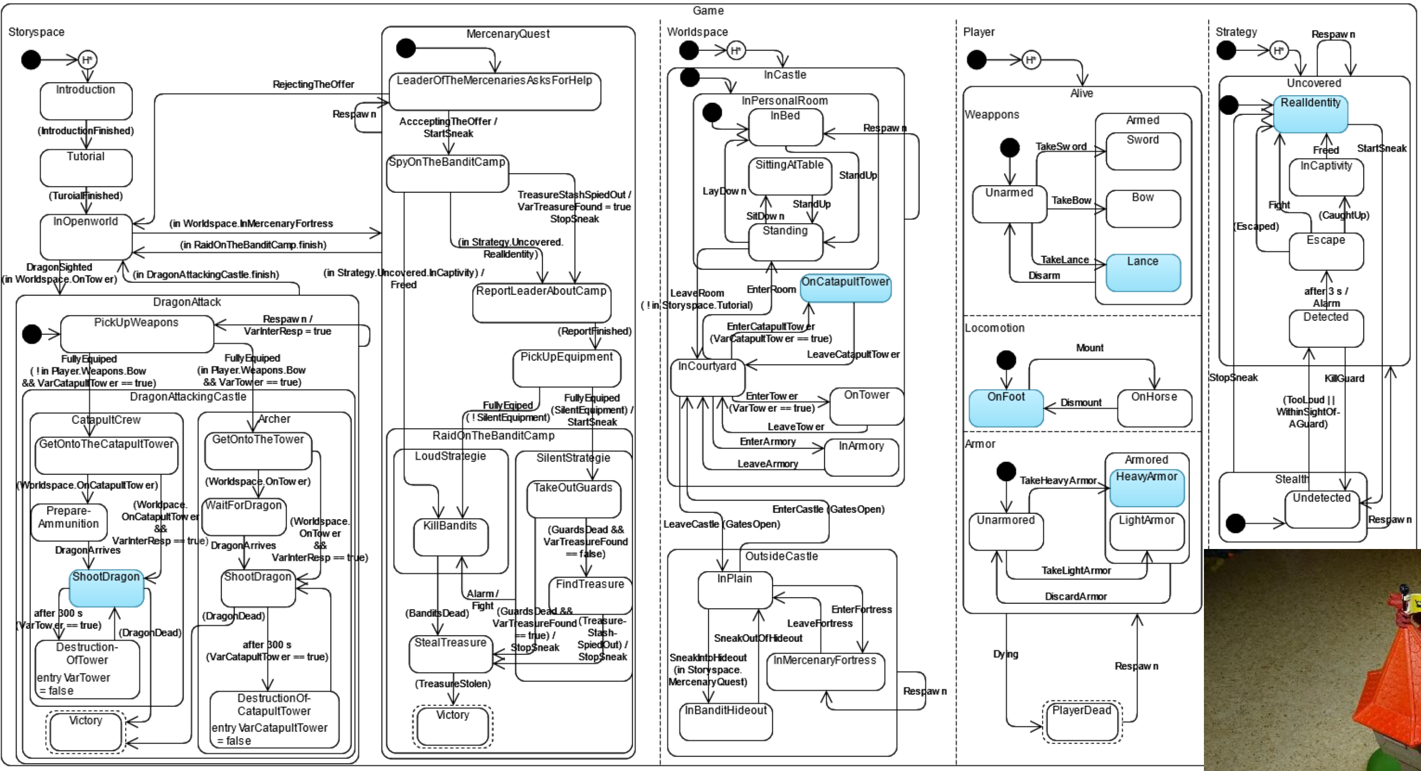


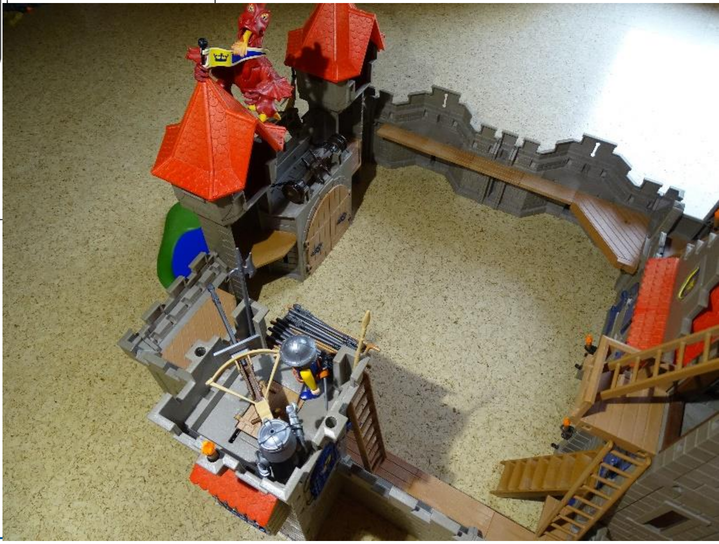


P.P. – Storyline: Dragon Arrives

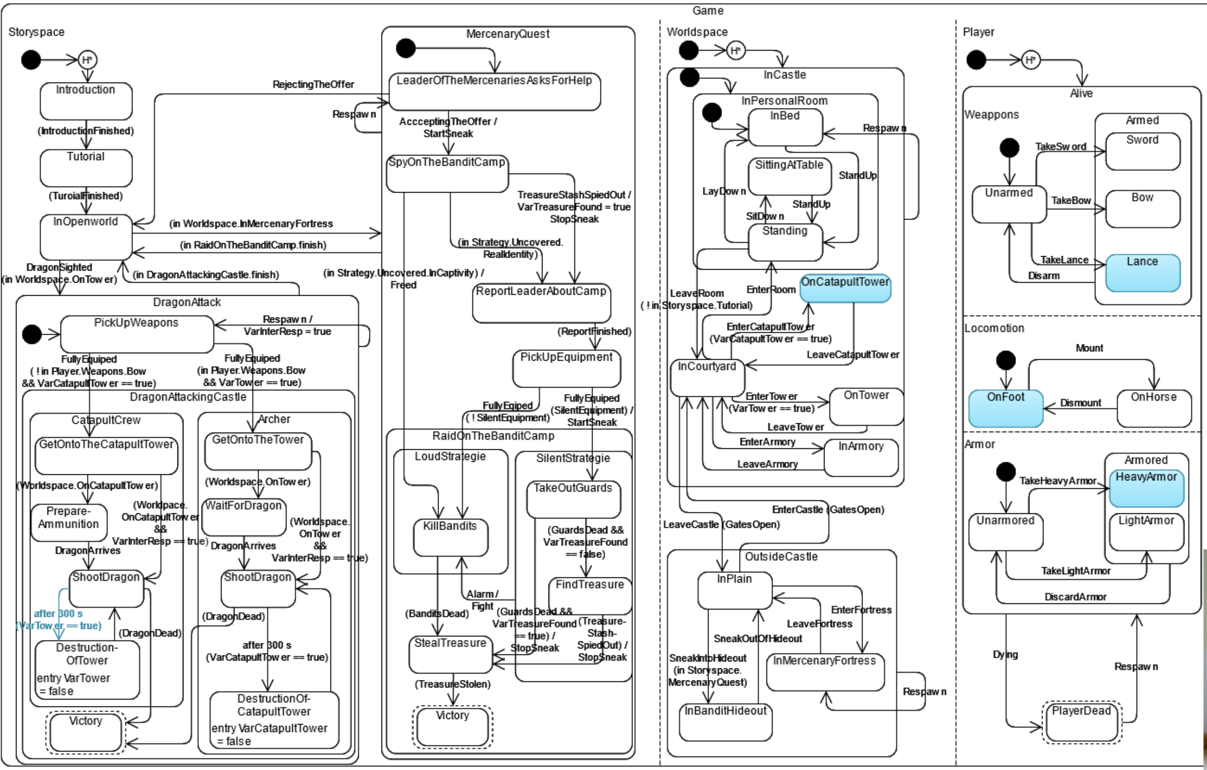


P.P. – Storyline: Shoot Dragon

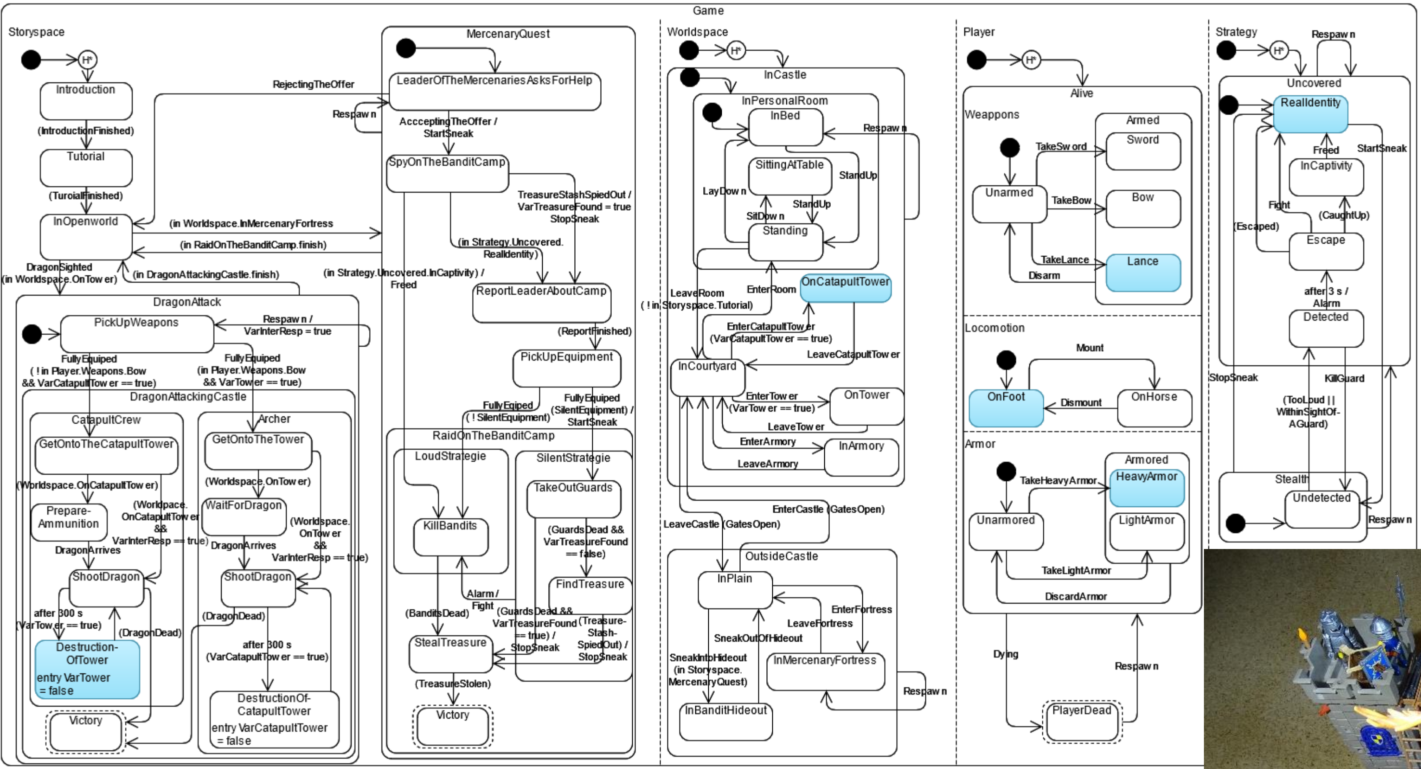




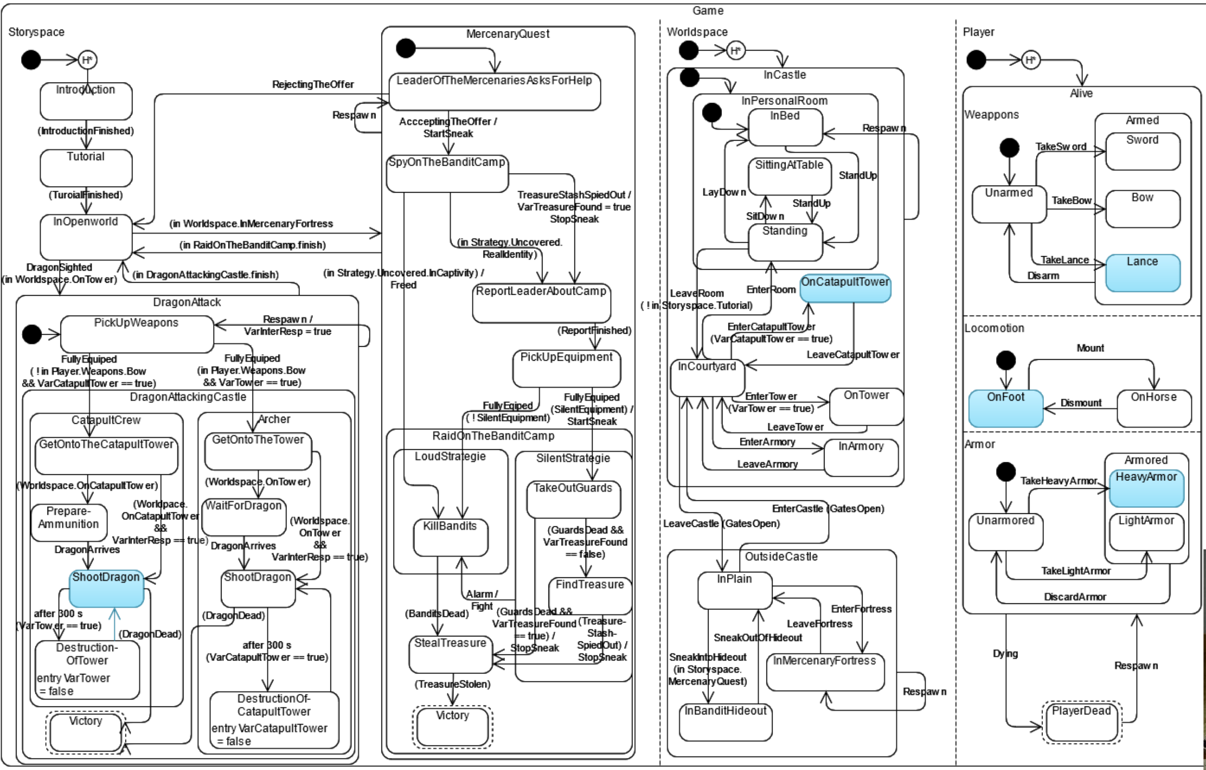
P.P. – Storyline: Shoot Dragon

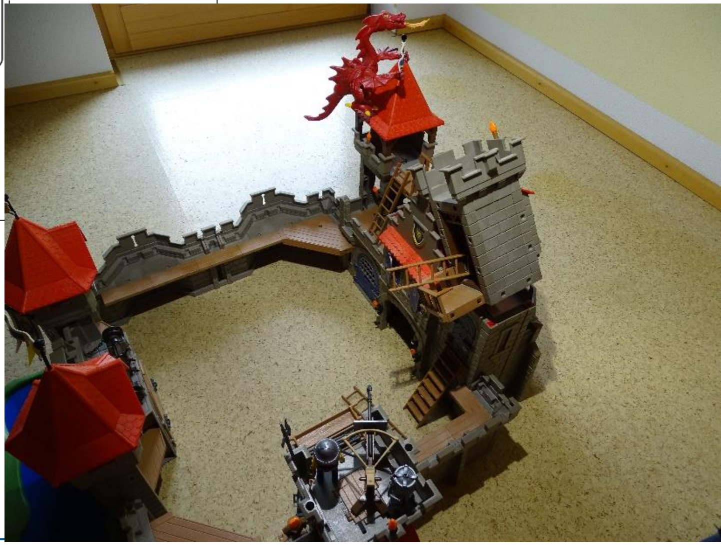


P.P. – Storyline: Dragon Destroys Tower

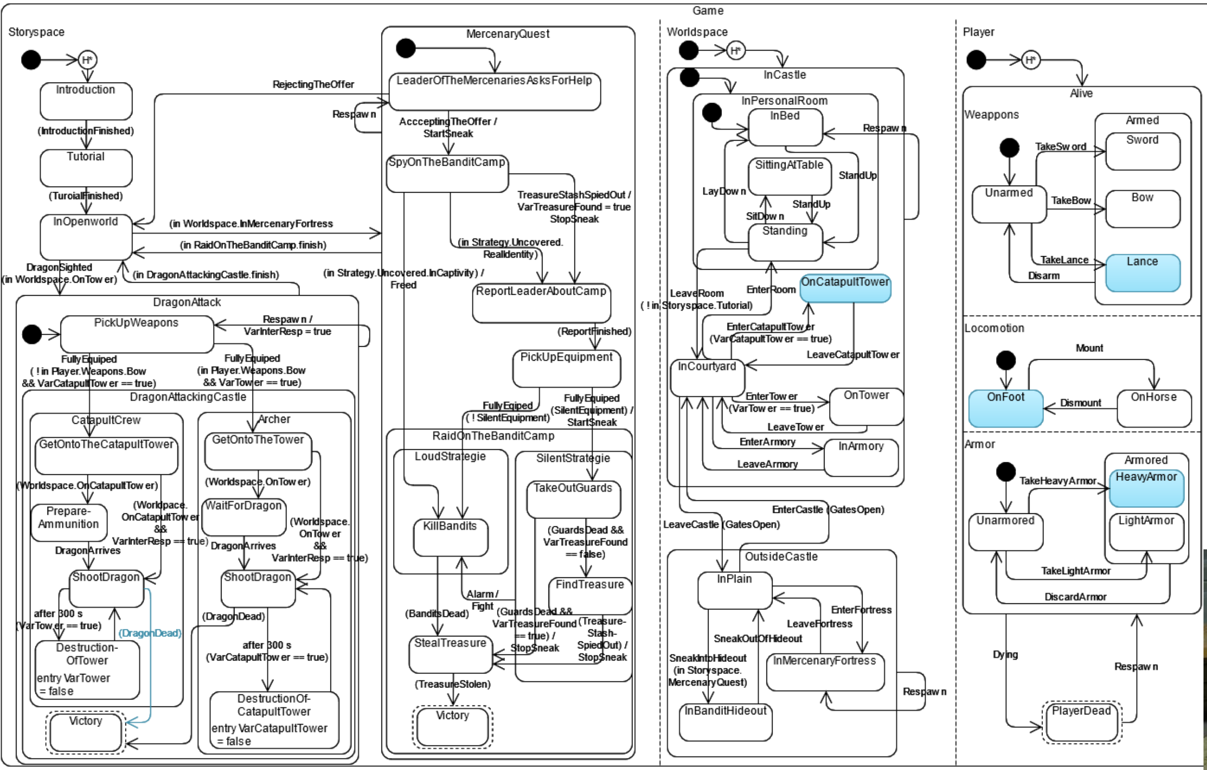


P.P. – Storyline: Shoot Dragon

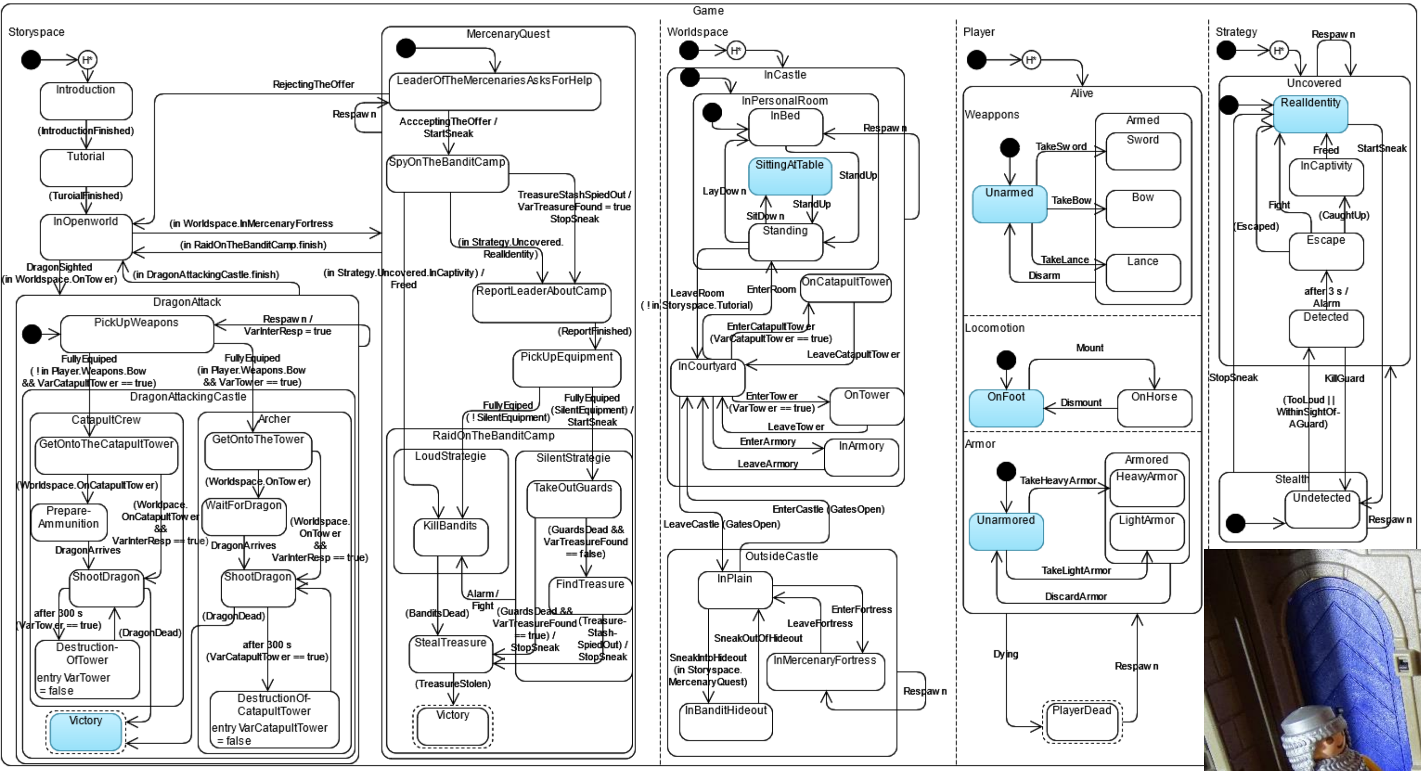




