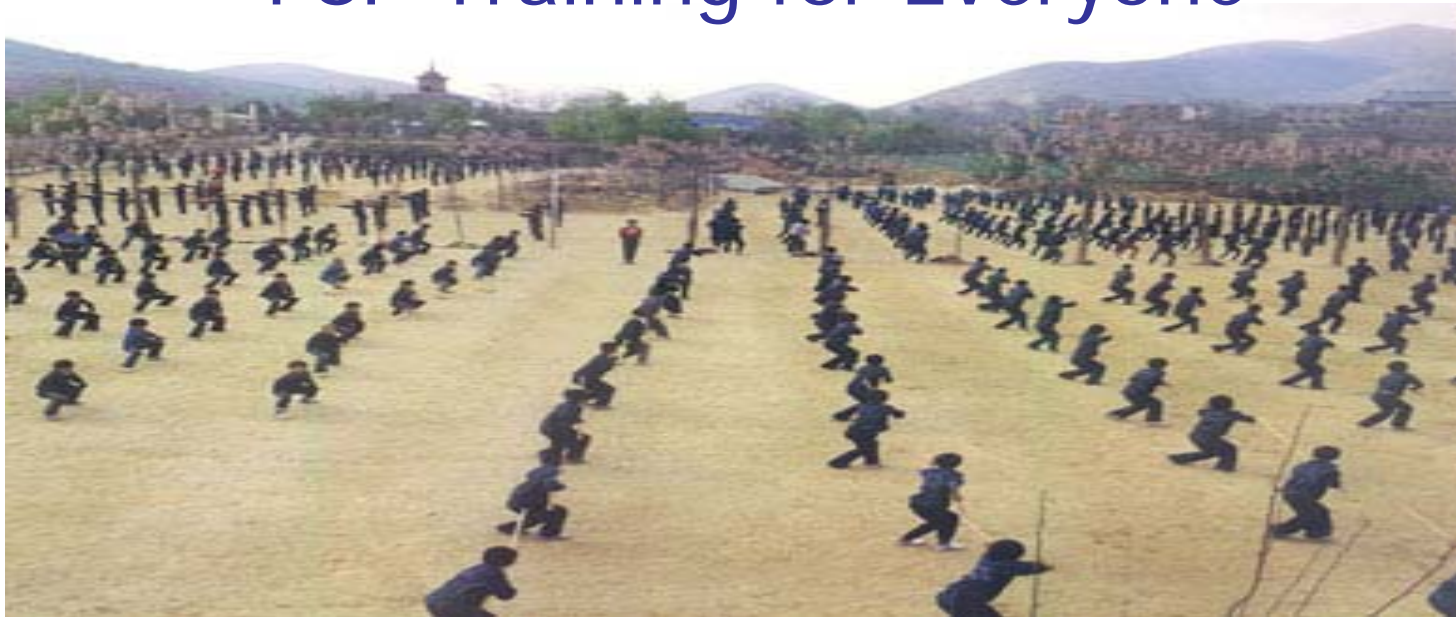


PSP Training for Everyone

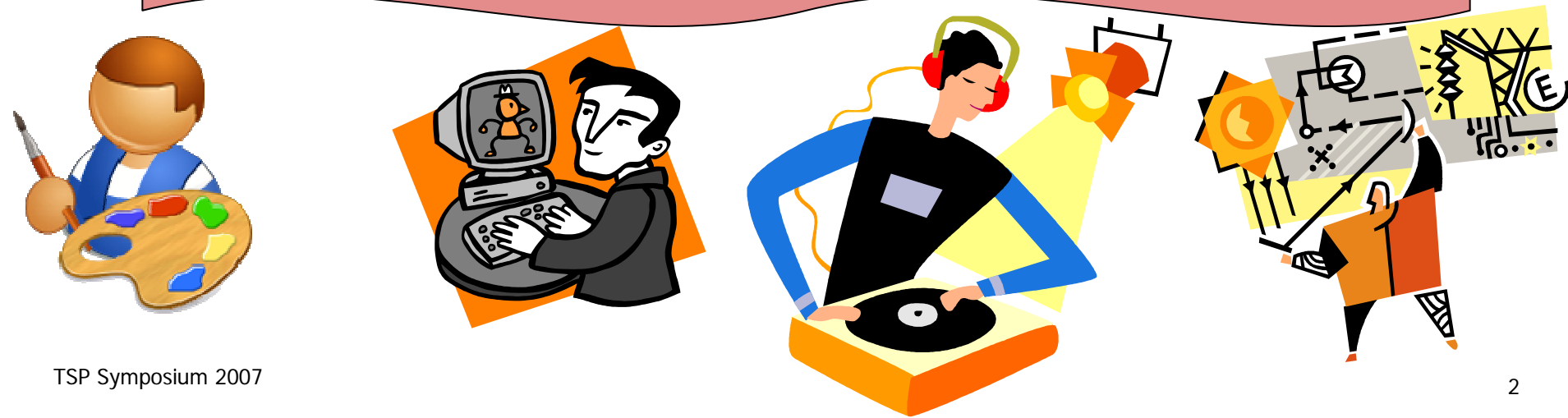


Dan Wall

More and more organizations are now adopting the TSP and attempting to use it in non software situations.

The purpose of the PSP for Engineers course is to provide the fundamental skills and knowledge needed for software developers to participate on TSP teams.

I have at least 5 different creative disciplines that work together to craft a video game.



Development Personnel Challenges

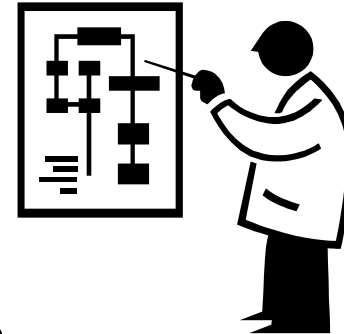
- Creative discovery, prototyping iterative development
 - Done when time is up
 - Can always make it better

- Game development culture
 - “In the garage” development mentality
 - Often immature practices
 - Fast, loose, casual, free thinking
 - Passionate

Agenda

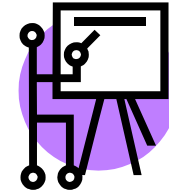


- Challenges
- Game Approach
- Demonstration
- Next Steps

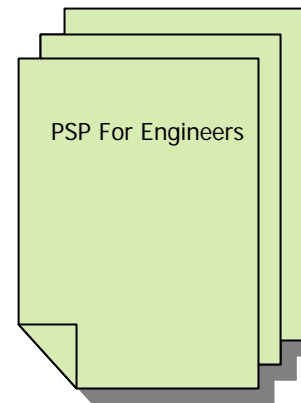


Training Challenges

PSP concepts and principles apply to any creative discipline



- Over 10 yrs of experience and refinement
- Training material is available