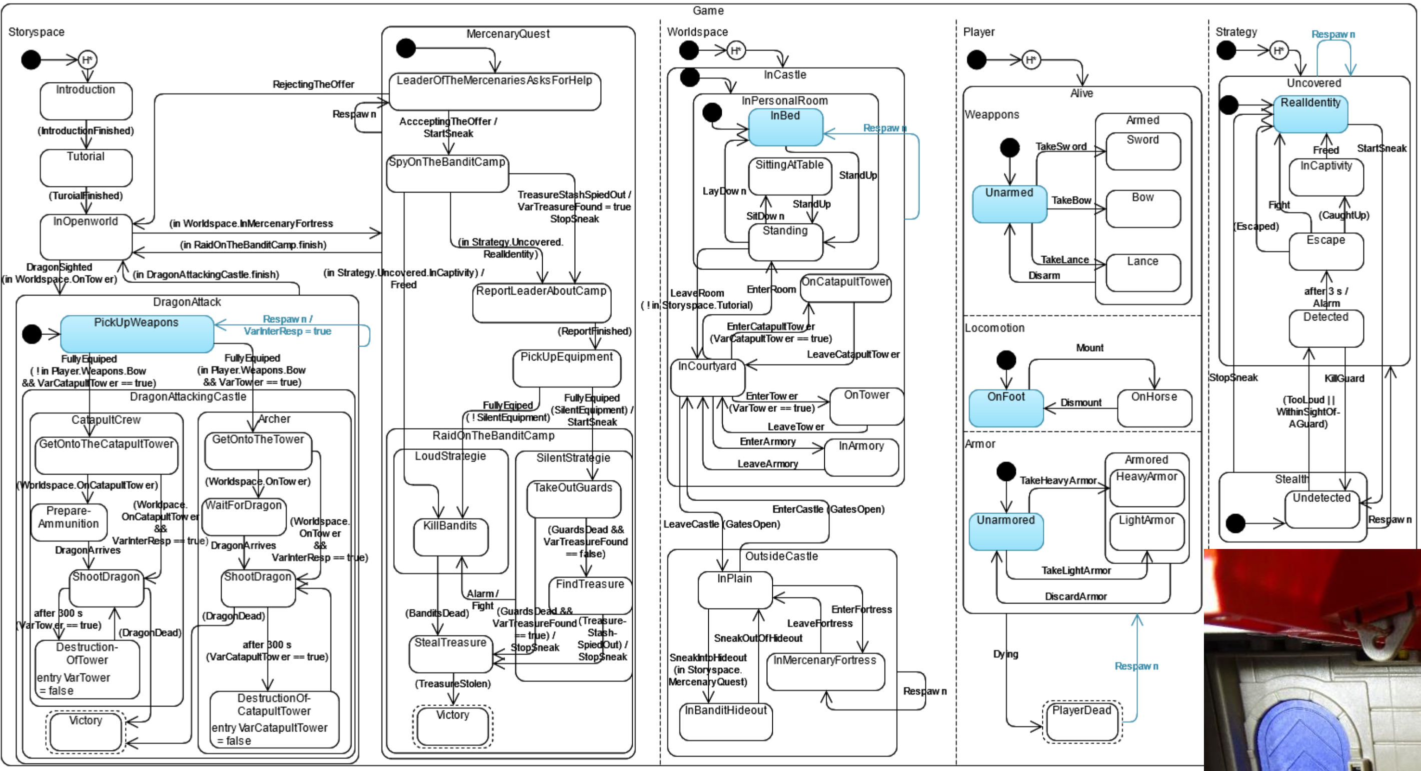
P.P. – Storyline: Dragon Dead



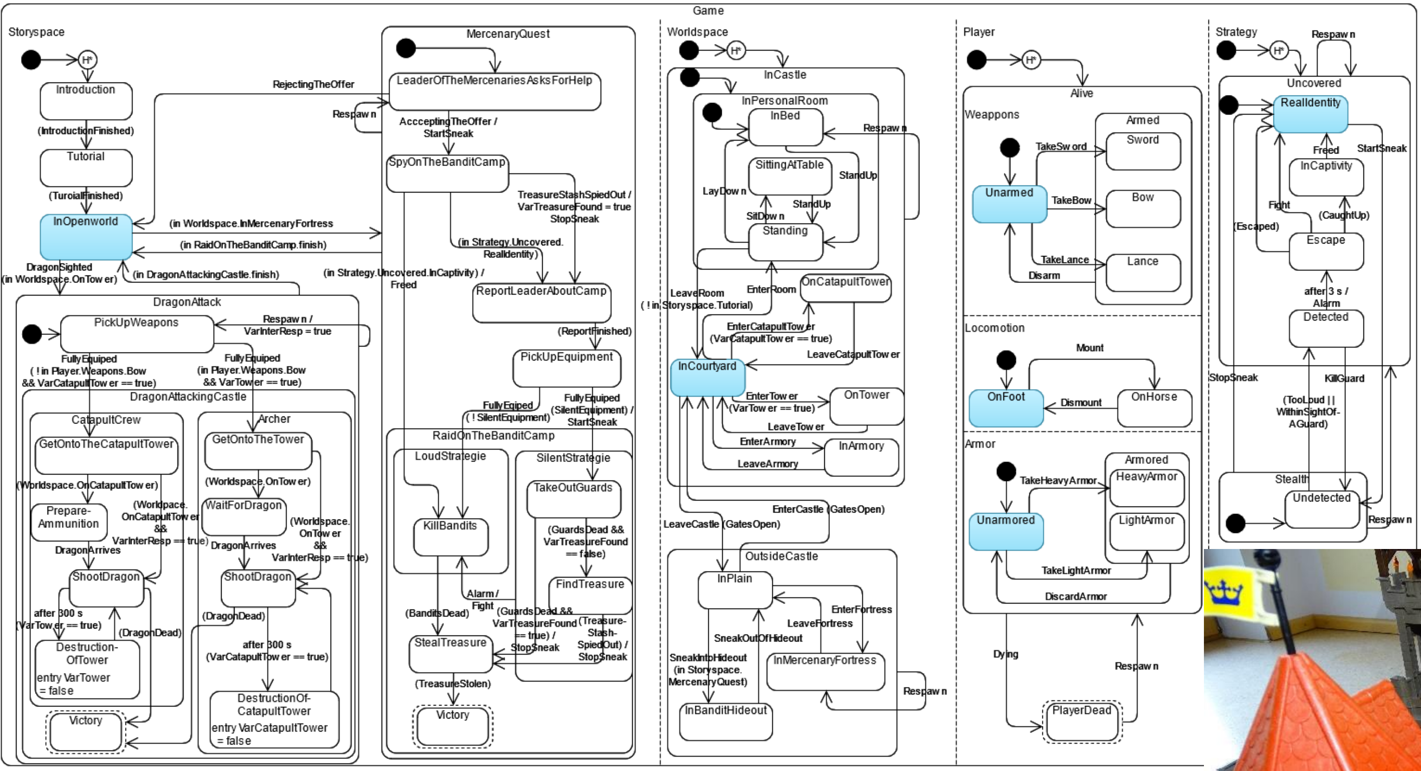
P.P. – Storyline: Dragon-Interrupt Victory



P.P. – Storyline: Dragon-Interrupt Respawn

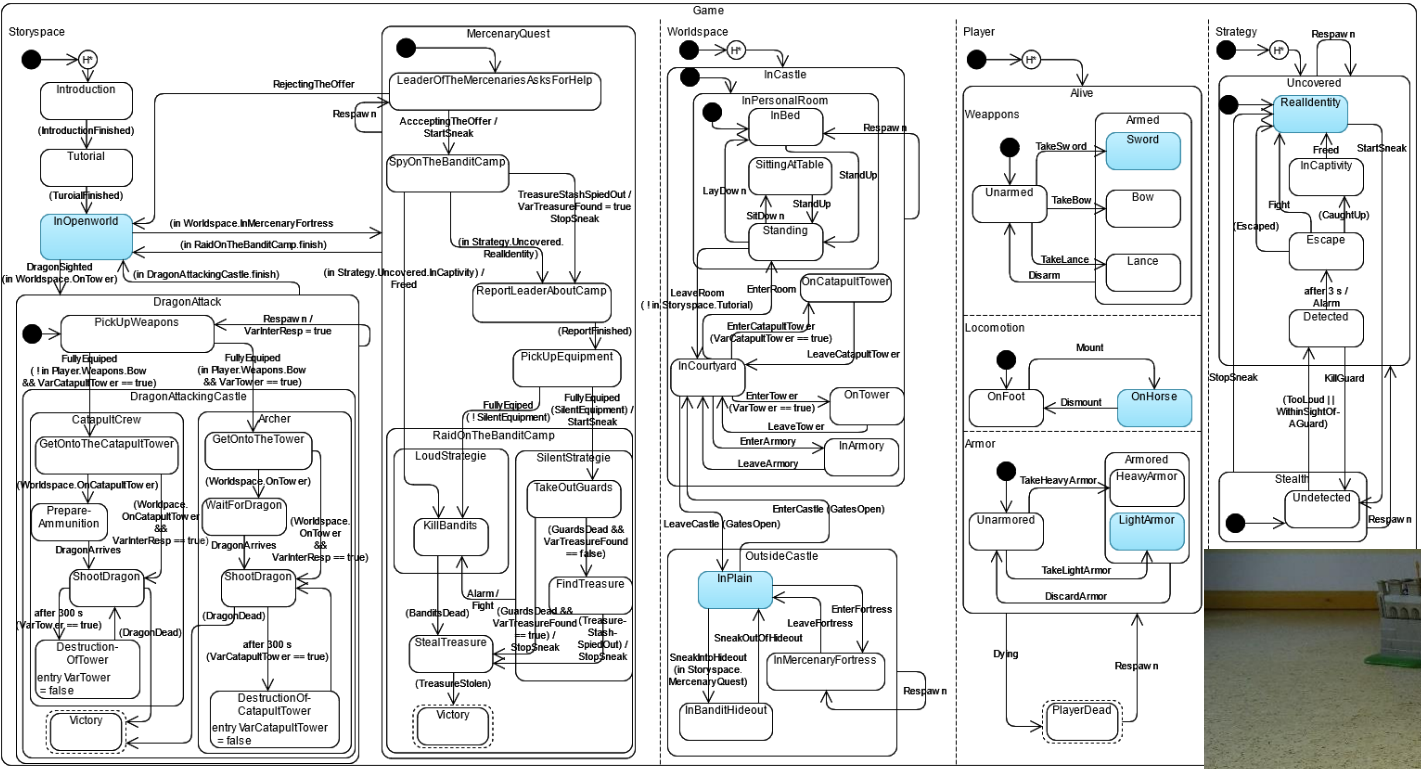


P.P. – Storyline: In Open World after Interrupt

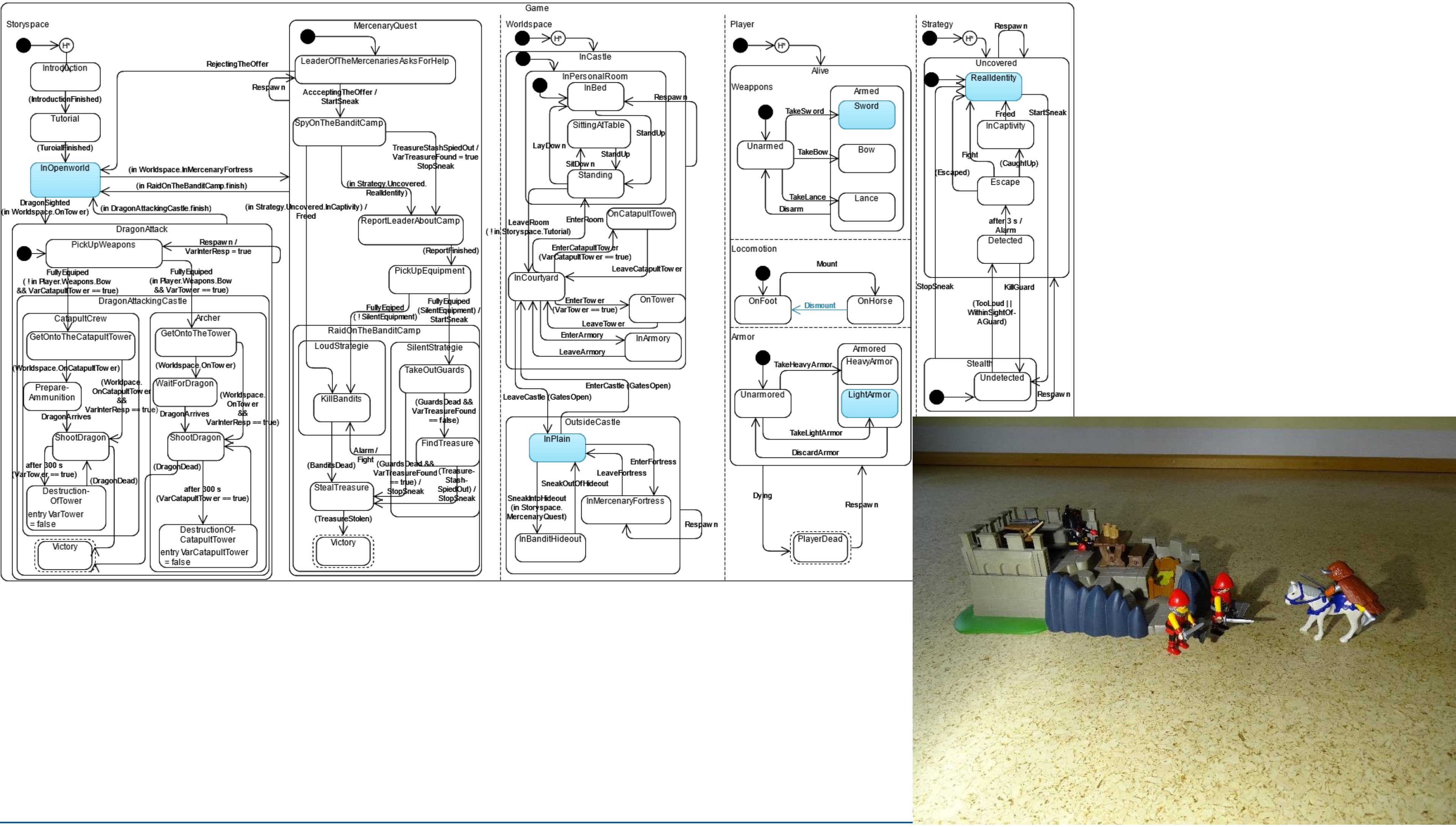


P.P. – Storyline: Mercenary Quest

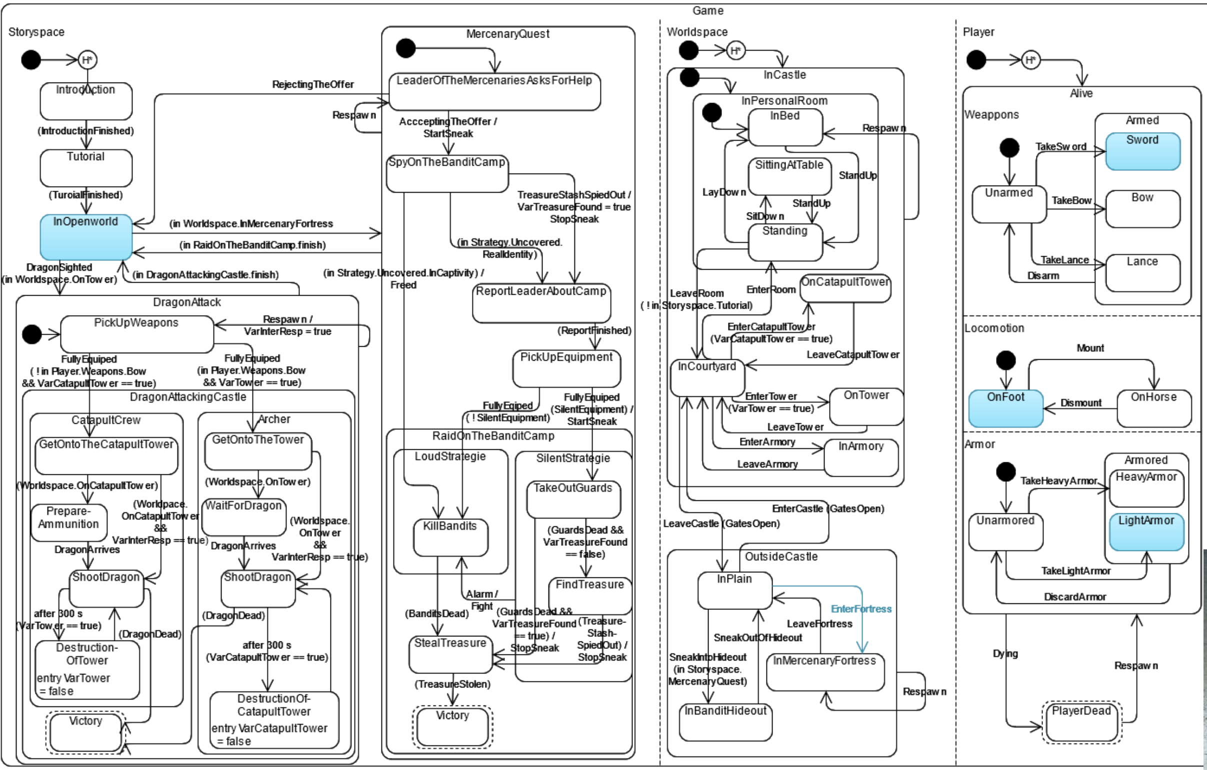
P.P. – Storyline: In Open World



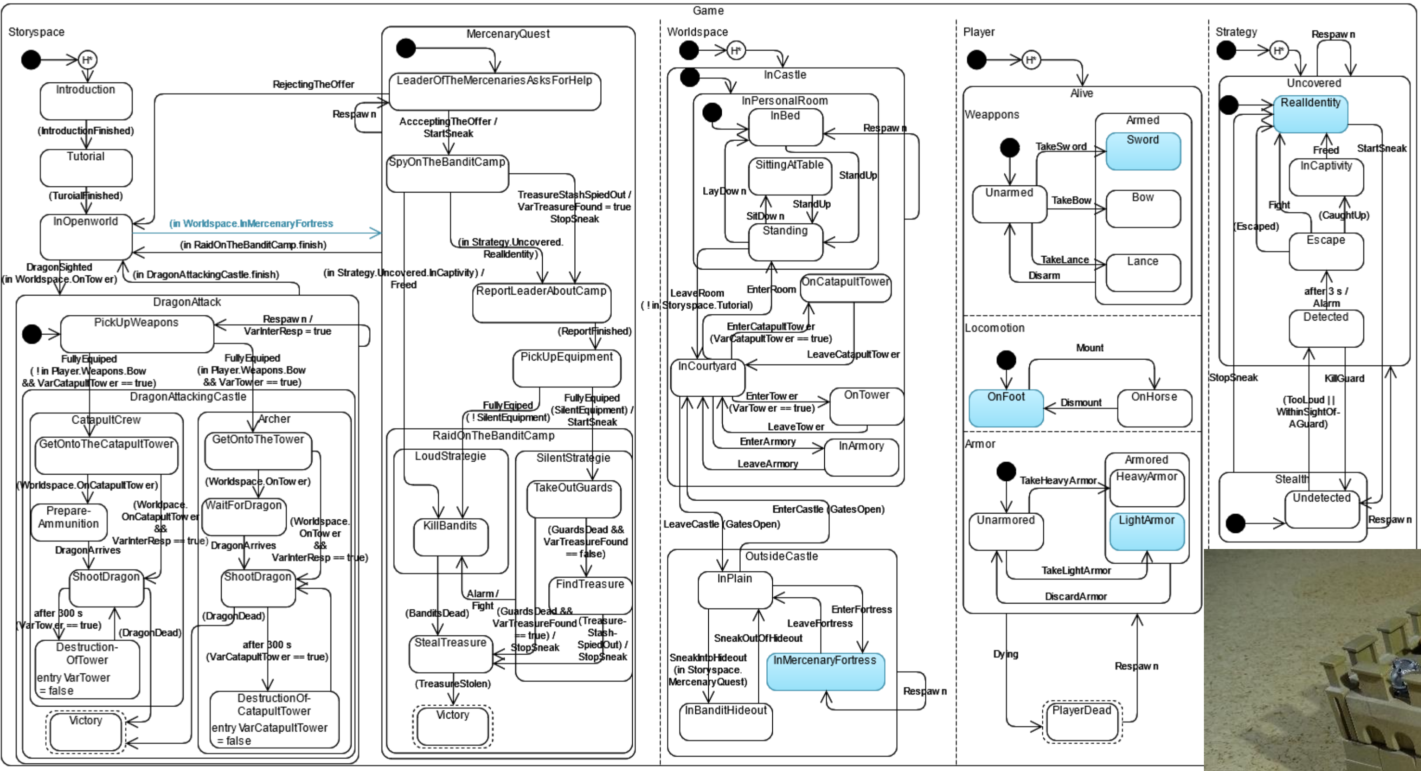
P.P. – Storyline: Entering Mercenary Fortress



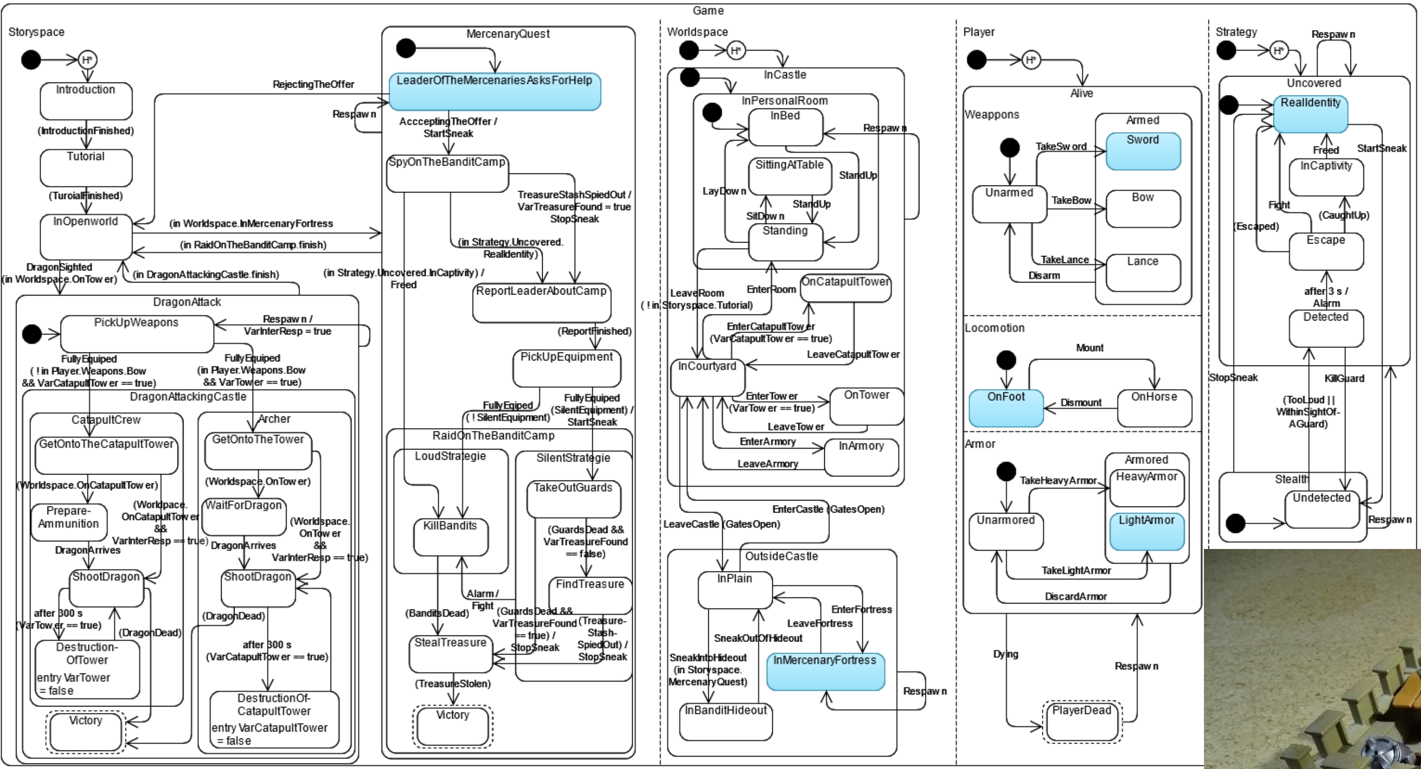
P.P. – Storyline: Entering Mercenary Fortress



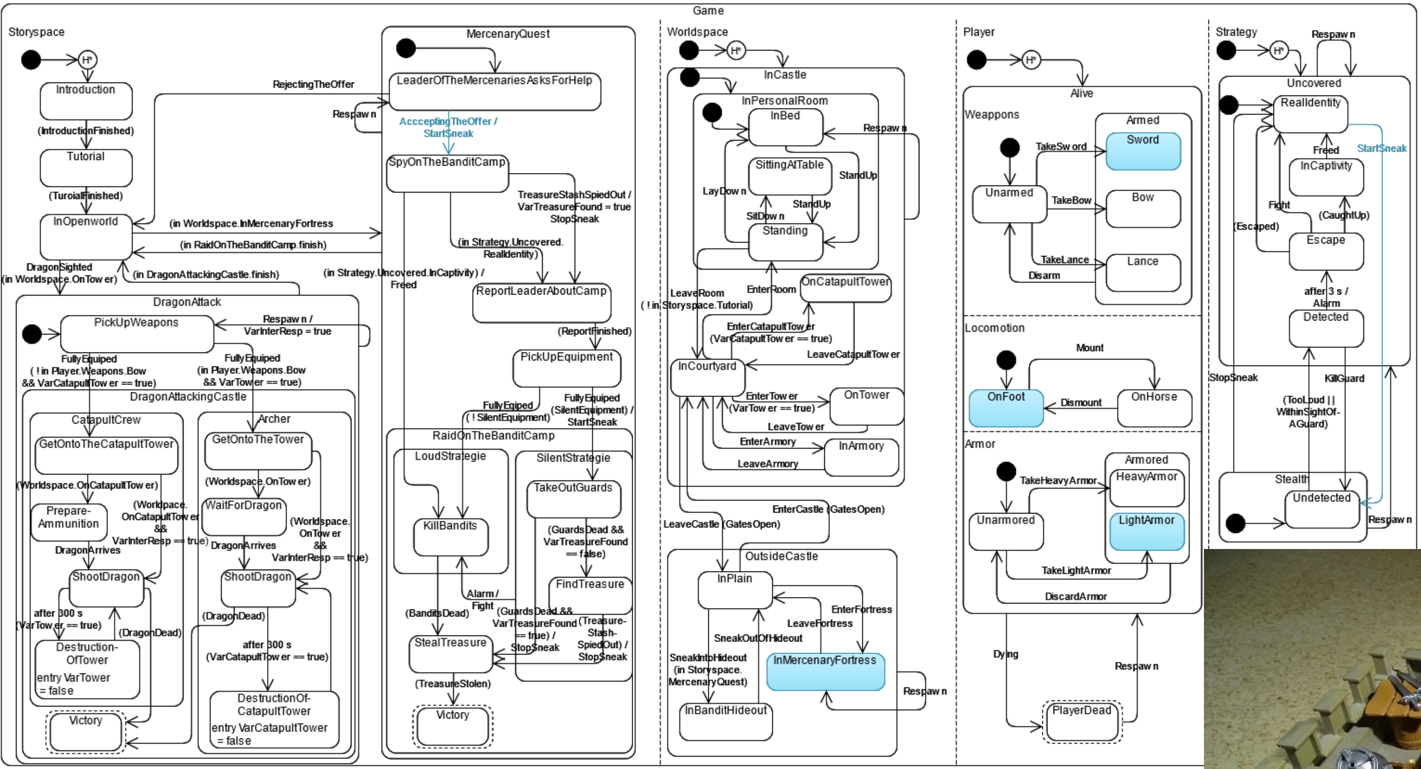
P.P. – Storyline: Entering Mercenary Fortress



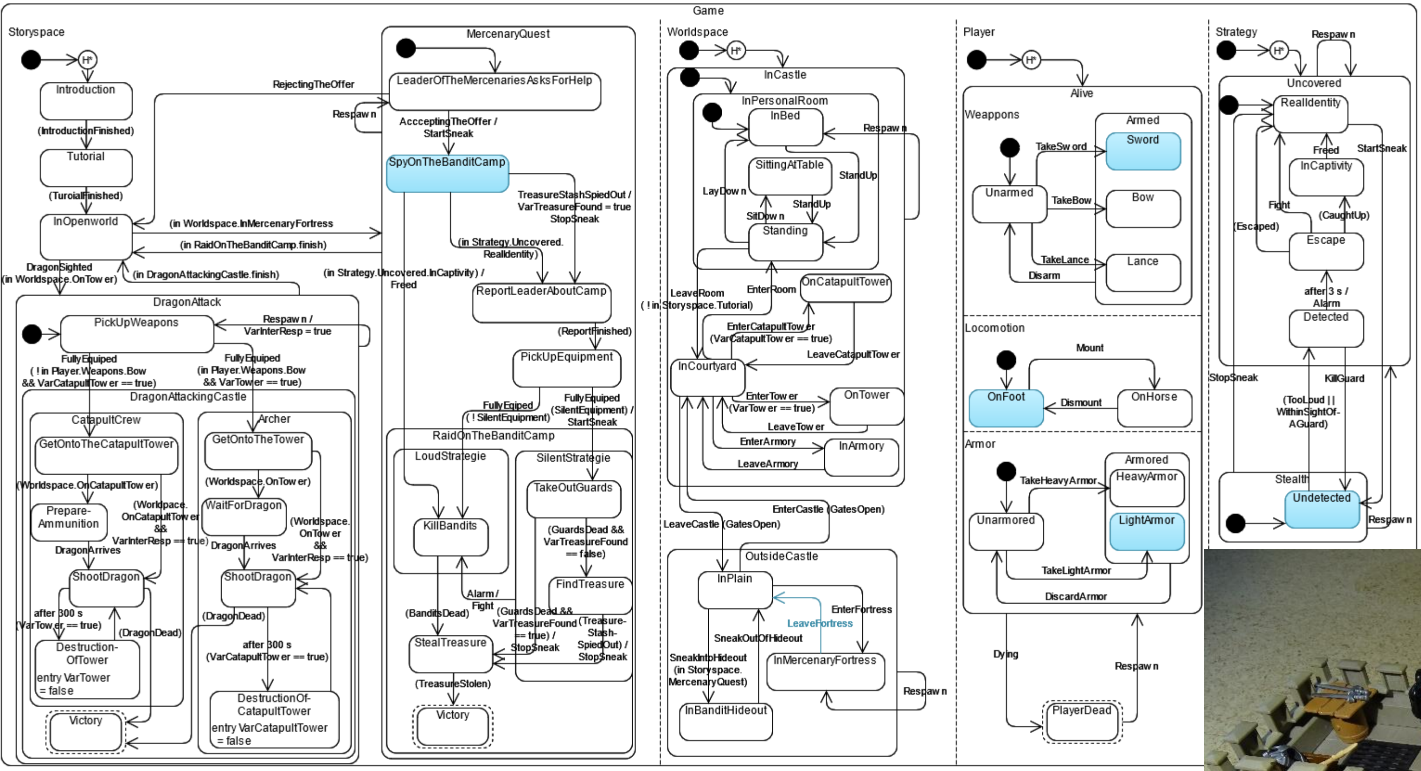
P.P. – Storyline: Leader Asks for Help



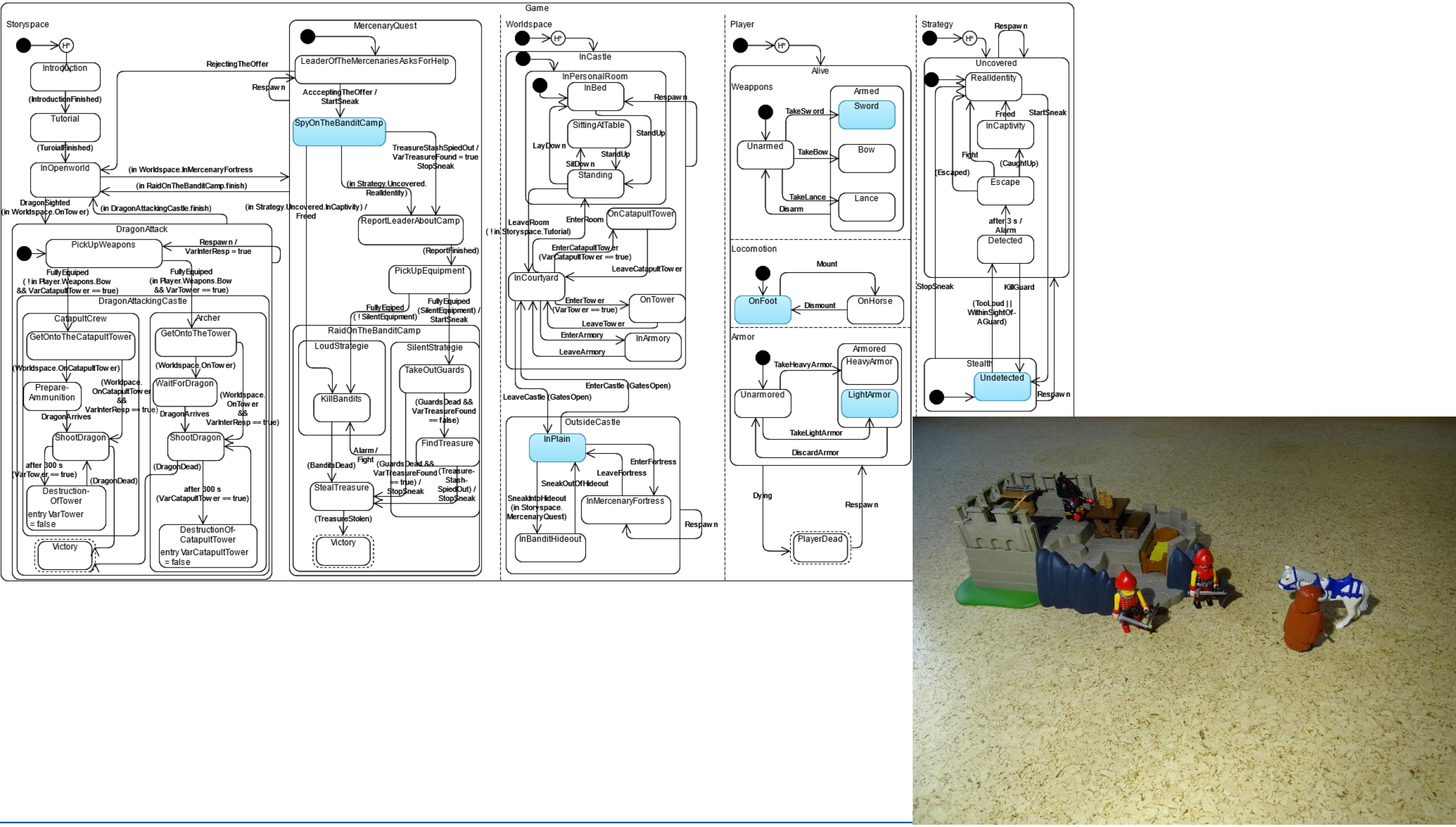
P.P. – Storyline: Accepting the Offer



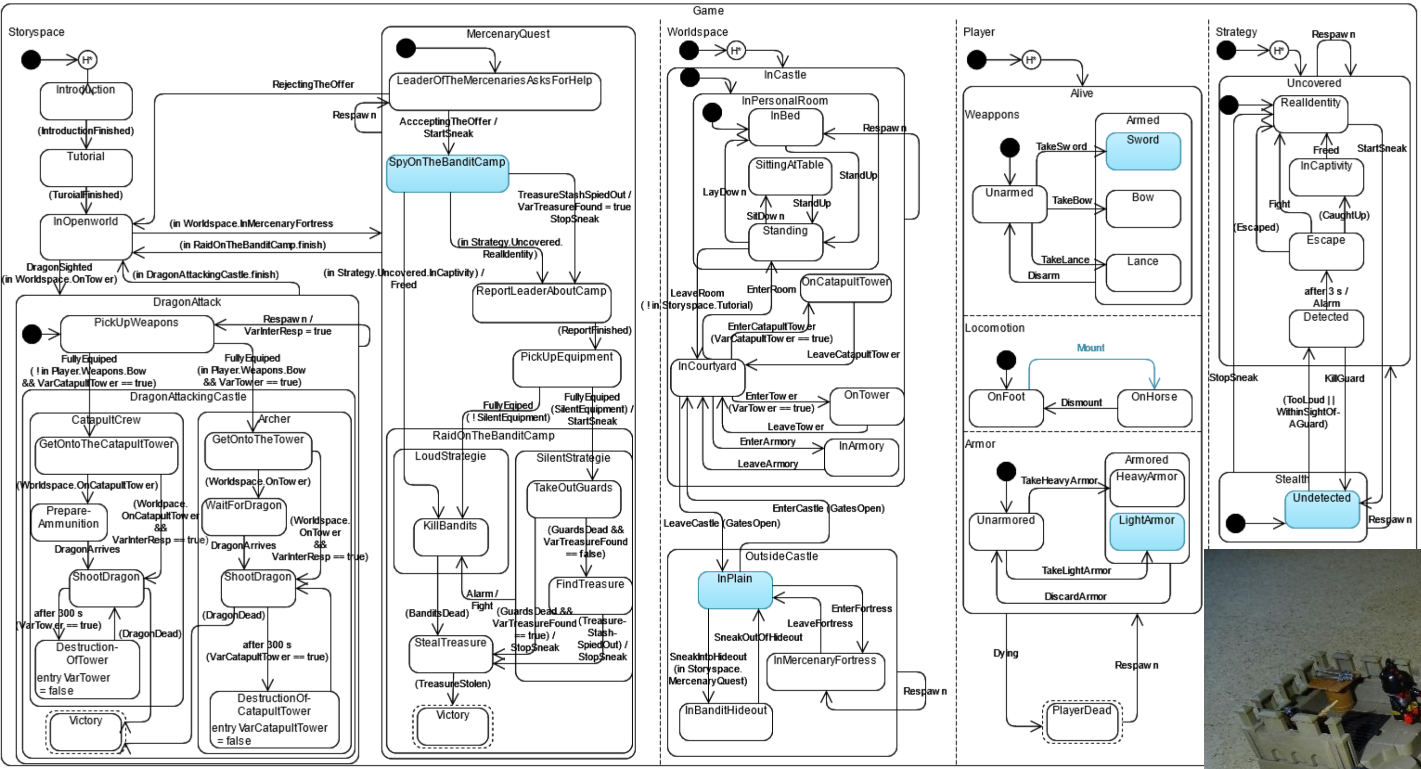
P.P. – Storyline: Spy on the Bandit Camp



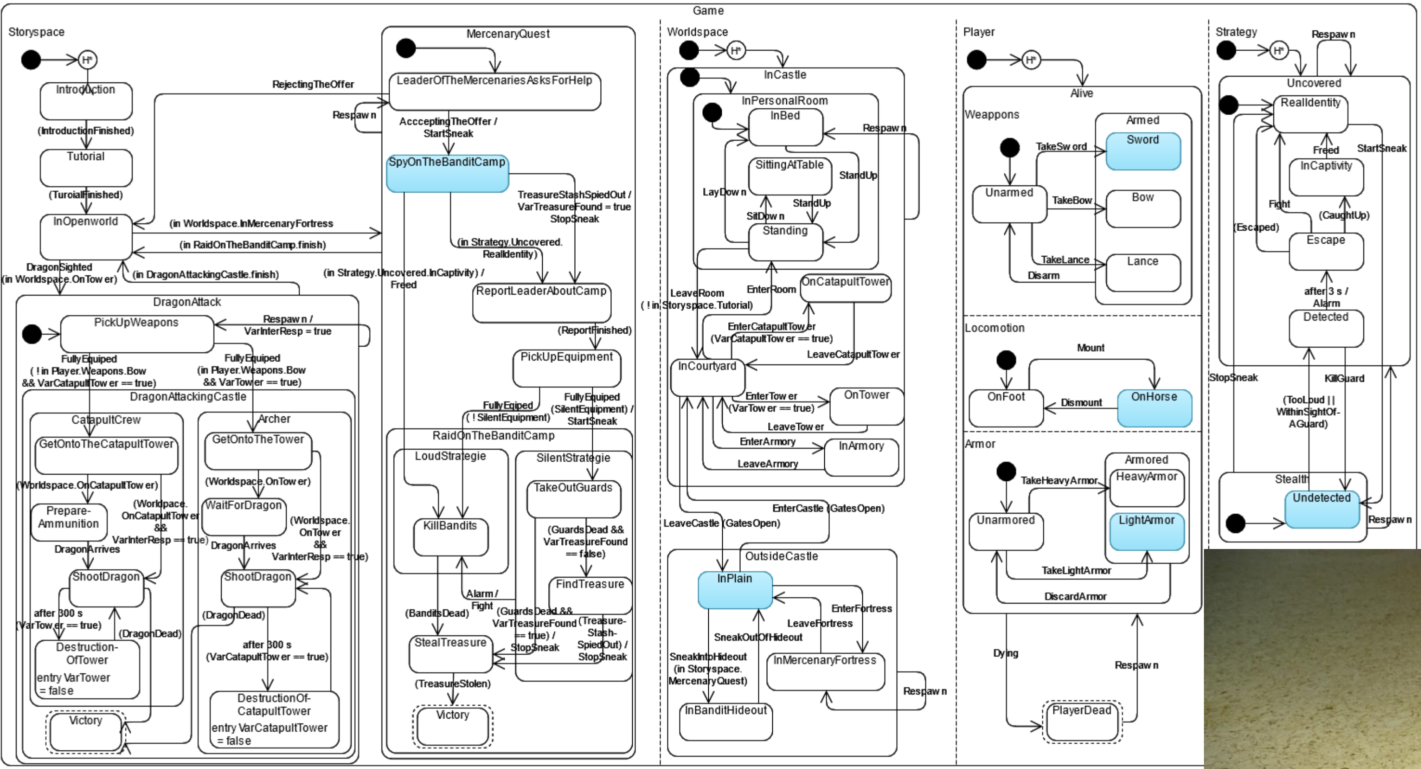
P.P. – Storyline: Spy on the Bandit Camp



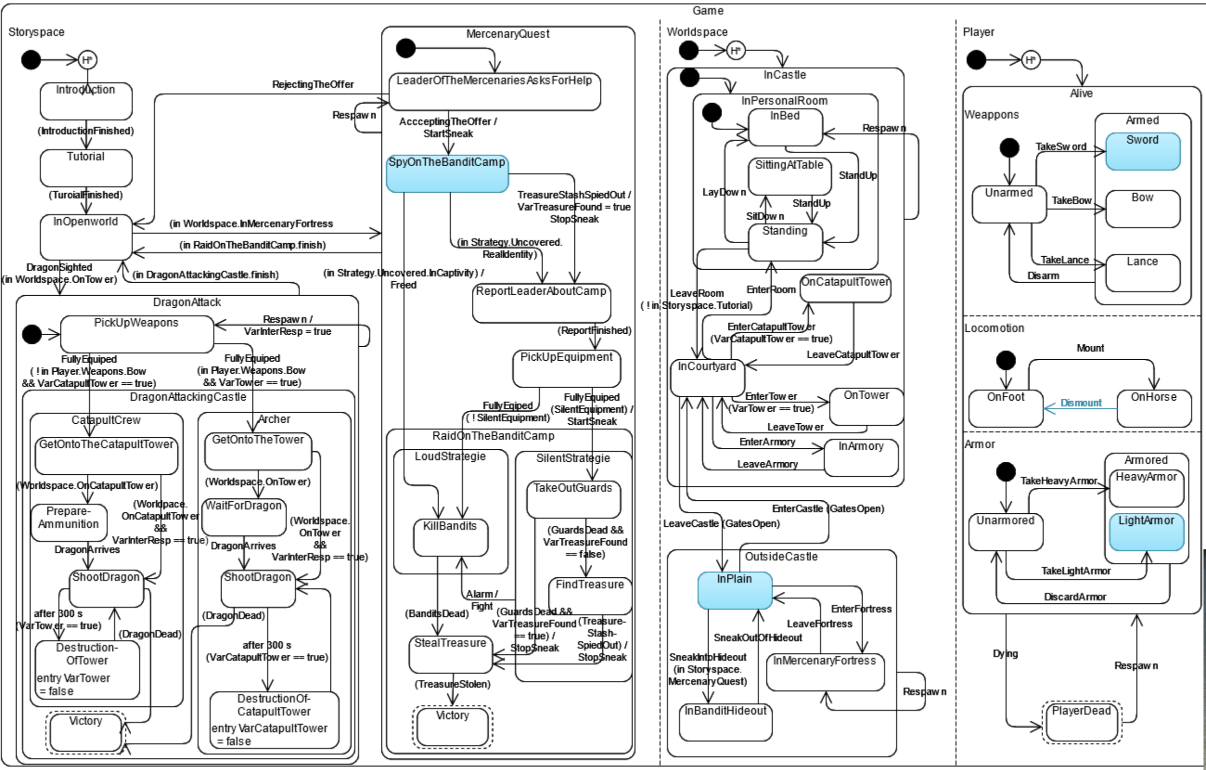
P.P. – Storyline: Spy on the Bandit Camp