- Mb of historical data is available



Training Challenges

Training material is available but requires heavy modification

- Concepts and principles apply to any creative discipline – need to be translated
- Language – software engineering vs. neutral
- Stories, examples
- Exercises germane to each discipline

Tool issues

- Measurement frameworks, terminology, processes

Need for each discipline different

- Size measures
- Proxy tables
- Checklists
- Defect definitions
- Defect types



■ Requirements

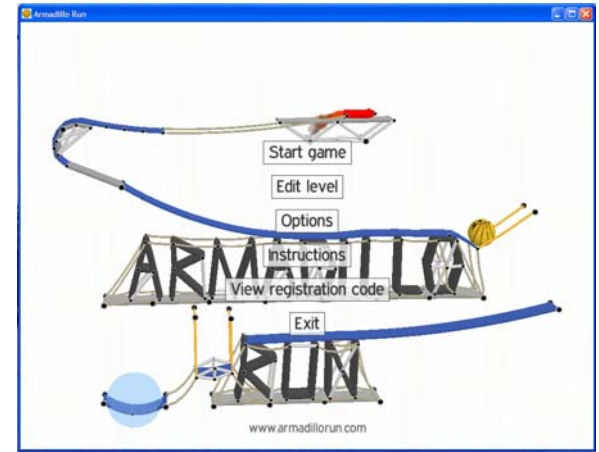
- Adheres to core PSP principles and practices
- Domain neutral exercises
- Short development time
- Support Tool available
- Easy, fun

■ Options

- Develop from scratch
- Find/Modify existing



- Use an existing video game
"Armadillo Run"
- Modify class material
- Use Process Dashboard



- Disadvantage
Context switch – game play to discipline work

- Modified class material to be domain neutral
 - Include discussions on how this applies to each discipline

- Normal progression thru PSP processes
 - 5-10 game levels as exercises
 - a game level takes ~5-60 minutes to complete