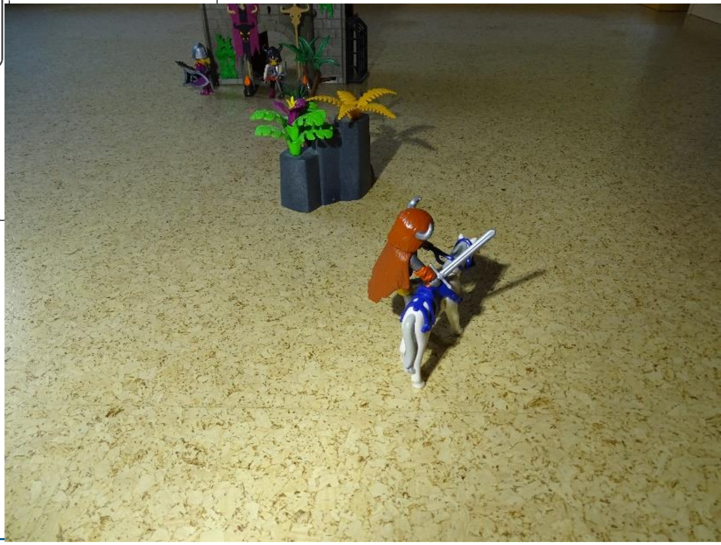


P.P. – Storyline: Spy on the Bandit Camp

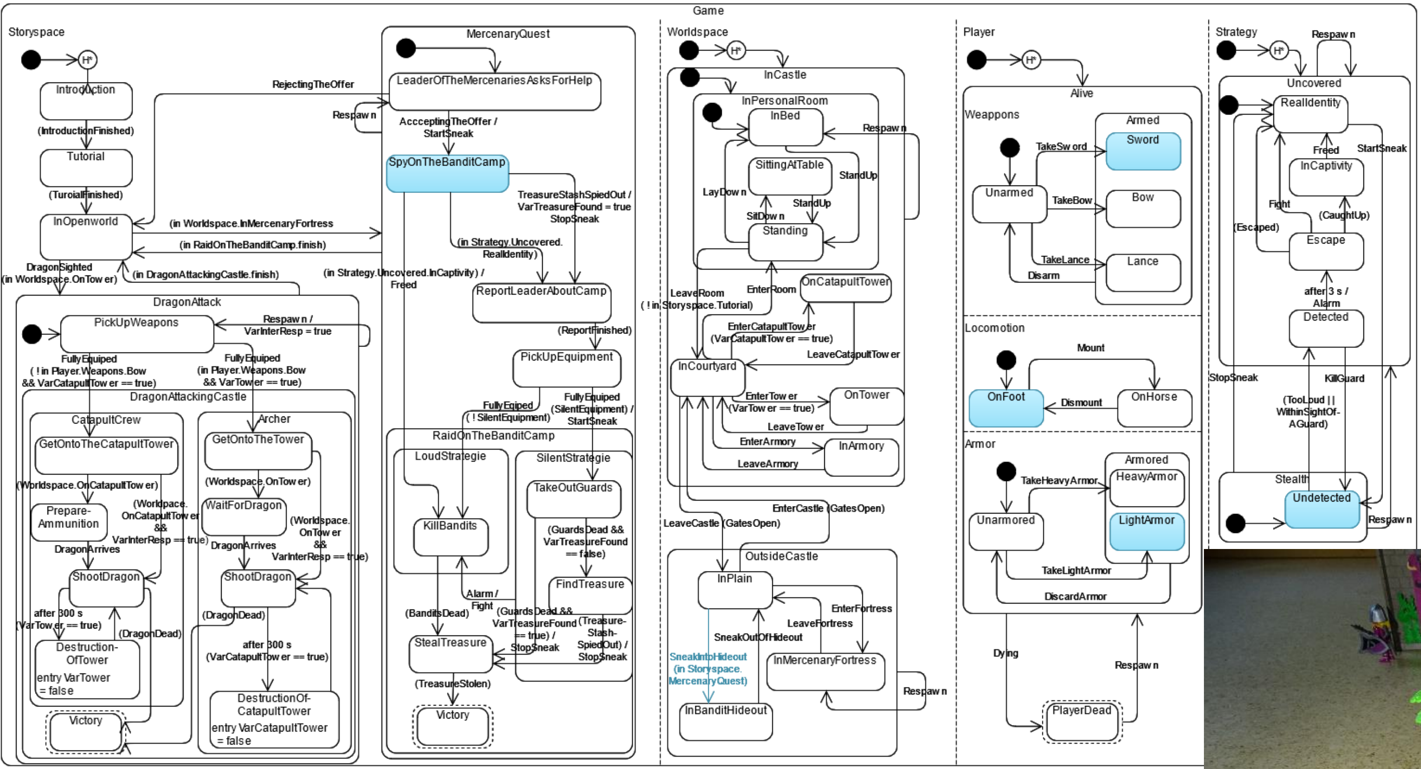


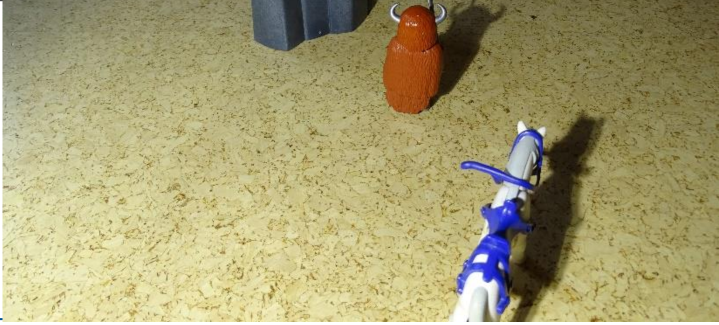
P.P. – Storyline: Spy on the Bandit Camp



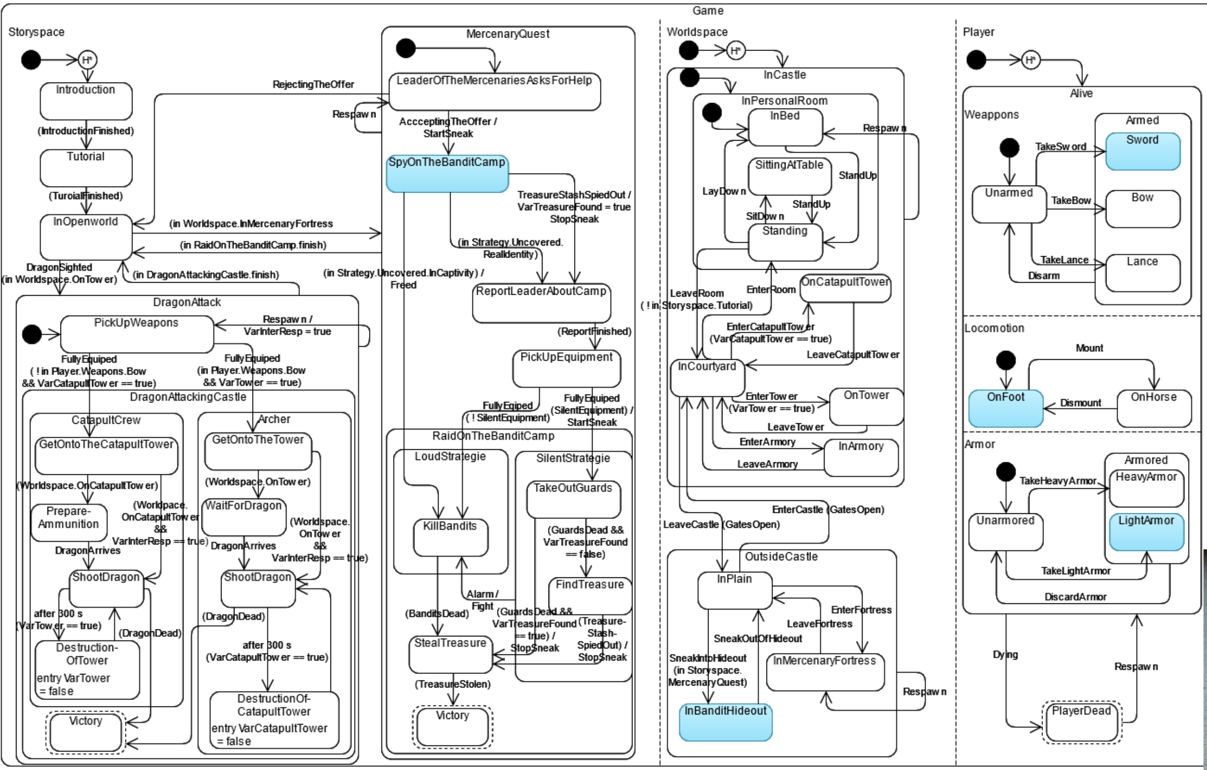


P.P. – Storyline: Spy on the Bandit Camp

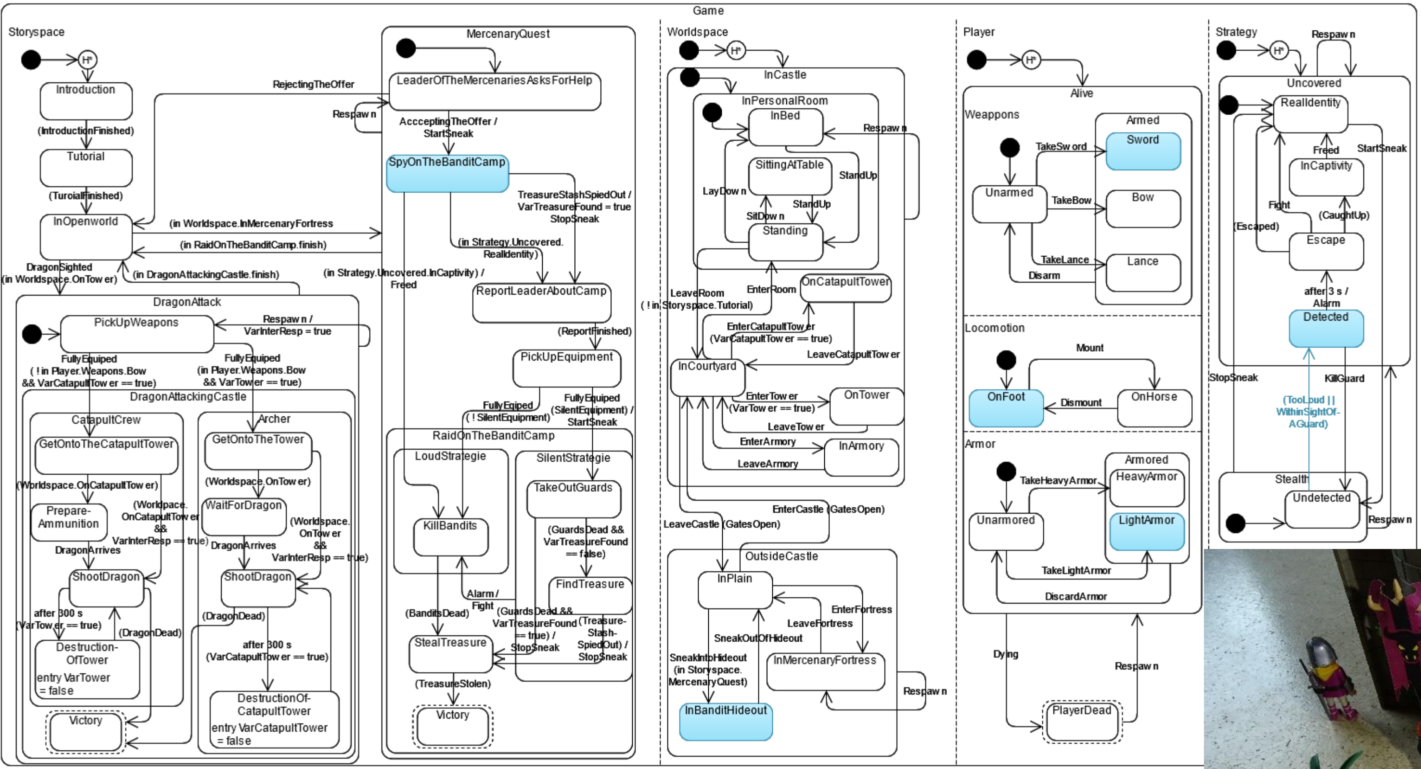




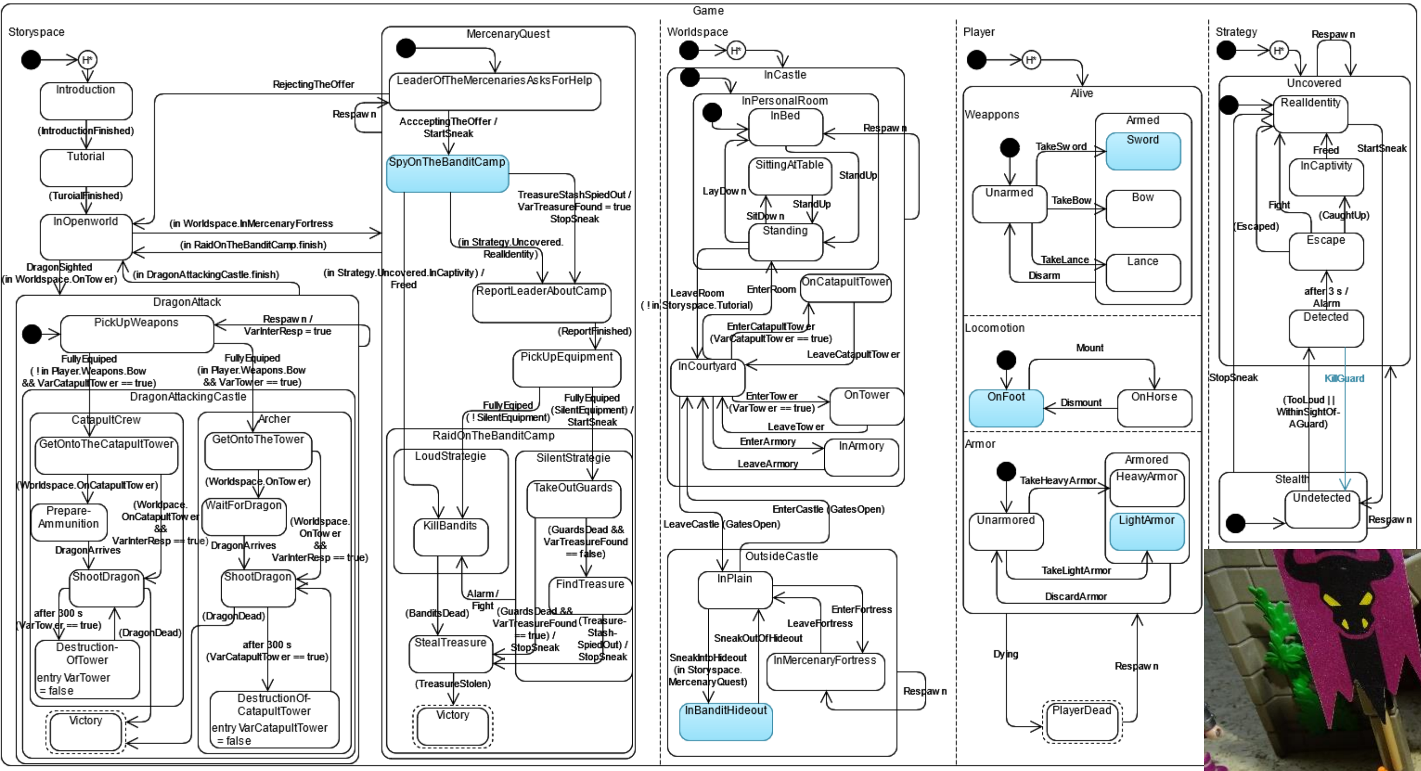
P.P. – Storyline: Spy on the Bandit Camp



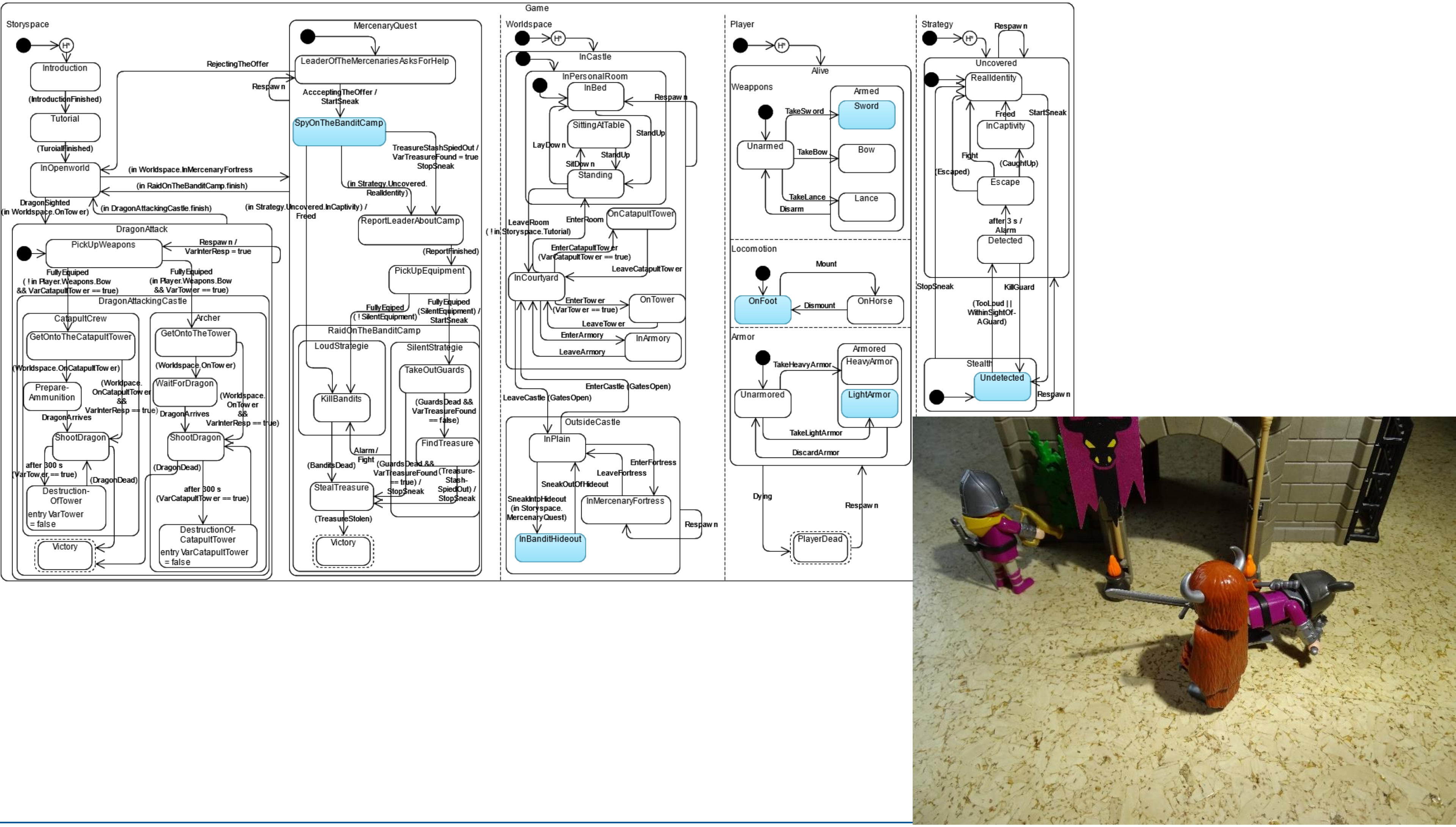
P.P. – Storyline: Detected



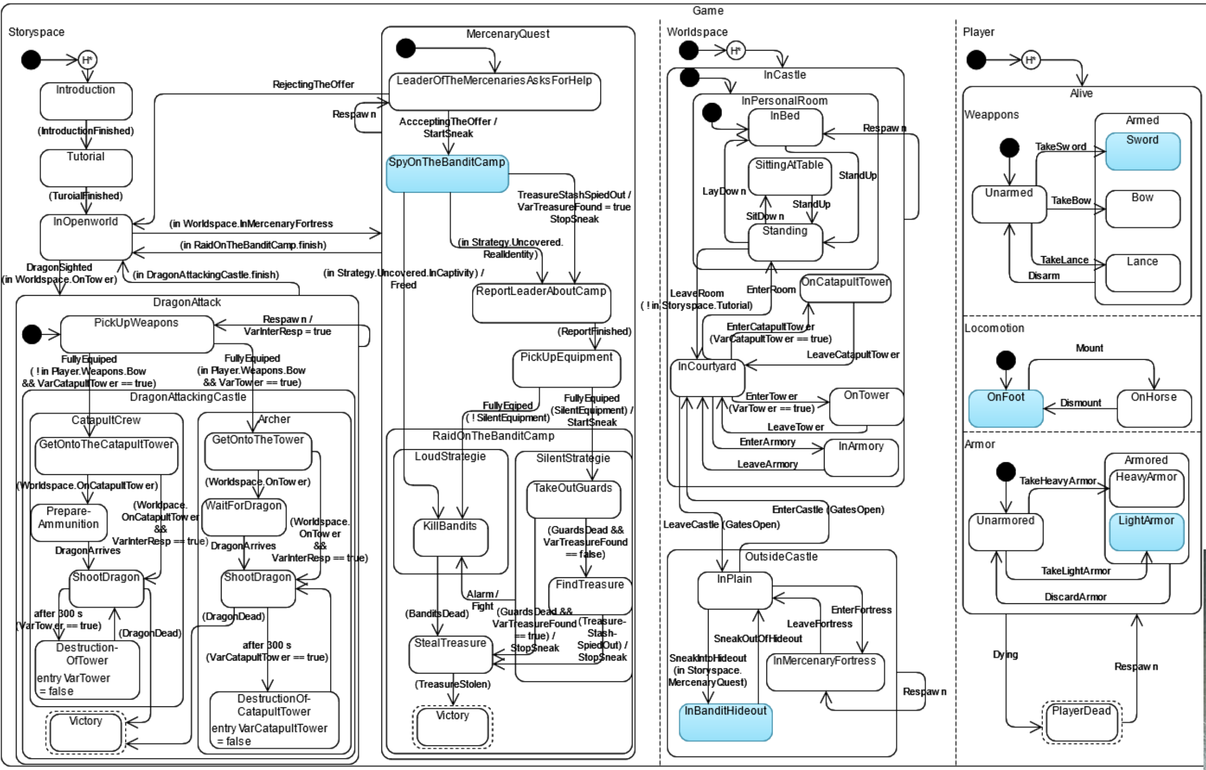
P.P. – Storyline: Kill Guard



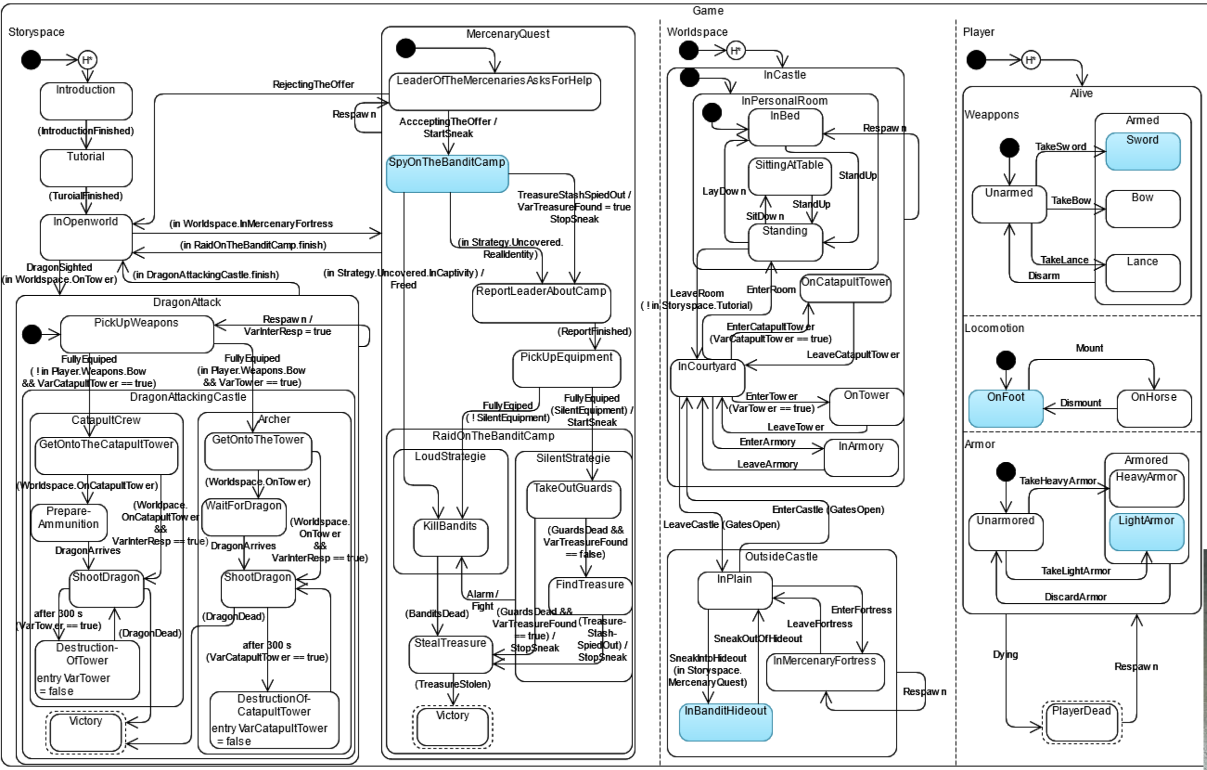
P.P. – Storyline: Undetected



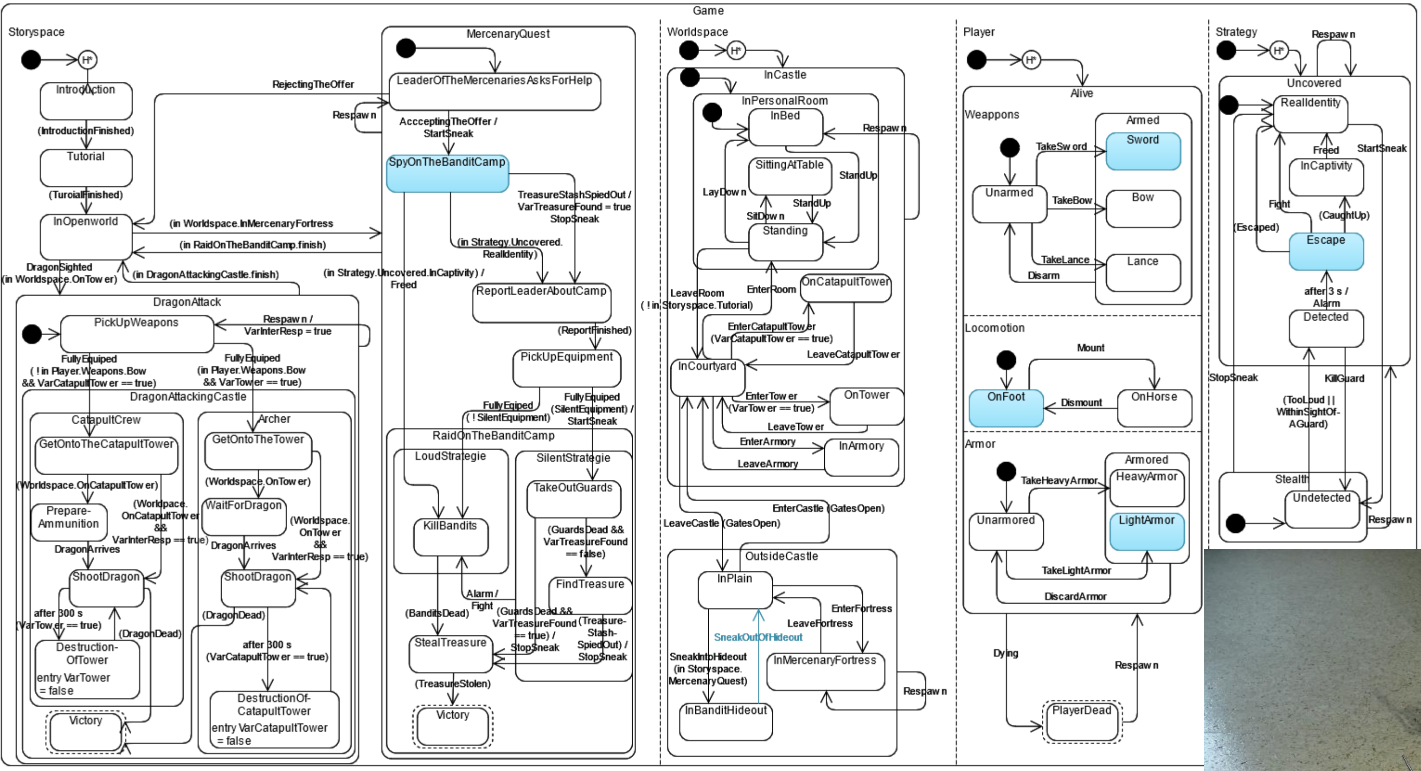
P.P. – Storyline: Detected



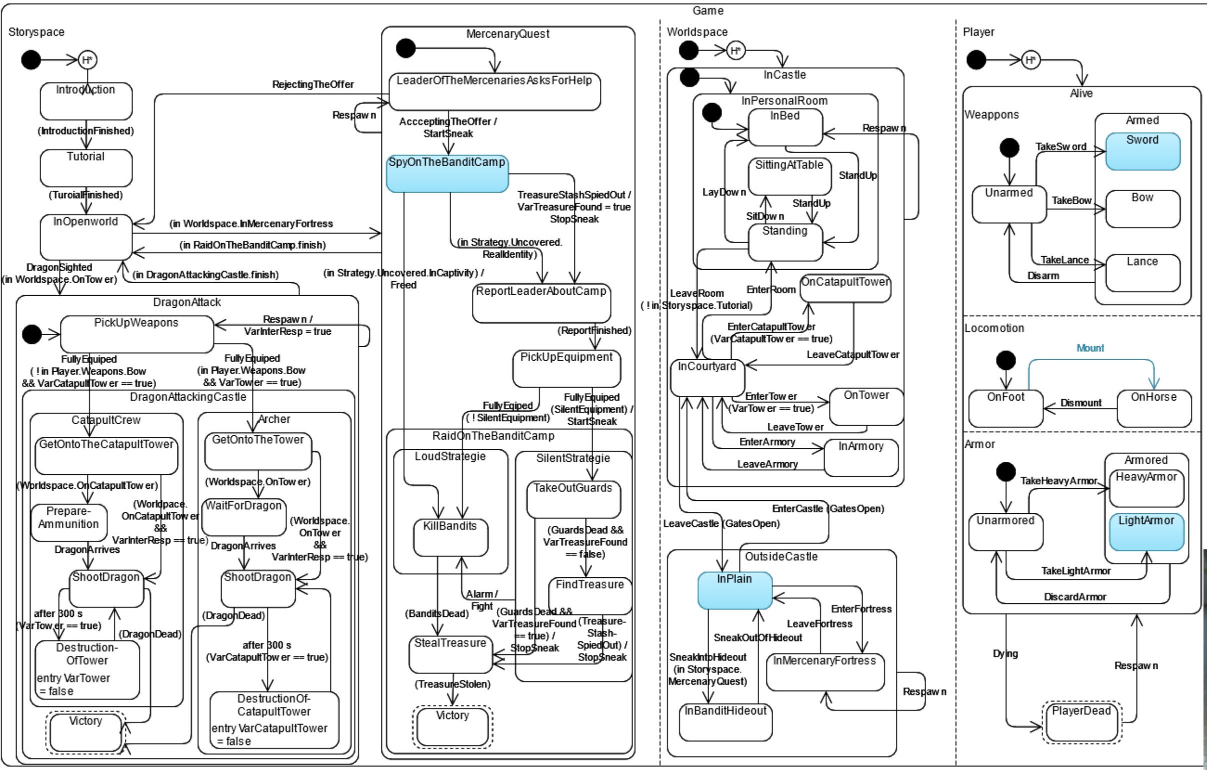
P.P. – Storyline: Alarm after 3 Seconds



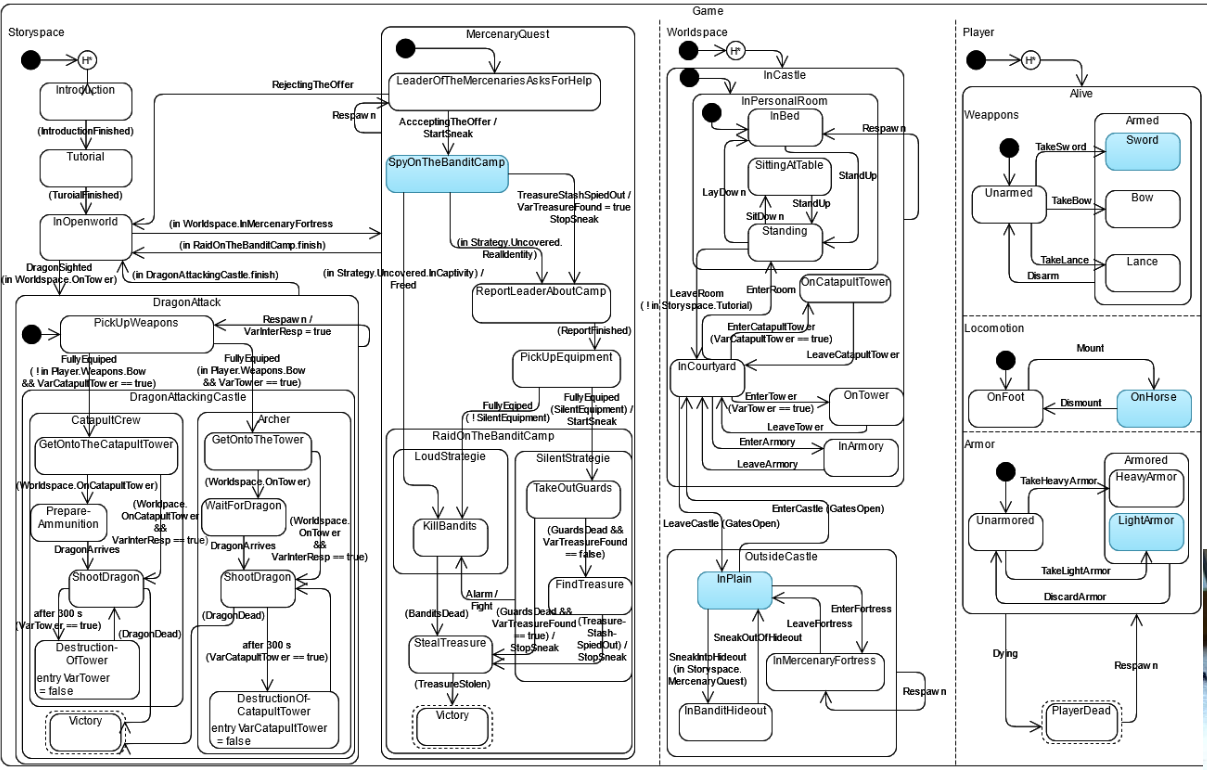
P.P. – Storyline: Escape



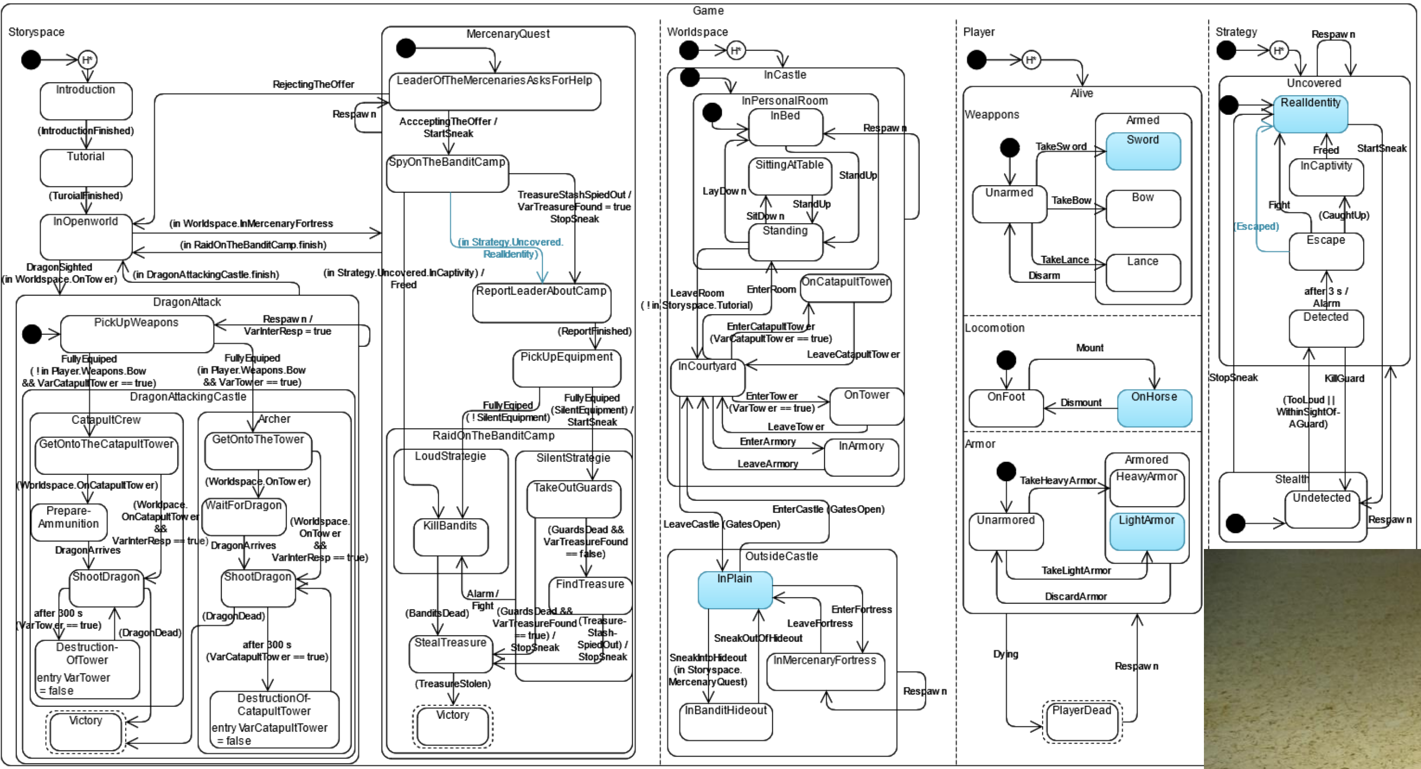
P.P. – Storyline: Escape



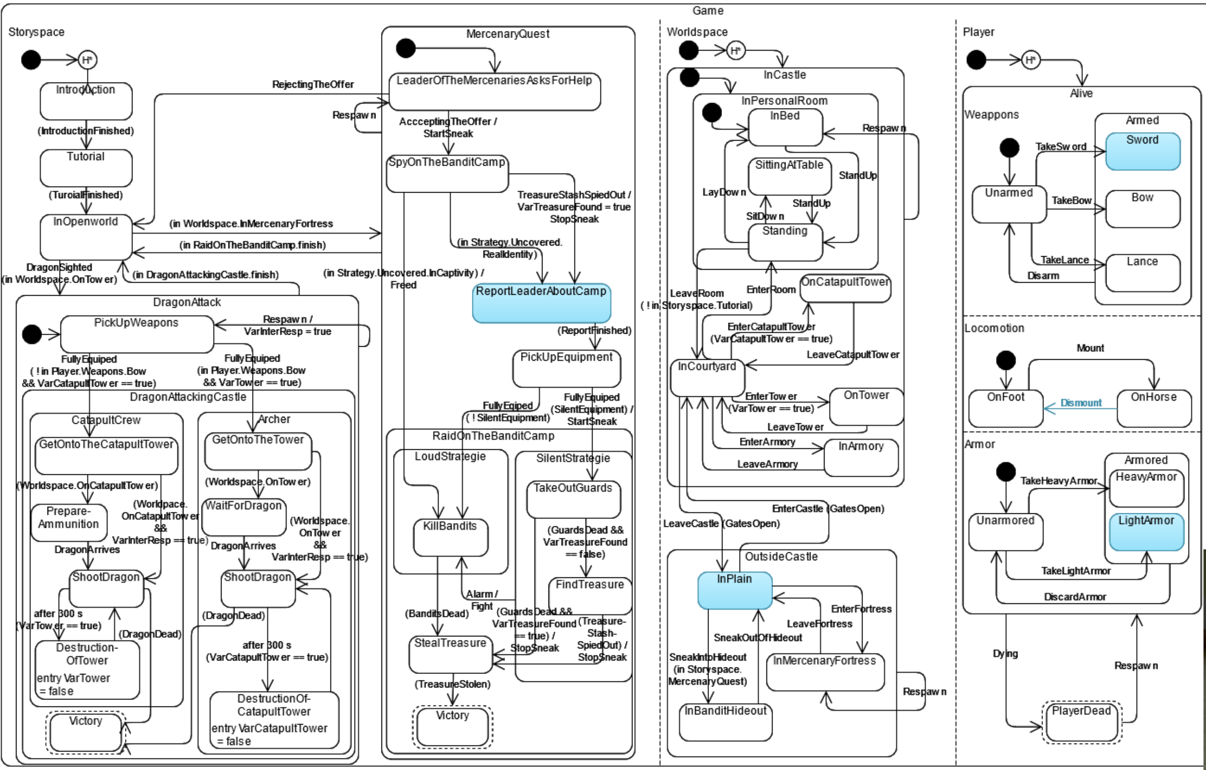
P.P. – Storyline: Escape



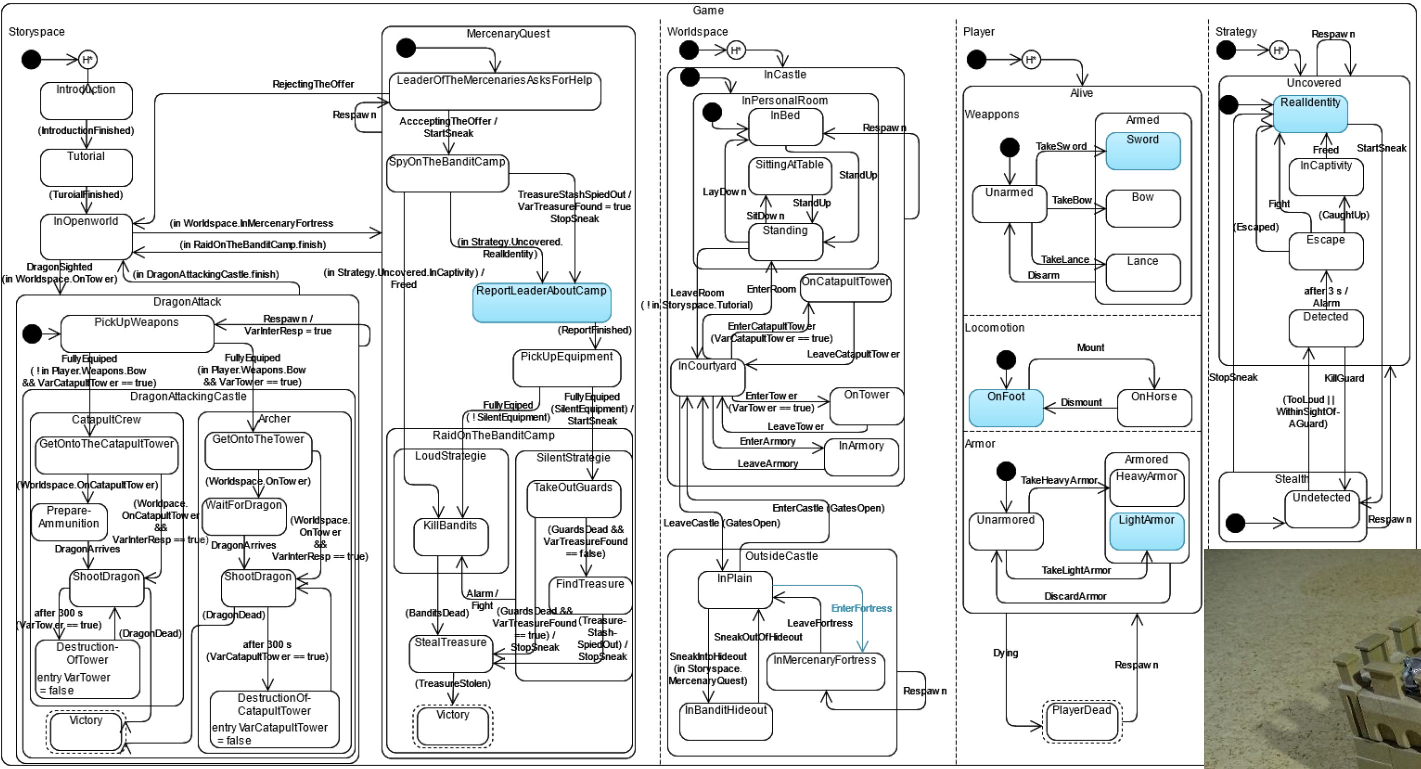
P.P. – Storyline: Successfully Escaped



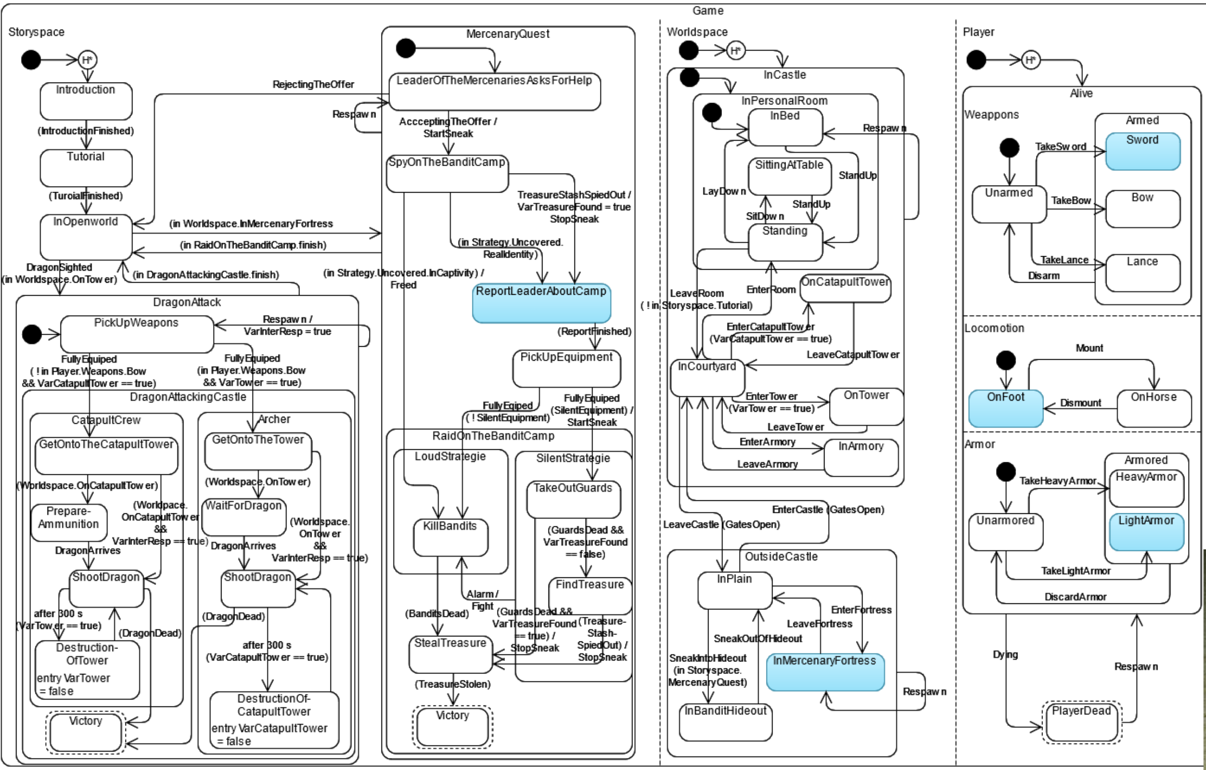
P.P. – Storyline: Report Leader



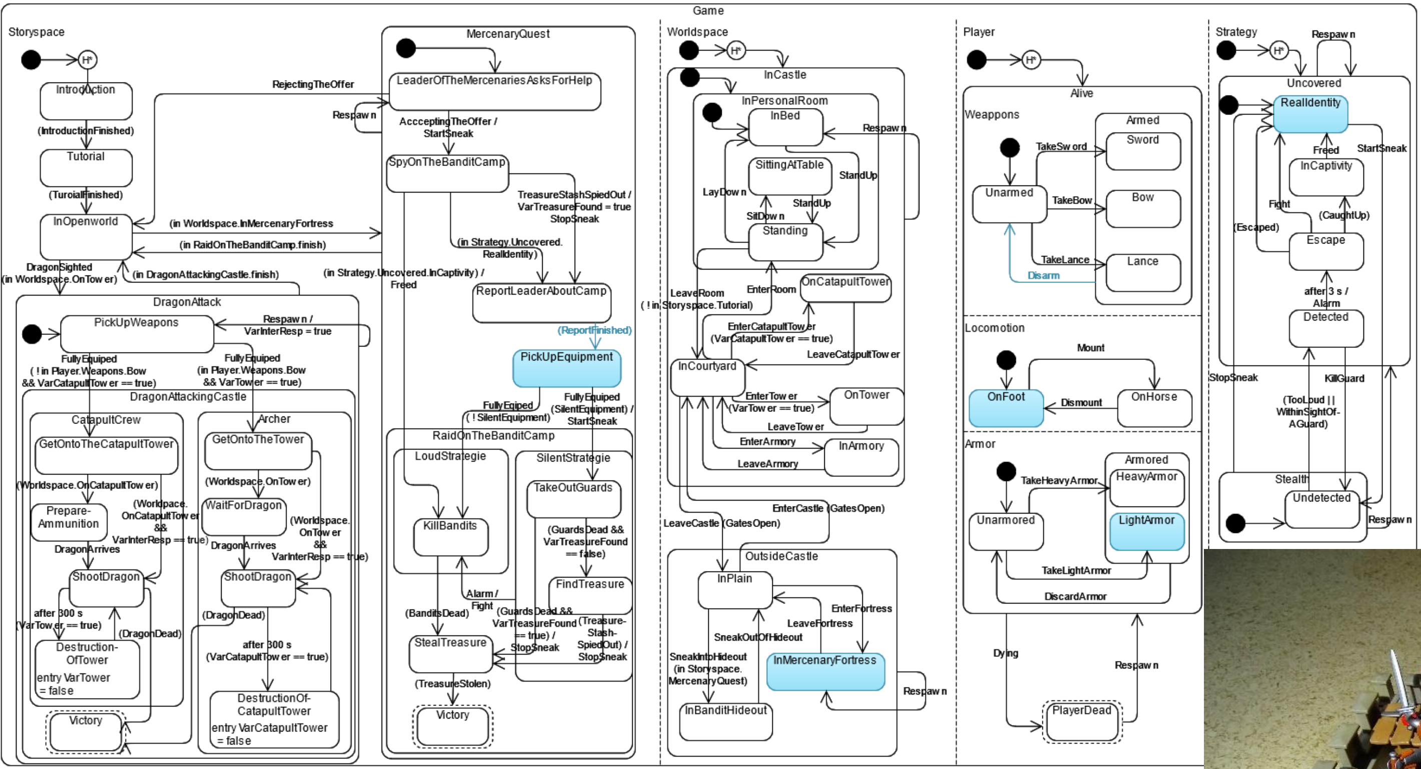
P.P. – Storyline: Report Leader



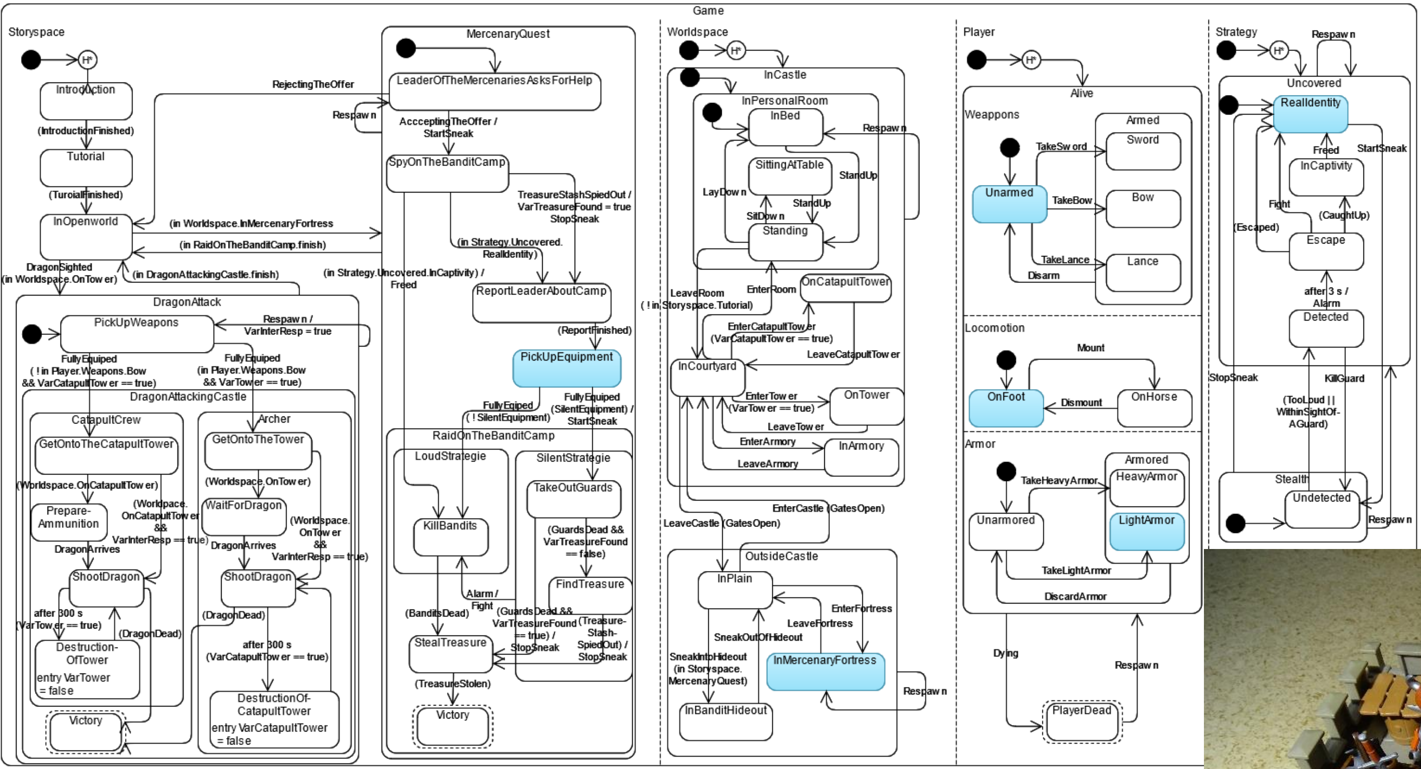
P.P. – Storyline: Report Leader



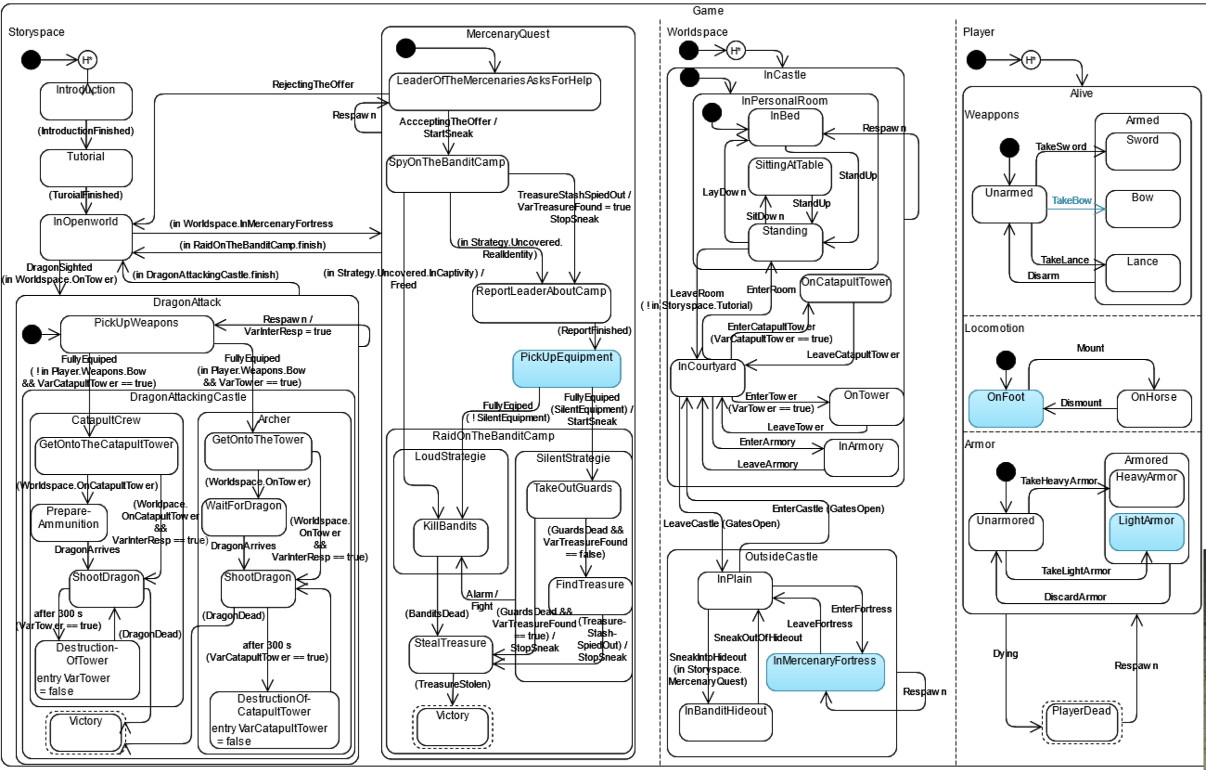
P.P. – Storyline: Pick Up Weapons



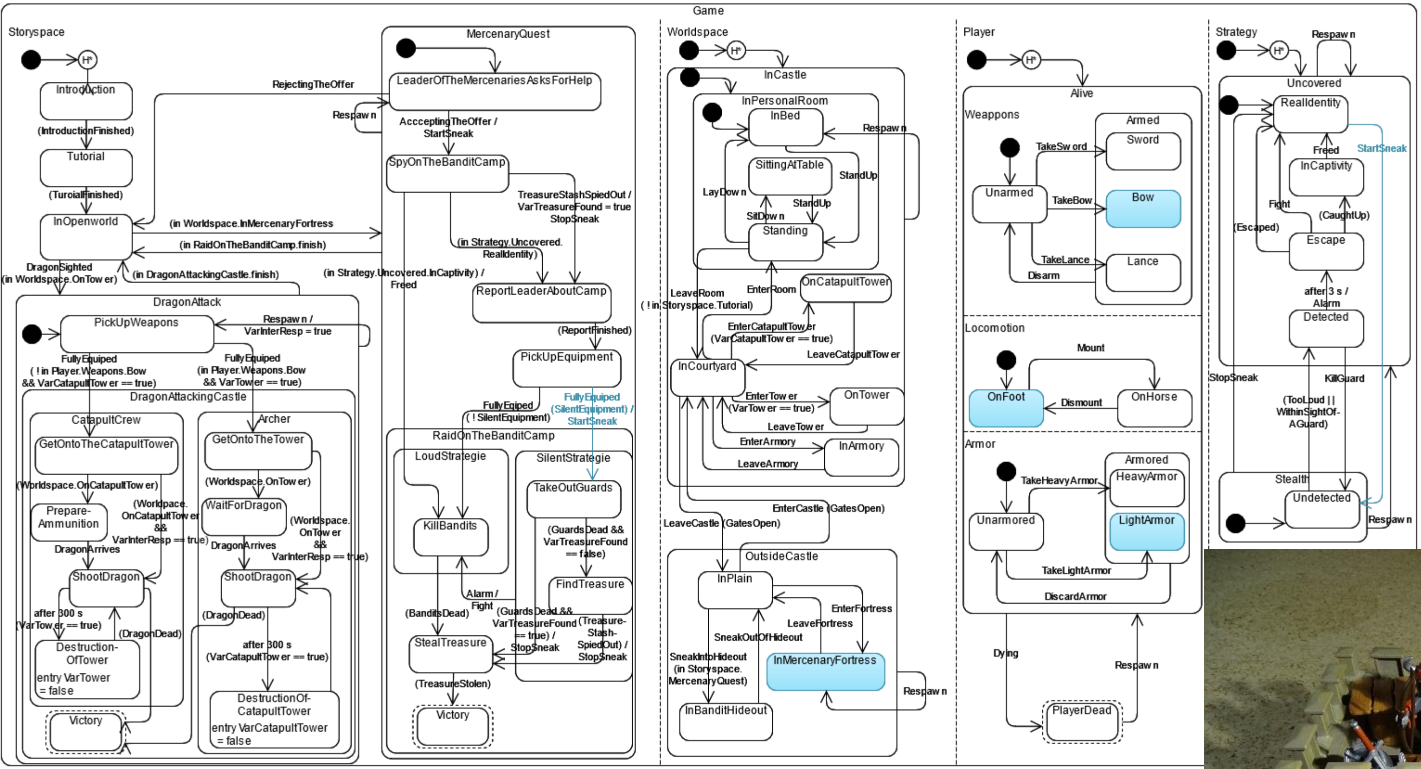
P.P. – Storyline: Pick Up Weapons



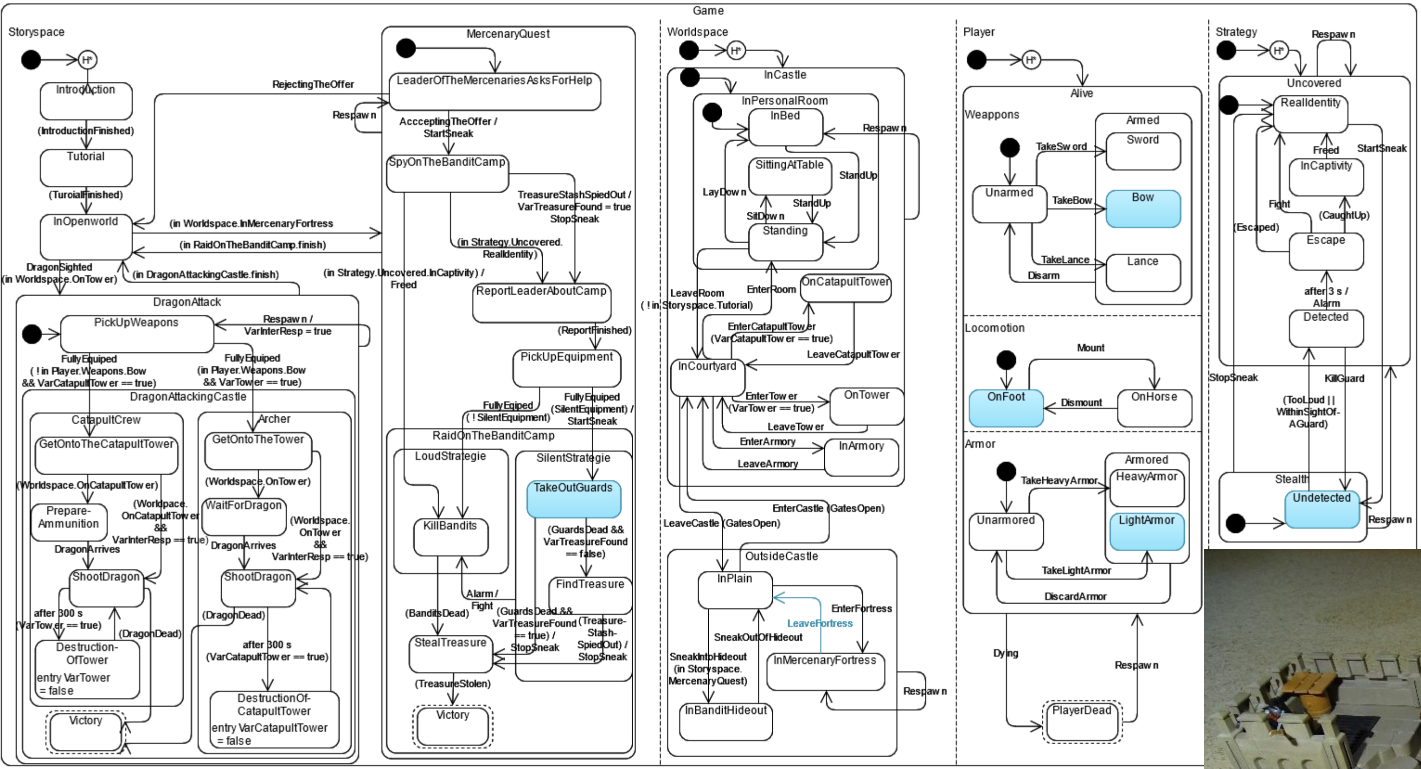
P.P. – Storyline: Pick Up Weapons



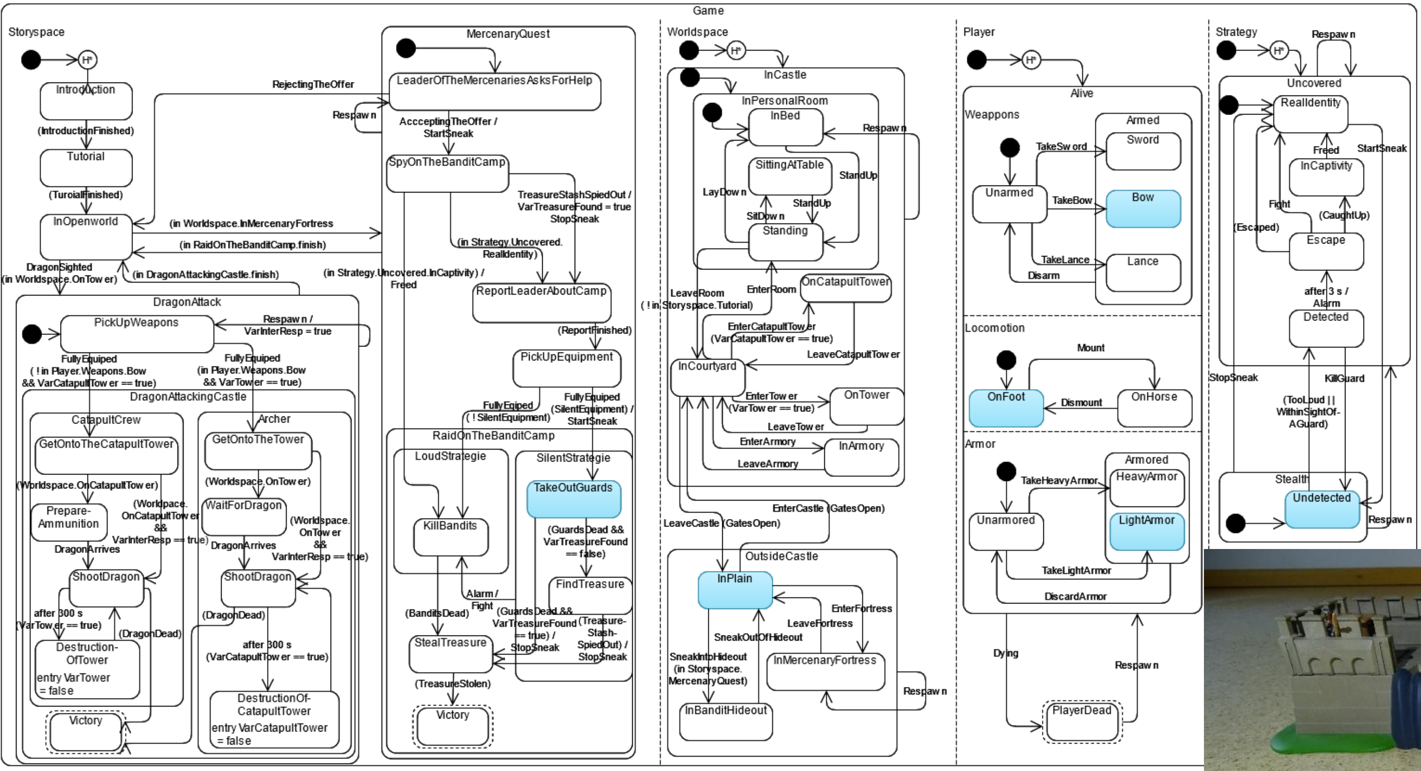
P.P. – Storyline: Fully Equipped



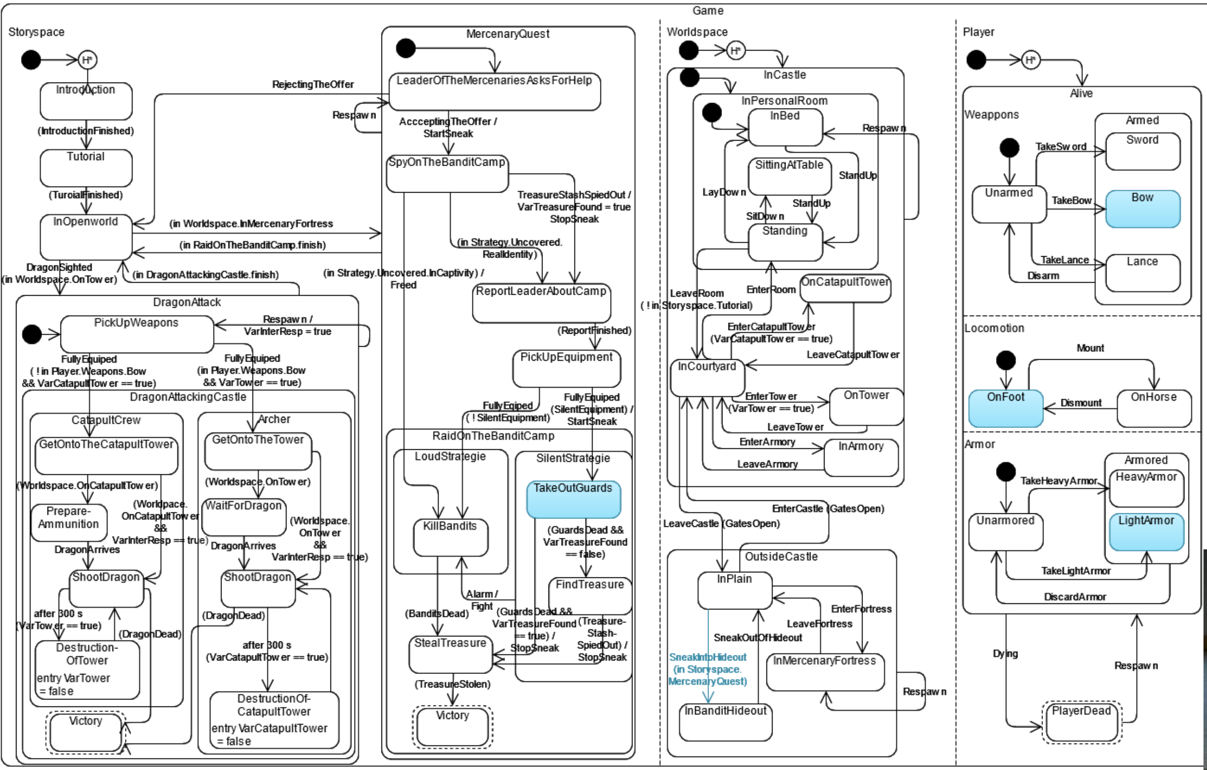
P.P. – Storyline: Take Out Guards



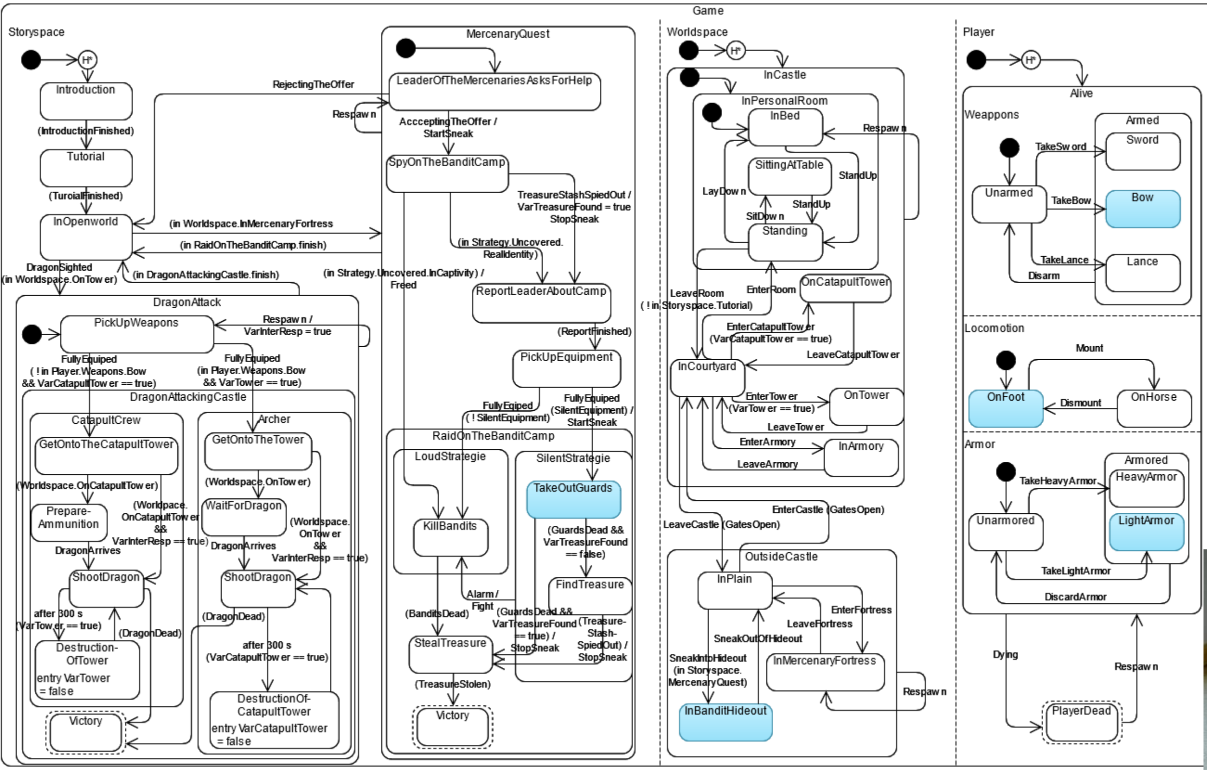
P.P. – Storyline: Take Out Guards



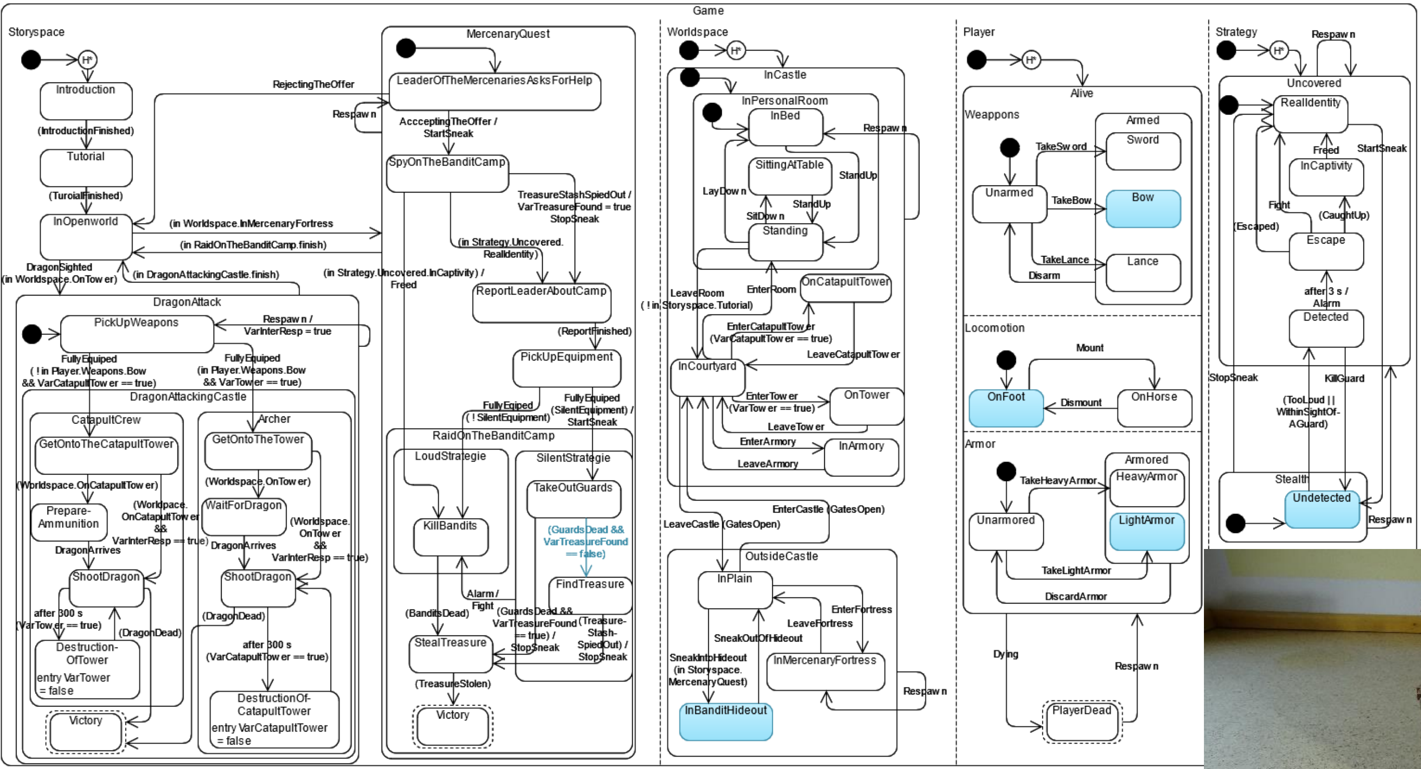
P.P. – Storyline: Take Out Guards



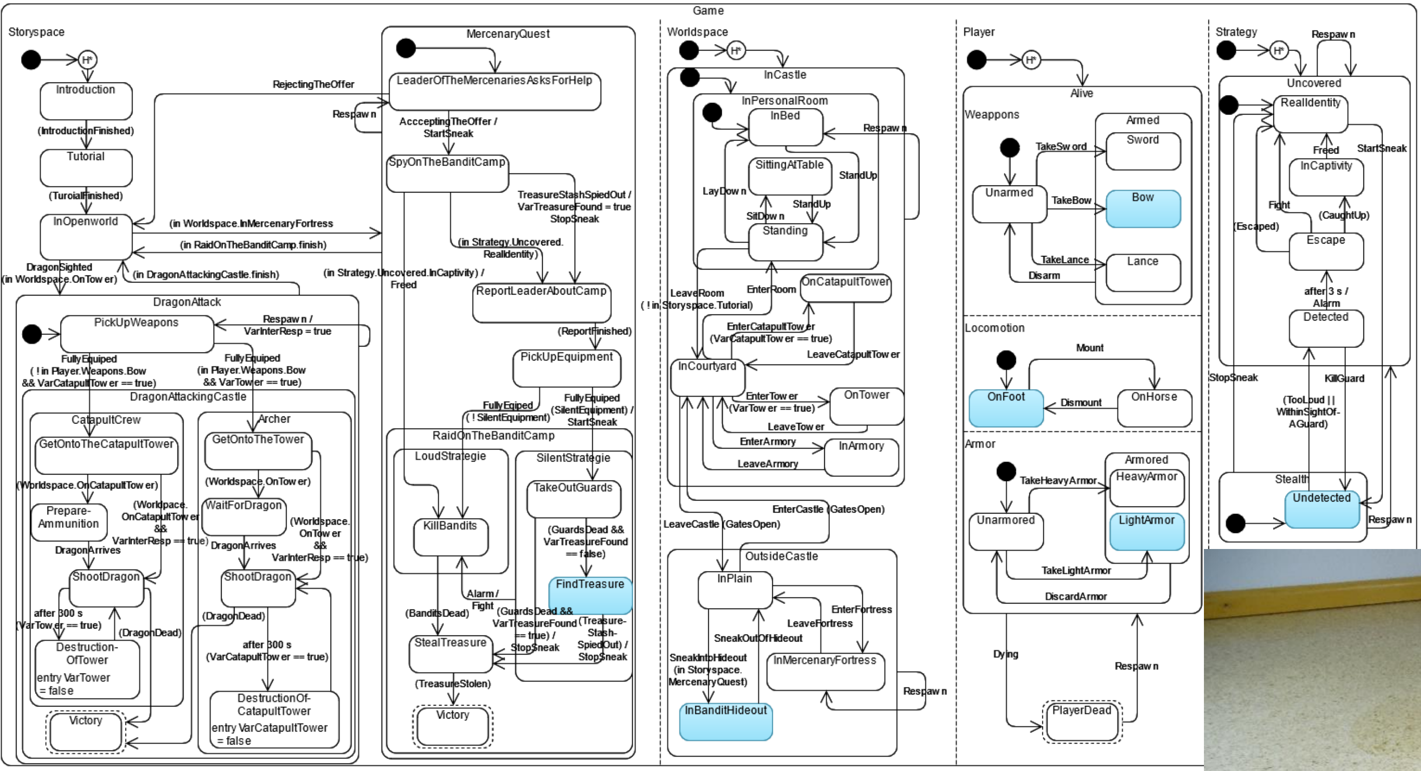
P.P. – Storyline: Take Out Guards



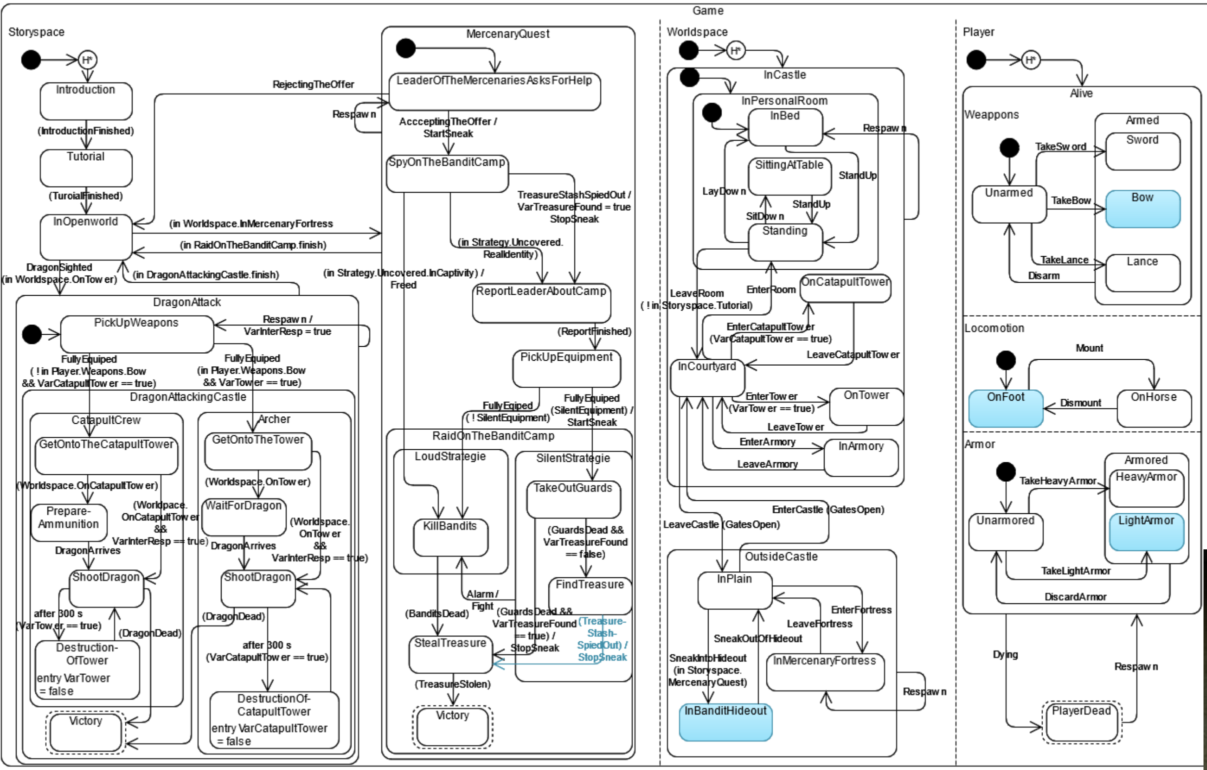
P.P. – Storyline: Guards Dead



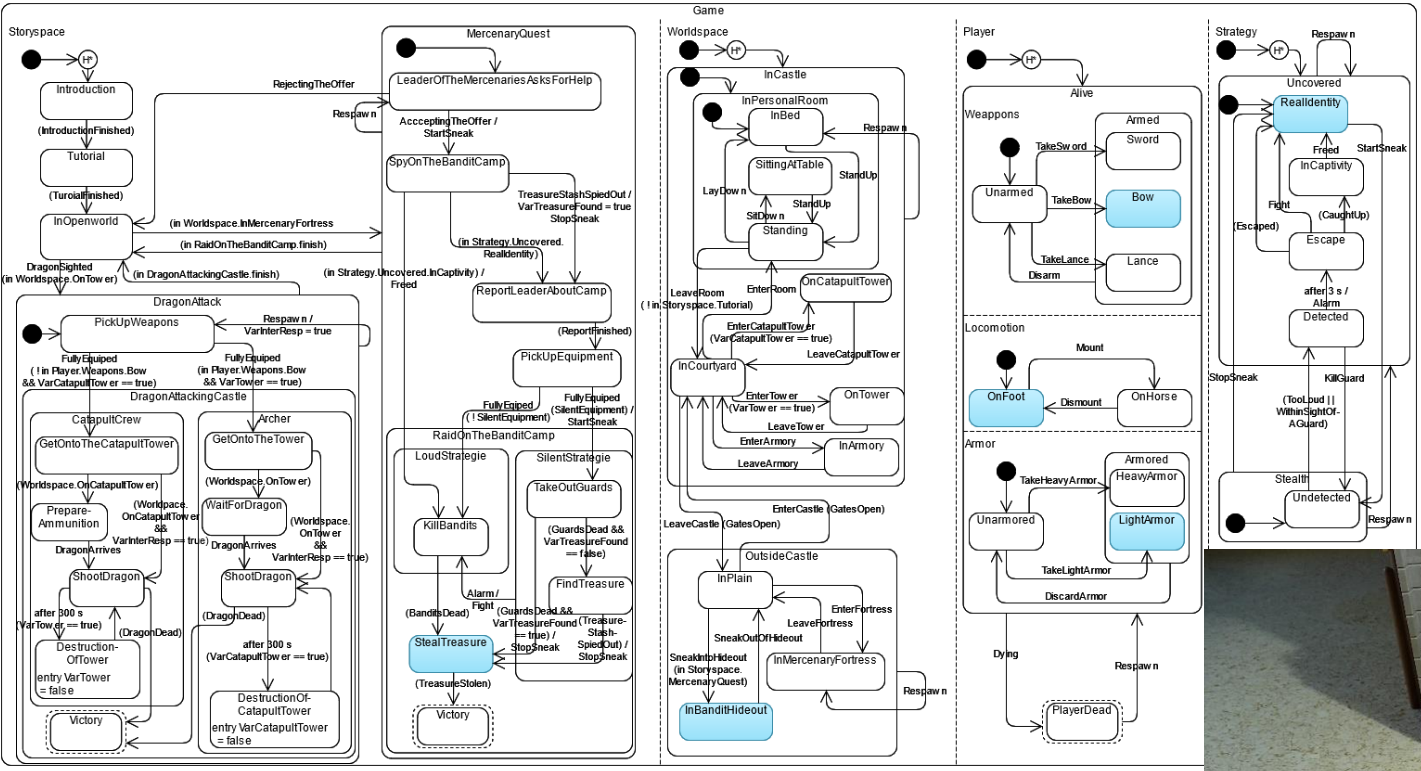
P.P. – Storyline: Find Treasure



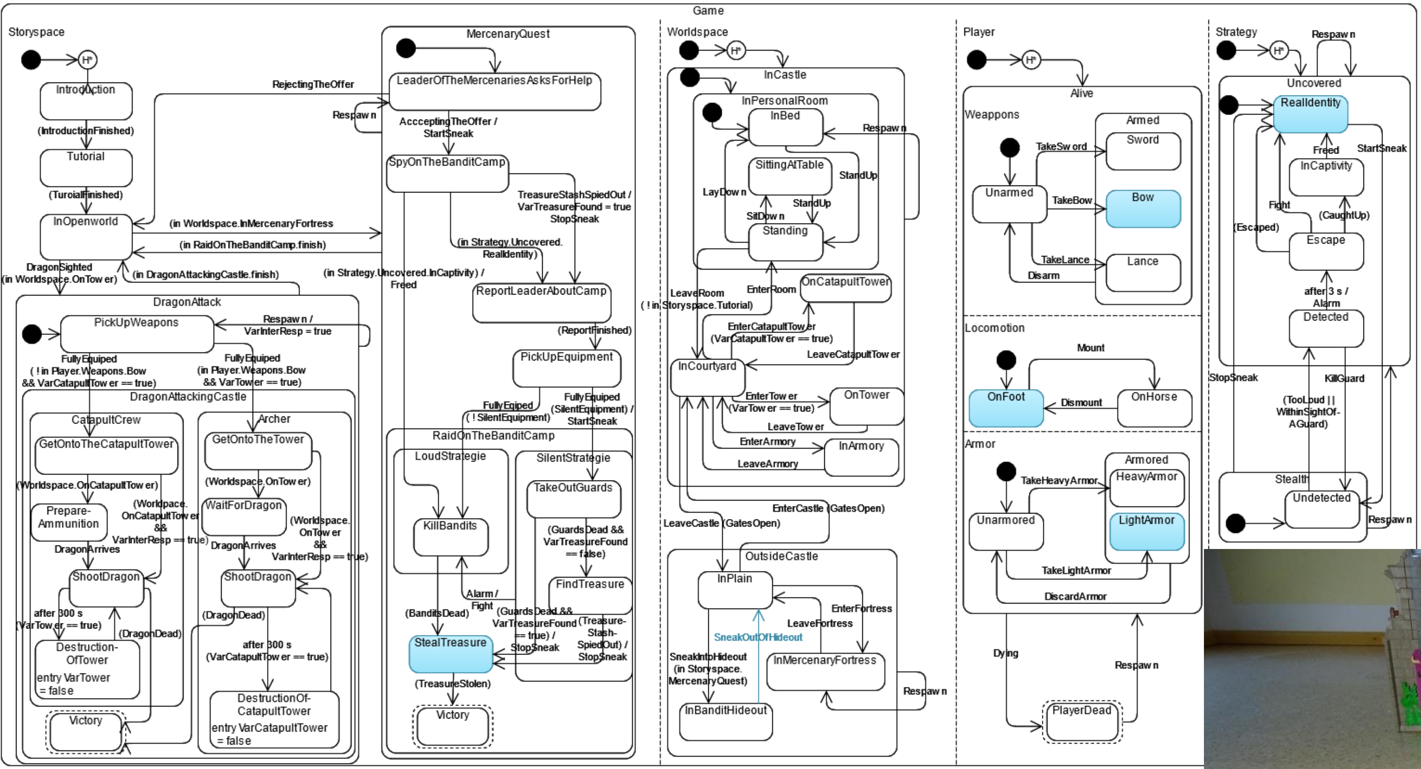