- Process Dashboard for planning, tracking, and data analysis
 - Will need minor tweaks

- Size Proxy tables
- Defect types
- Checklists

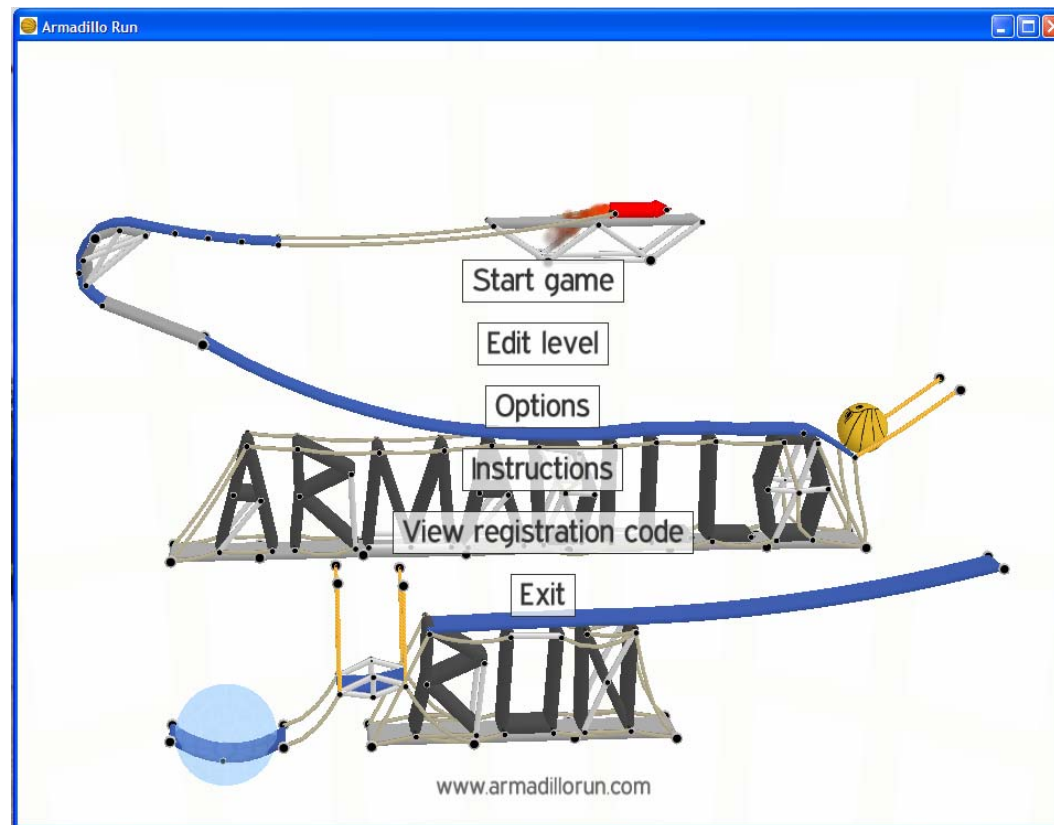
10 Movement
 11 - blockage
 12- speed
 13 - missed need
 14 - placement
 20 Structure
 21 - anchor

10 General
 20 Movement
 30 Catch
 40 Structure
 50 Force

need anchor

	Complete	Verify that the design covers all of the applicable requirements. - Cost						
	Movement	Verify that armadillo movement is correct Verify placement of the mechanism Ensure movement needed Ensure the correct direction Verify angles Ensure mechanism will stop Ensure mechanism will not block Armadillo Verify there is enough force						much much
VS								
S								
M								much
L								
VL								ing nodes
	Catch	- Verify a catch is needed. - Verify placement - Ensure armadillo will stop. - Ensure correct hard/softness						
mc								
ne								
su								
ca	Structure	Where the design assumes or relies upon external limits, determine if behavior is correct at nominal values, at limits, and beyond limits. Ensure structure will support itself Correct materials are used Anchor points are used Balance is correct						
ur								
	Run	- Verify a run is needed - Ensure placement - Ensure firmness - Ensure movement is not blocked.						

Demo



Next Steps

1. Review with Educational design
2. Modify Course Material
3. Try this in a pilot environment with multiple disciplines
4. Modify Process Dashboard
5. Get feedback from this community – You can help.

Questions ???

Contact Information

Dan Wall

VP Production Methods

150 Broadway Suite 205

Albany, NY 12204

DWall@vvisions.com

(518)-701-2512