P.P. – Storyline: Treasure Spotted



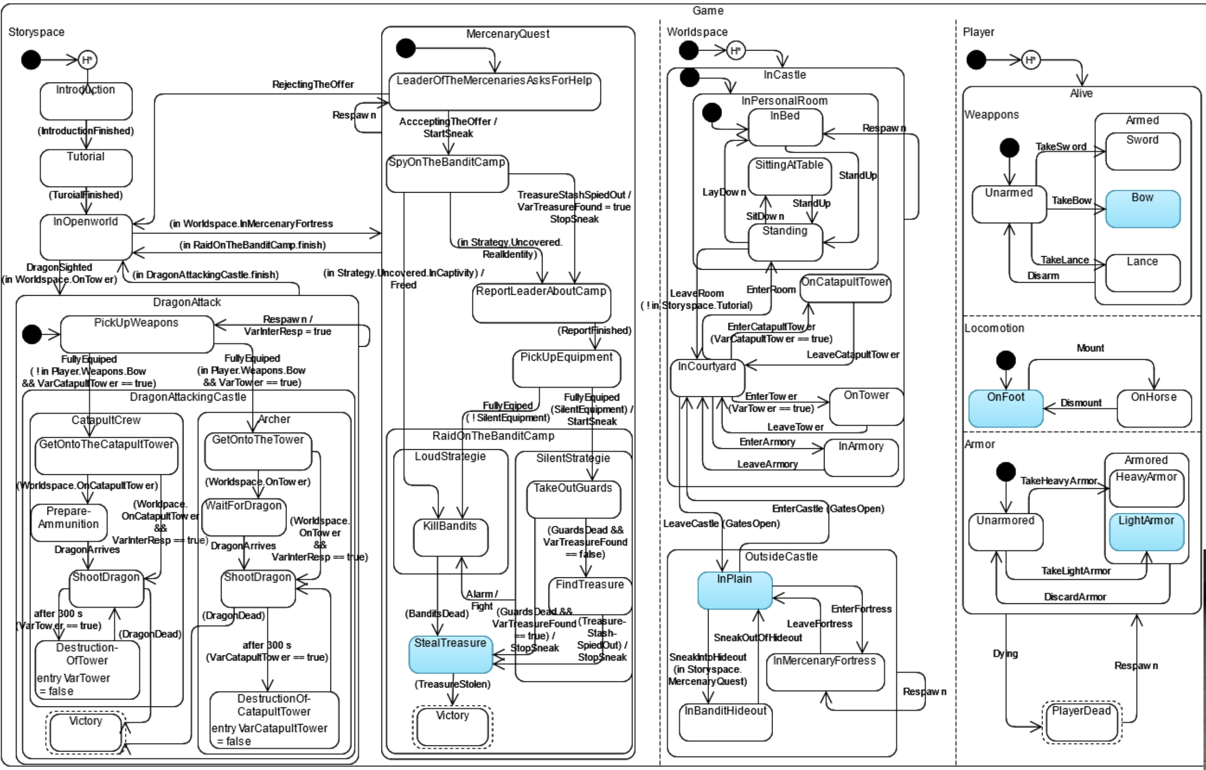
P.P. – Storyline: Steal Treasure



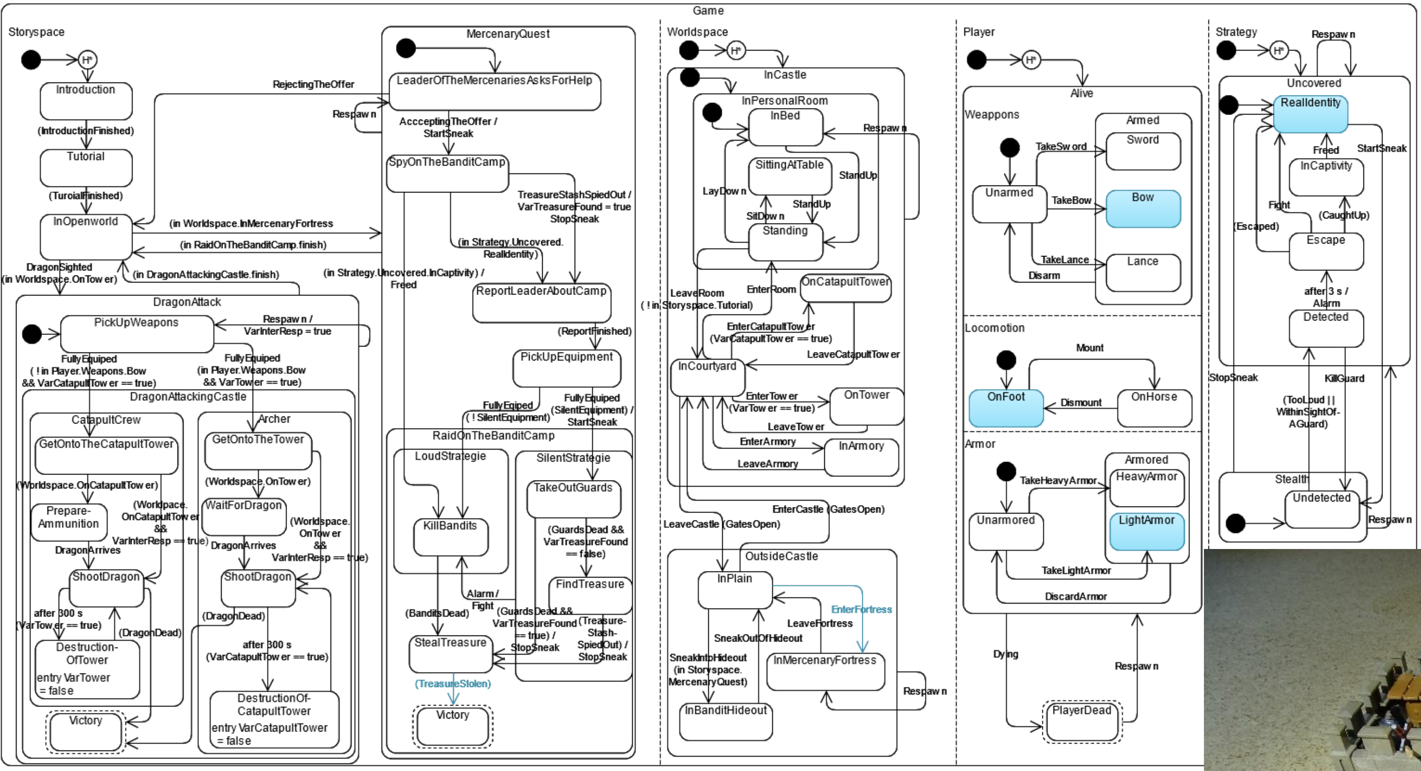
P.P. – Storyline: Steal Treasure



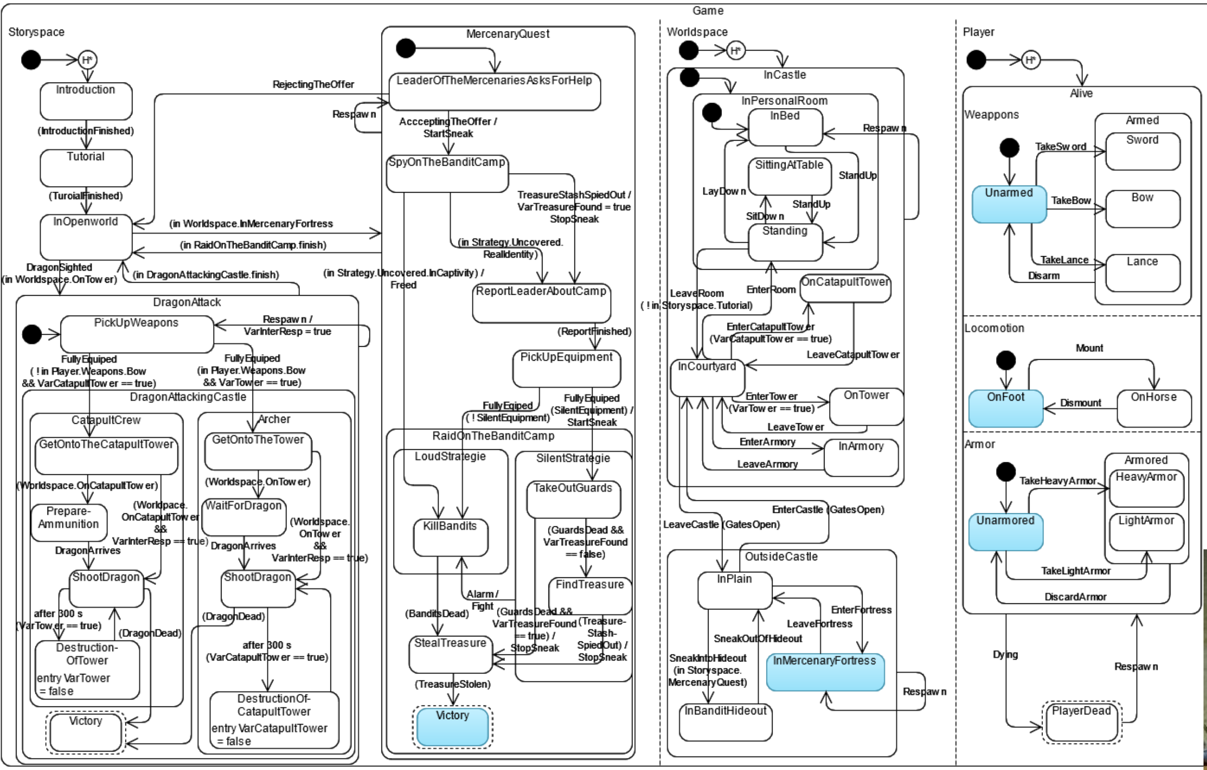
P.P. – Storyline: Steal Treasure



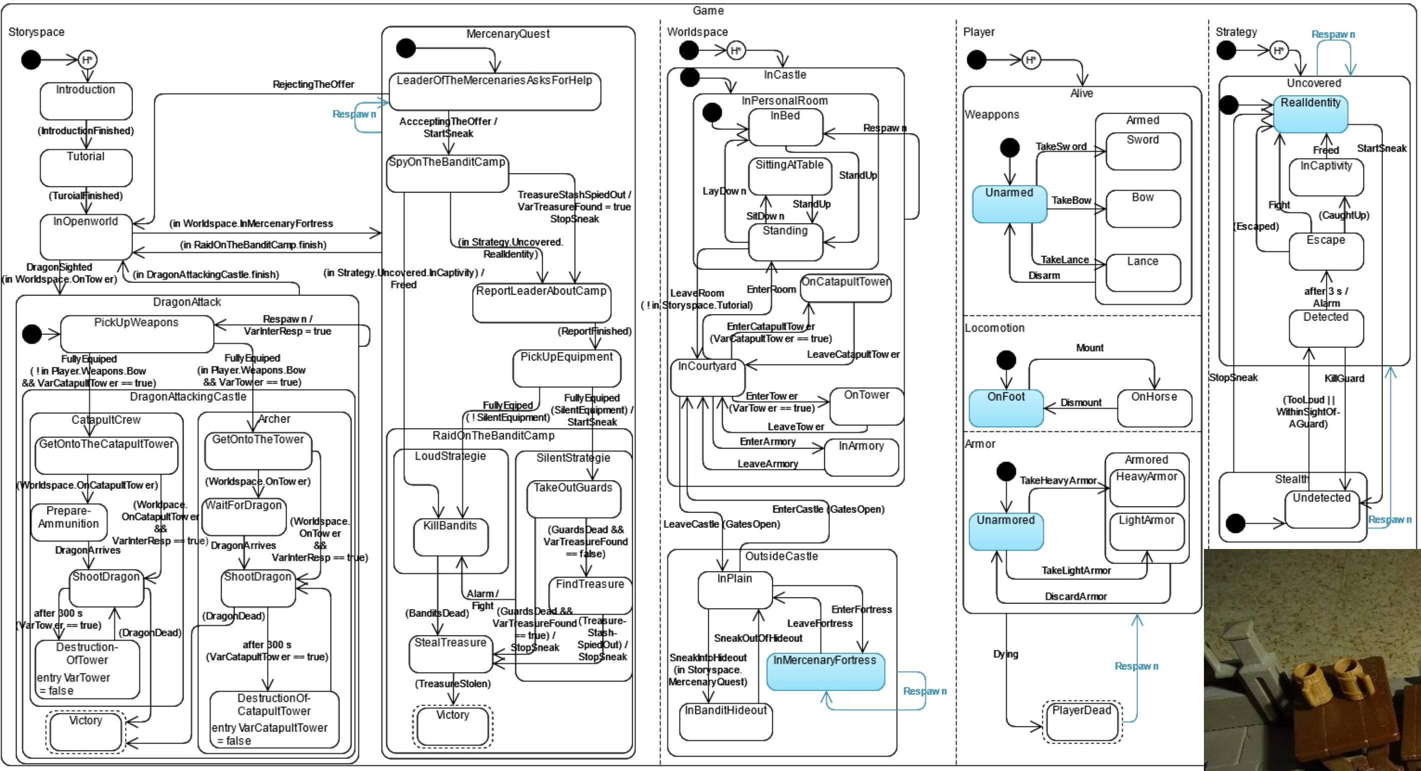
P.P. – Storyline: Steal Treasure



P.P. – Storyline: Mercenary-Quest Victory



P.P. – Storyline: Mercenary-Quest Respawn



Suggested Future Work

- Implementation of the prototype in a game engine
- Implementation of a statechart-framework for a game engine
- Developing a more efficient statechart feature for modeling dependencies between orthogonal components
- Developing tool in statechart-framework for adding and visualizing dependencies between parallel components more efficiently
- Unlimited event propagation in exits: Dynamic modifying of events, concealment of exceptions

Conclusion

- Statecharts can be successfully used as environment for representing dynamic storytelling in games

→ see Mapping table and prototype

Conclusion

Advantages of using statecharts for representing game-stories	Disadvantages of using statecharts for representing game-stories
Structured overview and visualization of hierarchical structure reveals logic caps in the story as soon as they are created	Representation of game scenarios quickly becomes complex and sprawling → many dependencies between parallel components
Effective implementation of interactive elements and the interaction of the story with the game-mechanics or with the game itself by actions, activities, or trigger events	Risk of modelling an event flow diagram instead of a statechart
Good format to model parallel storylines	Lack of uniform syntax
Extendibility of statechart-feature-canon by further features and direct Usability of existing statechart features for representing storytelling elements	Ambiguous semantic

List of References

- H. Züllighoven, “Object-oriented construction handbook. dpunkt. verlag,” 2005.
- PowerPyx, “Detroit become human full walkthrough – all chapters,” 2018, Retrieved on 15.07.2021. [Online]. Available: https://www.powerpyx.com/detroit-become-human-full-walkthrough-all-chapters/
- S. Halliwell, “What is a flowchart?” 2019, Retrieved on 15.07.2021. [Online]. Available: https://github.com/snozbot/fungus/wiki/flowcharts
- M. Cavazza, F. Charles, and S. Mead, “Character-based interactive storytelling,” IEEE Intelligent systems, vol. 17, no. 4, pp. 17–24, 2002.
- M. Merabti, and A. El Rhalibi, Y. Shen, J. Daniel, A. Melendez, and M. Price, “Interactive storytelling: Approaches and techniques to achieve dynamic stories,” in Transactions on Edutainment I. Springer, 2008, pp. 118–134.
- J. Brusk, and T. Lager, “Developing natural language enabled games in (extended) scxml,” in Proceedings from the international symposium on intelligence techniques in computer games and simulations (Pre-GAMEON-ASIA and Pre-ASTEC), Shiga, Japan, March, 2007, pp. 1–3.
- J. Brusk, T. Lager, A. Hjalmarsson, and P.Wik, “Deal: dialogue management in scxml for believable game characters,” in Proceedings of the 2007 conference on Future Play, 2007, pp. 137–144.

List of References

- A. Leclerc von Bonin, “Dynamic storytelling at internet briefing in zürich,” 2015, Retrieved on 15.07.2021. [Online]. Available: https://www.dizmo.com/dynamic-storytelling/
- B. Ip, “Narrative structures in computer and video games: Part 1: Context, definitions, and initial findings,” Games and Culture, vol. 6, no. 2, pp. 103–134, 2011. [Online]. Available: https://doi.org/10.1177/1555412010364982
- D. Harel, “Statecharts: A visual formalism for complex systems,” Science of computer programming, vol. 8, no. 3, pp. 231–274, 1987.
- Visual Paradigm, “Online state machine diagram tool,” 2021, Retrieved on 15.07.2021. [Online]. Available: https://online.visual-paradigm.com/de/diagrams/features/state-machine-diagram-software/#
- E. Mogensen, “Full list of glossary terms,” Retrieved on 15.07.2021. [Online]. Available: https://statecharts.dev/glossary/
- S. Brown, “Paper prototypes and beyond,” Visible Language, vol. 43, no. 2, p. 198, 2009.
- B. Buxton, “Sketching user experience: Getting the right design and the design right,” São Francisco: Morgan-Kaufmann, 2007.
- S. Yadav, S. Sikka, and U. Shrivastava, “A review of object-oriented coupling and cohesion metrics,” International Journal of Computer Science Trends and Technology, vol. 2, no. 5, pp. 45–55, 2014.

List of References – Images and Video

1. [Blackboard Video] – YouTube, “Tafel stürzt beinahe auf Dozenten | TU München | Elektrotechnik” 2017, Retrieved on 23.07.2021. [Online]. Available: https://www.youtube.com/watch?v=NOtLb34Osl0
2. [Powerpyx.com – Detroit: Become Human] – PowerPyx, “Detroit become human full walkthrough – all chapters,” 2018, Retrieved on 15.07.2021. [Online]. Available: https://www.powerpyx.com/detroit-become-human-full-walkthrough-all-chapters/
3. [Example game scenario from “Fungus” in “Unity 3D”] – S. Halliwell, “What is a flowchart?” 2019, Retrieved on 15.07.2021. [Online]. Available: https://github.com/snozbot/fungus/wiki/flowcharts